



Red Hat OpenStack Platform 17.0

Configuration Reference

Configuring Red Hat OpenStack Platform environments

Red Hat OpenStack Platform 17.0 Configuration Reference

Configuring Red Hat OpenStack Platform environments

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

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.

Table of Contents

PREFACE	10
MAKING OPEN SOURCE MORE INCLUSIVE	11
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	12
CHAPTER 1. BARBICAN	13
1.1. BARBICAN.CONF	13
1.1.1. DEFAULT	13
1.1.2. certificate	20
1.1.3. certificate_event	20
1.1.4. cors	21
1.1.5. crypto	21
1.1.6. dogtag_plugin	22
1.1.7. keystone_authtoken	23
1.1.8. keystone_notifications	26
1.1.9. kmip_plugin	27
1.1.10. oslo_messaging_amqp	28
1.1.11. oslo_messaging_kafka	31
1.1.12. oslo_messaging_notifications	32
1.1.13. oslo_messaging_rabbit	33
1.1.14. oslo_middleware	35
1.1.15. oslo_policy	35
1.1.16. p11_crypto_plugin	37
1.1.17. queue	38
1.1.18. quotas	39
1.1.19. retry_scheduler	39
1.1.20. secretstore	39
1.1.21. simple_crypto_plugin	40
1.1.22. snakeoil_ca_plugin	40
1.1.23. ssl	41
CHAPTER 2. CINDER	42
2.1. CINDER.CONF	42
2.1.1. DEFAULT	42
2.1.2. backend	72
2.1.3. backend_defaults	72
2.1.4. barbican	111
2.1.5. brcd_fabric_example	111
2.1.6. cisco_fabric_example	112
2.1.7. coordination	113
2.1.8. cors	113
2.1.9. database	114
2.1.10. fc-zone-manager	116
2.1.11. healthcheck	116
2.1.12. key_manager	117
2.1.13. keystone_authtoken	118
2.1.14. nova	122
2.1.15. oslo_concurrency	123
2.1.16. oslo_messaging_amqp	123
2.1.17. oslo_messaging_kafka	126
2.1.18. oslo_messaging_notifications	128

2.1.19. oslo_messaging_rabbit	128
2.1.20. oslo_middleware	130
2.1.21. oslo_policy	131
2.1.22. oslo_reports	132
2.1.23. oslo_versionedobjects	132
2.1.24. privsep	133
2.1.25. profiler	133
2.1.26. sample_castellan_source	136
2.1.27. sample_remote_file_source	136
2.1.28. service_user	137
2.1.29. ssl	138
2.1.30. vault	139
CHAPTER 3. GLANCE	140
3.1. GLANCE-API.CONF	140
3.1.1. DEFAULT	140
3.1.2. cinder	173
3.1.3. cors	181
3.1.4. database	182
3.1.5. file	184
3.1.6. glance.store.http.store	188
3.1.7. glance.store.rbd.store	190
3.1.8. glance.store.s3.store	193
3.1.9. glance.store.swift.store	198
3.1.10. glance.store.vmware_datastore.store	213
3.1.11. glance_store	218
3.1.12. image_format	261
3.1.13. keystone_authtoken	261
3.1.14. oslo_concurrency	265
3.1.15. oslo_messaging_amqp	265
3.1.16. oslo_messaging_kafka	268
3.1.17. oslo_messaging_notifications	270
3.1.18. oslo_messaging_rabbit	270
3.1.19. oslo_middleware	272
3.1.20. oslo_policy	273
3.1.21. paste_deploy	274
3.1.22. profiler	276
3.1.23. store_type_location_strategy	279
3.1.24. task	280
3.1.25. taskflow_executor	282
3.1.26. wsgi	284
3.2. GLANCE-SCRUBBER.CONF	285
3.2.1. DEFAULT	285
3.2.2. database	307
3.2.3. glance_store	309
3.2.4. oslo_concurrency	350
3.2.5. oslo_policy	351
3.3. GLANCE-CACHE.CONF	352
3.3.1. DEFAULT	352
3.3.2. glance_store	373
3.3.3. oslo_policy	415
CHAPTER 4. HEAT	418

4.1. HEAT.CONF	418
4.1.1. DEFAULT	418
4.1.2. auth_password	429
4.1.3. clients	429
4.1.4. clients_aodh	429
4.1.5. clients_barbican	430
4.1.6. clients_cinder	430
4.1.7. clients_designate	431
4.1.8. clients_glance	431
4.1.9. clients_heat	432
4.1.10. clients_keystone	432
4.1.11. clients_magnum	433
4.1.12. clients_manila	433
4.1.13. clients_mistral	434
4.1.14. clients_monasca	434
4.1.15. clients_neutron	435
4.1.16. clients_nova	435
4.1.17. clients_octavia	436
4.1.18. clients_sahara	436
4.1.19. clients_senlin	437
4.1.20. clients_swift	437
4.1.21. clients_trove	438
4.1.22. clients_vitrage	438
4.1.23. clients_zaqar	439
4.1.24. cors	439
4.1.25. database	440
4.1.26. ec2authtoken	442
4.1.27. eventlet_opts	442
4.1.28. healthcheck	443
4.1.29. heat_api	443
4.1.30. heat_api_cfn	444
4.1.31. heat_api_cloudwatch	445
4.1.32. keystone_authtoken	446
4.1.33. noauth	449
4.1.34. oslo_messaging_amqp	450
4.1.35. oslo_messaging_kafka	453
4.1.36. oslo_messaging_notifications	454
4.1.37. oslo_messaging_rabbit	455
4.1.38. oslo_middleware	457
4.1.39. oslo_policy	457
4.1.40. paste_deploy	459
4.1.41. profiler	459
4.1.42. revision	461
4.1.43. ssl	462
4.1.44. trustee	462
4.1.45. volumes	463
CHAPTER 5. IRONIC	465
5.1. IRONIC.CONF	465
5.1.1. DEFAULT	465
5.1.2. agent	480
5.1.3. anaconda	482
5.1.4. ansible	482

5.1.5. api	484
5.1.6. audit	485
5.1.7. cinder	486
5.1.8. conductor	489
5.1.9. console	493
5.1.10. cors	494
5.1.11. database	495
5.1.12. deploy	496
5.1.13. dhcp	499
5.1.14. disk_partitioner	499
5.1.15. disk_utils	500
5.1.16. drac	500
5.1.17. glance	501
5.1.18. healthcheck	505
5.1.19. ilo	506
5.1.20. inspector	508
5.1.21. ipmi	511
5.1.22. irmc	512
5.1.23. ironic_lib	514
5.1.24. iscsi	514
5.1.25. json_rpc	515
5.1.26. keystone_auth token	517
5.1.27. mdns	520
5.1.28. metrics	520
5.1.29. metrics_statsd	521
5.1.30. molds	522
5.1.31. neutron	522
5.1.32. nova	527
5.1.33. oslo_concurrency	529
5.1.34. oslo_messaging_amqp	530
5.1.35. oslo_messaging_kafka	533
5.1.36. oslo_messaging_notifications	534
5.1.37. oslo_messaging_rabbit	535
5.1.38. oslo_middleware	537
5.1.39. oslo_policy	537
5.1.40. oslo_reports	539
5.1.41. profiler	539
5.1.42. pxe	542
5.1.43. redfish	545
5.1.44. service_catalog	546
5.1.45. snmp	549
5.1.46. ssl	549
5.1.47. swift	550
5.1.48. xclarity	552
CHAPTER 6. IRONIC-INSPECTOR	554
6.1. INSPECTOR.CONF	554
6.1.1. DEFAULT	554
6.1.2. capabilities	558
6.1.3. coordination	559
6.1.4. cors	559
6.1.5. database	560
6.1.6. discovery	562

6.1.7. dnsmasq_pxe_filter	562
6.1.8. extra_hardware	563
6.1.9. iptables	563
6.1.10. ironic	563
6.1.11. keystone_authtoken	566
6.1.12. oslo_policy	569
6.1.13. pci_devices	571
6.1.14. port_physnet	571
6.1.15. processing	571
6.1.16. pxe_filter	573
6.1.17. service_catalog	574
6.1.18. swift	576
CHAPTER 7. KEYSTONE	580
7.1. KEYSTONE.CONF	580
7.1.1. DEFAULT	580
7.1.2. application_credential	586
7.1.3. assignment	586
7.1.4. auth	587
7.1.5. cache	588
7.1.6. catalog	591
7.1.7. cors	592
7.1.8. credential	593
7.1.9. database	594
7.1.10. domain_config	596
7.1.11. endpoint_filter	596
7.1.12. endpoint_policy	597
7.1.13. eventlet_server	597
7.1.14. federation	598
7.1.15. fernet_receipts	599
7.1.16. fernet_tokens	600
7.1.17. healthcheck	601
7.1.18. identity	602
7.1.19. identity_mapping	604
7.1.20. jwt_tokens	605
7.1.21. ldap	606
7.1.22. memcache	612
7.1.23. oauth1	613
7.1.24. oslo_messaging_amqp	613
7.1.25. oslo_messaging_kafka	617
7.1.26. oslo_messaging_notifications	618
7.1.27. oslo_messaging_rabbit	618
7.1.28. oslo_middleware	620
7.1.29. oslo_policy	621
7.1.30. policy	622
7.1.31. profiler	623
7.1.32. receipt	625
7.1.33. resource	626
7.1.34. revoke	627
7.1.35. role	628
7.1.36. saml	628
7.1.37. security_compliance	630
7.1.38. shadow_users	632

7.1.39. token	633
7.1.40. tokenless_auth	635
7.1.41. totp	636
7.1.42. trust	636
7.1.43. unified_limit	637
7.1.44. wsgi	638
CHAPTER 8. NEUTRON	639
8.1. DHCP_AGENT.INI	639
8.1.1. DEFAULT	639
8.1.2. agent	645
8.1.3. ovs	645
8.2. L3_AGENT.INI	646
8.2.1. DEFAULT	646
8.2.2. agent	654
8.2.3. network_log	654
8.2.4. ovs	654
8.3. LINUXBRIDGE_AGENT.INI	655
8.3.1. DEFAULT	655
8.3.2. agent	659
8.3.3. linux_bridge	660
8.3.4. network_log	660
8.3.5. securitygroup	660
8.3.6. vxlan	661
8.4. METADATA_AGENT.INI	662
8.4.1. DEFAULT	663
8.4.2. agent	667
8.4.3. cache	668
8.5. METERING_AGENT.INI	671
8.5.1. DEFAULT	671
8.5.2. agent	675
8.5.3. ovs	676
8.6. ML2_CONF.INI	677
8.6.1. DEFAULT	677
8.6.2. ml2	681
8.6.3. ml2_type_flat	682
8.6.4. ml2_type_geneve	682
8.6.5. ml2_type_gre	683
8.6.6. ml2_type_vlan	683
8.6.7. ml2_type_vxlan	683
8.6.8. ovs_driver	684
8.6.9. securitygroup	684
8.6.10. sriov_driver	685
8.7. NEUTRON.CONF	685
8.7.1. DEFAULT	685
8.7.2. agent	697
8.7.3. cors	699
8.7.4. database	699
8.7.5. healthcheck	701
8.7.6. ironic	702
8.7.7. keystone_authtoken	703
8.7.8. nova	707
8.7.9. oslo_concurrency	709

8.7.10. oslo_messaging_amqp	709
8.7.11. oslo_messaging_kafka	712
8.7.12. oslo_messaging_notifications	714
8.7.13. oslo_messaging_rabbit	714
8.7.14. oslo_middleware	716
8.7.15. oslo_policy	716
8.7.16. oslo_reports	718
8.7.17. privsep	718
8.7.18. quotas	719
8.7.19. ssl	720
8.8. OPENVSWITCH_AGENT.INI	720
8.8.1. DEFAULT	720
8.8.2. agent	724
8.8.3. network_log	725
8.8.4. ovs	725
8.8.5. securitygroup	728
8.9. SRIOV_AGENT.INI	729
8.9.1. DEFAULT	729
8.9.2. agent	733
8.9.3. sriov_nic	733
CHAPTER 9. NOVA	735
9.1. NOVA.CONF	735
9.1.1. DEFAULT	735
9.1.2. api	778
9.1.3. api_database	785
9.1.4. barbican	786
9.1.5. cache	787
9.1.6. cinder	790
9.1.7. compute	794
9.1.8. conductor	802
9.1.9. console	802
9.1.10. consoleauth	803
9.1.11. cors	803
9.1.12. cyborg	804
9.1.13. database	805
9.1.14. devices	807
9.1.15. ephemeral_storage_encryption	808
9.1.16. filter_scheduler	809
9.1.17. glance	822
9.1.18. guestfs	827
9.1.19. healthcheck	828
9.1.20. hyperv	828
9.1.21. image_cache	835
9.1.22. ironic	837
9.1.23. key_manager	839
9.1.24. keystone	841
9.1.25. keystone_authtoken	842
9.1.26. libvirt	845
9.1.27. metrics	876
9.1.28. mks	880
9.1.29. neutron	881
9.1.30. notifications	886

9.1.31. oslo_concurrency	887
9.1.32. oslo_messaging_amqp	887
9.1.33. oslo_messaging_kafka	891
9.1.34. oslo_messaging_notifications	892
9.1.35. oslo_messaging_rabbit	892
9.1.36. oslo_middleware	894
9.1.37. oslo_policy	895
9.1.38. pci	896
9.1.39. placement	899
9.1.40. powervm	902
9.1.41. privsep	902
9.1.42. profiler	903
9.1.43. quota	905
9.1.44. rdp	910
9.1.45. remote_debug	911
9.1.46. scheduler	912
9.1.47. serial_console	917
9.1.48. service_user	919
9.1.49. spice	921
9.1.50. upgrade_levels	924
9.1.51. vault	927
9.1.52. vendordata_dynamic_auth	927
9.1.53. vmware	929
9.1.54. vnc	934
9.1.55. workarounds	937
9.1.56. wsgi	949
9.1.57. zvm	953

PREFACE

This document describes the options available in the configuration files for each of the major services in Red Hat OpenStack Platform. The content is automatically generated based on the values in the configuration files themselves, and is provided for reference purposes only.



WARNING

Manually editing configuration files is not supported. All configuration changes must be made through the Director. Red Hat provides this guide as a technical reference only.

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Tell us how we can make it better.

Using the Direct Documentation Feedback (DDF) function

Use the **Add Feedback** DDF function for direct comments on specific sentences, paragraphs, or code blocks.

1. View the documentation in the *Multi-page HTML* format.
2. Ensure that you see the **Feedback** button in the upper right corner of the document.
3. Highlight the part of text that you want to comment on.
4. Click **Add Feedback**.
5. Complete the **Add Feedback** field with your comments.
6. Optional: Add your email address so that the documentation team can contact you for clarification on your issue.
7. Click **Submit**.

CHAPTER 1. BARBICAN

The following chapter contains information about the configuration options in the **barbican** service.

1.1. BARBICAN.CONF

This section contains options for the `/etc/barbican/barbican.conf` file.

1.1.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/barbican/barbican.conf` file.

Configuration option = Default value	Type	Description
admin_role = admin	string value	Role used to identify an authenticated user as administrator.
allow_anonymous_access = False	boolean value	Allow unauthenticated users to access the API with read-only privileges. This only applies when using ContextMiddleware.
api_paste_config = api-paste.ini	string value	File name for the paste.deploy config for api service
backdoor_port = None	string value	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.
backdoor_socket = None	string value	Enable eventlet backdoor, using the provided path as a unix socket that can receive connections. This option is mutually exclusive with <i>backdoor_port</i> in that only one should be provided. If both are provided then the existence of this option overrides the usage of that option. Inside the path {pid} will be replaced with the PID of the current process.
client_socket_timeout = 900	integer value	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.
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy

Configuration option = Default value	Type	Description
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
control_exchange = openstack	string value	The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.
db_auto_create = False	boolean value	Create the Barbican database on service startup.
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.
default_limit_paging = 10	integer value	Default page size for the <i>limit</i> paging URL parameter.
default_log_levels = ['amqp=WARN', 'amqp-lib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
executor_thread_pool_size = 64	integer value	Size of executor thread pool when executor is threading or eventlet.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.

Configuration option = Default value	Type	Description
host_href = http://localhost:9311	string value	Host name, for use in HATEOAS-style references Note: Typically this would be the load balanced endpoint that clients would use to communicate back with this service. If a deployment wants to derive host from wsgi request instead then make this blank. Blank is needed to override default config value which is http://localhost:9311
<code>`instance_format = [instance: %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.
<code>`instance_uuid_format = [instance: %(uuid)s] `</code>	string value	The format for an instance UUID that is passed with the log message.
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code> . This option is ignored if log_config_append is set.
log-dir = None	string value	(Optional) The base directory used for relative log_file paths. This option is ignored if log_config_append is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to stderr as defined by use_stderr. This option is ignored if log_config_append is set.
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless log_rotation_type is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.

Configuration option = Default value	Type	Description
logging_context_format_string = <code>%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s</code>	string value	Format string to use for log messages with context. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_debug_format_s uffix = <code>%(funcName)s %(pathname)s:%(lineno)d</code>	string value	Additional data to append to log message when logging level for the message is <code>DEBUG</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_default_format_s tring = <code>%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s</code>	string value	Format string to use for log messages when context is undefined. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_exception_prefix = <code>%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s</code>	string value	Prefix each line of exception output with this format. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_user_identity_for mat = <code>%(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s</code>	string value	Defines the format string for <code>%(user_identity)s</code> that is used in <code>logging_context_format_string</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
max_allowed_request_size_in_bytes = 25000	integer value	Maximum allowed http request size against the <code>barbican-api</code> .
max_allowed_secret_in_bytes = 20000	integer value	Maximum allowed secret size in bytes.
max_header_line = 16384	integer value	Maximum line size of message headers to be accepted. <code>max_header_line</code> may need to be increased when using large tokens (typically those generated when keystone is configured to use PKI tokens with big service catalogs).
max_limit_paging = 100	integer value	Maximum page size for the <i>limit</i> paging URL parameter.
max_logfile_count = 30	integer value	Maximum number of rotated log files.

Configuration option = Default value	Type	Description
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
publish_errors = False	boolean value	Enables or disables publication of error events.
rate_limit_burst = 0	integer value	Maximum number of logged messages per rate_limit_interval.
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
rpc_conn_pool_size = 30	integer value	Size of RPC connection pool.
rpc_ping_enabled = False	boolean value	Add an endpoint to answer to ping calls. Endpoint is named oslo_rpc_server_ping
rpc_response_timeout = 60	integer value	Seconds to wait for a response from a call.
run_external_periodic_tasks = True	boolean value	Some periodic tasks can be run in a separate process. Should we run them here?
sql_connection = sqlite:///barbican.sqlite	string value	SQLAlchemy connection string for the reference implementation registry server. Any valid SQLAlchemy connection string is fine. See: http://www.sqlalchemy.org/docs/05/reference/sqlalchemy/connections.html#sqlalchemy.create_engine . Note: For absolute addresses, use //// slashes after <i>sqlite</i> ..
sql_idle_timeout = 3600	integer value	Period in seconds after which SQLAlchemy should reestablish its connection to the database. MySQL uses a default wait_timeout of 8 hours, after which it will drop idle connections. This can result in <i>MySQL Gone Away</i> exceptions. If you notice this, you can lower this value to ensure that SQLAlchemy reconnects before MySQL can drop the connection.
sql_max_retries = 60	integer value	Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.

Configuration option = Default value	Type	Description
sql_pool_class = QueuePool	string value	Accepts a class imported from the sqlalchemy.pool module, and handles the details of building the pool for you. If commented out, SQLAlchemy will select based on the database dialect. Other options are QueuePool (for SQLAlchemy-managed connections) and NullPool (to disabled SQLAlchemy management of connections). See http://docs.sqlalchemy.org/en/latest/core/pooling.html for more details
sql_pool_logging = False	boolean value	Show SQLAlchemy pool-related debugging output in logs (sets DEBUG log level output) if specified.
sql_pool_max_overflow = 10	integer value	The maximum overflow size of the pool used by SQLAlchemy. When the number of checked-out connections reaches the size set in sql_pool_size, additional connections will be returned up to this limit. It follows then that the total number of simultaneous connections the pool will allow is sql_pool_size + sql_pool_max_overflow. Can be set to -1 to indicate no overflow limit, so no limit will be placed on the total number of concurrent connections. Comment out to allow SQLAlchemy to select the default.
sql_pool_size = 5	integer value	Size of pool used by SQLAlchemy. This is the largest number of connections that will be kept persistently in the pool. Can be set to 0 to indicate no size limit. To disable pooling, use a NullPool with sql_pool_class instead. Comment out to allow SQLAlchemy to select the default.
sql_retry_interval = 1	integer value	Interval between retries of opening a SQL connection.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
tcp_keepidle = 600	integer value	Sets the value of TCP_KEEPIDLE in seconds for each server socket. Not supported on OS X.

Configuration option = Default value	Type	Description
transport_url = rabbit://	string value	The network address and optional user credentials for connecting to the messaging backend, in URL format. The expected format is: <pre>driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query</pre> Example: <pre>rabbit://rabbitmq:password@127.0.0.1:5672//</pre> For full details on the fields in the URL see the documentation of oslo_messaging.TransportURL at https://docs.openstack.org/oslo.messaging/latest/reference/transport.html
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.
wsgi_default_pool_size = 100	integer value	Size of the pool of greenthreads used by wsgi
wsgi_keep_alive = True	boolean value	If False, closes the client socket connection explicitly.

Configuration option = Default value	Type	Description
wsgi_log_format = % (client_ip)s "% (request_line)s" status: % (status_code)s len: % (body_length)s time: % (wall_seconds).7f	string value	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.
wsgi_server_debug = False	boolean value	True if the server should send exception tracebacks to the clients on 500 errors. If False, the server will respond with empty bodies.

1.1.2. certificate

The following table outlines the options available under the **[certificate]** group in the **/etc/barbican/barbican.conf** file.

Table 1.1. certificate

Configuration option = Default value	Type	Description
enabled_certificate_plugins = ['simple_certificate']	multi valued	List of certificate plugins to load.
namespace = barbican.certificate.plugin	string value	Extension namespace to search for plugins.

1.1.3. certificate_event

The following table outlines the options available under the **[certificate_event]** group in the **/etc/barbican/barbican.conf** file.

Table 1.2. certificate_event

Configuration option = Default value	Type	Description
enabled_certificate_event_plugins = ['simple_certificate_event']	multi valued	List of certificate plugins to load.
namespace = barbican.certificate.event.plugin	string value	Extension namespace to search for eventing plugins.

1.1.4. cors

The following table outlines the options available under the **[cors]** group in the **/etc/barbican/barbican.conf** file.

Table 1.3. cors

Configuration option = Default value	Type	Description
allow_credentials = True	boolean value	Indicate that the actual request can include user credentials
allow_headers = ['X-Auth-Token', 'X-Openstack-Request-Id', 'X-Project-Id', 'X-Identity-Status', 'X-User-Id', 'X-Storage-Token', 'X-Domain-Id', 'X-User-Domain-Id', 'X-Project-Domain-Id', 'X-Roles']	list value	Indicate which header field names may be used during the actual request.
allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH']	list value	Indicate which methods can be used during the actual request.
allowed_origin = None	list value	Indicate whether this resource may be shared with the domain received in the requests "origin" header. Format: "<protocol>://<host>[:<port>]", no trailing slash. Example: https://horizon.example.com
expose_headers = ['X-Auth-Token', 'X-Openstack-Request-Id', 'X-Project-Id', 'X-Identity-Status', 'X-User-Id', 'X-Storage-Token', 'X-Domain-Id', 'X-User-Domain-Id', 'X-Project-Domain-Id', 'X-Roles']	list value	Indicate which headers are safe to expose to the API. Defaults to HTTP Simple Headers.
max_age = 3600	integer value	Maximum cache age of CORS preflight requests.

1.1.5. crypto

The following table outlines the options available under the **[crypto]** group in the **/etc/barbican/barbican.conf** file.

Table 1.4. crypto

Configuration option = Default value	Type	Description
enabled_crypto_plugins = ['simple_crypto']	multi valued	List of crypto plugins to load.
namespace = barbican.crypto.plugin	string value	Extension namespace to search for plugins.

1.1.6. dogtag_plugin

The following table outlines the options available under the **[dogtag_plugin]** group in the **/etc/barbican/barbican.conf** file.

Table 1.5. dogtag_plugin

Configuration option = Default value	Type	Description
auto_approved_profiles = caServerCert	string value	List of automatically approved enrollment profiles
ca_expiration_time = 1	integer value	Time in days for CA entries to expire
dogtag_host = localhost	string value	Hostname for the Dogtag instance
dogtag_port = 8443	port value	Port for the Dogtag instance
nss_db_path = /etc/barbican/alias	string value	Path to the NSS certificate database
nss_password = None	string value	Password for the NSS certificate databases
pem_path = /etc/barbican/kra_admin_ cert.pem	string value	Path to PEM file for authentication
plugin_name = Dogtag KRA	string value	User friendly plugin name
plugin_working_dir = /etc/barbican/dogtag	string value	Working directory for Dogtag plugin
retries = 3	integer value	Retries when storing or generating secrets
simple_cmc_profile = caOtherCert	string value	Profile for simple CMC requests

1.1.7. keystone_authtoken

The following table outlines the options available under the **[keystone_authtoken]** group in the **/etc/barbican/barbican.conf** file.

Table 1.6. keystone_authtoken

Configuration option = Default value	Type	Description
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load
auth_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint. This option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release. Deprecated since: Queens <i>*Reason:*</i>The <code>auth_uri</code> option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release.
auth_version = None	string value	API version of the Identity API endpoint.
cache = None	string value	Request environment key where the Swift cache object is stored. When <code>auth_token</code> middleware is deployed with a Swift cache, use this option to have the middleware share a caching backend with swift. Otherwise, use the memcached_servers option instead.
cafile = None	string value	A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
certfile = None	string value	Required if identity server requires client certificate
delay_auth_decision = False	boolean value	Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components.

Configuration option = Default value	Type	Description
enforce_token_bind = permissive	string value	Used to control the use and type of token binding. Can be set to: "disabled" to not check token binding. "permissive" (default) to validate binding information if the bind type is of a form known to the server and ignore it if not. "strict" like "permissive" but if the bind type is unknown the token will be rejected. "required" any form of token binding is needed to be allowed. Finally the name of a binding method that must be present in tokens.
http_connect_timeout = None	integer value	Request timeout value for communicating with Identity API server.
http_request_max_retries = 3	integer value	How many times are we trying to reconnect when communicating with Identity API Server.
include_service_catalog = True	boolean value	(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	boolean value	Verify HTTPS connections.
interface = internal	string value	Interface to use for the Identity API endpoint. Valid values are "public", "internal" (default) or "admin".
keyfile = None	string value	Required if identity server requires client certificate
memcache_pool_conn_get_timeout = 10	integer value	(Optional) Number of seconds that an operation will wait to get a memcached client connection from the pool.
memcache_pool_dead_retry = 300	integer value	(Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = 10	integer value	(Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = 3	integer value	(Optional) Socket timeout in seconds for communicating with a memcached server.
memcache_pool_unused_timeout = 60	integer value	(Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
memcache_secret_key = None	string value	(Optional, mandatory if memcache_security_strategy is defined) This string is used for key derivation.

Configuration option = Default value	Type	Description
memcache_security_strategy = None	string value	(Optional) If defined, indicate whether token data should be authenticated or authenticated and encrypted. If MAC, token data is authenticated (with HMAC) in the cache. If ENCRYPT, token data is encrypted and authenticated in the cache. If the value is not one of these options or empty, auth_token will raise an exception on initialization.
memcache_use_advanced_pool = False	boolean value	(Optional) Use the advanced (eventlet safe) memcached client pool. The advanced pool will only work under python 2.x.
memcached_servers = None	list value	Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
region_name = None	string value	The region in which the identity server can be found.
service_token_roles = ['service']	list value	A choice of roles that must be present in a service token. Service tokens are allowed to request that an expired token can be used and so this check should tightly control that only actual services should be sending this token. Roles here are applied as an ANY check so any role in this list must be present. For backwards compatibility reasons this currently only affects the allow_expired check.
service_token_roles_required = False	boolean value	For backwards compatibility reasons we must let valid service tokens pass that don't pass the service_token_roles check as valid. Setting this true will become the default in a future release and should be enabled if possible.
service_type = None	string value	The name or type of the service as it appears in the service catalog. This is used to validate tokens that have restricted access rules.
token_cache_time = 300	integer value	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.

Configuration option = Default value	Type	Description
www_authenticate_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint.

1.1.8. keystone_notifications

The following table outlines the options available under the **[keystone_notifications]** group in the **/etc/barbican/barbican.conf** file.

Table 1.7. keystone_notifications

Configuration option = Default value	Type	Description
allow_requeue = False	boolean value	True enables requeue feature in case of notification processing error. Enable this only when underlying transport supports this feature.
control_exchange = keystone	string value	The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.
enable = False	boolean value	True enables keystone notification listener functionality.
pool_name = None	string value	Pool name for notifications listener. Setting this to a distinctive value will allow barbican notifications listener to receive its own copy of all messages from the topic without interfering with other services listening on the same topic. This feature is supported only by some oslo.messaging backends (in particular by rabbitmq) and for those it is preferable to use it instead of separate notification topic for barbican.
thread_pool_size = 10	integer value	Define the number of max threads to be used for notification server processing functionality.

Configuration option = Default value	Type	Description
topic = notifications	string value	Keystone notification queue topic name. This name needs to match one of values mentioned in Keystone deployment's <i>notification_topics</i> configuration e.g. <i>notification_topics=notifications</i> , <i>barbican_notifications</i> Multiple servers may listen on a topic and messages will be dispatched to one of the servers in a round-robin fashion. That's why Barbican service should have its own dedicated notification queue so that it receives all of Keystone notifications. Alternatively if the chosen oslo.messaging backend supports listener pooling (for example rabbitmq), setting a non-default <i>pool_name</i> option should be preferred.
version = 1.0	string value	Version of tasks invoked via notifications

1.1.9. kmip_plugin

The following table outlines the options available under the **[kmip_plugin]** group in the */etc/barbican/barbican.conf* file.

Table 1.8. kmip_plugin

Configuration option = Default value	Type	Description
ca_certs = None	string value	File path to concatenated "certification authority" certificates
certfile = None	string value	File path to local client certificate
host = localhost	string value	Address of the KMIP server
keyfile = None	string value	File path to local client certificate keyfile
password = None	string value	Password for authenticating with KMIP server
pkcs1_only = False	boolean value	Only support PKCS#1 encoding of asymmetric keys
plugin_name = KMIP HSM	string value	User friendly plugin name
port = 5696	port value	Port for the KMIP server
ssl_version = PROTOCOL_TLSv1_2	string value	SSL version, maps to the module ssl's constants

Configuration option = Default value	Type	Description
username = None	string value	Username for authenticating with KMIP server

1.1.10. oslo_messaging_amqp

The following table outlines the options available under the **[oslo_messaging_amqp]** group in the **/etc/barbican/barbican.conf** file.

Table 1.9. oslo_messaging_amqp

Configuration option = Default value	Type	Description
addressing_mode = dynamic	string value	Indicates the addressing mode used by the driver. Permitted values: <i>legacy</i> - use legacy non-routable addressing <i>routable</i> - use routable addresses <i>dynamic</i> - use legacy addresses if the message bus does not support routing otherwise use routable addressing
anycast_address = anycast	string value	Appended to the address prefix when sending to a group of consumers. Used by the message bus to identify messages that should be delivered in a round-robin fashion across consumers.
broadcast_prefix = broadcast	string value	address prefix used when broadcasting to all servers
connection_retry_backoff = 2	integer value	Increase the connection_retry_interval by this many seconds after each unsuccessful failover attempt.
connection_retry_interval = 1	integer value	Seconds to pause before attempting to re-connect.
connection_retry_interval_max = 30	integer value	Maximum limit for connection_retry_interval + connection_retry_backoff
container_name = None	string value	Name for the AMQP container. must be globally unique. Defaults to a generated UUID
default_notification_exchange = None	string value	Exchange name used in notification addresses. Exchange name resolution precedence: Target.exchange if set else default_notification_exchange if set else control_exchange if set else <i>notify</i>

Configuration option = Default value	Type	Description
default_notify_timeout = 30	integer value	The deadline for a sent notification message delivery. Only used when caller does not provide a timeout expiry.
default_reply_retry = 0	integer value	The maximum number of attempts to re-send a reply message which failed due to a recoverable error.
default_reply_timeout = 30	integer value	The deadline for an rpc reply message delivery.
default_rpc_exchange = None	string value	Exchange name used in RPC addresses. Exchange name resolution precedence: Target.exchange if set else default_rpc_exchange if set else control_exchange if set else <i>rpc</i>
default_send_timeout = 30	integer value	The deadline for an rpc cast or call message delivery. Only used when caller does not provide a timeout expiry.
default_sender_link_timeout = 600	integer value	The duration to schedule a purge of idle sender links. Detach link after expiry.
group_request_prefix = unicast	string value	address prefix when sending to any server in group
idle_timeout = 0	integer value	Timeout for inactive connections (in seconds)
link_retry_delay = 10	integer value	Time to pause between re-connecting an AMQP 1.0 link that failed due to a recoverable error.
multicast_address = multicast	string value	Appended to the address prefix when sending a fanout message. Used by the message bus to identify fanout messages.
notify_address_prefix = openstack.org/om/notify	string value	Address prefix for all generated Notification addresses
notify_server_credit = 100	integer value	Window size for incoming Notification messages
pre_settled = ['rpc-cast', 'rpc-reply']	multi valued	Send messages of this type pre-settled. Pre-settled messages will not receive acknowledgement from the peer. Note well: pre-settled messages may be silently discarded if the delivery fails. Permitted values: <i>rpc-call</i> - send RPC Calls pre-settled <i>rpc-reply</i> - send RPC Replies pre-settled <i>rpc-cast</i> - Send RPC Casts pre-settled <i>notify</i> - Send Notifications pre-settled

Configuration option = Default value	Type	Description
pseudo_vhost = True	boolean value	Enable virtual host support for those message buses that do not natively support virtual hosting (such as qpid). When set to true the virtual host name will be added to all message bus addresses, effectively creating a private <i>subnet</i> per virtual host. Set to False if the message bus supports virtual hosting using the <i>hostname</i> field in the AMQP 1.0 Open performative as the name of the virtual host.
reply_link_credit = 200	integer value	Window size for incoming RPC Reply messages.
rpc_address_prefix = openstack.org/om/rpc	string value	Address prefix for all generated RPC addresses
rpc_server_credit = 100	integer value	Window size for incoming RPC Request messages
<code>`sasldb_config_dir = `</code>	string value	Path to directory that contains the SASL configuration
<code>`sasldb_config_name = `</code>	string value	Name of configuration file (without .conf suffix)
<code>`sasldb_default_realm = `</code>	string value	SASL realm to use if no realm present in username
<code>`sasldb_mechanisms = `</code>	string value	Space separated list of acceptable SASL mechanisms
server_request_prefix = exclusive	string value	address prefix used when sending to a specific server
ssl = False	boolean value	Attempt to connect via SSL. If no other ssl-related parameters are given, it will use the system's CA-bundle to verify the server's certificate.
<code>`ssl_ca_file = `</code>	string value	CA certificate PEM file used to verify the server's certificate
<code>`ssl_cert_file = `</code>	string value	Self-identifying certificate PEM file for client authentication
<code>`ssl_key_file = `</code>	string value	Private key PEM file used to sign <code>ssl_cert_file</code> certificate (optional)
ssl_key_password = None	string value	Password for decrypting <code>ssl_key_file</code> (if encrypted)

Configuration option = Default value	Type	Description
ssl_verify_vhost = False	boolean value	By default SSL checks that the name in the server's certificate matches the hostname in the transport_url. In some configurations it may be preferable to use the virtual hostname instead, for example if the server uses the Server Name Indication TLS extension (rfc6066) to provide a certificate per virtual host. Set ssl_verify_vhost to True if the server's SSL certificate uses the virtual host name instead of the DNS name.
trace = False	boolean value	Debug: dump AMQP frames to stdout
unicast_address = unicast	string value	Appended to the address prefix when sending to a particular RPC/Notification server. Used by the message bus to identify messages sent to a single destination.

1.1.11. oslo_messaging_kafka

The following table outlines the options available under the **[oslo_messaging_kafka]** group in the **/etc/barbican/barbican.conf** file.

Table 1.10. oslo_messaging_kafka

Configuration option = Default value	Type	Description
compression_codec = none	string value	The compression codec for all data generated by the producer. If not set, compression will not be used. Note that the allowed values of this depend on the kafka version
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
consumer_group = oslo_messaging_consumer	string value	Group id for Kafka consumer. Consumers in one group will coordinate message consumption
enable_auto_commit = False	boolean value	Enable asynchronous consumer commits
kafka_consumer_timeout = 1.0	floating point value	Default timeout(s) for Kafka consumers

Configuration option = Default value	Type	Description
kafka_max_fetch_bytes = 1048576	integer value	Max fetch bytes of Kafka consumer
max_poll_records = 500	integer value	The maximum number of records returned in a poll call
pool_size = 10	integer value	Pool Size for Kafka Consumers
producer_batch_size = 16384	integer value	Size of batch for the producer async send
producer_batch_timeout = 0.0	floating point value	Upper bound on the delay for KafkaProducer batching in seconds
sasl_mechanism = PLAIN	string value	Mechanism when security protocol is SASL
security_protocol = PLAINTEXT	string value	Protocol used to communicate with brokers
<code>`ssl_cafile = `</code>	string value	CA certificate PEM file used to verify the server certificate
<code>`ssl_client_cert_file = `</code>	string value	Client certificate PEM file used for authentication.
<code>`ssl_client_key_file = `</code>	string value	Client key PEM file used for authentication.
<code>`ssl_client_key_password = `</code>	string value	Client key password file used for authentication.

1.1.12. oslo_messaging_notifications

The following table outlines the options available under the **[oslo_messaging_notifications]** group in the **/etc/barbican/barbican.conf** file.

Table 1.11. oslo_messaging_notifications

Configuration option = Default value	Type	Description
driver = []	multi valued	The Drivers(s) to handle sending notifications. Possible values are messaging, messagingv2, routing, log, test, noop
retry = -1	integer value	The maximum number of attempts to re-send a notification message which failed to be delivered due to a recoverable error. 0 - No retry, -1 - indefinite

Configuration option = Default value	Type	Description
topics = ['notifications']	list value	AMQP topic used for OpenStack notifications.
transport_url = None	string value	A URL representing the messaging driver to use for notifications. If not set, we fall back to the same configuration used for RPC.

1.1.13. oslo_messaging_rabbit

The following table outlines the options available under the **[oslo_messaging_rabbit]** group in the **/etc/barbican/barbican.conf** file.

Table 1.12. oslo_messaging_rabbit

Configuration option = Default value	Type	Description
amqp_auto_delete = False	boolean value	Auto-delete queues in AMQP.
amqp_durable_queues = False	boolean value	Use durable queues in AMQP.
direct_mandatory_flag = True	boolean value	(DEPRECATED) Enable/Disable the RabbitMQ mandatory flag for direct send. The direct send is used as reply, so the MessageUndeliverable exception is raised in case the client queue does not exist. MessageUndeliverable exception will be used to loop for a timeout to let a chance to sender to recover. This flag is deprecated and it will not be possible to deactivate this functionality anymore
enable_cancel_on_failover = False	boolean value	Enable x-cancel-on-ha-failover flag so that rabbitmq server will cancel and notify consumers when queue is down
heartbeat_in_pthread = True	boolean value	Run the health check heartbeat thread through a native python thread by default. If this option is equal to False then the health check heartbeat will inherit the execution model from the parent process. For example if the parent process has monkey patched the stdlib by using eventlet/greenlet then the heartbeat will be run through a green thread.
heartbeat_rate = 2	integer value	How often times during the heartbeat_timeout_threshold we check the heartbeat.

Configuration option = Default value	Type	Description
heartbeat_timeout_thresh old = 60	integer value	Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables heartbeat).
kombu_compression = None	string value	EXPERIMENTAL: Possible values are: gzip, bz2. If not set compression will not be used. This option may not be available in future versions.
kombu_failover_strategy = round-robin	string value	Determines how the next RabbitMQ node is chosen in case the one we are currently connected to becomes unavailable. Takes effect only if more than one RabbitMQ node is provided in config.
kombu_missing_consumer_retry_timeout = 60	integer value	How long to wait a missing client before abandoning to send it its replies. This value should not be longer than <code>rpc_response_timeout</code> .
kombu_reconnect_delay = 1.0	floating point value	How long to wait before reconnecting in response to an AMQP consumer cancel notification.
rabbit_ha_queues = False	boolean value	Try to use HA queues in RabbitMQ (<code>x-ha-policy: all</code>). If you change this option, you must wipe the RabbitMQ database. In RabbitMQ 3.0, queue mirroring is no longer controlled by the <code>x-ha-policy</code> argument when declaring a queue. If you just want to make sure that all queues (except those with auto-generated names) are mirrored across all nodes, run: <code>"rabbitmqctl set_policy HA ^(?!amq\.).* {\"ha-mode\": \"all\"}"</code>
rabbit_interval_max = 30	integer value	Maximum interval of RabbitMQ connection retries. Default is 30 seconds.
rabbit_login_method = AMQPLAIN	string value	The RabbitMQ login method.
rabbit_qos_prefetch_count = 0	integer value	Specifies the number of messages to prefetch. Setting to zero allows unlimited messages.
rabbit_retry_backoff = 2	integer value	How long to backoff for between retries when connecting to RabbitMQ.
rabbit_retry_interval = 1	integer value	How frequently to retry connecting with RabbitMQ.

Configuration option = Default value	Type	Description
rabbit_transient_queues_ttl = 1800	integer value	Positive integer representing duration in seconds for queue TTL (x-expire). Queues which are unused for the duration of the TTL are automatically deleted. The parameter affects only reply and fanout queues.
ssl = False	boolean value	Connect over SSL.
<code>`ssl_ca_file = `</code>	string value	SSL certification authority file (valid only if SSL enabled).
<code>`ssl_cert_file = `</code>	string value	SSL cert file (valid only if SSL enabled).
<code>`ssl_key_file = `</code>	string value	SSL key file (valid only if SSL enabled).
<code>`ssl_version = `</code>	string value	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.

1.1.14. oslo_middleware

The following table outlines the options available under the **[oslo_middleware]** group in the **/etc/barbican/barbican.conf** file.

Table 1.13. oslo_middleware

Configuration option = Default value	Type	Description
enable_proxy_headers_parsing = False	boolean value	Whether the application is behind a proxy or not. This determines if the middleware should parse the headers or not.

1.1.15. oslo_policy

The following table outlines the options available under the **[oslo_policy]** group in the **/etc/barbican/barbican.conf** file.

Table 1.14. oslo_policy

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
enforce_new_defaults = False	boolean value	This option controls whether or not to use old deprecated defaults when evaluating policies. If True , the old deprecated defaults are not going to be evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be disallowed. It is encouraged to enable this flag along with the enforce_scope flag so that you can get the benefits of new defaults and scope_type together
enforce_scope = False	boolean value	This option controls whether or not to enforce scope when evaluating policies. If True , the scope of the token used in the request is compared to the scope_types of the policy being enforced. If the scopes do not match, an InvalidScope exception will be raised. If False , a message will be logged informing operators that policies are being invoked with mismatching scope.
policy_default_rule = default	string value	Default rule. Enforced when a requested rule is not found.
policy_dirs = ['policy.d']	multi valued	Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the <code>config_dir</code> option, or absolute paths. The file defined by <code>policy_file</code> must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = policy.yaml	string value	The relative or absolute path of a file that maps roles to permissions for a given service. Relative paths must be specified in relation to the configuration file setting this option.
remote_content_type = application/x-www-form-urlencoded	string value	Content Type to send and receive data for REST based policy check
remote_ssl_ca_cert_file = None	string value	Absolute path to ca cert file for REST based policy check
remote_ssl_client_cert_file = None	string value	Absolute path to client cert for REST based policy check
remote_ssl_client_key_file = None	string value	Absolute path client key file REST based policy check

Configuration option = Default value	Type	Description
remote_ssl_verify_server _cert = False	boolean value	server identity verification for REST based policy check

1.1.16. p11_crypto_plugin

The following table outlines the options available under the **[p11_crypto_plugin]** group in the **/etc/barbican/barbican.conf** file.

Table 1.15. p11_crypto_plugin

Configuration option = Default value	Type	Description
aes_gcm_generate_iv = True	boolean value	Generate IVs for CKM_AES_GCM mechanism.
always_set_cka_sensitive = True	boolean value	Always set CKA_SENSITIVE=CK_TRUE including CKA_EXTRACTABLE=CK_TRUE keys.
encryption_mechanism = CKM_AES_CBC	string value	Secret encryption mechanism
hmac_key_type = CKK_AES	string value	HMAC Key Type
hmac_keygen_mechanism = CKM_AES_KEY_GEN	string value	HMAC Key Generation Algorithm
hmac_keywrap_mechanism = CKM_SHA256_HMAC	string value	HMAC key wrap mechanism
hmac_label = None	string value	Master HMAC Key label (as stored in the HSM)
library_path = None	string value	Path to vendor PKCS11 library
login = None	string value	Password to login to PKCS11 session
mkek_label = None	string value	Master KEK label (as stored in the HSM)
mkek_length = None	integer value	Master KEK length in bytes.
os_locking_ok = False	boolean value	Enable CKF_OS_LOCKING_OK flag when initializing the PKCS#11 client library.
pkek_cache_limit = 100	integer value	Project KEK Cache Item Limit

Configuration option = Default value	Type	Description
pkek_cache_ttl = 900	integer value	Project KEK Cache Time To Live, in seconds
pkek_length = 32	integer value	Project KEK length in bytes.
plugin_name = PKCS11 HSM	string value	User friendly plugin name
rw_session = True	boolean value	Flag for Read/Write Sessions
seed_file = `	string value	File to pull entropy for seeding RNG
seed_length = 32	integer value	Amount of data to read from file for seed
slot_id = 1	integer value	(Optional) HSM Slot ID that contains the token device to be used.
token_label = None	string value	DEPRECATED: Use token_labels instead. Token label used to identify the token to be used.
token_labels = None	list value	List of labels for one or more tokens to be used. Typically this is a single label, but some HSM devices may require more than one label for Load Balancing or High Availability configurations.
token_serial_number = None	string value	Token serial number used to identify the token to be used.

1.1.17. queue

The following table outlines the options available under the **[queue]** group in the **/etc/barbican/barbican.conf** file.

Table 1.16. queue

Configuration option = Default value	Type	Description
asynchronous_workers = 1	integer value	Number of asynchronous worker processes
enable = False	boolean value	True enables queuing, False invokes workers synchronously
namespace = barbican	string value	Queue namespace

Configuration option = Default value	Type	Description
server_name = barbican.queue	string value	Server name for RPC task processing server
topic = barbican.workers	string value	Queue topic name
version = 1.1	string value	Version of tasks invoked via queue

1.1.18. quotas

The following table outlines the options available under the **[quotas]** group in the **/etc/barbican/barbican.conf** file.

Table 1.17. quotas

Configuration option = Default value	Type	Description
quota_cas = -1	integer value	Number of CAs allowed per project
quota_consumers = -1	integer value	Number of consumers allowed per project
quota_containers = -1	integer value	Number of containers allowed per project
quota_orders = -1	integer value	Number of orders allowed per project
quota_secrets = -1	integer value	Number of secrets allowed per project

1.1.19. retry_scheduler

The following table outlines the options available under the **[retry_scheduler]** group in the **/etc/barbican/barbican.conf** file.

Table 1.18. retry_scheduler

Configuration option = Default value	Type	Description
initial_delay_seconds = 10.0	floating point value	Seconds (float) to wait before starting retry scheduler
periodic_interval_max_se conds = 10.0	floating point value	Seconds (float) to wait between periodic schedule events

1.1.20. secretstore

The following table outlines the options available under the **[secretstore]** group in the **/etc/barbican/barbican.conf** file.

Table 1.19. secretstore

Configuration option = Default value	Type	Description
enable_multiple_secret_stores = False	boolean value	Flag to enable multiple secret store plugin backend support. Default is False
enabled_secretstore_plugins = ['store_crypto']	multi valued	List of secret store plugins to load.
namespace = barbican.secretstore.plugin	string value	Extension namespace to search for plugins.
stores_lookup_suffix = None	list value	List of suffix to use for looking up plugins which are supported with multiple backend support.

1.1.21. simple_crypto_plugin

The following table outlines the options available under the **[simple_crypto_plugin]** group in the **/etc/barbican/barbican.conf** file.

Table 1.20. simple_crypto_plugin

Configuration option = Default value	Type	Description
kek = dGhpcnR5X3R3b19ieXRI X2tleWJsYWhibGFoYmxh aGg=	string value	Key encryption key to be used by Simple Crypto Plugin
plugin_name = Software Only Crypto	string value	User friendly plugin name

1.1.22. snakeoil_ca_plugin

The following table outlines the options available under the **[snakeoil_ca_plugin]** group in the **/etc/barbican/barbican.conf** file.

Table 1.21. snakeoil_ca_plugin

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
ca_cert_chain_path = None	string value	Path to CA certificate chain file
ca_cert_key_path = None	string value	Path to CA certificate key file
ca_cert_path = None	string value	Path to CA certificate file
ca_cert_pkcs7_path = None	string value	Path to CA chain pkcs7 file
subca_cert_key_directory = /etc/barbican/snakeoil-cas	string value	Directory in which to store certs/keys for subcas

1.1.23. ssl

The following table outlines the options available under the **[ssl]** group in the **/etc/barbican/barbican.conf** file.

Table 1.22. ssl

Configuration option = Default value	Type	Description
ca_file = None	string value	CA certificate file to use to verify connecting clients.
cert_file = None	string value	Certificate file to use when starting the server securely.
ciphers = None	string value	Sets the list of available ciphers. value should be a string in the OpenSSL cipher list format.
key_file = None	string value	Private key file to use when starting the server securely.
version = None	string value	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.

CHAPTER 2. CINDER

The following chapter contains information about the configuration options in the **cinder** service.

2.1. CINDER.CONF

This section contains options for the `/etc/cinder/cinder.conf` file.

2.1.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/cinder/cinder.conf` file.

Configuration option = Default value	Type	Description
acs5000_copy_interval = 5	integer value	When volume copy task is going on,refresh volume status interval
acs5000_volpool_name = ['pool01']	list value	Comma separated list of storage system storage pools for volumes.
allocated_capacity_weight_multiplier = -1.0	floating point value	Multiplier used for weighing allocated capacity. Positive numbers mean to stack vs spread.
allow_availability_zone_fallback = False	boolean value	If the requested Cinder availability zone is unavailable, fall back to the value of <code>default_availability_zone</code> , then <code>storage_availability_zone</code> , instead of failing.
allow_compression_on_image_upload = False	boolean value	The strategy to use for image compression on upload. Default is disallow compression.
allowed_direct_url_schemes = []	list value	A list of url schemes that can be downloaded directly via the <code>direct_url</code> . Currently supported schemes: [file, cinder].
api_paste_config = api-paste.ini	string value	File name for the paste.deploy config for api service
api_rate_limit = True	boolean value	Enables or disables rate limit of the API.
as13000_ipsan_pools = ['Pool0']	list value	The Storage Pools Cinder should use, a comma separated list.

Configuration option = Default value	Type	Description
as13000_meta_pool = None	string value	The pool which is used as a meta pool when creating a volume, and it should be a replication pool at present. If not set, the driver will choose a replication pool from the value of <code>as13000_ipsan_pools</code> .
as13000_token_available_time = 3300	integer value	The effective time of token validity in seconds.
auth_strategy = keystone	string value	The strategy to use for auth. Supports <code>noauth</code> or <code>keystone</code> .
az_cache_duration = 3600	integer value	Cache volume availability zones in memory for the provided duration in seconds
backdoor_port = None	string value	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.
backdoor_socket = None	string value	Enable eventlet backdoor, using the provided path as a unix socket that can receive connections. This option is mutually exclusive with <code>backdoor_port</code> in that only one should be provided. If both are provided then the existence of this option overrides the usage of that option. Inside the path {pid} will be replaced with the PID of the current process.
backend_availability_zone = None	string value	Availability zone for this volume backend. If not set, the <code>storage_availability_zone</code> option value is used as the default for all backends.
backend_stats_polling_interval = 60	integer value	Time in seconds between requests for usage statistics from the backend. Be aware that generating usage statistics is expensive for some backends, so setting this value too low may adversely affect performance.
backup_api_class = cinder.backup.api.API	string value	The full class name of the volume backup API class
backup_ceph_chunk_size = 134217728	integer value	The chunk size, in bytes, that a backup is broken into before transfer to the Ceph object store.

Configuration option = Default value	Type	Description
backup_ceph_conf = /etc/ceph/ceph.conf	string value	Ceph configuration file to use.
backup_ceph_image_journals = False	boolean value	If True, apply JOURNALING and EXCLUSIVE_LOCK feature bits to the backup RBD objects to allow mirroring
backup_ceph_pool = backups	string value	The Ceph pool where volume backups are stored.
backup_ceph_stripe_count = 0	integer value	RBD stripe count to use when creating a backup image.
backup_ceph_stripe_unit = 0	integer value	RBD stripe unit to use when creating a backup image.
backup_ceph_user = cinder	string value	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.
backup_compression_algorithm = zlib	string value	Compression algorithm ("none" to disable)
backup_container = None	string value	Custom directory to use for backups.
backup_driver = cinder.backup.drivers.swift.SwiftBackupDriver	string value	Driver to use for backups.
backup_driver_init_check_interval = 60	integer value	Time in seconds between checks to see if the backup driver has been successfully initialized, any time the driver is restarted.
backup_driver_stats_polling_interval = 60	integer value	Time in seconds between checks of the backup driver status. If does not report as working, it is restarted.
backup_enable_progress_timer = True	boolean value	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	integer value	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.

Configuration option = Default value	Type	Description
backup_manager = cinder.backup.manager.B ackupManager	string value	Full class name for the Manager for volume backup
backup_max_operations = 15	integer value	Maximum number of concurrent memory heavy operations: backup and restore. Value of 0 means unlimited
backup_metadata_versio n = 2	integer value	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_mount_attempts = 3	integer value	The number of attempts to mount NFS shares before raising an error.
backup_mount_options = None	string value	Mount options passed to the NFS client. See NFS man page for details.
backup_mount_point_bas e = \$state_path/backup_mou nt	string value	Base dir containing mount point for NFS share.
backup_name_template = backup-%s	string value	Template string to be used to generate backup names
backup_native_threads_p ool_size = 60	integer value	Size of the native threads pool for the backups. Most backup drivers rely heavily on this, it can be decreased for specific drivers that don't.
backup_object_number_p er_notification = 10	integer value	The number of chunks or objects, for which one Ceilometer notification will be sent
backup_posix_path = \$state_path/backup	string value	Path specifying where to store backups.
backup_s3_block_size = 32768	integer value	The size in bytes that changes are tracked for incremental backups. backup_s3_object_size has to be multiple of backup_s3_block_size.
backup_s3_ca_cert_file = None	string value	path/to/cert/bundle.pem - A filename of the CA cert bundle to use.

Configuration option = Default value	Type	Description
backup_s3_enable_progress_timer = True	boolean value	Enable or Disable the timer to send the periodic progress notifications to Ceilometer when backing up the volume to the S3 backend storage. The default value is True to enable the timer.
backup_s3_endpoint_url = None	string value	The url where the S3 server is listening.
<code>`backup_s3_http_proxy = `</code>	string value	Address or host for the http proxy server.
<code>`backup_s3_https_proxy = `</code>	string value	Address or host for the https proxy server.
backup_s3_max_pool_connections = 10	integer value	The maximum number of connections to keep in a connection pool.
backup_s3_md5_validation = True	boolean value	Enable or Disable md5 validation in the s3 backend.
backup_s3_object_size = 52428800	integer value	The size in bytes of S3 backup objects
backup_s3_retry_max_attempts = 4	integer value	An integer representing the maximum number of retry attempts that will be made on a single request.
backup_s3_retry_mode = legacy	string value	A string representing the type of retry mode. e.g: legacy, standard, adaptive
backup_s3_sse_customer_algorithm = None	string value	The SSECustomerAlgorithm. backup_s3_sse_customer_key must be set at the same time to enable SSE.
backup_s3_sse_customer_key = None	string value	The SSECustomerKey. backup_s3_sse_customer_algorithm must be set at the same time to enable SSE.
backup_s3_store_access_key = None	string value	The S3 query token access key.
backup_s3_store_bucket = volumebackups	string value	The S3 bucket to be used to store the Cinder backup data.
backup_s3_store_secret_key = None	string value	The S3 query token secret key.

Configuration option = Default value	Type	Description
backup_s3_timeout = 60	floating point value	The time in seconds till a timeout exception is thrown.
backup_s3_verify_ssl = True	boolean value	Enable or Disable ssl verify.
backup_service_inithost_offload = True	boolean value	Offload pending backup delete during backup service startup. If false, the backup service will remain down until all pending backups are deleted.
backup_sha_block_size_bytes = 32768	integer value	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	string value	NFS share in hostname:path, ipv4addr:path, or "[ipv6addr]:path" format.
backup_swift_auth = per_user	string value	Swift authentication mechanism (per_user or single_user).
backup_swift_auth_insecure = False	boolean value	Bypass verification of server certificate when making SSL connection to Swift.
backup_swift_auth_url = None	uri value	The URL of the Keystone endpoint
backup_swift_auth_version = 1	string value	Swift authentication version. Specify "1" for auth 1.0, or "2" for auth 2.0 or "3" for auth 3.0
backup_swift_block_size = 32768	integer value	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_ca_cert_file = None	string value	Location of the CA certificate file to use for swift client requests.
backup_swift_container = volumebackups	string value	The default Swift container to use
backup_swift_enable_progress_timer = True	boolean value	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	string value	Swift key for authentication

Configuration option = Default value	Type	Description
backup_swift_object_size = 52428800	integer value	The size in bytes of Swift backup objects
backup_swift_project = None	string value	Swift project/account name. Required when connecting to an auth 3.0 system
backup_swift_project_domain = None	string value	Swift project domain name. Required when connecting to an auth 3.0 system
backup_swift_retry_attempts = 3	integer value	The number of retries to make for Swift operations
backup_swift_retry_backoff = 2	integer value	The backoff time in seconds between Swift retries
backup_swift_tenant = None	string value	Swift tenant/account name. Required when connecting to an auth 2.0 system
backup_swift_url = None	uri value	The URL of the Swift endpoint
backup_swift_user = None	string value	Swift user name
backup_swift_user_domain = None	string value	Swift user domain name. Required when connecting to an auth 3.0 system
backup_timer_interval = 120	integer value	Interval, in seconds, between two progress notifications reporting the backup status
backup_use_same_host = False	boolean value	Backup services use same backend.
backup_use_temp_snapshot_hot = False	boolean value	If this is set to True, a temporary snapshot will be created for performing non-disruptive backups. Otherwise a temporary volume will be cloned in order to perform a backup.
backup_workers = 1	integer value	Number of backup processes to launch. Improves performance with concurrent backups.
capacity_weight_multiplier = 1.0	floating point value	Multiplier used for weighing free capacity. Negative numbers mean to stack vs spread.
`chap_password = `	string value	Password for specified CHAP account name.

Configuration option = Default value	Type	Description
chap_password_len = 12	integer value	Length of the random string for CHAP password.
<code>`chap_username = `</code>	string value	CHAP user name.
chiscsi_conf = /etc/chelsio-iscsi/chiscsi.conf	string value	Chiscsi (CXT) global defaults configuration file
cinder_internal_tenant_project_id = None	string value	ID of the project which will be used as the Cinder internal tenant.
cinder_internal_tenant_user_id = None	string value	ID of the user to be used in volume operations as the Cinder internal tenant.
client_socket_timeout = 900	integer value	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.
clone_volume_timeout = 680	integer value	Create clone volume timeout Deprecated since: 14.0.0 *Reason:*FusionStorage cinder driver refactored the code with Restful method and the old CLI mode has been abandon. So those configuration items are no longer used.
cloned_volume_same_az = True	boolean value	Ensure that the new volumes are the same AZ as snapshot or source volume
cluster = None	string value	Name of this cluster. Used to group volume hosts that share the same backend configurations to work in HA Active-Active mode.
compression_format = gzip	string value	Image compression format on image upload
compute_api_class = cinder.compute.nova.API	string value	The full class name of the compute API class to use
config-dir = ['~/project/project.conf.d/ '; '~/project.conf.d', '/etc/project/project.conf.d', '/etc/project.conf.d']	list value	Path to a config directory to pull *.conf files from. This file set is sorted, so as to provide a predictable parse order if individual options are over-ridden. The set is parsed after the file(s) specified via previous --config-file, arguments hence over-ridden options in the directory take precedence. This option must be set from the command-line.

Configuration option = Default value	Type	Description
config-file = ['~/project/project.conf', '~/project.conf', '/etc/project/project.conf', '/etc/project.conf']	unknown value	Path to a config file to use. Multiple config files can be specified, with values in later files taking precedence. Defaults to %(default)s. This option must be set from the command-line.
config_source = []	list value	Lists configuration groups that provide more details for accessing configuration settings from locations other than local files.
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
consistencygroup_api_class = cinder.consistencygroup. api.API	string value	The full class name of the consistencygroup API class
control_exchange = openstack	string value	The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.
datera_503_interval = 5	integer value	Interval between 503 retries
datera_503_timeout = 120	integer value	Timeout for HTTP 503 retry messages
datera_api_port = 7717	string value	Datera API port.
datera_api_version = 2.2	string value	Datera API version.
datera_debug = False	boolean value	True to set function arg and return logging
datera_debug_replica_count_override = False	boolean value	ONLY FOR DEBUG/TESTING PURPOSES True to set replica_count to 1
datera_disable_extended_metadata = False	boolean value	Set to True to disable sending additional metadata to the Datera backend
datera_disable_profiler = False	boolean value	Set to True to disable profiling in the Datera driver
datera_disable_template_override = False	boolean value	Set to True to disable automatic template override of the size attribute when creating from a template

Configuration option = Default value	Type	Description
datera_enable_image_cache = False	boolean value	Set to True to enable Datera backend image caching
datera_image_cache_volume_type_id = None	string value	Cinder volume type id to use for cached volumes
datera_ldap_server = None	string value	LDAP authentication server
datera_tenant_id = None	string value	If set to <i>Map</i> --> OpenStack project ID will be mapped implicitly to Datera tenant ID If set to None --> Datera tenant ID will not be used during volume provisioning If set to anything else --> Datera tenant ID will be the provided value
datera_volume_type_defaults = {}	dict value	Settings here will be used as volume-type defaults if the volume-type setting is not provided. This can be used, for example, to set a very low total_iops_max value if none is specified in the volume-type to prevent accidental overusage. Options are specified via the following format, WITHOUT ANY DF: PREFIX: <i>datera_volume_type_defaults=iops_per_gb:100,bandwidth_per_gb:200...etc.</i>
db_driver = cinder.db	string value	Driver to use for database access
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.
default_availability_zone = None	string value	Default availability zone for new volumes. If not set, the storage_availability_zone option value is used as the default for new volumes.
default_group_type = None	string value	Default group type to use

Configuration option = Default value	Type	Description
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
default_sandstone_target _ips = []	list value	SandStone default target ip.
default_volume_type = __DEFAULT__	string value	Default volume type to use
driver_client_cert = None	string value	The path to the client certificate for verification, if the driver supports it.
driver_client_cert_key = None	string value	The path to the client certificate key for verification, if the driver supports it.
driver_data_namespace = None	string value	Namespace for driver private data values to be saved in.
driver_ssl_cert_path = None	string value	Can be used to specify a non default path to a CA_BUNDLE file or directory with certificates of trusted CAs, which will be used to validate the backend

Configuration option = Default value	Type	Description
driver_ssl_cert_verify = False	boolean value	If set to True the http client will validate the SSL certificate of the backend endpoint.
driver_use_ssl = False	boolean value	Tell driver to use SSL for connection to backend storage if the driver supports it.
dsware_isthin = False	boolean value	The flag of thin storage allocation. Deprecated since: 14.0.0 *Reason:*FusionStorage cinder driver refactored the code with Restful method and the old CLI mode has been abandon. So those configuration items are no longer used.
<code>`dsware_manager = `</code>	string value	Fusionstorage manager ip addr for cinder-volume. Deprecated since: 14.0.0 *Reason:*FusionStorage cinder driver refactored the code with Restful method and the old CLI mode has been abandon. So those configuration items are no longer used.
<code>`dsware_rest_url = `</code>	string value	The address of FusionStorage array. For example, "dsware_rest_url=xxx"
<code>`dsware_storage_pools = `</code>	string value	The list of pools on the FusionStorage array, the semicolon(;) was used to split the storage pools, "dsware_storage_pools = xxx1; xxx2; xxx3"
enable_force_upload = False	boolean value	Enables the Force option on upload_to_image. This enables running upload_volume on in-use volumes for backends that support it.
enable_new_services = True	boolean value	Services to be added to the available pool on create
enable_unsupported_driver = False	boolean value	Set this to True when you want to allow an unsupported driver to start. Drivers that haven't maintained a working CI system and testing are marked as unsupported until CI is working again. This also marks a driver as deprecated and may be removed in the next release.
enable_v2_api = True	boolean value	DEPRECATED: Deploy v2 of the Cinder API.

Configuration option = Default value	Type	Description
enable_v3_api = True	boolean value	Deploy v3 of the Cinder API.
enabled_backends = None	list value	A list of backend names to use. These backend names should be backed by a unique [CONFIG] group with its options
enforce_multipath_for_image_xfer = False	boolean value	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. This parameter needs to be configured for each backend section or in [backend_defaults] section as a common configuration for all backends.
executor_thread_pool_size = 64	integer value	Size of executor thread pool when executor is threading or eventlet.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
filter_function = None	string value	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.
<code>`fusionstorageagent = `</code>	string value	Fusionstorage agent ip addr range Deprecated since: 14.0.0 *Reason:*FusionStorage cinder driver refactored the code with Restful method and the old CLI mode has been abandon. So those configuration items are no longer used.
glance_api_insecure = False	boolean value	Allow to perform insecure SSL (https) requests to glance (https will be used but cert validation will not be performed).
glance_api_servers = None	list value	A list of the URLs of glance API servers available to cinder ([http[s]://][hostname ip]:port). If protocol is not specified it defaults to http.
glance_api_ssl_compression = False	boolean value	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_ca_certificates_file = None	string value	Location of ca certificates file to use for glance client requests.

Configuration option = Default value	Type	Description
glance_catalog_info = image:glance:publicURL	string value	Info to match when looking for glance in the service catalog. Format is: separated values of the form: <service_type>:<service_name>:<endpoint_type> - Only used if glance_api_servers are not provided.
glance_certfile = None	string value	Location of certificate 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']	list value	Default core properties of image
glance_keyfile = None	string value	Location of certificate key file to use for glance client requests.
glance_num_retries = 3	integer value	Number retries when downloading an image from glance
glance_request_timeout = None	integer value	http/https timeout value for glance operations. If no value (None) is supplied here, the glanceclient default value is used.
glusterfs_backup_mount _point = \$state_path/backup_mou nt	string value	Base dir containing mount point for gluster share.
glusterfs_backup_share = None	string value	GlusterFS share in <hostname ipv4addr ipv6addr>:<gluster_vol_name> format. Eg: 1.2.3.4:backup_vol
goodness_function = None	string value	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.
graceful_shutdown_timeo ut = 60	integer value	Specify a timeout after which a gracefully shutdown server will exit. Zero value means endless wait.
group_api_class = cinder.group.api.API	string value	The full class name of the group API class

Configuration option = Default value	Type	Description
host = <based on operating system>	string value	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	string value	DEPRECATED: IET configuration file
image_compress_on_upload = True	boolean value	When possible, compress images uploaded to the image service
image_conversion_address_space_limit = 1	integer value	Address space limit in gigabytes to convert the image
image_conversion_cpu_limit = 60	integer value	CPU time limit in seconds to convert the image
image_conversion_dir = \$state_path/conversion	string value	Directory used for temporary storage during image conversion
image_upload_use_cinder_backend = False	boolean value	If set to True, upload-to-image in raw format will create a cloned volume and register its location to the image service, instead of uploading the volume content. The cinder backend and locations support must be enabled in the image service.
image_upload_use_internal_tenant = False	boolean value	If set to True, the image volume created by upload-to-image will be placed in the internal tenant. Otherwise, the image volume is created in the current context's tenant.
image_volume_cache_enabled = False	boolean value	Enable the image volume cache for this backend.
image_volume_cache_max_count = 0	integer value	Max number of entries allowed in the image volume cache. 0 ⇒ unlimited.
image_volume_cache_max_size_gb = 0	integer value	Max size of the image volume cache for this backend in GB. 0 ⇒ unlimited.
infortrend_cli_cache = False	boolean value	The Infortrend CLI cache. While set True, the RAID status report will use cache stored in the CLI. Never enable this unless the RAID is managed only by Openstack and only by one infortrend cinder-volume backend. Otherwise, CLI might report out-dated status to cinder and thus there might be some race condition among all backend/CLIs.

Configuration option = Default value	Type	Description
infortrend_cli_max_retries = 5	integer value	The maximum retry times if a command fails.
infortrend_cli_path = /opt/bin/Infortrend/raidcmd_ESDS10.jar	string value	The Infortrend CLI absolute path.
infortrend_cli_timeout = 60	integer value	The timeout for CLI in seconds.
infortrend_iqn_prefix = iqn.2002-10.com.infortrend	string value	Infortrend iqn prefix for iSCSI.
`infortrend_pools_name = `	list value	The Infortrend logical volumes name list. It is separated with comma.
`infortrend_slots_a_channels_id = `	list value	Infortrend raid channel ID list on Slot A for OpenStack usage. It is separated with comma.
`infortrend_slots_b_channels_id = `	list value	Infortrend raid channel ID list on Slot B for OpenStack usage. It is separated with comma.
init_host_max_objects_retrieval = 0	integer value	Max number of volumes and snapshots to be retrieved per batch during volume manager host initialization. Query results will be obtained in batches from the database and not in one shot to avoid extreme memory usage. Set 0 to turn off this functionality.
initiator_assign_sandstone_target_ip = {}	dict value	Support initiator assign target with assign ip.
`instance_format = [instance: %(uuid)s] `	string value	The format for an instance that is passed with the log message.
`instance_uuid_format = [instance: %(uuid)s] `	string value	The format for an instance UUID that is passed with the log message.
instorage_mcs_allow_tenant_qos = False	boolean value	Allow tenants to specify QOS on create
instorage_mcs_iscsi_chap_enabled = True	boolean value	Configure CHAP authentication for iSCSI connections (Default: Enabled)

Configuration option = Default value	Type	Description
instorage_mcs_localcopy_rate = 50	integer value	Specifies the InStorage LocalCopy copy rate to be used when creating a full volume copy. The default rate is 50, and the valid rates are 1-100.
instorage_mcs_localcopy_timeout = 120	integer value	Maximum number of seconds to wait for LocalCopy to be prepared.
instorage_mcs_vol_autoexpand = True	boolean value	Storage system autoexpand parameter for volumes (True/False)
instorage_mcs_vol_compression = False	boolean value	Storage system compression option for volumes
instorage_mcs_vol_grain_size = 256	integer value	Storage system grain size parameter for volumes (32/64/128/256)
instorage_mcs_vol_intier = True	boolean value	Enable InTier for volumes
instorage_mcs_vol_iogrp = 0	string value	The I/O group in which to allocate volumes. It can be a comma-separated list in which case the driver will select an io_group based on least number of volumes associated with the io_group.
instorage_mcs_vol_rsize = 2	integer value	Storage system space-efficiency parameter for volumes (percentage)
instorage_mcs_vol_warning = 0	integer value	Storage system threshold for volume capacity warnings (percentage)
instorage_mcs_volpool_name = ['volpool']	list value	Comma separated list of storage system storage pools for volumes.
instorage_san_secondary_ip = None	string value	Specifies secondary management IP or hostname to be used if san_ip is invalid or becomes inaccessible.
iscsi_iotype = fileio	string value	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_secondary_ip_addresses = []	list value	The list of secondary IP addresses of the iSCSI daemon
`iscsi_target_flags = `	string value	Sets the target-specific flags for the iSCSI target. Only used for tgtadm to specify backing device flags using bsflags option. The specified string is passed as is to the underlying tool.

Configuration option = Default value	Type	Description
iscsi_write_cache = on	string value	Sets the behavior of the iSCSI target to either perform write-back(on) or write-through(off). This parameter is valid if target_helper is set to tgtadm.
iser_helper = tgtadm	string value	The name of the iSER target user-land tool to use
iser_ip_address = \$my_ip	string value	The IP address that the iSER daemon is listening on
iser_port = 3260	port value	The port that the iSER daemon is listening on
iser_target_prefix = iqn.2010- 10.org.openstack:	string value	Prefix for iSER volumes
java_path = /usr/bin/java	string value	The Java absolute path.
jovian_block_size = 64K	string value	Block size can be: 32K, 64K, 128K, 256K, 512K, 1M
jovian_ignore_tpath = []	list value	List of multipath ip addresses to ignore.
jovian_pool = Pool-0	string value	JovianDSS pool that holds all cinder volumes
jovian_recovery_delay = 60	integer value	Time before HA cluster failure.
keystone_catalog_info = identity:Identity Service:publicURL	string value	Info to match when looking for keystone in the service catalog. Format is: separated values of the form: <service_type>:<service_name>:<endpoint_type> - Only used if backup_swift_auth_url is unset
kioxia_block_size = 4096	integer value	Volume block size in bytes - 512 or 4096 (Default).
kioxia_cafile = None	string value	Cert for provisioner REST API SSL
kioxia_desired_bw_per_gb = 0	integer value	Desired bandwidth in B/s per GB.
kioxia_desired_iops_per_gb = 0	integer value	Desired IOPS/GB.
kioxia_max_bw_per_gb = 0	integer value	Upper limit for bandwidth in B/s per GB.

Configuration option = Default value	Type	Description
kioxia_max_iops_per_gb = 0	integer value	Upper limit for IOPS/GB.
kioxia_max_replica_down_time = 0	integer value	Replicated volume max downtime for replica in minutes.
kioxia_num_replicas = 1	integer value	Number of volume replicas.
kioxia_provisioning_type = THICK	string value	Thin or thick volume, Default thick.
kioxia_same_rack_allowed = False	boolean value	Can more than one replica be allocated to same rack.
kioxia_snap_reserved_space_percentage = 0	integer value	Percentage of the parent volume to be used for log.
kioxia_snap_vol_reserved_space_percentage = 0	integer value	Writable snapshot percentage of parent volume used for log.
kioxia_snap_vol_span_allowed = True	boolean value	Allow span in snapshot volume - Default True.
kioxia_span_allowed = True	boolean value	Allow span - Default True.
kioxia_token = None	string value	KumoScale Provisioner auth token.
kioxia_url = None	string value	KumoScale provisioner REST API URL
kioxia_vol_reserved_space_percentage = 0	integer value	Thin volume reserved capacity allocation percentage.
kioxia_writable = False	boolean value	Volumes from snapshot writeable or not.
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for %(asctime)s in log records. Default: %(default)s . This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
log-dir = None	string value	(Optional) The base directory used for relative log_file paths. This option is ignored if log_config_append is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to stderr as defined by use_stderr. This option is ignored if log_config_append is set.
log_options = True	boolean value	Enables or disables logging values of all registered options when starting a service (at DEBUG level).
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless log_rotation_type is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.% (msecs)03d %(process)d %(levelname)s %(name)s [% (request_id)s % (user_identity)s] % (instance)s%(message)s	string value	Format string to use for log messages with context. Used by oslo_log.formatters.ContextFormatter
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is DEBUG. Used by oslo_log.formatters.ContextFormatter
logging_default_format_s tring = %(asctime)s.% (msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s% (message)s	string value	Format string to use for log messages when context is undefined. Used by oslo_log.formatters.ContextFormatter

Configuration option = Default value	Type	Description
logging_exception_prefix = <code>%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s</code>	string value	Prefix each line of exception output with this format. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_user_identity_format = <code>%(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s</code>	string value	Defines the format string for <code>%(user_identity)s</code> that is used in <code>logging_context_format_string</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
manager_ips = <code>{}</code>	dict value	This option is to support the FSA to mount across the different nodes. The parameters takes the standard dict config form, <code>manager_ips = host1:ip1, host2:ip2...</code>
max_age = 0	integer value	Number of seconds between subsequent usage refreshes
max_header_line = 16384	integer value	Maximum line size of message headers to be accepted. <code>max_header_line</code> may need to be increased when using large tokens (typically those generated when keystone is configured to use PKI tokens with big service catalogs).
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
max_over_subscription_ratio = 20.0	string value	Representation of the over subscription ratio when thin provisioning is enabled. 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. If ratio is <i>auto</i> , Cinder will automatically calculate the ratio based on the provisioned capacity and the used space. If not set to <i>auto</i> , the ratio has to be a minimum of 1.0.
message_reap_interval = 86400	integer value	interval between periodic task runs to clean expired messages in seconds.
message_ttl = 2592000	integer value	message minimum life in seconds.

Configuration option = Default value	Type	Description
migration_create_volume_timeout_secs = 300	integer value	Timeout for creating the volume to migrate to when performing volume migration (seconds)
monkey_patch = False	boolean value	Enable monkey patching
monkey_patch_modules = []	list value	List of modules/decorators to monkey patch
my_ip = <based on operating system>	host address value	IP address of this host
no_snapshot_gb_quota = False	boolean value	Whether snapshots count against gigabyte quota
num_iscsi_scan_tries = 3	integer value	The maximum number of times to rescan iSER target to find volume
num_shell_tries = 3	integer value	Number of times to attempt to run flakey shell commands
num_volume_device_scan_tries = 3	integer value	The maximum number of times to rescan targets to find volume
nvmet_ns_id = 10	integer value	The namespace id associated with the subsystem that will be created with the path for the LVM volume.
nvmet_port_id = 1	port value	The port that the NVMe target is listening on.
osapi_max_limit = 1000	integer value	The maximum number of items that a collection resource returns in a single response
osapi_volume_ext_list = []	list value	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']	multi valued	osapi volume extension to load
osapi_volume_listen = 0.0.0.0	string value	IP address on which OpenStack Volume API listens
osapi_volume_listen_port = 8776	port value	Port on which OpenStack Volume API listens

Configuration option = Default value	Type	Description
osapi_volume_use_ssl = False	boolean value	Wraps the socket in a SSL context if True is set. A certificate file and key file must be specified.
osapi_volume_workers = None	integer value	Number of workers for OpenStack Volume API service. The default is equal to the number of CPUs available.
per_volume_size_limit = -1	integer value	Max size allowed per volume, in gigabytes
periodic_fuzzy_delay = 60	integer value	Range, in seconds, to randomly delay when starting the periodic task scheduler to reduce stampeding. (Disable by setting to 0)
periodic_interval = 60	integer value	Interval, in seconds, between running periodic tasks
pool_id_filter = []	list value	Pool id permit to use Deprecated since: 14.0.0 *Reason:*FusionStorage cinder driver refactored the code with Restful method and the old CLI mode has been abandon. So those configuration items are no longer used.
pool_type = default	string value	Pool type, like sata-2copy Deprecated since: 14.0.0 *Reason:*FusionStorage cinder driver refactored the code with Restful method and the old CLI mode has been abandon. So those configuration items are no longer used.
public_endpoint = None	string value	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.
publish_errors = False	boolean value	Enables or disables publication of error events.
quota_backup_gigabytes = 1000	integer value	Total amount of storage, in gigabytes, allowed for backups per project
quota_backups = 10	integer value	Number of volume backups allowed per project
quota_consistencygroups = 10	integer value	Number of consistencygroups allowed per project

Configuration option = Default value	Type	Description
quota_driver = cinder.quota.DbQuotaDriver	string value	Default driver to use for quota checks
quota_gigabytes = 1000	integer value	Total amount of storage, in gigabytes, allowed for volumes and snapshots per project
quota_groups = 10	integer value	Number of groups allowed per project
quota_snapshots = 10	integer value	Number of volume snapshots allowed per project
quota_volumes = 10	integer value	Number of volumes allowed per project
rate_limit_burst = 0	integer value	Maximum number of logged messages per rate_limit_interval.
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
reinit_driver_count = 3	integer value	Maximum times to reinitialize the driver if volume initialization fails. The interval of retry is exponentially backoff, and will be 1s, 2s, 4s etc.
replication_device = None	dict value	Multi opt of dictionaries to represent a replication target device. This option may be specified multiple times in a single config section to specify multiple replication target devices. Each entry takes the standard dict config form: replication_device = target_device_id:<required>,key1:value1,key2:value2...
report_discard_supported = False	boolean value	Report to clients of Cinder that the backend supports discard (aka. trim/unmap). This will not actually change the behavior of the backend or the client directly, it will only notify that it can be used.
report_interval = 10	integer value	Interval, in seconds, between nodes reporting state to datastore
reservation_clean_interval = \$reservation_expire	integer value	Interval between periodic task runs to clean expired reservations in seconds.

Configuration option = Default value	Type	Description
reservation_expire = 86400	integer value	Number of seconds until a reservation expires
reserved_percentage = 0	integer value	The percentage of backend capacity is reserved
resource_query_filters_file = /etc/cinder/resource_filters.json	string value	Json file indicating user visible filter parameters for list queries.
restore_discard_excess_bytes = True	boolean value	If True, always discard excess bytes when restoring volumes i.e. pad with zeroes.
rootwrap_config = /etc/cinder/rootwrap.conf	string value	Path to the rootwrap configuration file to use for running commands as root
rpc_conn_pool_size = 30	integer value	Size of RPC connection pool.
rpc_ping_enabled = False	boolean value	Add an endpoint to answer to ping calls. Endpoint is named oslo_rpc_server_ping
rpc_response_timeout = 60	integer value	Seconds to wait for a response from a call.
run_external_periodic_tasks = True	boolean value	Some periodic tasks can be run in a separate process. Should we run them here?
`san_hosts = `	list value	IP address of Open-E JovianDSS SA
`sandstone_pool = `	string value	SandStone storage pool resource name.
scheduler_default_filters = ['AvailabilityZoneFilter', 'CapacityFilter', 'CapabilitiesFilter']	list value	Which filter class names to use for filtering hosts when not specified in the request.
scheduler_default_weighters = ['CapacityWeighter']	list value	Which weighter class names to use for weighing hosts.
scheduler_driver = cinder.scheduler.filter_scheduler.FilterScheduler	string value	Default scheduler driver to use

Configuration option = Default value	Type	Description
scheduler_driver_init_wait_time = 60	integer value	Maximum time in seconds to wait for the driver to report as ready
scheduler_host_manager = cinder.scheduler.host_manager.HostManager	string value	The scheduler host manager class to use
^scheduler_json_config_location = ^	string value	Absolute path to scheduler configuration JSON file.
scheduler_manager = cinder.scheduler.manager.SchedulerManager	string value	Full class name for the Manager for scheduler
scheduler_max_attempts = 3	integer value	Maximum number of attempts to schedule a volume
scheduler_weight_handler = cinder.scheduler.weights.OrderedHostWeightHandler	string value	Which handler to use for selecting the host/pool after weighing
scst_target_driver = iscsi	string value	SCST target implementation can choose from multiple SCST target drivers.
scst_target_iqn_name = None	string value	Certain ISCSI targets have predefined target names, SCST target driver uses this name.
service_down_time = 60	integer value	Maximum time since last check-in for a service to be considered up
snapshot_name_template = snapshot-%s	string value	Template string to be used to generate snapshot names
snapshot_same_host = True	boolean value	Create volume from snapshot at the host where snapshot resides
split_loggers = False	boolean value	Log requests to multiple loggers.
ssh_hosts_key_file = \$state_path/ssh_known_hosts	string value	File containing SSH host keys for the systems with which Cinder needs to communicate. OPTIONAL: Default=\$state_path/ssh_known_hosts
state_path = /var/lib/cinder	string value	Top-level directory for maintaining cinder's state

Configuration option = Default value	Type	Description
storage_availability_zone = nova	string value	Availability zone of this node. Can be overridden per volume backend with the option "backend_availability_zone".
storage_protocol = iscsi	string value	Protocol for transferring data between host and storage back-end.
strict_ssh_host_key_policy = False	boolean value	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
swift_catalog_info = object-store:swift:publicURL	string value	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
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
target_helper = tgtadm	string value	Target user-land tool to use. tgtadm is default, use lioadm for LIO iSCSI support, scstadmin for SCST target support, ietadm for iSCSI Enterprise Target, iscsictl for Chelsio iSCSI Target, nvmet for NVMeoF support, spdk-nvmeof for SPDK NVMe-oF, or fake for testing. Note: The IET driver is deprecated and will be removed in the V release.
target_ip_address = \$my_ip	string value	The IP address that the iSCSI daemon is listening on
target_port = 3260	port value	The port that the iSCSI daemon is listening on
target_prefix = iqn.2010-10.org.openstack:	string value	Prefix for iSCSI volumes
target_protocol = iscsi	string value	Determines the target protocol for new volumes, created with tgtadm, lioadm and nvmet 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", in case of nvmet target set to "nvmet_rdma".
tcp_keepalive = True	boolean value	Sets the value of TCP_KEEPALIVE (True/False) for each server socket.

Configuration option = Default value	Type	Description
tcp_keepalive_count = None	integer value	Sets the value of TCP_KEEPCNT for each server socket. Not supported on OS X.
tcp_keepalive_interval = None	integer value	Sets the value of TCP_KEEPINTVL in seconds for each server socket. Not supported on OS X.
tcp_keepidle = 600	integer value	Sets the value of TCP_KEEPIDLE in seconds for each server socket. Not supported on OS X.
trace_flags = None	list value	List of options that control which trace info is written to the DEBUG log level to assist developers. Valid values are method and api.
transfer_api_class = cinder.transfer.api.API	string value	The full class name of the volume transfer API class
transport_url = rabbit://	string value	The network address and optional user credentials for connecting to the messaging backend, in URL format. The expected format is: <pre>driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query</pre> Example: rabbit://rabbitmq:password@127.0.0.1:5672// For full details on the fields in the URL see the documentation of oslo_messaging.TransportURL at https://docs.openstack.org/oslo.messaging/latest/reference/transport.html
until_refresh = 0	integer value	Count of reservations until usage is refreshed
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
use_chap_auth = False	boolean value	Option to enable/disable CHAP authentication for targets.
use_default_quota_class = True	boolean value	Enables or disables use of default quota class with default quota.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_forwarded_for = False	boolean value	Treat X-Forwarded-For as the canonical remote address. Only enable this if you have a sanitizing proxy.
use_multipath_for_image_xfer = False	boolean value	Do we attach/detach volumes in cinder using multipath for volume to image and image to volume transfers? This parameter needs to be configured for each backend section or in [backend_defaults] section as a common configuration for all backends.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
verify_glance_signatures = enabled	string value	Enable image signature verification. Cinder uses the image signature metadata from Glance and verifies the signature of a signed image while downloading that image. There are two options here. 1. enabled: verify when image has signature metadata. 2. disabled: verification is turned off. If the image signature cannot be verified or if the image signature metadata is incomplete when required, then Cinder will not create the volume and update it into an error state. This provides end users with stronger assurances of the integrity of the image data they are using to create volumes.
vmdk_allowed_types = ['streamOptimized', 'monolithicSparse']	list value	A list of strings describing the VMDK createType subformats that are allowed. We recommend that you only include single-file-with-sparse-header variants to avoid potential host file exposure when processing named extents when an image is converted to raw format as it is written to a volume. If this list is empty, no VMDK images are allowed.
volume_api_class = cinder.volume.api.API	string value	The full class name of the volume API class to use

Configuration option = Default value	Type	Description
volume_backend_name = None	string value	The backend name for a given driver implementation
volume_clear = zero	string value	Method used to wipe old volumes
volume_clear_ionice = None	string value	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	integer value	Size in MiB to wipe at start of old volumes. 1024 MiB at max. 0 ⇒ all
volume_copy_blkio_cgrou p_name = cinder- volume-copy	string value	The blkio cgroup name to be used to limit bandwidth of volume copy
volume_copy_bps_limit = 0	integer value	The upper limit of bandwidth of volume copy. 0 ⇒ unlimited
volume_dd_blocksize = 1M	string value	The default block size used when copying/clearing volumes
volume_manager = cinder.volume.manager.V olumeManager	string value	Full class name for the Manager for volume
volume_name_template = volume-%s	string value	Template string to be used to generate volume names
volume_number_multiplie r = -1.0	floating point value	Multiplier used for weighing volume number. Negative numbers mean to spread vs stack.
volume_service_inithost_ offload = False	boolean value	Offload pending volume delete during volume service startup
volume_transfer_key_len gth = 16	integer value	The number of characters in the autogenerated auth key.
volume_transfer_salt_len gth = 8	integer value	The number of characters in the salt.
volume_usage_audit_peri od = month	string value	Time period for which to generate volume usages. The options are hour, day, month, or year.
volumes_dir = \$state_path/volumes	string value	Volume configuration file storage directory

Configuration option = Default value	Type	Description
vrts_lun_sparse = True	boolean value	Create sparse Lun.
vrts_target_config = /etc/cinder/vrts_target.xml	string value	VA config file.
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.
wsgi_default_pool_size = 100	integer value	Size of the pool of greenthreads used by wsgi
wsgi_keep_alive = True	boolean value	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	string value	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.
wsgi_server_debug = False	boolean value	True if the server should send exception tracebacks to the clients on 500 errors. If False, the server will respond with empty bodies.
zoning_mode = None	string value	FC Zoning mode configured, only <i>fabric</i> is supported now.

2.1.2. backend

The following table outlines the options available under the **[backend]** group in the **/etc/cinder/cinder.conf** file.

Table 2.1. backend

Configuration option = Default value	Type	Description
backend_host = None	string value	Backend override of host value.

2.1.3. backend_defaults

The following table outlines the options available under the **[backend_defaults]** group in the **/etc/cinder/cinder.conf** file.

Table 2.2. backend_defaults

Configuration option = Default value	Type	Description
auto_calc_max_oversubscription_ratio = False	boolean value	K2 driver will calculate max_oversubscription_ratio on setting this option as True.
backend_availability_zone = None	string value	Availability zone for this volume backend. If not set, the storage_availability_zone option value is used as the default for all backends.
backend_native_threads_pool_size = 20	integer value	Size of the native threads pool for the backend. Increase for backends that heavily rely on this, like the RBD driver.
chap = disabled	string value	CHAP authentication mode, effective only for iscsi (disabled enabled)
<code>`chap_password = `</code>	string value	Password for specified CHAP account name.
<code>`chap_username = `</code>	string value	CHAP user name.
check_max_pool_luns_threshold = False	boolean value	DEPRECATED: Report free_capacity_gb as 0 when the limit to maximum number of pool LUNs is reached. By default, the value is False.
chiscsi_conf = /etc/chelsio-iscsi/chiscsi.conf	string value	Chiscsi (CXT) global defaults configuration file
cinder_eternus_config_file = /etc/cinder/cinder_fujitsu_eternus_dx.xml	string value	Config file for cinder eternus_dx volume driver.
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml	string value	The configuration file for the Cinder Huawei driver.
connection_type = iscsi	string value	Connection type to the IBM Storage Array

Configuration option = Default value	Type	Description
cycle_period_seconds = 300	integer value	This defines an optional cycle period that applies to Global Mirror relationships with a cycling mode of multi. A Global Mirror relationship using the multi cycling_mode performs a complete cycle at most once each period. The default is 300 seconds, and the valid seconds are 60-86400.
datacore_api_timeout = 300	integer value	Seconds to wait for a response from a DataCore API call.
datacore_disk_failed_delay = 300	integer value	Seconds to wait for DataCore virtual disk to come out of the "Failed" state.
datacore_disk_pools = []	list value	List of DataCore disk pools that can be used by volume driver.
datacore_disk_type = single	string value	DataCore virtual disk type (single/mirrored). Mirrored virtual disks require two storage servers in the server group.
datacore_fc_unallowed_targets = []	list value	List of FC targets that cannot be used to attach volume. To prevent the DataCore FibreChannel volume driver from using some front-end targets in volume attachment, specify this option and list the iqn and target machine for each target as the value, such as <wwpns:target name>, <wwpns:target name>, <wwpns:target name>.
datacore_iscsi_chap_storage = \$state_path/.datacore_chap	string value	Fully qualified file name where dynamically generated iSCSI CHAP secrets are stored.
datacore_iscsi_unallowed_targets = []	list value	List of iSCSI targets that cannot be used to attach volume. To prevent the DataCore iSCSI volume driver from using some front-end targets in volume attachment, specify this option and list the iqn and target machine for each target as the value, such as <iqn:target name>, <iqn:target name>, <iqn:target name>.
datacore_storage_profile = None	string value	DataCore virtual disk storage profile.

Configuration option = Default value	Type	Description
default_timeout = 31536000	integer value	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.
deferred_deletion_delay = 0	integer value	Time delay in seconds before a volume is eligible for permanent removal after being tagged for deferred deletion.
deferred_deletion_purge_interval = 60	integer value	Number of seconds between runs of the periodic task to purge volumes tagged for deletion.
dell_api_async_rest_timeout = 15	integer value	Dell SC API async call default timeout in seconds.
dell_api_sync_rest_timeout = 30	integer value	Dell SC API sync call default timeout in seconds.
dell_sc_api_port = 3033	port value	Dell API port
dell_sc_server_folder = openstack	string value	Name of the server folder to use on the Storage Center
dell_sc_ssn = 64702	integer value	Storage Center System Serial Number
dell_sc_verify_cert = False	boolean value	Enable HTTPS SC certificate verification
dell_sc_volume_folder = openstack	string value	Name of the volume folder to use on the Storage Center
dell_server_os = Red Hat Linux 6.x	string value	Server OS type to use when creating a new server on the Storage Center.
destroy_empty_storage_group = False	boolean value	To destroy storage group when the last LUN is removed from it. By default, the value is False.
disable_discovery = False	boolean value	Disabling iSCSI discovery (sendtargets) for multipath connections on K2 driver.
dpl_pool = `	string value	DPL pool uuid in which DPL volumes are stored.
dpl_port = 8357	port value	DPL port number.

Configuration option = Default value	Type	Description
driver_client_cert = None	string value	The path to the client certificate for verification, if the driver supports it.
driver_client_cert_key = None	string value	The path to the client certificate key for verification, if the driver supports it.
driver_data_namespace = None	string value	Namespace for driver private data values to be saved in.
driver_ssl_cert_path = None	string value	Can be used to specify a non default path to a CA_BUNDLE file or directory with certificates of trusted CAs, which will be used to validate the backend
driver_ssl_cert_verify = False	boolean value	If set to True the http client will validate the SSL certificate of the backend endpoint.
driver_use_ssl = False	boolean value	Tell driver to use SSL for connection to backend storage if the driver supports it.
<code>ds8k_devadd_unitadd_mapping = `</code>	string value	Mapping between IODevice address and unit address.
ds8k_host_type = auto	string value	Set to zLinux if your OpenStack version is prior to Liberty and you're connecting to zLinux systems. Otherwise set to auto. Valid values for this parameter are: <i>auto, AMDLinuxRHEL, AMDLinuxSuse, AppleOSX, Fujitsu, Hp, HpTru64, HpVms, LinuxDT, LinuxRF, LinuxRHEL, LinuxSuse, Novell, SGI, SVC, SanFsAIX, SanFsLinux, Sun, VMWare, Win2000, Win2003, Win2008, Win2012, iLinux, nSeries, pLinux, pSeries, pSeriesPowerswap, zLinux, iSeries.</i>
ds8k_ssid_prefix = FF	string value	Set the first two digits of SSID.
enable_deferred_deletion = False	boolean value	Enable deferred deletion. Upon deletion, volumes are tagged for deletion but will only be removed asynchronously at a later time.
enable_unsupported_driver = False	boolean value	Set this to True when you want to allow an unsupported driver to start. Drivers that haven't maintained a working CI system and testing are marked as unsupported until CI is working again. This also marks a driver as deprecated and may be removed in the next release.

Configuration option = Default value	Type	Description
enforce_multipath_for_image_xfer = False	boolean value	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. This parameter needs to be configured for each backend section or in [backend_defaults] section as a common configuration for all backends.
excluded_domain_ip = None	IP address value	DEPRECATED: Fault Domain IP to be excluded from iSCSI returns. Deprecated since: Stein *Reason:* Replaced by excluded_domain_ips option
excluded_domain_ips = []	list value	Comma separated Fault Domain IPs to be excluded from iSCSI returns.
expiry_thres_minutes = 720	integer value	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.
extra_capabilities = {}	string value	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.
filter_function = None	string value	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.
flashsystem_connection_protocol = FC	string value	Connection protocol should be FC. (Default is FC.)
flashsystem_iscsi_portid = 0	integer value	Default iSCSI Port ID of FlashSystem. (Default port is 0.)
flashsystem_multihostmap_enabled = True	boolean value	Allows vdisk to multi host mapping. (Default is True)
force_delete_lun_in_storage_group = True	boolean value	Delete a LUN even if it is in Storage Groups.

Configuration option = Default value	Type	Description
goodness_function = None	string value	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.
gpfs_hosts = []	list value	Comma-separated list of IP address or hostnames of GPFS nodes.
gpfs_hosts_key_file = \$state_path/ssh_known_hosts	string value	File containing SSH host keys for the gpfs nodes with which driver needs to communicate. Default=\$state_path/ssh_known_hosts
gpfs_images_dir = None	string value	Specifies the path of the Image service repository in GPFS. Leave undefined if not storing images in GPFS.
gpfs_images_share_mode = None	string value	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	integer value	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.
gpfs_mount_point_base = None	string value	Specifies the path of the GPFS directory where Block Storage volume and snapshot files are stored.
gpfs_private_key = `	string value	Filename of private key to use for SSH authentication.
gpfs_sparse_volumes = True	boolean value	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_ssh_port = 22	port value	SSH port to use.
gpfs_storage_pool = system	string value	Specifies the storage pool that volumes are assigned to. By default, the system storage pool is used.

Configuration option = Default value	Type	Description
gpfs_strict_host_key_policy = False	boolean value	Option to enable strict gpfs host key checking while connecting to gpfs nodes. Default=False
gpfs_user_login = root	string value	Username for GPFS nodes.
<code>`gpfs_user_password = `</code>	string value	Password for GPFS node user.
hitachi_compute_target_ports = []	list value	IDs of the storage ports used to attach volumes to compute nodes. To specify multiple ports, connect them by commas (e.g. CL1-A,CL2-A).
hitachi_discard_zero_page = True	boolean value	Enable or disable zero page reclamation in a DP-VOL.
hitachi_group_create = False	boolean value	If True, the driver will create host groups or iSCSI targets on storage ports as needed.
hitachi_group_delete = False	boolean value	If True, the driver will delete host groups or iSCSI targets on storage ports as needed.
hitachi_ldev_range = None	string value	Range of the LDEV numbers in the format of xxxx-yyyy that can be used by the driver. Values can be in decimal format (e.g. 1000) or in colon-separated hexadecimal format (e.g. 00:03:E8).
hitachi_pool = None	string value	Pool number or pool name of the DP pool.
hitachi_rest_tcp_keepalive = True	boolean value	Enables or disables use of REST API tcp keepalive
hitachi_snap_pool = None	string value	Pool number or pool name of the snapshot pool.
hitachi_storage_id = None	string value	Product number of the storage system.
hitachi_target_ports = []	list value	IDs of the storage ports used to attach volumes to the controller node. To specify multiple ports, connect them by commas (e.g. CL1-A,CL2-A).
hitachi_zoning_request = False	boolean value	If True, the driver will configure FC zoning between the server and the storage system provided that FC zoning manager is enabled.

Configuration option = Default value	Type	Description
<code>`hpe3par_api_url = `</code>	string value	WSAPI Server URL. This setting applies to both 3PAR and Primera. Example 1: for 3PAR, URL is: <a href="https://<3par ip>:8080/api/v1">https://<3par ip>:8080/api/v1 Example 2: for Primera, URL is: <a href="https://<primera ip>:443/api/v1">https://<primera ip>:443/api/v1
hpe3par_cpg = ['OpenStack']	list value	List of the 3PAR / Primera CPG(s) to use for volume creation
<code>`hpe3par_cpg_snap = `</code>	string value	The 3PAR / Primera CPG to use for snapshots of volumes. If empty the userCPG will be used.
hpe3par_debug = False	boolean value	Enable HTTP debugging to 3PAR / Primera
hpe3par_iscsi_chap_enabled = False	boolean value	Enable CHAP authentication for iSCSI connections.
hpe3par_iscsi_ips = []	list value	List of target iSCSI addresses to use.
<code>`hpe3par_password = `</code>	string value	3PAR / Primera password for the user specified in <code>hpe3par_username</code>
<code>`hpe3par_snapshot_expiration = `</code>	string value	The time in hours when a snapshot expires and is deleted. This must be larger than expiration
<code>`hpe3par_snapshot_retention = `</code>	string value	The time in hours to retain a snapshot. You can't delete it before this expires.
<code>`hpe3par_target_nsp = `</code>	string value	The nsp of 3PAR backend to be used when: (1) multipath is not enabled in <code>cinder.conf</code> . (2) Fiber Channel Zone Manager is not used. (3) the 3PAR backend is prezoned with this specific nsp only. For example if nsp is 2 1 2, the format of the option's value is 2:1:2
<code>`hpe3par_username = `</code>	string value	3PAR / Primera username with the <i>edit</i> role
hpmsa_api_protocol = https	string value	HPMSA API interface protocol.
hpmsa_iscsi_ips = []	list value	List of comma-separated target iSCSI IP addresses.
hpmsa_pool_name = A	string value	Pool or Vdisk name to use for volume creation.
hpmsa_pool_type = virtual	string value	linear (for Vdisk) or virtual (for Pool).

Configuration option = Default value	Type	Description
hpmsa_verify_certificate = False	boolean value	Whether to verify HPMSA array SSL certificate.
hpmsa_verify_certificate_ path = None	string value	HPMSA array SSL certificate path.
hypermetro_devices = None	string value	The remote device hypermetro will use.
iet_conf = /etc/iet/ietd.conf	string value	DEPRECATED: IET configuration file
ignore_pool_full_threshol d = False	boolean value	Force LUN creation even if the full threshold of pool is reached. By default, the value is False.
image_upload_use_cinde r_backend = False	boolean value	If set to True, upload-to-image in raw format will create a cloned volume and register its location to the image service, instead of uploading the volume content. The cinder backend and locations support must be enabled in the image service.
image_upload_use_intern al_tenant = False	boolean value	If set to True, the image volume created by upload-to-image will be placed in the internal tenant. Otherwise, the image volume is created in the current context's tenant.
image_volume_cache_en abled = False	boolean value	Enable the image volume cache for this backend.
image_volume_cache_ma x_count = 0	integer value	Max number of entries allowed in the image volume cache. 0 ⇒ unlimited.
image_volume_cache_ma x_size_gb = 0	integer value	Max size of the image volume cache for this backend in GB. 0 ⇒ unlimited.
included_domain_ips = []	list value	Comma separated Fault Domain IPs to be included from iSCSI returns.
infinidat_iscsi_netspaces = []	list value	List of names of network spaces to use for iSCSI connectivity
infinidat_pool_name = None	string value	Name of the pool from which volumes are allocated
infinidat_storage_protoco l = fc	string value	Protocol for transferring data between host and storage back-end.

Configuration option = Default value	Type	Description
infinidat_use_compression = False	boolean value	Specifies whether to turn on compression for newly created volumes.
initiator_auto_deregistration = False	boolean value	Automatically deregister initiators after the related storage group is destroyed. By default, the value is False.
initiator_auto_registration = False	boolean value	Automatically register initiators. By default, the value is False.
initiator_check = False	boolean value	Use this value to enable the initiator_check.
interval = 3	integer value	Use this value to specify length of the interval in seconds.
io_port_list = None	list value	Comma separated iSCSI or FC ports to be used in Nova or Cinder.
iscsi_initiators = None	string value	Mapping between hostname and its iSCSI initiator IP addresses.
iscsi_iotype = fileio	string value	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_secondary_ip_addresses = []	list value	The list of secondary IP addresses of the iSCSI daemon
`iscsi_target_flags = `	string value	Sets the target-specific flags for the iSCSI target. Only used for tgtadm to specify backing device flags using bsoflags option. The specified string is passed as is to the underlying tool.
iscsi_write_cache = on	string value	Sets the behavior of the iSCSI target to either perform write-back(on) or write-through(off). This parameter is valid if target_helper is set to tgtadm.
iser_helper = tgtadm	string value	The name of the iSER target user-land tool to use
iser_ip_address = \$my_ip	string value	The IP address that the iSER daemon is listening on
iser_port = 3260	port value	The port that the iSER daemon is listening on
iser_target_prefix = iqn.2010-10.org.openstack:	string value	Prefix for iSER volumes

Configuration option = Default value	Type	Description
lenovo_api_protocol = https	string value	Lenovo api interface protocol.
lenovo_iscsi_ips = []	list value	List of comma-separated target iSCSI IP addresses.
lenovo_pool_name = A	string value	Pool or Vdisk name to use for volume creation.
lenovo_pool_type = virtual	string value	linear (for VDisk) or virtual (for Pool).
lenovo_verify_certificate = False	boolean value	Whether to verify Lenovo array SSL certificate.
lenovo_verify_certificate_path = None	string value	Lenovo array SSL certificate path.
linstor_autoplace_count = 0	integer value	Autoplace replication count on volume deployment. 0 = Full cluster replication without autoplace, 1 = Single node deployment without replication, 2 or greater = Replicated deployment with autoplace.
linstor_controller_diskless = True	boolean value	True means Cinder node is a diskless LINSTOR node.
linstor_default_blocksize = 4096	integer value	Default Block size for Image restoration. When using iSCSI transport, this option specifies the block size.
linstor_default_storage_pool_name = DfltStorPool	string value	Default Storage Pool name for LINSTOR.
linstor_default_uri = linstor://localhost	string value	Default storage URI for LINSTOR.
linstor_default_volume_group_name = drbd-vg	string value	Default Volume Group name for LINSTOR. Not Cinder Volume.
linstor_volume_downsize_factor = 4096	floating point value	Default volume downscale size in KiB = 4 MiB.
load_balance = False	boolean value	Enable/disable load balancing for a PowerMax backend.
load_balance_real_time = False	boolean value	Enable/disable real-time performance metrics for Port level load balancing for a PowerMax backend.

Configuration option = Default value	Type	Description
load_data_format = Avg	string value	Performance data format, not applicable for real-time metrics. Available options are "avg" and "max".
load_look_back = 60	integer value	How far in minutes to look back for diagnostic performance metrics in load calculation, minimum of 0 maximum of 1440 (24 hours).
load_look_back_real_time = 1	integer value	How far in minutes to look back for real-time performance metrics in load calculation, minimum of 1 maximum of 10.
lss_range_for_cg = `	string value	Reserve LSSs for consistency group.
lvm_conf_file = /etc/cinder/lvm.conf	string value	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 <i>None</i> to not use a conf file even if one exists).
lvm_mirrors = 0	integer value	If >0, create LVs with multiple mirrors. Note that this requires lvm_mirrors + 2 PVs with available space
lvm_suppress_fd_warnings = False	boolean value	Suppress leaked file descriptor warnings in LVM commands.
lvm_type = auto	string value	Type of LVM volumes to deploy; (default, thin, or auto). Auto defaults to thin if thin is supported.
macrosan_client = None	list value	Macrosan iscsi_clients list. You can configure multiple clients. You can configure it in this format: (host; client_name; sp1_iscsi_port; sp2_iscsi_port), (host; client_name; sp1_iscsi_port; sp2_iscsi_port) Important warning, Client_name has the following requirements: [a-zA-Z0-9.-_:], the maximum number of characters is 31 E.g: (controller1; device1; eth-1:0; eth-2:0), (controller2; device2; eth-1:0/eth-1:1; eth-2:0/eth-2:1),
macrosan_client_default = None	string value	This is the default connection ports' name for iscsi. This default configuration is used when no host related information is obtained.E.g: eth-1:0/eth-1:1; eth-2:0/eth-2:1
macrosan_fc_keep_mapped_ports = True	boolean value	In the case of an FC connection, the configuration item associated with the port is maintained.

Configuration option = Default value	Type	Description
macrosan_fc_use_sp_port_nr = 1	integer value	The use_sp_port_nr parameter is the number of online FC ports used by the single-ended memory when the FC connection is established in the switch non-all-pass mode. The maximum is 4
macrosan_force_unmap_image = True	boolean value	Force disconnect while deleting volume
macrosan_log_timing = True	boolean value	Whether enable log timing
macrosan_pool = None	string value	Pool to use for volume creation
macrosan_replication_destination_ports = None	list value	Slave device
macrosan_replication_ipaddresses = None	list value	MacroSAN replication devices' ip addresses
macrosan_replication_password = None	string value	MacroSAN replication devices' password
macrosan_replication_username = None	string value	MacroSAN replication devices' username
macrosan_sdas_ipaddresses = None	list value	MacroSAN sdas devices' ip addresses
macrosan_sdas_password = None	string value	MacroSAN sdas devices' password
macrosan_sdas_username = None	string value	MacroSAN sdas devices' username
macrosan_snapshot_resource_ratio = 1.0	floating point value	Set snapshot's resource ratio
macrosan_thin_lun_extent_size = 8	integer value	Set the thin lun's extent size
macrosan_thin_lun_high_watermark = 20	integer value	Set the thin lun's high watermark
macrosan_thin_lun_low_watermark = 5	integer value	Set the thin lun's low watermark

Configuration option = Default value	Type	Description
<code>`management_ips = `</code>	string value	List of Management IP addresses (separated by commas)
max_luns_per_storage_group = 255	integer value	Default max number of LUNs in a storage group. By default, the value is 255.
max_over_subscription_ratio = 20.0	string value	Representation of the over subscription ratio when thin provisioning is enabled. 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. If ratio is <i>auto</i> , Cinder will automatically calculate the ratio based on the provisioned capacity and the used space. If not set to auto, the ratio has to be a minimum of 1.0.
metro_domain_name = None	string value	The remote metro device domain name.
metro_san_address = None	string value	The remote metro device request url.
metro_san_password = None	string value	The remote metro device san password.
metro_san_user = None	string value	The remote metro device san user.
metro_storage_pools = None	string value	The remote metro device pool names.
<code>`nas_host = `</code>	string value	IP address or Hostname of NAS system.
nas_login = admin	string value	User name to connect to NAS system.
nas_mount_options = None	string value	Options used to mount the storage backend file system where Cinder volumes are stored.
<code>`nas_password = `</code>	string value	Password to connect to NAS system.
<code>`nas_private_key = `</code>	string value	Filename of private key to use for SSH authentication.

Configuration option = Default value	Type	Description
nas_secure_file_operations = auto	string value	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	string value	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.
<code>`nas_share_path = `</code>	string value	Path to the share to use for storing Cinder volumes. For example: <code>"/srv/export1"</code> for an NFS server export available at 10.0.5.10:/srv/export1.
nas_ssh_port = 22	port value	SSH port to use to connect to NAS system.
nas_volume_prov_type = thin	string value	Provisioning type that will be used when creating volumes.
naviseccli_path = None	string value	Naviseccli Path.
netapp_api_trace_pattern = (.*)	string value	A regular expression to limit the API tracing. This option is honored only if enabling api tracing with the trace_flags option. By default, all APIs will be traced.
netapp_copyoffload_tool_path = None	string value	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_host_type = None	string value	This option defines the type of operating system for all initiators that can access a LUN. This information is used when mapping LUNs to individual hosts or groups of hosts.
netapp_login = None	string value	Administrative user account name used to access the storage system or proxy server.

Configuration option = Default value	Type	Description
netapp_lun_ostype = None	string value	This option defines the type of operating system that will access a LUN exported from Data ONTAP; it is assigned to the LUN at the time it is created.
netapp_lun_space_reservation = enabled	string value	This option determines if storage space is reserved for LUN allocation. If enabled, LUNs are thick provisioned. If space reservation is disabled, storage space is allocated on demand.
netapp_password = None	string value	Password for the administrative user account specified in the netapp_login option.
netapp_pool_name_search_pattern = (.+)	string value	This option is used to restrict provisioning to the specified pools. Specify the value of this option to be a regular expression which will be applied to the names of objects from the storage backend which represent pools in Cinder. This option is only utilized when the storage protocol is configured to use iSCSI or FC.
netapp_replication_aggregate_map = None	dict value	Multi opt of dictionaries to represent the aggregate mapping between source and destination back ends when using whole back end replication. For every source aggregate associated with a cinder pool (NetApp FlexVol/FlexGroup), you would need to specify the destination aggregate on the replication target device. A replication target device is configured with the configuration option replication_device. Specify this option as many times as you have replication devices. Each entry takes the standard dict config form: netapp_replication_aggregate_map = backend_id: <name_of_replication_device_section>,src_aggr_name1:dest_aggr_name1,src_aggr_name2:dest_aggr_name2,...
netapp_replication_volume_online_timeout = 360	integer value	Sets time in seconds to wait for a replication volume create to complete and go online.
netapp_server_hostname = None	string value	The hostname (or IP address) for the storage system or proxy server.
netapp_server_port = None	integer value	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.

Configuration option = Default value	Type	Description
netapp_size_multiplier = 1.2	floating point value	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. Note: this option is deprecated and will be removed in favor of "reserved_percentage" in the Mitaka release.
netapp_snapmirror_quiesce_timeout = 3600	integer value	The maximum time in seconds to wait for existing SnapMirror transfers to complete before aborting during a failover.
netapp_storage_family = ontap_cluster	string value	The storage family type used on the storage system; the only valid value is ontap_cluster for using clustered Data ONTAP.
netapp_storage_protocol = None	string value	The storage protocol to be used on the data path with the storage system.
netapp_transport_type = http	string value	The transport protocol used when communicating with the storage system or proxy server.
netapp_vserver = None	string value	This option specifies the virtual storage server (Vserver) name on the storage cluster on which provisioning of block storage volumes should occur.
nexenta_blocksize = 4096	integer value	Block size for datasets
nexenta_chunksize = 32768	integer value	NexentaEdge iSCSI LUN object chunk size
<code>`nexenta_client_address = `</code>	string value	NexentaEdge iSCSI Gateway client address for non-VIP service
nexenta_dataset_compression = on	string value	Compression value for new ZFS folders.
nexenta_dataset_dedup = off	string value	Deduplication value for new ZFS folders.
<code>`nexenta_dataset_description = `</code>	string value	Human-readable description for the folder.
nexenta_encryption = False	boolean value	Defines whether NexentaEdge iSCSI LUN object has encryption enabled.
<code>`nexenta_folder = `</code>	string value	A folder where cinder created datasets will reside.

Configuration option = Default value	Type	Description
nexenta_group_snapshot_template = group-snapshot-%s	string value	Template string to generate group snapshot name
<code>`nexenta_host = `</code>	string value	IP address of NexentaStor Appliance
nexenta_host_group_prefix = cinder	string value	Prefix for iSCSI host groups on NexentaStor
nexenta_iops_limit = 0	integer value	NexentaEdge iSCSI LUN object IOPS limit
<code>`nexenta_iscsi_service = `</code>	string value	NexentaEdge iSCSI service name
nexenta_iscsi_target_host_group = all	string value	Group of hosts which are allowed to access volumes
<code>`nexenta_iscsi_target_portal_groups = `</code>	string value	NexentaStor target portal groups
nexenta_iscsi_target_portal_port = 3260	integer value	Nexenta appliance iSCSI target portal port
<code>`nexenta_iscsi_target_portals = `</code>	string value	Comma separated list of portals for NexentaStor5, in format of IP1:port1,IP2:port2. Port is optional, default=3260. Example: 10.10.10.1:3267,10.10.1.2
nexenta_lu_writebackcache_disabled = False	boolean value	Postponed write to backing store or not
<code>`nexenta_lun_container = `</code>	string value	NexentaEdge logical path of bucket for LUNs
nexenta_luns_per_target = 100	integer value	Amount of LUNs per iSCSI target
nexenta_mount_point_base = \$state_path/mnt	string value	Base directory that contains NFS share mount points
nexenta_nbd_symlinks_dir = /dev/disk/by-path	string value	NexentaEdge logical path of directory to store symbolic links to NBDs
nexenta_nms_cache_volroot = True	boolean value	If set True cache NexentaStor appliance volroot option value.
nexenta_ns5_blocksize = 32	integer value	Block size for datasets

Configuration option = Default value	Type	Description
nexenta_origin_snapshot_template = origin-snapshot-%s	string value	Template string to generate origin name of clone
nexenta_password = nexenta	string value	Password to connect to NexentaStor management REST API server
nexenta_qcow2_volumes = False	boolean value	Create volumes as QCOW2 files rather than raw files
nexenta_replication_count = 3	integer value	NexentaEdge iSCSI LUN object replication count.
nexenta_rest_address = `	string value	IP address of NexentaStor management REST API endpoint
nexenta_rest_backoff_factor = 0.5	floating point value	Specifies the backoff factor to apply between connection attempts to NexentaStor management REST API server
nexenta_rest_connect_timeout = 30	floating point value	Specifies the time limit (in seconds), within which the connection to NexentaStor management REST API server must be established
nexenta_rest_password = nexenta	string value	Password to connect to NexentaEdge.
nexenta_rest_port = 0	integer value	HTTP(S) port to connect to NexentaStor management REST API server. If it is equal zero, 8443 for HTTPS and 8080 for HTTP is used
nexenta_rest_protocol = auto	string value	Use http or https for NexentaStor management REST API connection (default auto)
nexenta_rest_read_timeout = 300	floating point value	Specifies the time limit (in seconds), within which NexentaStor management REST API server must send a response
nexenta_rest_retry_count = 3	integer value	Specifies the number of times to repeat NexentaStor management REST API call in case of connection errors and NexentaStor appliance EBUSY or ENOENT errors
nexenta_rest_user = admin	string value	User name to connect to NexentaEdge.

Configuration option = Default value	Type	Description
nexenta_rrmgr_compression = 0	integer value	Enable stream compression, level 1..9. 1 – gives best speed; 9 – gives best compression.
nexenta_rrmgr_connections = 2	integer value	Number of TCP connections.
nexenta_rrmgr_tcp_buffer_size = 4096	integer value	TCP Buffer size in KiloBytes.
nexenta_shares_config = /etc/cinder/nfs_shares	string value	File with the list of available nfs shares
nexenta_sparse = False	boolean value	Enables or disables the creation of sparse datasets
nexenta_sparsed_volumes = True	boolean value	Enables or disables the creation of volumes as sparsed files that take no space. If disabled (False), volume is created as a regular file, which takes a long time.
nexenta_target_group_prefix = cinder	string value	Prefix for iSCSI target groups on NexentaStor
nexenta_target_prefix = iqn.1986-03.com.sun:02:cinder	string value	iqn prefix for NexentaStor iSCSI targets
nexenta_use_https = True	boolean value	Use HTTP secure protocol for NexentaStor management REST API connections
nexenta_user = admin	string value	User name to connect to NexentaStor management REST API server
nexenta_volume = cinder	string value	NexentaStor pool name that holds all volumes
nexenta_volume_group = iscsi	string value	Volume group for NexentaStor5 iSCSI
nfs_mount_attempts = 3	integer value	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	string value	Mount options passed to the NFS client. See the NFS(5) man page for details.

Configuration option = Default value	Type	Description
nfs_mount_point_base = \$state_path/mnt	string value	Base dir containing mount points for NFS shares.
nfs_qcow2_volumes = False	boolean value	Create volumes as QCOW2 files rather than raw files.
nfs_shares_config = /etc/cinder/nfs_shares	string value	File with the list of available NFS shares.
nfs_snapshot_support = False	boolean value	Enable support for snapshots on the NFS driver. Platforms using libvirt <1.2.7 will encounter issues with this feature.
nfs_sparsed_volumes = True	boolean value	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.
nimble_pool_name = default	string value	Nimble Controller pool name
nimble_subnet_label = *	string value	Nimble Subnet Label
nimble_verify_cert_path = None	string value	Path to Nimble Array SSL certificate
nimble_verify_certificate = False	boolean value	Whether to verify Nimble SSL Certificate
num_iser_scan_tries = 3	integer value	The maximum number of times to rescan iSER target to find volume
num_shell_tries = 3	integer value	Number of times to attempt to run flakey shell commands
num_volume_device_scan_tries = 3	integer value	The maximum number of times to rescan targets to find volume
nvmet_ns_id = 10	integer value	The namespace id associated with the subsystem that will be created with the path for the LVM volume.
nvmet_port_id = 1	port value	The port that the NVMe target is listening on.
port_group_load_metric = PercentBusy	string value	Metric used for port group load calculation.

Configuration option = Default value	Type	Description
port_load_metric = PercentBusy	string value	Metric used for port load calculation.
powerflex_allow_migration_during_rebuild = False	boolean value	Allow volume migration during rebuild.
powerflex_allow_non_padded_volumes = False	boolean value	Allow volumes to be created in Storage Pools when zero padding is disabled. This option should not be enabled if multiple tenants will utilize volumes from a shared Storage Pool.
powerflex_max_over_subscription_ratio = 10.0	floating point value	max_over_subscription_ratio setting for the driver. Maximum value allowed is 10.0.
powerflex_rest_server_port = 443	port value	Gateway REST server port.
powerflex_round_volume_capacity = True	boolean value	Round volume sizes up to 8GB boundaries. PowerFlex/VxFlex OS requires volumes to be sized in multiples of 8GB. If set to False, volume creation will fail for volumes not sized properly
powerflex_server_api_version = None	string value	PowerFlex/ScaleIO API version. This value should be left as the default value unless otherwise instructed by technical support.
powerflex_storage_pools = None	string value	Storage Pools. Comma separated list of storage pools used to provide volumes. Each pool should be specified as a protection_domain_name:storage_pool_name value
powerflex_unmap_volume_before_deletion = False	boolean value	Unmap volumes before deletion.
powermax_array = None	string value	Serial number of the array to connect to.
powermax_array_tag_list = None	list value	List of user assigned name for storage array.
powermax_port_group_name_template = portGroupName	string value	User defined override for port group name.
powermax_port_groups = None	list value	List of port groups containing frontend ports configured prior for server connection.

Configuration option = Default value	Type	Description
powermax_service_level = None	string value	Service level to use for provisioning storage. Setting this as an extra spec in pool_name is preferable.
powermax_short_host_name_template = shortHostName	string value	User defined override for short host name.
powermax_srp = None	string value	Storage resource pool on array to use for provisioning.
powerstore_appliances = []	list value	Appliances names. Comma separated list of PowerStore appliances names used to provision volumes. Deprecated since: Wallaby *Reason:*Is not used anymore. PowerStore Load Balancer is used to provision volumes instead.
powerstore_ports = []	list value	Allowed ports. Comma separated list of PowerStore iSCSI IPs or FC WWNs (ex. 58:cc:f0:98:49:22:07:02) to be used. If option is not set all ports are allowed.
proxy = cinder.volume.drivers.ibm.ibm_storage.proxy.IBMStorageProxy	string value	Proxy driver that connects to the IBM Storage Array
pure_api_token = None	string value	REST API authorization token.
pure_automatic_max_over_subscription_ratio = True	boolean value	Automatically determine an oversubscription ratio based on the current total data reduction values. If used this calculated value will override the max_over_subscription_ratio config option.
pure_eradicate_on_delete = False	boolean value	When enabled, all Pure volumes, snapshots, and protection groups will be eradicated at the time of deletion in Cinder. Data will NOT be recoverable after a delete with this set to True! When disabled, volumes and snapshots will go into pending eradication state and can be recovered.
pure_host_personality = None	string value	Determines how the Purity system tunes the protocol used between the array and the initiator.
pure_iscsi_cidr = 0.0.0.0/0	string value	CIDR of FlashArray iSCSI targets hosts are allowed to connect to. Default will allow connection to any IP address.

Configuration option = Default value	Type	Description
pure_replica_interval_default = 3600	integer value	Snapshot replication interval in seconds.
pure_replica_retention_long_term_default = 7	integer value	Retain snapshots per day on target for this time (in days.)
pure_replica_retention_long_term_per_day_default = 3	integer value	Retain how many snapshots for each day.
pure_replica_retention_short_term_default = 14400	integer value	Retain all snapshots on target for this time (in seconds.)
pure_replication_pg_name = cinder-group	string value	Pure Protection Group name to use for async replication (will be created if it does not exist).
pure_replication_pod_name = cinder-pod	string value	Pure Pod name to use for sync replication (will be created if it does not exist).
pvme_iscsi_ips = []	list value	List of comma-separated target iSCSI IP addresses.
pvme_pool_name = A	string value	Pool or Vdisk name to use for volume creation.
qnap_management_url = None	uri value	The URL to management QNAP Storage. Driver does not support IPv6 address in URL.
qnap_poolname = None	string value	The pool name in the QNAP Storage
qnap_storage_protocol = iscsi	string value	Communication protocol to access QNAP storage
quobyte_client_cfg = None	string value	Path to a Quobyte Client configuration file.
quobyte_mount_point_base = \$state_path/mnt	string value	Base dir containing the mount point for the Quobyte volume.
quobyte_overlay_volumes = False	boolean value	Create new volumes from the volume_from_snapshot_cache by creating overlay files instead of full copies. This speeds up the creation of volumes from this cache. This feature requires the options quobyte_qcow2_volumes and quobyte_volume_from_snapshot_cache to be set to True. If one of these is set to False this option is ignored.

Configuration option = Default value	Type	Description
quobyte_qcow2_volumes = True	boolean value	Create volumes as QCOW2 files rather than raw files.
quobyte_sparsed_volumes = True	boolean value	Create volumes as sparse files which take no space. If set to False, volume is created as regular file.
quobyte_volume_from_snapshot_cache = False	boolean value	Create a cache of volumes from merged snapshots to speed up creation of multiple volumes from a single snapshot.
quobyte_volume_url = None	string value	Quobyte URL to the Quobyte volume using e.g. a DNS SRV record (preferred) or a host list (alternatively) like quobyte://<DIR host1>, <DIR host2>/<volume name>
rados_connect_timeout = -1	integer value	Timeout value (in seconds) used when connecting to ceph cluster. If value < 0, no timeout is set and default librados value is used.
rados_connection_interval = 5	integer value	Interval value (in seconds) between connection retries to ceph cluster.
rados_connection_retries = 3	integer value	Number of retries if connection to ceph cluster failed.
<code>`rbd_ceph_conf = `</code>	string value	Path to the ceph configuration file
rbd_cluster_name = ceph	string value	The name of ceph cluster
rbd_exclusive_cinder_pool = True	boolean value	Set to False if the pool is shared with other usages. On exclusive use driver won't query images' provisioned size as they will match the value calculated by the Cinder core code for allocated_capacity_gb. This reduces the load on the Ceph cluster as well as on the volume service. On non exclusive use driver will query the Ceph cluster for per image used disk, this is an intensive operation having an independent request for each image.
rbd_flatten_volume_from_snapshot = False	boolean value	Flatten volumes created from snapshots to remove dependency from volume to snapshot
rbd_iscsi_api_debug = False	boolean value	Enable client request debugging.
<code>`rbd_iscsi_api_password = `</code>	string value	The username for the rbd_target_api service

Configuration option = Default value	Type	Description
<code>`rd_iscsi_api_url = `</code>	string value	The url to the rbd_target_api service
<code>`rd_iscsi_api_user = `</code>	string value	The username for the rbd_target_api service
rd_iscsi_target_iqn = None	string value	The preconfigured target_iqn on the iscsi gateway.
rd_max_clone_depth = 5	integer value	Maximum number of nested volume clones that are taken before a flatten occurs. Set to 0 to disable cloning. Note: lowering this value will not affect existing volumes whose clone depth exceeds the new value.
rd_pool = rbd	string value	The RADOS pool where rbd volumes are stored
rd_secret_uuid = None	string value	The libvirt uuid of the secret for the rbd_user volumes
rd_store_chunk_size = 4	integer value	Volumes will be chunked into objects of this size (in megabytes).
rd_user = None	string value	The RADOS client name for accessing rbd volumes - only set when using cephx authentication
remove_empty_host = False	boolean value	To remove the host from Unity when the last LUN is detached from it. By default, it is False.
replication_connect_timeout = 5	integer value	Timeout value (in seconds) used when connecting to ceph cluster to do a demotion/promotion of volumes. If value < 0, no timeout is set and default librados value is used.
replication_device = None	dict value	Multi opt of dictionaries to represent a replication target device. This option may be specified multiple times in a single config section to specify multiple replication target devices. Each entry takes the standard dict config form: replication_device = target_device_id:<required>,key1:value1,key2:value2...
report_discard_supported = False	boolean value	Report to clients of Cinder that the backend supports discard (aka. trim/unmap). This will not actually change the behavior of the backend or the client directly, it will only notify that it can be used.

Configuration option = Default value	Type	Description
report_dynamic_total_capacity = True	boolean value	Set to True for driver to report total capacity as a dynamic value (used + current free) and to False to report a static value (quota max bytes if defined and global size of cluster if not).
reserved_percentage = 0	integer value	The percentage of backend capacity is reserved
retries = 200	integer value	Use this value to specify number of retries.
san_api_port = None	port value	Port to use to access the SAN API
<code>`san_clustername = `</code>	string value	Cluster name to use for creating volumes
<code>`san_ip = `</code>	string value	IP address of SAN controller
san_is_local = False	boolean value	Execute commands locally instead of over SSH; use if the volume service is running on the SAN device
san_login = admin	string value	Username for SAN controller
<code>`san_password = `</code>	string value	Password for SAN controller
<code>`san_private_key = `</code>	string value	Filename of private key to use for SSH authentication
san_ssh_port = 22	port value	SSH port to use with SAN
san_thin_provision = True	boolean value	Use thin provisioning for SAN volumes?
scst_target_driver = iscsi	string value	SCST target implementation can choose from multiple SCST target drivers.
scst_target_iqn_name = None	string value	Certain iSCSI targets have predefined target names, SCST target driver uses this name.
seagate_iscsi_ips = []	list value	List of comma-separated target iSCSI IP addresses.
seagate_pool_name = A	string value	Pool or vdisk name to use for volume creation.
seagate_pool_type = virtual	string value	linear (for vdisk) or virtual (for virtual pool).
<code>`secondary_san_ip = `</code>	string value	IP address of secondary DSM controller

Configuration option = Default value	Type	Description
secondary_san_login = Admin	string value	Secondary DSM user name
<code>`secondary_san_password = `</code>	string value	Secondary DSM user password name
secondary_sc_api_port = 3033	port value	Secondary Dell API port
sf_account_prefix = None	string value	Create SolidFire accounts with this prefix. Any string can be used here, but the string "hostname" is special and will create a prefix using the cinder node hostname (previous default behavior). The default is NO prefix.
sf_allow_tenant_qos = False	boolean value	Allow tenants to specify QOS on create
sf_api_port = 443	port value	SolidFire API port. Useful if the device api is behind a proxy on a different port.
sf_api_request_timeout = 30	integer value	Sets time in seconds to wait for an api request to complete.
sf_cluster_pairing_timeout = 60	integer value	Sets time in seconds to wait for clusters to complete pairing.
sf_emulate_512 = True	boolean value	Set 512 byte emulation on volume creation;
sf_enable_vag = False	boolean value	Utilize volume access groups on a per-tenant basis.
sf_provisioning_calc = maxProvisionedSpace	string value	Change how SolidFire reports used space and provisioning calculations. If this parameter is set to <i>usedSpace</i> , the driver will report correct values as expected by Cinder thin provisioning.
sf_svip = None	string value	Overrides default cluster SVIP with the one specified. This is required for deployments that have implemented the use of VLANs for iSCSI networks in their cloud.
sf_volume_clone_timeout = 600	integer value	Sets time in seconds to wait for a clone of a volume or snapshot to complete.
sf_volume_create_timeout = 60	integer value	Sets time in seconds to wait for a create volume operation to complete.

Configuration option = Default value	Type	Description
sf_volume_pairing_timeout = 3600	integer value	Sets time in seconds to wait for a migrating volume to complete pairing and sync.
sf_volume_prefix = UUID-	string value	Create SolidFire volumes with this prefix. Volume names are of the form <sf_volume_prefix><cinder-volume-id>. The default is to use a prefix of <i>UUID-</i> .
smbfs_default_volume_format = vhd	string value	Default format that will be used when creating volumes if no volume format is specified.
smbfs_mount_point_base = C:\OpenStack\mnt	string value	Base dir containing mount points for smbfs shares.
smbfs_pool_mappings = {}	dict value	Mappings between share locations and pool names. If not specified, the share names will be used as pool names. Example: //addr/share:pool_name, //addr/share2:pool_name2
smbfs_shares_config = C:\OpenStack\smbfs_shares.txt	string value	File with the list of available smbfs shares.
spdk_max_queue_depth = 64	integer value	Queue depth for rdma transport.
spdk_rpc_ip = None	string value	The NVMe target remote configuration IP address.
spdk_rpc_password = None	string value	The NVMe target remote configuration password.
spdk_rpc_port = 8000	port value	The NVMe target remote configuration port.
spdk_rpc_protocol = http	string value	Protocol to be used with SPDK RPC proxy
spdk_rpc_username = None	string value	The NVMe target remote configuration username.
ssh_conn_timeout = 30	integer value	SSH connection timeout in seconds
ssh_max_pool_conn = 5	integer value	Maximum ssh connections in the pool
ssh_min_pool_conn = 1	integer value	Minimum ssh connections in the pool
storage_protocol = iscsi	string value	Protocol for transferring data between host and storage back-end.

Configuration option = Default value	Type	Description
storage_vnx_authentication_type = global	string value	VNX authentication scope type. By default, the value is global.
storage_vnx_pool_names = None	list value	Comma-separated list of storage pool names to be used.
storage_vnx_security_file_dir = None	string value	Directory path that contains the VNX security file. Make sure the security file is generated first.
storpool_replication = 3	integer value	The default StorPool chain replication value. Used when creating a volume with no specified type if storpool_template is not set. Also used for calculating the apparent free space reported in the stats.
storpool_template = None	string value	The StorPool template for volumes with no type.
storwize_peer_pool = None	string value	Specifies the name of the peer pool for hyperswap volume, the peer pool must exist on the other site.
storwize_preferred_host_site = {}	dict value	Specifies the site information for host. One WWPN or multi WWPNs used in the host can be specified. For example: storwize_preferred_host_site=site1:wwpn1,site2:wwpn2&wwpn3 or storwize_preferred_host_site=site1:iqn1,site2:iqn2
storwize_san_secondary_ip = None	string value	Specifies secondary management IP or hostname to be used if san_ip is invalid or becomes inaccessible.
storwize_svc_allow_tenant_qos = False	boolean value	Allow tenants to specify QOS on create
storwize_svc_flashcopy_rate = 50	integer value	Specifies the Storwize FlashCopy copy rate to be used when creating a full volume copy. The default is rate is 50, and the valid rates are 1-150.
storwize_svc_flashcopy_timeout = 120	integer value	Maximum number of seconds to wait for FlashCopy to be prepared.
storwize_svc_iscsi_chap_enabled = True	boolean value	Configure CHAP authentication for iSCSI connections (Default: Enabled)
storwize_svc_mirror_pool = None	string value	Specifies the name of the pool in which mirrored copy is stored. Example: "pool2"

Configuration option = Default value	Type	Description
storwize_svc_multihostmap_enabled = True	boolean value	This option no longer has any affect. It is deprecated and will be removed in the next release.
storwize_svc_multipath_enabled = False	boolean value	Connect with multipath (FC only; iSCSI multipath is controlled by Nova)
storwize_svc_retain_aux_volume = False	boolean value	Enable or disable retaining of aux volume on secondary storage during delete of the volume on primary storage or moving the primary volume from mirror to non-mirror with replication enabled. This option is valid for Spectrum Virtualize Family.
storwize_svc_stretched_cluster_partner = None	string value	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	boolean value	Storage system autoexpand parameter for volumes (True/False)
storwize_svc_vol_compression = False	boolean value	Storage system compression option for volumes
storwize_svc_vol_easytier = True	boolean value	Enable Easy Tier for volumes
storwize_svc_vol_grainsize = 256	integer value	Storage system grain size parameter for volumes (8/32/64/128/256)
storwize_svc_vol_iogrp = 0	string value	The I/O group in which to allocate volumes. It can be a comma-separated list in which case the driver will select an io_group based on least number of volumes associated with the io_group.
storwize_svc_vol_nofmtdisk = False	boolean value	Specifies that the volume not be formatted during creation.
storwize_svc_vol_rsize = 2	integer value	Storage system space-efficiency parameter for volumes (percentage)
storwize_svc_vol_warning = 0	integer value	Storage system threshold for volume capacity warnings (percentage)
storwize_svc_volpool_name = ['volpool']	list value	Comma separated list of storage system storage pools for volumes.

Configuration option = Default value	Type	Description
suppress_requests_ssl_warnings = False	boolean value	Suppress requests library SSL certificate warnings.
synology_admin_port = 5000	port value	Management port for Synology storage.
synology_device_id = None	string value	Device id for skip one time password check for logging in Synology storage if OTP is enabled.
synology_one_time_pass = None	string value	One time password of administrator for logging in Synology storage if OTP is enabled.
<code>`synology_password = `</code>	string value	Password of administrator for logging in Synology storage.
<code>`synology_pool_name = `</code>	string value	Volume on Synology storage to be used for creating lun.
synology_ssl_verify = True	boolean value	Do certificate validation or not if \$driver_use_ssl is True
synology_username = admin	string value	Administrator of Synology storage.
target_helper = tgtadm	string value	Target user-land tool to use. tgtadm is default, use lioadm for LIO iSCSI support, scstadmin for SCST target support, ietadm for iSCSI Enterprise Target, iscsictl for Chelsio iSCSI Target, nvmet for NVMe-oF support, spdk-nvmeof for SPDK NVMe-oF, or fake for testing. Note: The IET driver is deprecated and will be removed in the V release.
target_ip_address = \$my_ip	string value	The IP address that the iSCSI daemon is listening on
target_port = 3260	port value	The port that the iSCSI daemon is listening on
target_prefix = iqn.2010-10.org.openstack:	string value	Prefix for iSCSI volumes
target_protocol = iscsi	string value	Determines the target protocol for new volumes, created with tgtadm, lioadm and nvmet 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", in case of nvmet target set to "nvmet_rdma".

Configuration option = Default value	Type	Description
thres_avl_size_perc_start = 20	integer value	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	integer value	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 <code>expiry_thres_minutes</code> configuration option.
trace_flags = None	list value	List of options that control which trace info is written to the DEBUG log level to assist developers. Valid values are <code>method</code> and <code>api</code> .
u4p_failover_autofailback = True	boolean value	If the driver should automatically failback to the primary instance of Unisphere when a successful connection is re-established.
u4p_failover_backoff_factor = 1	integer value	A backoff factor to apply between attempts after the second try (most errors are resolved immediately by a second try without a delay). Retries will sleep for: $\{\text{backoff factor}\} * (2 ^ (\{\text{number of total retries}\} - 1))$ seconds.
u4p_failover_retries = 3	integer value	The maximum number of retries each connection should attempt. Note, this applies only to failed DNS lookups, socket connections and connection timeouts, never to requests where data has made it to the server.
u4p_failover_target = None	dict value	Dictionary of Unisphere failover target info.
u4p_failover_timeout = 20.0	integer value	How long to wait for the server to send data before giving up.
unique_fqdn_network = True	boolean value	Whether or not our private network has unique FQDN on each initiator or not. For example networks with QA systems usually have multiple servers/VMs with the same FQDN. When true this will create host entries on 3PAR using the FQDN, when false it will use the reversed IQN/WWNN.
unity_io_ports = []	list value	A comma-separated list of iSCSI or FC ports to be used. Each port can be Unix-style glob expressions.

Configuration option = Default value	Type	Description
unity_storage_pool_names = []	list value	A comma-separated list of storage pool names to be used.
use_chap_auth = False	boolean value	Option to enable/disable CHAP authentication for targets.
use_multipath_for_image_xfer = False	boolean value	Do we attach/detach volumes in cinder using multipath for volume to image and image to volume transfers? This parameter needs to be configured for each backend section or in [backend_defaults] section as a common configuration for all backends.
vmax_workload = None	string value	Workload, setting this as an extra spec in pool_name is preferable.
vmware_adapter_type = lsiLogic	string value	Default adapter type to be used for attaching volumes.
vmware_api_retry_count = 10	integer value	Number of times VMware vCenter server API must be retried upon connection related issues.
vmware_ca_file = None	string value	CA bundle file to use in verifying the vCenter server certificate.
vmware_cluster_name = None	multi valued	Name of a vCenter compute cluster where volumes should be created.
vmware_connection_pool_size = 10	integer value	Maximum number of connections in http connection pool.
vmware_datastore_regex = None	string value	Regular expression pattern to match the name of datastores where backend volumes are created.
vmware_enable_volume_stats = False	boolean value	If true, this enables the fetching of the volume stats from the backend. This has potential performance issues at scale. When False, the driver will not collect ANY stats about the backend.
vmware_host_ip = None	string value	IP address for connecting to VMware vCenter server.
vmware_host_password = None	string value	Password for authenticating with VMware vCenter server.
vmware_host_port = 443	port value	Port number for connecting to VMware vCenter server.

Configuration option = Default value	Type	Description
vmware_host_username = None	string value	Username for authenticating with VMware vCenter server.
vmware_host_version = None	string value	Optional string specifying the VMware vCenter server version. The driver attempts to retrieve the version from VMware vCenter server. Set this configuration only if you want to override the vCenter server version.
vmware_image_transfer_timeout_secs = 7200	integer value	Timeout in seconds for VMDK volume transfer between Cinder and Glance.
vmware_insecure = False	boolean value	If true, the vCenter server certificate is not verified. If false, then the default CA truststore is used for verification. This option is ignored if "vmware_ca_file" is set.
vmware_lazy_create = True	boolean value	If true, the backend volume in vCenter server is created lazily when the volume is created without any source. The backend volume is created when the volume is attached, uploaded to image service or during backup.
vmware_max_objects_retrieval = 100	integer value	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_snapshot_format = template	string value	Volume snapshot format in vCenter server.
vmware_storage_profile = None	multi valued	Names of storage profiles to be monitored. Only used when vmware_enable_volume_stats is True.
vmware_task_poll_interval = 2.0	floating point value	The interval (in seconds) for polling remote tasks invoked on VMware vCenter server.
vmware_tmp_dir = /tmp	string value	Directory where virtual disks are stored during volume backup and restore.
vmware_volume_folder = Volumes	string value	Name of the vCenter inventory folder that will contain Cinder volumes. This folder will be created under "OpenStack/<project_folder>", where project_folder is of format "Project (<volume_project_id>)".

Configuration option = Default value	Type	Description
vmware_wsdl_location = None	string value	Optional VIM service WSDL Location e.g <a href="http://<server>/vimService.wsdl">http://<server>/vimService.wsdl . Optional over-ride to default location for bug work-arounds.
vnx_async_migrate = True	boolean value	Always use asynchronous migration during volume cloning and creating from snapshot. As described in configuration doc, async migration has some constraints. Besides using metadata, customers could use this option to disable async migration. Be aware that async_migrate in metadata overrides this option when both are set. By default, the value is True.
volume_backend_name = None	string value	The backend name for a given driver implementation
volume_clear = zero	string value	Method used to wipe old volumes
volume_clear_ionice = None	string value	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	integer value	Size in MiB to wipe at start of old volumes. 1024 MiB at max. 0 ⇒ all
volume_copy_blkio_cgrou p_name = cinder- volume-copy	string value	The blkio cgroup name to be used to limit bandwidth of volume copy
volume_copy_bps_limit = 0	integer value	The upper limit of bandwidth of volume copy. 0 ⇒ unlimited
volume_dd_blocksize = 1M	string value	The default block size used when copying/clearing volumes
volume_driver = cinder.volume.drivers.lvm .LVMVolumeDriver	string value	Driver to use for volume creation
volume_group = cinder- volumes	string value	Name for the VG that will contain exported volumes
volumes_dir = \$state_path/volumes	string value	Volume configuration file storage directory
vxflexos_allow_migration _during_rebuild = False	boolean value	renamed to powerflex_allow_migration_during_rebuild.

Configuration option = Default value	Type	Description
vxflexos_allow_non_padded_volumes = False	boolean value	renamed to powerflex_allow_non_padded_volumes.
vxflexos_max_over_subscription_ratio = 10.0	floating point value	renamed to powerflex_max_over_subscription_ratio.
vxflexos_rest_server_port = 443	port value	renamed to powerflex_rest_server_port.
vxflexos_round_volume_capacity = True	boolean value	renamed to powerflex_round_volume_capacity.
vxflexos_server_api_version = None	string value	renamed to powerflex_server_api_version.
vxflexos_storage_pools = None	string value	renamed to powerflex_storage_pools.
vxflexos_unmap_volume_before_deletion = False	boolean value	renamed to powerflex_round_volume_capacity.
vzstorage_default_volume_format = raw	string value	Default format that will be used when creating volumes if no volume format is specified.
vzstorage_mount_options = None	list value	Mount options passed to the vzstorage client. See section of the pstorage-mount man page for details.
vzstorage_mount_point_base = \$state_path/mnt	string value	Base dir containing mount points for vzstorage shares.
vzstorage_shares_config = /etc/cinder/vzstorage_shares	string value	File with the list of available vzstorage shares.
vzstorage_sparsed_volumes = True	boolean value	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.
vzstorage_used_ratio = 0.95	floating point value	Percent of ACTUAL usage of the underlying volume before no new volumes can be allocated to the volume destination.
windows_iscsi_lun_path = C:\iSCSIVirtualDisks	string value	Path to store VHD backed volumes

Configuration option = Default value	Type	Description
xtremio_array_busy_retry_count = 5	integer value	Number of retries in case array is busy
xtremio_array_busy_retry_interval = 5	integer value	Interval between retries in case array is busy
xtremio_clean_unused_ig = False	boolean value	Should the driver remove initiator groups with no volumes after the last connection was terminated. Since the behavior till now was to leave the IG be, we default to False (not deleting IGs without connected volumes); setting this parameter to True will remove any IG after terminating its connection to the last volume.
xtremio_cluster_name = `	string value	XMS cluster id in multi-cluster environment
xtremio_ports = []	list value	Allowed ports. Comma separated list of XtremIO iSCSI IPs or FC WWNs (ex. 58:cc:f0:98:49:22:07:02) to be used. If option is not set all ports are allowed.
xtremio_volumes_per_glance_cache = 100	integer value	Number of volumes created from each cached glance image
zadara_access_key = None	string value	VPSA access key
zadara_default_snap_policy = False	boolean value	VPSA - Attach snapshot policy for volumes. If the option is neither configured nor provided as metadata, the VPSA will inherit the default value.
zadara_gen3_vol_compression = False	boolean value	VPSA - Enable compression for volumes. If the option is neither configured nor provided as metadata, the VPSA will inherit the default value.
zadara_gen3_vol_dedupe = False	boolean value	VPSA - Enable deduplication for volumes. If the option is neither configured nor provided as metadata, the VPSA will inherit the default value.
zadara_ssl_cert_verify = True	boolean value	If set to True the http client will validate the SSL certificate of the VPSA endpoint.
zadara_vol_encrypt = False	boolean value	VPSA - Default encryption policy for volumes. If the option is neither configured nor provided as metadata, the VPSA will inherit the default value.
zadara_vpsa_host = None	host address value	VPSA - Management Host name or IP address

Configuration option = Default value	Type	Description
zadara_vpsa_poolname = None	string value	VPSA - Storage Pool assigned for volumes
zadara_vpsa_port = None	port value	VPSA - Port number
zadara_vpsa_use_ssl = False	boolean value	VPSA - Use SSL connection

2.1.4. barbican

The following table outlines the options available under the **[barbican]** group in the `/etc/cinder/cinder.conf` file.

Table 2.3. barbican

Configuration option = Default value	Type	Description
auth_endpoint = http://localhost/identity/v3	string value	Use this endpoint to connect to Keystone
barbican_api_version = None	string value	Version of the Barbican API, for example: "v1"
barbican_endpoint = None	string value	Use this endpoint to connect to Barbican, for example: "http://localhost:9311/"
barbican_endpoint_type = public	string value	Specifies the type of endpoint. Allowed values are: public, private, and admin
number_of_retries = 60	integer value	Number of times to retry poll for key creation completion
retry_delay = 1	integer value	Number of seconds to wait before retrying poll for key creation completion
verify_ssl = True	boolean value	Specifies if insecure TLS (https) requests. If False, the server's certificate will not be validated, if True, we can set the verify_ssl_path config meanwhile.
verify_ssl_path = None	string value	A path to a bundle or CA certs to check against, or None for requests to attempt to locate and use certificates which verify_ssh is True. If verify_ssl is False, this is ignored.

2.1.5. brcd_fabric_example

The following table outlines the options available under the **[brcd_fabric_example]** group in the **/etc/cinder/cinder.conf** file.

Table 2.4. brcd_fabric_example

Configuration option = Default value	Type	Description
<code>`fc_fabric_address = `</code>	string value	Management IP of fabric.
<code>`fc_fabric_password = `</code>	string value	Password for user.
fc_fabric_port = 22	port value	Connecting port
<code>`fc_fabric_ssh_cert_path = `</code>	string value	Local SSH certificate Path.
<code>`fc_fabric_user = `</code>	string value	Fabric user ID.
fc_southbound_protocol = REST_HTTP	string value	South bound connector for the fabric.
fc_virtual_fabric_id = None	string value	Virtual Fabric ID.
zone_activate = True	boolean value	Overridden zoning activation state.
zone_name_prefix = openstack	string value	Overridden zone name prefix.
zoning_policy = initiator-target	string value	Overridden zoning policy.

2.1.6. cisco_fabric_example

The following table outlines the options available under the **[cisco_fabric_example]** group in the **/etc/cinder/cinder.conf** file.

Table 2.5. cisco_fabric_example

Configuration option = Default value	Type	Description
<code>`cisco_fc_fabric_address = `</code>	string value	Management IP of fabric
<code>`cisco_fc_fabric_password = `</code>	string value	Password for user
cisco_fc_fabric_port = 22	port value	Connecting port
<code>`cisco_fc_fabric_user = `</code>	string value	Fabric user ID

Configuration option = Default value	Type	Description
cisco_zone_activate = True	boolean value	overridden zoning activation state
cisco_zone_name_prefix = None	string value	overridden zone name prefix
cisco_zoning_policy = initiator-target	string value	overridden zoning policy
cisco_zoning_vsan = None	string value	VSAN of the Fabric

2.1.7. coordination

The following table outlines the options available under the **[coordination]** group in the `/etc/cinder/cinder.conf` file.

Table 2.6. coordination

Configuration option = Default value	Type	Description
backend_url = file://\$state_path	string value	The backend URL to use for distributed coordination.

2.1.8. cors

The following table outlines the options available under the **[cors]** group in the `/etc/cinder/cinder.conf` file.

Table 2.7. cors

Configuration option = Default value	Type	Description
allow_credentials = True	boolean value	Indicate that the actual request can include user credentials
allow_headers = ['X-Auth-Token', 'X-Identity-Status', 'X-Roles', 'X-Service-Catalog', 'X-User-Id', 'X-Tenant-Id', 'X-OpenStack-Request-ID', 'X-Trace-Info', 'X-Trace-HMAC', 'OpenStack-API-Version']	list value	Indicate which header field names may be used during the actual request.

Configuration option = Default value	Type	Description
allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH', 'HEAD']	list value	Indicate which methods can be used during the actual request.
allowed_origin = None	list value	Indicate whether this resource may be shared with the domain received in the requests "origin" header. Format: "<protocol>://<host>[:<port>]", no trailing slash. Example: https://horizon.example.com
expose_headers = ['X-Auth-Token', 'X-Subject-Token', 'X-Service-Token', 'X-OpenStack-Request-ID', 'OpenStack-API-Version']	list value	Indicate which headers are safe to expose to the API. Defaults to HTTP Simple Headers.
max_age = 3600	integer value	Maximum cache age of CORS preflight requests.

2.1.9. database

The following table outlines the options available under the **[database]** group in the **/etc/cinder/cinder.conf** file.

Table 2.8. database

Configuration option = Default value	Type	Description
backend = sqlalchemy	string value	The back end to use for the database.
connection = None	string value	The SQLAlchemy connection string to use to connect to the database.
connection_debug = 0	integer value	Verbosity of SQL debugging information: 0=None, 100=Everything.
connection_parameters = {}	string value	Optional URL parameters to append onto the connection URL at connect time; specify as param1=value1&param2=value2&...
connection_recycle_time = 3600	integer value	Connections which have been present in the connection pool longer than this number of seconds will be replaced with a new one the next time they are checked out from the pool.
connection_trace = False	boolean value	Add Python stack traces to SQL as comment strings.

Configuration option = Default value	Type	Description
db_inc_retry_interval = True	boolean value	If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = 20	integer value	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	integer value	If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = 1	integer value	Seconds between retries of a database transaction.
max_overflow = 50	integer value	If set, use this value for max_overflow with SQLAlchemy.
max_pool_size = 5	integer value	Maximum number of SQL connections to keep open in a pool. Setting a value of 0 indicates no limit.
max_retries = 10	integer value	Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.
mysql_enable_ndb = False	boolean value	If True, transparently enables support for handling MySQL Cluster (NDB).
mysql_sql_mode = TRADITIONAL	string value	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	integer value	If set, use this value for pool_timeout with SQLAlchemy.
retry_interval = 10	integer value	Interval between retries of opening a SQL connection.
slave_connection = None	string value	The SQLAlchemy connection string to use to connect to the slave database.
sqlite_synchronous = True	boolean value	If True, SQLite uses synchronous mode.
use_db_reconnect = False	boolean value	Enable the experimental use of database reconnect on connection lost.

2.1.10. fc-zone-manager

The following table outlines the options available under the **[fc-zone-manager]** group in the **/etc/cinder/cinder.conf** file.

Table 2.9. fc-zone-manager

Configuration option = Default value	Type	Description
brcd_sb_connector = HTTP	string value	South bound connector for zoning operation
cisco_sb_connector = cinder.zonemanager.drivers.cisco.cisco_fc_zone_client_cli.CiscoFCZoneClientCLI	string value	Southbound connector for zoning operation
enable_unsupported_driver = False	boolean value	Set this to True when you want to allow an unsupported zone manager driver to start. Drivers that haven't maintained a working CI system and testing are marked as unsupported until CI is working again. This also marks a driver as deprecated and may be removed in the next release.
fc_fabric_names = None	string value	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	string value	FC SAN Lookup Service
zone_driver = cinder.zonemanager.drivers.brocade.brcd_fc_zone_driver.BrcdFCZoneDriver	string value	FC Zone Driver responsible for zone management
zoning_policy = initiator-target	string value	Zoning policy configured by user; valid values include "initiator-target" or "initiator"

2.1.11. healthcheck

The following table outlines the options available under the **[healthcheck]** group in the **/etc/cinder/cinder.conf** file.

Table 2.10. healthcheck

Configuration option = Default value	Type	Description
backends = []	list value	Additional backends that can perform health checks and report that information back as part of a request.
detailed = False	boolean value	Show more detailed information as part of the response. Security note: Enabling this option may expose sensitive details about the service being monitored. Be sure to verify that it will not violate your security policies.
disable_by_file_path = None	string value	Check the presence of a file to determine if an application is running on a port. Used by DisableByFileHealthcheck plugin.
disable_by_file_paths = []	list value	Check the presence of a file based on a port to determine if an application is running on a port. Expects a "port:path" list of strings. Used by DisableByFilesPortsHealthcheck plugin.
path = /healthcheck	string value	The path to respond to healthcheck requests on.

2.1.12. key_manager

The following table outlines the options available under the **[key_manager]** group in the `/etc/cinder/cinder.conf` file.

Table 2.11. key_manager

Configuration option = Default value	Type	Description
auth_type = None	string value	The type of authentication credential to create. Possible values are <i>token</i> , <i>password</i> , <i>keystone_token</i> , and <i>keystone_password</i> . Required if no context is passed to the credential factory.
auth_url = None	string value	Use this endpoint to connect to Keystone.
backend = barbican	string value	Specify the key manager implementation. Options are "barbican" and "vault". Default is "barbican". Will support the values earlier set using <code>[key_manager]/api_class</code> for some time.
domain_id = None	string value	Domain ID for domain scoping. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
domain_name = None	string value	Domain name for domain scoping. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.

Configuration option = Default value	Type	Description
fixed_key = None	string value	Fixed key returned by key manager, specified in hex
password = None	string value	Password for authentication. Required for <i>password</i> and <i>keystone_password</i> auth_type.
project_domain_id = None	string value	Project's domain ID for project. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
project_domain_name = None	string value	Project's domain name for project. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
project_id = None	string value	Project ID for project scoping. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
project_name = None	string value	Project name for project scoping. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
reauthenticate = True	boolean value	Allow fetching a new token if the current one is going to expire. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
token = None	string value	Token for authentication. Required for <i>token</i> and <i>keystone_token</i> auth_type if no context is passed to the credential factory.
trust_id = None	string value	Trust ID for trust scoping. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
user_domain_id = None	string value	User's domain ID for authentication. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
user_domain_name = None	string value	User's domain name for authentication. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
user_id = None	string value	User ID for authentication. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
username = None	string value	Username for authentication. Required for <i>password</i> auth_type. Optional for the <i>keystone_password</i> auth_type.

2.1.13. keystone_authtoken

The following table outlines the options available under the **[keystone_authtoken]** group in the **/etc/cinder/cinder.conf** file.

Table 2.12. keystone_authtoken

Configuration option = Default value	Type	Description
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load
auth_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint. This option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release. Deprecated since: Queens <i>*Reason:</i>*The <code>auth_uri</code> option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release.
auth_version = None	string value	API version of the Identity API endpoint.
cache = None	string value	Request environment key where the Swift cache object is stored. When <code>auth_token</code> middleware is deployed with a Swift cache, use this option to have the middleware share a caching backend with swift. Otherwise, use the <code>memcached_servers</code> option instead.
cafile = None	string value	A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
certfile = None	string value	Required if identity server requires client certificate
delay_auth_decision = False	boolean value	Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components.

Configuration option = Default value	Type	Description
enforce_token_bind = permissive	string value	Used to control the use and type of token binding. Can be set to: "disabled" to not check token binding. "permissive" (default) to validate binding information if the bind type is of a form known to the server and ignore it if not. "strict" like "permissive" but if the bind type is unknown the token will be rejected. "required" any form of token binding is needed to be allowed. Finally the name of a binding method that must be present in tokens.
http_connect_timeout = None	integer value	Request timeout value for communicating with Identity API server.
http_request_max_retries = 3	integer value	How many times are we trying to reconnect when communicating with Identity API Server.
include_service_catalog = True	boolean value	(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	boolean value	Verify HTTPS connections.
interface = internal	string value	Interface to use for the Identity API endpoint. Valid values are "public", "internal" (default) or "admin".
keyfile = None	string value	Required if identity server requires client certificate
memcache_pool_conn_get_timeout = 10	integer value	(Optional) Number of seconds that an operation will wait to get a memcached client connection from the pool.
memcache_pool_dead_retry = 300	integer value	(Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = 10	integer value	(Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = 3	integer value	(Optional) Socket timeout in seconds for communicating with a memcached server.
memcache_pool_unused_timeout = 60	integer value	(Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.

Configuration option = Default value	Type	Description
memcache_secret_key = None	string value	(Optional, mandatory if memcache_security_strategy is defined) This string is used for key derivation.
memcache_security_strategy = None	string value	(Optional) If defined, indicate whether token data should be authenticated or authenticated and encrypted. If MAC, token data is authenticated (with HMAC) in the cache. If ENCRYPT, token data is encrypted and authenticated in the cache. If the value is not one of these options or empty, auth_token will raise an exception on initialization.
memcache_use_advanced_pool = False	boolean value	(Optional) Use the advanced (eventlet safe) memcached client pool. The advanced pool will only work under python 2.x.
memcached_servers = None	list value	Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
region_name = None	string value	The region in which the identity server can be found.
service_token_roles = ['service']	list value	A choice of roles that must be present in a service token. Service tokens are allowed to request that an expired token can be used and so this check should tightly control that only actual services should be sending this token. Roles here are applied as an ANY check so any role in this list must be present. For backwards compatibility reasons this currently only affects the allow_expired check.
service_token_roles_required = False	boolean value	For backwards compatibility reasons we must let valid service tokens pass that don't pass the service_token_roles check as valid. Setting this true will become the default in a future release and should be enabled if possible.
service_type = None	string value	The name or type of the service as it appears in the service catalog. This is used to validate tokens that have restricted access rules.
token_cache_time = 300	integer value	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.

Configuration option = Default value	Type	Description
www_authenticate_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint.

2.1.14. nova

The following table outlines the options available under the **[nova]** group in the **/etc/cinder/cinder.conf** file.

Table 2.13. nova

Configuration option = Default value	Type	Description
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
insecure = False	boolean value	Verify HTTPS connections.
interface = public	string value	Type of the nova endpoint to use. This endpoint will be looked up in the keystone catalog and should be one of public, internal or admin.
keyfile = None	string value	PEM encoded client certificate key file
region_name = None	string value	Name of nova region to use. Useful if keystone manages more than one region.
split-loggers = False	boolean value	Log requests to multiple loggers.

Configuration option = Default value	Type	Description
timeout = None	integer value	Timeout value for http requests
token_auth_url = None	string value	The authentication URL for the nova connection when using the current users token

2.1.15. oslo_concurrency

The following table outlines the options available under the **[oslo_concurrency]** group in the **/etc/cinder/cinder.conf** file.

Table 2.14. oslo_concurrency

Configuration option = Default value	Type	Description
disable_process_locking = False	boolean value	Enables or disables inter-process locks.
lock_path = None	string value	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.

2.1.16. oslo_messaging_amqp

The following table outlines the options available under the **[oslo_messaging_amqp]** group in the **/etc/cinder/cinder.conf** file.

Table 2.15. oslo_messaging_amqp

Configuration option = Default value	Type	Description
addressing_mode = dynamic	string value	Indicates the addressing mode used by the driver. Permitted values: <i>legacy</i> - use legacy non-routable addressing <i>routable</i> - use routable addresses <i>dynamic</i> - use legacy addresses if the message bus does not support routing otherwise use routable addressing
anycast_address = anycast	string value	Appended to the address prefix when sending to a group of consumers. Used by the message bus to identify messages that should be delivered in a round-robin fashion across consumers.

Configuration option = Default value	Type	Description
broadcast_prefix = broadcast	string value	address prefix used when broadcasting to all servers
connection_retry_backoff = 2	integer value	Increase the connection_retry_interval by this many seconds after each unsuccessful failover attempt.
connection_retry_interval = 1	integer value	Seconds to pause before attempting to re-connect.
connection_retry_interval_max = 30	integer value	Maximum limit for connection_retry_interval + connection_retry_backoff
container_name = None	string value	Name for the AMQP container. must be globally unique. Defaults to a generated UUID
default_notification_exchange = None	string value	Exchange name used in notification addresses. Exchange name resolution precedence: Target.exchange if set else default_notification_exchange if set else control_exchange if set else <i>notify</i>
default_notify_timeout = 30	integer value	The deadline for a sent notification message delivery. Only used when caller does not provide a timeout expiry.
default_reply_retry = 0	integer value	The maximum number of attempts to re-send a reply message which failed due to a recoverable error.
default_reply_timeout = 30	integer value	The deadline for an rpc reply message delivery.
default_rpc_exchange = None	string value	Exchange name used in RPC addresses. Exchange name resolution precedence: Target.exchange if set else default_rpc_exchange if set else control_exchange if set else <i>rpc</i>
default_send_timeout = 30	integer value	The deadline for an rpc cast or call message delivery. Only used when caller does not provide a timeout expiry.
default_sender_link_timeout = 600	integer value	The duration to schedule a purge of idle sender links. Detach link after expiry.
group_request_prefix = unicast	string value	address prefix when sending to any server in group

Configuration option = Default value	Type	Description
idle_timeout = 0	integer value	Timeout for inactive connections (in seconds)
link_retry_delay = 10	integer value	Time to pause between re-connecting an AMQP 1.0 link that failed due to a recoverable error.
multicast_address = multicast	string value	Appended to the address prefix when sending a fanout message. Used by the message bus to identify fanout messages.
notify_address_prefix = openstack.org/om/notify	string value	Address prefix for all generated Notification addresses
notify_server_credit = 100	integer value	Window size for incoming Notification messages
pre_settled = ['rpc-cast', 'rpc-reply']	multi valued	Send messages of this type pre-settled. Pre-settled messages will not receive acknowledgement from the peer. Note well: pre-settled messages may be silently discarded if the delivery fails. Permitted values: <i>rpc-call</i> - send RPC Calls pre-settled <i>rpc-reply</i> - send RPC Replies pre-settled <i>rpc-cast</i> - Send RPC Casts pre-settled <i>notify</i> - Send Notifications pre-settled
pseudo_vhost = True	boolean value	Enable virtual host support for those message buses that do not natively support virtual hosting (such as qpidd). When set to true the virtual host name will be added to all message bus addresses, effectively creating a private <i>subnet</i> per virtual host. Set to False if the message bus supports virtual hosting using the <i>hostname</i> field in the AMQP 1.0 Open performative as the name of the virtual host.
reply_link_credit = 200	integer value	Window size for incoming RPC Reply messages.
rpc_address_prefix = openstack.org/om/rpc	string value	Address prefix for all generated RPC addresses
rpc_server_credit = 100	integer value	Window size for incoming RPC Request messages
<code>`sasldb_config_dir = `</code>	string value	Path to directory that contains the SASL configuration
<code>`sasldb_config_name = `</code>	string value	Name of configuration file (without .conf suffix)
<code>`sasldb_default_realm = `</code>	string value	SASL realm to use if no realm present in username

Configuration option = Default value	Type	Description
<code>`sasl_mechanisms = `</code>	string value	Space separated list of acceptable SASL mechanisms
server_request_prefix = exclusive	string value	address prefix used when sending to a specific server
ssl = False	boolean value	Attempt to connect via SSL. If no other ssl-related parameters are given, it will use the system's CA-bundle to verify the server's certificate.
<code>`ssl_ca_file = `</code>	string value	CA certificate PEM file used to verify the server's certificate
<code>`ssl_cert_file = `</code>	string value	Self-identifying certificate PEM file for client authentication
<code>`ssl_key_file = `</code>	string value	Private key PEM file used to sign <code>ssl_cert_file</code> certificate (optional)
ssl_key_password = None	string value	Password for decrypting <code>ssl_key_file</code> (if encrypted)
ssl_verify_vhost = False	boolean value	By default SSL checks that the name in the server's certificate matches the hostname in the <code>transport_url</code> . In some configurations it may be preferable to use the virtual hostname instead, for example if the server uses the Server Name Indication TLS extension (rfc6066) to provide a certificate per virtual host. Set <code>ssl_verify_vhost</code> to True if the server's SSL certificate uses the virtual host name instead of the DNS name.
trace = False	boolean value	Debug: dump AMQP frames to stdout
unicast_address = unicast	string value	Appended to the address prefix when sending to a particular RPC/Notification server. Used by the message bus to identify messages sent to a single destination.

2.1.17. oslo_messaging_kafka

The following table outlines the options available under the **[oslo_messaging_kafka]** group in the `/etc/cinder/cinder.conf` file.

Table 2.16. oslo_messaging_kafka

Configuration option = Default value	Type	Description
compression_codec = none	string value	The compression codec for all data generated by the producer. If not set, compression will not be used. Note that the allowed values of this depend on the kafka version
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
consumer_group = oslo_messaging_consumer	string value	Group id for Kafka consumer. Consumers in one group will coordinate message consumption
enable_auto_commit = False	boolean value	Enable asynchronous consumer commits
kafka_consumer_timeout = 1.0	floating point value	Default timeout(s) for Kafka consumers
kafka_max_fetch_bytes = 1048576	integer value	Max fetch bytes of Kafka consumer
max_poll_records = 500	integer value	The maximum number of records returned in a poll call
pool_size = 10	integer value	Pool Size for Kafka Consumers
producer_batch_size = 16384	integer value	Size of batch for the producer async send
producer_batch_timeout = 0.0	floating point value	Upper bound on the delay for KafkaProducer batching in seconds
sasl_mechanism = PLAIN	string value	Mechanism when security protocol is SASL
security_protocol = PLAINTEXT	string value	Protocol used to communicate with brokers
<code>`ssl_cafile = `</code>	string value	CA certificate PEM file used to verify the server certificate
<code>`ssl_client_cert_file = `</code>	string value	Client certificate PEM file used for authentication.
<code>`ssl_client_key_file = `</code>	string value	Client key PEM file used for authentication.
<code>`ssl_client_key_password = `</code>	string value	Client key password file used for authentication.

2.1.18. oslo_messaging_notifications

The following table outlines the options available under the **[oslo_messaging_notifications]** group in the **/etc/cinder/cinder.conf** file.

Table 2.17. oslo_messaging_notifications

Configuration option = Default value	Type	Description
driver = []	multi valued	The Drivers(s) to handle sending notifications. Possible values are messaging, messagingv2, routing, log, test, noop
retry = -1	integer value	The maximum number of attempts to re-send a notification message which failed to be delivered due to a recoverable error. 0 - No retry, -1 - indefinite
topics = ['notifications']	list value	AMQP topic used for OpenStack notifications.
transport_url = None	string value	A URL representing the messaging driver to use for notifications. If not set, we fall back to the same configuration used for RPC.

2.1.19. oslo_messaging_rabbit

The following table outlines the options available under the **[oslo_messaging_rabbit]** group in the **/etc/cinder/cinder.conf** file.

Table 2.18. oslo_messaging_rabbit

Configuration option = Default value	Type	Description
amqp_auto_delete = False	boolean value	Auto-delete queues in AMQP.
amqp_durable_queues = False	boolean value	Use durable queues in AMQP.
direct_mandatory_flag = True	boolean value	(DEPRECATED) Enable/Disable the RabbitMQ mandatory flag for direct send. The direct send is used as reply, so the MessageUndeliverable exception is raised in case the client queue does not exist. MessageUndeliverable exception will be used to loop for a timeout to let a chance to sender to recover. This flag is deprecated and it will not be possible to deactivate this functionality anymore
enable_cancel_on_failover = False	boolean value	Enable x-cancel-on-ha-failover flag so that rabbitmq server will cancel and notify consumers when queue is down

Configuration option = Default value	Type	Description
heartbeat_in_pthread = True	boolean value	Run the health check heartbeat thread through a native python thread by default. If this option is equal to False then the health check heartbeat will inherit the execution model from the parent process. For example if the parent process has monkey patched the stdlib by using eventlet/greenlet then the heartbeat will be run through a green thread.
heartbeat_rate = 2	integer value	How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = 60	integer value	Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables heartbeat).
kombu_compression = None	string value	EXPERIMENTAL: Possible values are: gzip, bz2. If not set compression will not be used. This option may not be available in future versions.
kombu_failover_strategy = round-robin	string value	Determines how the next RabbitMQ node is chosen in case the one we are currently connected to becomes unavailable. Takes effect only if more than one RabbitMQ node is provided in config.
kombu_missing_consumer_retry_timeout = 60	integer value	How long to wait a missing client before abandoning to send it its replies. This value should not be longer than rpc_response_timeout.
kombu_reconnect_delay = 1.0	floating point value	How long to wait before reconnecting in response to an AMQP consumer cancel notification.
rabbit_ha_queues = False	boolean value	Try to use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database. In RabbitMQ 3.0, queue mirroring is no longer controlled by the x-ha-policy argument when declaring a queue. If you just want to make sure that all queues (except those with auto-generated names) are mirrored across all nodes, run: <code>"rabbitmqctl set_policy HA ^(?!amq\.).* {\"ha-mode\": \"all\"}"</code>
rabbit_interval_max = 30	integer value	Maximum interval of RabbitMQ connection retries. Default is 30 seconds.
rabbit_login_method = AMQPLAIN	string value	The RabbitMQ login method.

Configuration option = Default value	Type	Description
rabbit_qos_prefetch_count = 0	integer value	Specifies the number of messages to prefetch. Setting to zero allows unlimited messages.
rabbit_retry_backoff = 2	integer value	How long to backoff for between retries when connecting to RabbitMQ.
rabbit_retry_interval = 1	integer value	How frequently to retry connecting with RabbitMQ.
rabbit_transient_queues_ttl = 1800	integer value	Positive integer representing duration in seconds for queue TTL (x-expires). Queues which are unused for the duration of the TTL are automatically deleted. The parameter affects only reply and fanout queues.
ssl = False	boolean value	Connect over SSL.
<code>`ssl_ca_file = `</code>	string value	SSL certification authority file (valid only if SSL enabled).
<code>`ssl_cert_file = `</code>	string value	SSL cert file (valid only if SSL enabled).
<code>`ssl_key_file = `</code>	string value	SSL key file (valid only if SSL enabled).
<code>`ssl_version = `</code>	string value	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.

2.1.20. oslo_middleware

The following table outlines the options available under the **[oslo_middleware]** group in the `/etc/cinder/cinder.conf` file.

Table 2.19. oslo_middleware

Configuration option = Default value	Type	Description
enable_proxy_headers_parsing = False	boolean value	Whether the application is behind a proxy or not. This determines if the middleware should parse the headers or not.
max_request_body_size = 114688	integer value	The maximum body size for each request, in bytes.

Configuration option = Default value	Type	Description
secure_proxy_ssl_header = X-Forwarded-Proto	string value	The HTTP Header that will be used to determine what the original request protocol scheme was, even if it was hidden by a SSL termination proxy.

2.1.21. oslo_policy

The following table outlines the options available under the **[oslo_policy]** group in the **/etc/cinder/cinder.conf** file.

Table 2.20. oslo_policy

Configuration option = Default value	Type	Description
enforce_new_defaults = False	boolean value	This option controls whether or not to use old deprecated defaults when evaluating policies. If True , the old deprecated defaults are not going to be evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be disallowed. It is encouraged to enable this flag along with the enforce_scope flag so that you can get the benefits of new defaults and scope_type together
enforce_scope = False	boolean value	This option controls whether or not to enforce scope when evaluating policies. If True , the scope of the token used in the request is compared to the scope_types of the policy being enforced. If the scopes do not match, an InvalidScope exception will be raised. If False , a message will be logged informing operators that policies are being invoked with mismatching scope.
policy_default_rule = default	string value	Default rule. Enforced when a requested rule is not found.
policy_dirs = ['policy.d']	multi valued	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.yaml	string value	The relative or absolute path of a file that maps roles to permissions for a given service. Relative paths must be specified in relation to the configuration file setting this option.

Configuration option = Default value	Type	Description
remote_content_type = application/x-www-form-urlencoded	string value	Content Type to send and receive data for REST based policy check
remote_ssl_ca_cert_file = None	string value	Absolute path to ca cert file for REST based policy check
remote_ssl_client_cert_file = None	string value	Absolute path to client cert for REST based policy check
remote_ssl_client_key_file = None	string value	Absolute path client key file REST based policy check
remote_ssl_verify_server_cert = False	boolean value	server identity verification for REST based policy check

2.1.22. oslo_reports

The following table outlines the options available under the **[oslo_reports]** group in the **/etc/cinder/cinder.conf** file.

Table 2.21. oslo_reports

Configuration option = Default value	Type	Description
file_event_handler = None	string value	The path to a file to watch for changes to trigger the reports, instead of signals. Setting this option disables the signal trigger for the reports. If application is running as a WSGI application it is recommended to use this instead of signals.
file_event_handler_interval = 1	integer value	How many seconds to wait between polls when file_event_handler is set
log_dir = None	string value	Path to a log directory where to create a file

2.1.23. oslo_versionedobjects

The following table outlines the options available under the **[oslo_versionedobjects]** group in the **/etc/cinder/cinder.conf** file.

Table 2.22. oslo_versionedobjects

Configuration option = Default value	Type	Description
fatal_exception_format_errors = False	boolean value	Make exception message format errors fatal

2.1.24. privsep

The following table outlines the options available under the **[privsep]** group in the **/etc/cinder/cinder.conf** file.

Table 2.23. privsep

Configuration option = Default value	Type	Description
capabilities = []	list value	List of Linux capabilities retained by the privsep daemon.
group = None	string value	Group that the privsep daemon should run as.
helper_command = None	string value	Command to invoke to start the privsep daemon if not using the "fork" method. If not specified, a default is generated using "sudo privsep-helper" and arguments designed to recreate the current configuration. This command must accept suitable <code>--privsep_context</code> and <code>--privsep_sock_path</code> arguments.
thread_pool_size = <based on operating system>	integer value	The number of threads available for privsep to concurrently run processes. Defaults to the number of CPU cores in the system.
user = None	string value	User that the privsep daemon should run as.

2.1.25. profiler

The following table outlines the options available under the **[profiler]** group in the **/etc/cinder/cinder.conf** file.

Table 2.24. profiler

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
connection_string = messaging://	string value	Connection string for a notifier backend. Default value is messaging:// which sets the notifier to oslo_messaging. Examples of possible values: ● messaging:// - use oslo_messaging driver for sending spans. ● redis://127.0.0.1:6379 - use redis driver for sending spans. ● mongodb://127.0.0.1:27017 - use mongodb driver for sending spans. ● elasticsearch://127.0.0.1:9200 - use elasticsearch driver for sending spans. ● jaeger://127.0.0.1:6831 - use jaeger tracing as driver for sending spans.
enabled = False	boolean value	Enable the profiling for all services on this node. Default value is False (fully disable the profiling feature). Possible values: ● True: Enables the feature ● False: Disables the feature. The profiling cannot be started via this project operations. If the profiling is triggered by another project, this project part will be empty.
es_doc_type = notification	string value	Document type for notification indexing in elasticsearch.
es_scroll_size = 10000	integer value	Elasticsearch splits large requests in batches. This parameter defines maximum size of each batch (for example: es_scroll_size=10000).
es_scroll_time = 2m	string value	This parameter is a time value parameter (for example: es_scroll_time=2m), indicating for how long the nodes that participate in the search will maintain relevant resources in order to continue and support it.

Configuration option = Default value	Type	Description
filter_error_trace = False	boolean value	Enable filter traces that contain error/exception to a separated place. Default value is set to False. Possible values: • True: Enable filter traces that contain error/exception. • False: Disable the filter.
hmac_keys = SECRET_KEY	string value	Secret key(s) to use for encrypting context data for performance profiling. This string value should have the following format: <key1>[,<key2>,...<keyn>], where each key is some random string. A user who triggers the profiling via the REST API has to set one of these keys in the headers of the REST API call to include profiling results of this node for this particular project. Both "enabled" flag and "hmac_keys" config options should be set to enable profiling. Also, to generate correct profiling information across all services at least one key needs to be consistent between OpenStack projects. This ensures it can be used from client side to generate the trace, containing information from all possible resources.
sentinel_service_name = mymaster	string value	Redis sentinel uses a service name to identify a master redis service. This parameter defines the name (for example: sentinal_service_name=mymaster).
socket_timeout = 0.1	floating point value	Redis sentinel provides a timeout option on the connections. This parameter defines that timeout (for example: socket_timeout=0.1).

Configuration option = Default value	Type	Description
trace_sqlalchemy = False	boolean value	Enable SQL requests profiling in services. Default value is False (SQL requests won't be traced). Possible values: • True: Enables SQL requests profiling. Each SQL query will be part of the trace and can be analyzed by how much time was spent for that. • False: Disables SQL requests profiling. The spent time is only shown on a higher level of operations. Single SQL queries cannot be analyzed this way.

2.1.26. sample_castellan_source

The following table outlines the options available under the **[sample_castellan_source]** group in the **/etc/cinder/cinder.conf** file.

Table 2.25. sample_castellan_source

Configuration option = Default value	Type	Description
config_file = None	string value	The path to a castellan configuration file.
driver = None	string value	The name of the driver that can load this configuration source.
mapping_file = None	string value	The path to a configuration/castellan_id mapping file.

2.1.27. sample_remote_file_source

The following table outlines the options available under the **[sample_remote_file_source]** group in the **/etc/cinder/cinder.conf** file.

Table 2.26. sample_remote_file_source

Configuration option = Default value	Type	Description
ca_path = None	string value	The path to a CA_BUNDLE file or directory with certificates of trusted CAs.

Configuration option = Default value	Type	Description
client_cert = None	string value	Client side certificate, as a single file path containing either the certificate only or the private key and the certificate.
client_key = None	string value	Client side private key, in case client_cert is specified but does not includes the private key.
driver = None	string value	The name of the driver that can load this configuration source.
uri = None	uri value	Required option with the URI of the extra configuration file's location.

2.1.28. service_user

The following table outlines the options available under the **[service_user]** group in the `/etc/cinder/cinder.conf` file.

Table 2.27. service_user

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project

Configuration option = Default value	Type	Description
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
send_service_user_token = False	boolean value	When True, if sending a user token to an REST API, also send a service token.
split-loggers = False	boolean value	Log requests to multiple loggers.
system-scope = None	string value	Scope for system operations
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User ID
username = None	string value	Username

2.1.29. ssl

The following table outlines the options available under the **[ssl]** group in the `/etc/cinder/cinder.conf` file.

Table 2.28. ssl

Configuration option = Default value	Type	Description
ca_file = None	string value	CA certificate file to use to verify connecting clients.
cert_file = None	string value	Certificate file to use when starting the server securely.
ciphers = None	string value	Sets the list of available ciphers. value should be a string in the OpenSSL cipher list format.

Configuration option = Default value	Type	Description
key_file = None	string value	Private key file to use when starting the server securely.
version = None	string value	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.

2.1.30. vault

The following table outlines the options available under the **[vault]** group in the `/etc/cinder/cinder.conf` file.

Table 2.29. vault

Configuration option = Default value	Type	Description
approle_role_id = None	string value	AppRole role_id for authentication with vault
approle_secret_id = None	string value	AppRole secret_id for authentication with vault
kv_mountpoint = secret	string value	Mountpoint of KV store in Vault to use, for example: secret
kv_version = 2	integer value	Version of KV store in Vault to use, for example: 2
root_token_id = None	string value	root token for vault
ssl_ca_cert_file = None	string value	Absolute path to ca cert file
use_ssl = False	boolean value	SSL Enabled/Disabled
vault_url = http://127.0.0.1:8200	string value	Use this endpoint to connect to Vault, for example: "http://127.0.0.1:8200"

CHAPTER 3. GLANCE

The following chapter contains information about the configuration options in the **glance** service.

3.1. GLANCE-API.CONF

This section contains options for the `/etc/glance/glance-api.conf` file.

3.1.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/glance/glance-api.conf` file.

Configuration option = Default value	Type	Description
allow_additional_image_properties = True	boolean value	Allow users to add additional/custom properties to images. Glance defines a standard set of properties (in its schema) that appear on every image. These properties are also known as base properties. In addition to these properties, Glance allows users to add custom properties to images. These are known as additional properties. By default, this configuration option is set to True and users are allowed to add additional properties. The number of additional properties that can be added to an image can be controlled via image_property_quota configuration option. Possible values: • True • False Related options: • image_property_quota Deprecated since: Ussuri Reason: This option is redundant. Control custom image property usage via the <i>image_property_quota</i> configuration option. This option is scheduled to be removed during the Victoria development cycle.

Configuration option = Default value	Type	Description
allow_anonymous_access = False	boolean value	Allow limited access to unauthenticated users. Assign a boolean to determine API access for unauthenticated users. When set to False, the API cannot be accessed by unauthenticated users. When set to True, unauthenticated users can access the API with read-only privileges. This however only applies when using ContextMiddleware. Possible values: • True • False Related options: • None
api_limit_max = 1000	integer value	Maximum number of results that could be returned by a request. As described in the help text of limit_param_default, some requests may return multiple results. The number of results to be returned are governed either by the limit parameter in the request or the limit_param_default configuration option. The value in either case, can't be greater than the absolute maximum defined by this configuration option. Anything greater than this value is trimmed down to the maximum value defined here.  NOTE Setting this to a very large value may slow down database queries and increase response times. Setting this to a very low value may result in poor user experience. Possible values: • Any positive integer Related options: • limit_param_default

Configuration option = Default value	Type	Description
backlog = 4096	integer value	Set the number of incoming connection requests. Provide a positive integer value to limit the number of requests in the backlog queue. The default queue size is 4096. An incoming connection to a TCP listener socket is queued before a connection can be established with the server. Setting the backlog for a TCP socket ensures a limited queue size for incoming traffic. Possible values: ● Positive integer Related options: ● None
bind_host = 0.0.0.0	host address value	IP address to bind the glance servers to. Provide an IP address to bind the glance server to. The default value is 0.0.0.0. Edit this option to enable the server to listen on one particular IP address on the network card. This facilitates selection of a particular network interface for the server. Possible values: ● A valid IPv4 address ● A valid IPv6 address Related options: ● None

Configuration option = Default value	Type	Description
bind_port = None	port value	Port number on which the server will listen. Provide a valid port number to bind the server's socket to. This port is then set to identify processes and forward network messages that arrive at the server. The default bind_port value for the API server is 9292 and for the registry server is 9191. Possible values: • A valid port number (0 to 65535) Related options: • None
client_socket_timeout = 900	integer value	Timeout for client connections' socket operations. Provide a valid integer value representing time in seconds to set the period of wait before an incoming connection can be closed. The default value is 900 seconds. The value zero implies wait forever. Possible values: • Zero • Positive integer Related options: • None
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
control_exchange = openstack	string value	The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.

Configuration option = Default value	Type	Description
default_log_levels = <code>['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' ', 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']</code>	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
default_publisher_id = image.localhost	string value	Default publisher_id for outgoing Glance notifications. This is the value that the notification driver will use to identify messages for events originating from the Glance service. Typically, this is the hostname of the instance that generated the message. Possible values: - Any reasonable instance identifier, for example: image.host1 Related options: - None

Configuration option = Default value	Type	Description
delayed_delete = False	boolean value	Turn on/off delayed delete. Typically when an image is deleted, the glance-api service puts the image into deleted state and deletes its data at the same time. Delayed delete is a feature in Glance that delays the actual deletion of image data until a later point in time (as determined by the configuration option scrub_time). When delayed delete is turned on, the glance-api service puts the image into pending_delete state upon deletion and leaves the image data in the storage backend for the image scrubber to delete at a later time. The image scrubber will move the image into deleted state upon successful deletion of image data.  NOTE When delayed delete is turned on, image scrubber MUST be running as a periodic task to prevent the backend storage from filling up with undesired usage. Possible values: • True • False Related options: • scrub_time • wakeup_time • scrub_pool_size

Configuration option = Default value	Type	Description
digest_algorithm = sha256	string value	Digest algorithm to use for digital signature. Provide a string value representing the digest algorithm to use for generating digital signatures. By default, sha256 is used. To get a list of the available algorithms supported by the version of OpenSSL on your platform, run the command: openssl list-message-digest-algorithms. Examples are <i>sha1</i>, <i>sha256</i>, and <i>sha512</i>.  NOTE digest_algorithm is not related to Glance's image signing and verification. It is only used to sign the universally unique identifier (UUID) as a part of the certificate file and key file validation. Possible values: ● An OpenSSL message digest algorithm identifier Relation options: ● None

Configuration option = Default value	Type	Description
disabled_notifications = []	list value	List of notifications to be disabled. Specify a list of notifications that should not be emitted. A notification can be given either as a notification type to disable a single event notification, or as a notification group prefix to disable all event notifications within a group. Possible values: A comma-separated list of individual notification types or notification groups to be disabled. Currently supported groups: ● image ● image.member ● task ● metadef_namespace ● metadef_object ● metadef_property ● metadef_resource_type ● metadef_tag For a complete listing and description of each event refer to: http://docs.openstack.org/developer/glance/notifications.html The values must be specified as: <group_name>.<event_name> For example: image.create,task.success,metadef_tag Related options: ● None
enabled_backends = None	dict value	Key:Value pair of store identifier and store type. In case of multiple backends should be separated using comma.

Configuration option = Default value	Type	Description
enabled_import_methods = ['glance-direct', 'web-download', 'copy-image']	list value	List of enabled Image Import Methods 'glance-direct', 'copy-image' and 'web-download' are enabled by default. Related options: ** [DEFAULT]/node_staging_uri
enforce_secure_rbac = False	boolean value	Enforce API access based on common persona definitions used across OpenStack. Enabling this option formalizes project-specific read/write operations, like creating private images or updating the status of shared image, behind the member role. It also formalizes a read-only variant useful for project-specific API operations, like listing private images in a project, behind the reader role. Operators should take an opportunity to understand glance's new image policies, audit assignments in their deployment, and update permissions using the default roles in keystone (e.g., admin, member, and reader). Related options: • [oslo_policy]/enforce_new_defaults Deprecated since: Wallaby Reason: This option has been introduced to require operators to opt into enforcing authorization based on common RBAC personas, which is EXPERIMENTAL as of the Wallaby release. This behavior will be the default and STABLE in a future release, allowing this option to be removed.
executor_thread_pool_size = 64	integer value	Size of executor thread pool when executor is threading or eventlet.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.

Configuration option = Default value	Type	Description
hashing_algorithm = sha512	string value	Secure hashing algorithm used for computing the <i>os_hash_value</i> property. This option configures the Glance "multihash", which consists of two image properties: the <i>os_hash_algo</i> and the <i>os_hash_value</i>. The <i>os_hash_algo</i> will be populated by the value of this configuration option, and the <i>os_hash_value</i> will be populated by the hexdigest computed when the algorithm is applied to the uploaded or imported image data. The value must be a valid secure hash algorithm name recognized by the python <i>hashlib</i> library. You can determine what these are by examining the <i>hashlib.algorithms_available</i> data member of the version of the library being used in your Glance installation. For interoperability purposes, however, we recommend that you use the set of secure hash names supplied by the <i>hashlib.algorithms_guaranteed</i> data member because those algorithms are guaranteed to be supported by the <i>hashlib</i> library on all platforms. Thus, any image consumer using <i>hashlib</i> locally should be able to verify the <i>os_hash_value</i> of the image. The default value of <i>sha512</i> is a performant secure hash algorithm. If this option is misconfigured, any attempts to store image data will fail. For that reason, we recommend using the default value. Possible values: - Any secure hash algorithm name recognized by the Python <i>hashlib</i> library Related options: - None

Configuration option = Default value	Type	Description
http_keepalive = True	boolean value	Set keep alive option for HTTP over TCP. Provide a boolean value to determine sending of keep alive packets. If set to False, the server returns the header "Connection: close". If set to True, the server returns a "Connection: Keep-Alive" in its responses. This enables retention of the same TCP connection for HTTP conversations instead of opening a new one with each new request. This option must be set to False if the client socket connection needs to be closed explicitly after the response is received and read successfully by the client. Possible values: • True • False Related options: • None

Configuration option = Default value	Type	Description
image_cache_dir = None	string value	Base directory for image cache. This is the location where image data is cached and served out of. All cached images are stored directly under this directory. This directory also contains three subdirectories, namely, incomplete, invalid and queue. The incomplete subdirectory is the staging area for downloading images. An image is first downloaded to this directory. When the image download is successful it is moved to the base directory. However, if the download fails, the partially downloaded image file is moved to the invalid subdirectory. The queue subdirectory is used for queuing images for download. This is used primarily by the cache-prefetcher, which can be scheduled as a periodic task like cache-pruner and cache-cleaner, to cache images ahead of their usage. Upon receiving the request to cache an image, Glance touches a file in the queue directory with the image id as the file name. The cache-prefetcher, when running, polls for the files in queue directory and starts downloading them in the order they were created. When the download is successful, the zero-sized file is deleted from the queue directory. If the download fails, the zero-sized file remains and it'll be retried the next time cache-prefetcher runs. Possible values: • A valid path Related options: • image_cache_sqlite_db

Configuration option = Default value	Type	Description
image_cache_driver = sqlite	string value	The driver to use for image cache management. This configuration option provides the flexibility to choose between the different image-cache drivers available. An image-cache driver is responsible for providing the essential functions of image-cache like write images to/read images from cache, track age and usage of cached images, provide a list of cached images, fetch size of the cache, queue images for caching and clean up the cache, etc. The essential functions of a driver are defined in the base class glance.image_cache.drivers.base.Driver. All image-cache drivers (existing and prospective) must implement this interface. Currently available drivers are sqlite and xattr. These drivers primarily differ in the way they store the information about cached images: • The sqlite driver uses a sqlite database (which sits on every glance node locally) to track the usage of cached images. • The xattr driver uses the extended attributes of files to store this information. It also requires a filesystem that sets atime on the files when accessed. Possible values: • sqlite • xattr Related options: • None

Configuration option = Default value	Type	Description
image_cache_max_size = 10737418240	integer value	The upper limit on cache size, in bytes, after which the cache-pruner cleans up the image cache.  NOTE This is just a threshold for cache-pruner to act upon. It is NOT a hard limit beyond which the image cache would never grow. In fact, depending on how often the cache-pruner runs and how quickly the cache fills, the image cache can far exceed the size specified here very easily. Hence, care must be taken to appropriately schedule the cache-pruner and in setting this limit. Glance caches an image when it is downloaded. Consequently, the size of the image cache grows over time as the number of downloads increases. To keep the cache size from becoming unmanageable, it is recommended to run the cache-pruner as a periodic task. When the cache pruner is kicked off, it compares the current size of image cache and triggers a cleanup if the image cache grew beyond the size specified here. After the cleanup, the size of cache is less than or equal to size specified here. Possible values: - Any non-negative integer Related options: - None

Configuration option = Default value	Type	Description
image_cache_sqlite_db = cache.db	string value	The relative path to sqlite file database that will be used for image cache management. This is a relative path to the sqlite file database that tracks the age and usage statistics of image cache. The path is relative to image cache base directory, specified by the configuration option image_cache_dir. This is a lightweight database with just one table. Possible values: • A valid relative path to sqlite file database Related options: • image_cache_dir
image_cache_stall_time = 86400	integer value	The amount of time, in seconds, an incomplete image remains in the cache. Incomplete images are images for which download is in progress. Please see the description of configuration option image_cache_dir for more detail. Sometimes, due to various reasons, it is possible the download may hang and the incompletely downloaded image remains in the incomplete directory. This configuration option sets a time limit on how long the incomplete images should remain in the incomplete directory before they are cleaned up. Once an incomplete image spends more time than is specified here, it'll be removed by cache-cleaner on its next run. It is recommended to run cache-cleaner as a periodic task on the Glance API nodes to keep the incomplete images from occupying disk space. Possible values: • Any non-negative integer Related options: • None

Configuration option = Default value	Type	Description
image_location_quota = 10	integer value	Maximum number of locations allowed on an image. Any negative value is interpreted as unlimited. Related options: • None
image_member_quota = 128	integer value	Maximum number of image members per image. This limits the maximum of users an image can be shared with. Any negative value is interpreted as unlimited. Related options: • None
image_property_quota = 128	integer value	Maximum number of properties allowed on an image. This enforces an upper limit on the number of additional properties an image can have. Any negative value is interpreted as unlimited.  NOTE This won't have any impact if additional properties are disabled. Please refer to allow_additional_image_properties. Related options: • allow_additional_image_properties

Configuration option = Default value	Type	Description
image_size_cap = 1099511627776	integer value	Maximum size of image a user can upload in bytes. An image upload greater than the size mentioned here would result in an image creation failure. This configuration option defaults to 1099511627776 bytes (1 TiB). NOTES: • This value should only be increased after careful consideration and must be set less than or equal to 8 EiB (9223372036854775808). • This value must be set with careful consideration of the backend storage capacity. Setting this to a very low value may result in a large number of image failures. And, setting this to a very large value may result in faster consumption of storage. Hence, this must be set according to the nature of images created and storage capacity available. Possible values: • Any positive number less than or equal to 9223372036854775808
image_tag_quota = 128	integer value	Maximum number of tags allowed on an image. Any negative value is interpreted as unlimited. Related options: • None
<code>`instance_format = [instance: %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.
<code>`instance_uuid_format = [instance: %(uuid)s] `</code>	string value	The format for an instance UUID that is passed with the log message.

Configuration option = Default value	Type	Description
limit_param_default = 25	integer value	The default number of results to return for a request. Responses to certain API requests, like list images, may return multiple items. The number of results returned can be explicitly controlled by specifying the limit parameter in the API request. However, if a limit parameter is not specified, this configuration value will be used as the default number of results to be returned for any API request. NOTES: ● The value of this configuration option may not be greater than the value specified by api_limit_max. ● Setting this to a very large value may slow down database queries and increase response times. Setting this to a very low value may result in poor user experience. Possible values: ● Any positive integer Related options: ● <code>api_limit_max</code>

Configuration option = Default value	Type	Description
location_strategy = location_order	string value	Strategy to determine the preference order of image locations. This configuration option indicates the strategy to determine the order in which an image's locations must be accessed to serve the image's data. Glance then retrieves the image data from the first responsive active location it finds in this list. This option takes one of two possible values location_order and store_type. The default value is location_order, which suggests that image data be served by using locations in the order they are stored in Glance. The store_type value sets the image location preference based on the order in which the storage backends are listed as a comma separated list for the configuration option store_type_preference. Possible values: ● location_order ● store_type Related options: ● store_type_preference
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for %(asctime)s in log records. Default: %(default)s . This option is ignored if log_config_append is set.
log-dir = None	string value	(Optional) The base directory used for relative log_file paths. This option is ignored if log_config_append is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to stderr as defined by use_stderr. This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless log_rotation_type is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.% (msecs)03d %(process)d %(levelname)s %(name)s [% (request_id)s % (user_identity)s] % (instance)s%(message)s	string value	Format string to use for log messages with context. Used by oslo_log.formatters.ContextFormatter
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is DEBUG. Used by oslo_log.formatters.ContextFormatter
logging_default_format_s tring = %(asctime)s.% (msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s% (message)s	string value	Format string to use for log messages when context is undefined. Used by oslo_log.formatters.ContextFormatter
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	string value	Prefix each line of exception output with this format. Used by oslo_log.formatters.ContextFormatter
logging_user_identity_for mat = %(user)s % (tenant)s %(domain)s % (user_domain)s % (project_domain)s	string value	Defines the format string for %(user_identity)s that is used in logging_context_format_string. Used by oslo_log.formatters.ContextFormatter

Configuration option = Default value	Type	Description
max_header_line = 16384	integer value	Maximum line size of message headers. Provide an integer value representing a length to limit the size of message headers. The default value is 16384.  NOTE max_header_line may need to be increased when using large tokens (typically those generated by the Keystone v3 API with big service catalogs). However, it is to be kept in mind that larger values for max_header_line would flood the logs. Setting max_header_line to 0 sets no limit for the line size of message headers. Possible values: • 0 • Positive integer Related options: • None
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
max_request_id_length = 64	integer value	Limit the request ID length. Provide an integer value to limit the length of the request ID to the specified length. The default value is 64. Users can change this to any integer value between 0 and 16384 however keeping in mind that a larger value may flood the logs. Possible values: • Integer value between 0 and 16384 Related options: • None

Configuration option = Default value	Type	Description
metadata_encryption_key = None	string value	AES key for encrypting store location metadata. Provide a string value representing the AES cipher to use for encrypting Glance store metadata. NOTE The AES key to use must be set to a random string of length 16, 24 or 32 bytes. Possible values: • String value representing a valid AES key Related options: • None

Configuration option = Default value	Type	Description
node_staging_uri = file:///tmp/staging/	string value	The URL provides location where the temporary data will be stored This option is for Glance internal use only. Glance will save the image data uploaded by the user to <i>staging</i> endpoint during the image import process. This option does not change the <i>staging</i> API endpoint by any means. NOTE It is discouraged to use same path as [task]/work_dir NOTE <i>file://<absolute-directory-path></i> is the only option api_image_import flow will support for now. NOTE The staging path must be on shared filesystem available to all Glance API nodes. Possible values: ● String starting with <i>file://</i> followed by absolute FS path Related options: ● [task]/work_dir

Configuration option = Default value	Type	Description
property_protection_file = None	string value	The location of the property protection file. Provide a valid path to the property protection file which contains the rules for property protections and the roles/policies associated with them. A property protection file, when set, restricts the Glance image properties to be created, read, updated and/or deleted by a specific set of users that are identified by either roles or policies. If this configuration option is not set, by default, property protections won't be enforced. If a value is specified and the file is not found, the glance-api service will fail to start. More information on property protections can be found at: https://docs.openstack.org/glance/latest/admin/property-protections.html Possible values: ● Empty string ● Valid path to the property protection configuration file Related options: ● property_protection_rule_format

Configuration option = Default value	Type	Description
property_protection_rule_format = roles	string value	Rule format for property protection. Provide the desired way to set property protection on Glance image properties. The two permissible values are roles and policies. The default value is roles. If the value is roles, the property protection file must contain a comma separated list of user roles indicating permissions for each of the CRUD operations on each property being protected. If set to policies, a policy defined in policy.yaml is used to express property protections for each of the CRUD operations. Examples of how property protections are enforced based on roles or policies can be found at: https://docs.openstack.org/glance/latest/admin/property-protections.html#examples Possible values: • roles • policies Related options: • property_protection_file

Configuration option = Default value	Type	Description
public_endpoint = None	string value	Public url endpoint to use for Glance versions response. This is the public url endpoint that will appear in the Glance "versions" response. If no value is specified, the endpoint that is displayed in the version's response is that of the host running the API service. Change the endpoint to represent the proxy URL if the API service is running behind a proxy. If the service is running behind a load balancer, add the load balancer's URL for this value. Possible values: • None • Proxy URL • Load balancer URL Related options: • None
publish_errors = False	boolean value	Enables or disables publication of error events.
pydev_worker_debug_host = None	host address value	Host address of the pydev server. Provide a string value representing the hostname or IP of the pydev server to use for debugging. The pydev server listens for debug connections on this address, facilitating remote debugging in Glance. Possible values: • Valid hostname • Valid IP address Related options: • None

Configuration option = Default value	Type	Description
pydev_worker_debug_port = 5678	port value	Port number that the pydev server will listen on. Provide a port number to bind the pydev server to. The pydev process accepts debug connections on this port and facilitates remote debugging in Glance. Possible values: • A valid port number Related options: • None
rate_limit_burst = 0	integer value	Maximum number of logged messages per rate_limit_interval.
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
rpc_conn_pool_size = 30	integer value	Size of RPC connection pool.
rpc_ping_enabled = False	boolean value	Add an endpoint to answer to ping calls. Endpoint is named oslo_rpc_server_ping
rpc_response_timeout = 60	integer value	Seconds to wait for a response from a call.

Configuration option = Default value	Type	Description
scrub_pool_size = 1	integer value	The size of thread pool to be used for scrubbing images. When there are a large number of images to scrub, it is beneficial to scrub images in parallel so that the scrub queue stays in control and the backend storage is reclaimed in a timely fashion. This configuration option denotes the maximum number of images to be scrubbed in parallel. The default value is one, which signifies serial scrubbing. Any value above one indicates parallel scrubbing. Possible values: - Any non-zero positive integer Related options: delayed_delete
scrub_time = 0	integer value	The amount of time, in seconds, to delay image scrubbing. When delayed delete is turned on, an image is put into pending_delete state upon deletion until the scrubber deletes its image data. Typically, soon after the image is put into pending_delete state, it is available for scrubbing. However, scrubbing can be delayed until a later point using this configuration option. This option denotes the time period an image spends in pending_delete state before it is available for scrubbing. It is important to realize that this has storage implications. The larger the scrub_time, the longer the time to reclaim backend storage from deleted images. Possible values: - Any non-negative integer Related options: delayed_delete
secure_proxy_ssl_header = None	string value	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".

Configuration option = Default value	Type	Description
show_image_direct_url = False	boolean value	Show direct image location when returning an image. This configuration option indicates whether to show the direct image location when returning image details to the user. The direct image location is where the image data is stored in backend storage. This image location is shown under the image property direct_url. When multiple image locations exist for an image, the best location is displayed based on the location strategy indicated by the configuration option location_strategy. NOTES: ● Revealing image locations can present a GRAVE SECURITY RISK as image locations can sometimes include credentials. Hence, this is set to False by default. Set this to True with EXTREME CAUTION and ONLY IF you know what you are doing! ● If an operator wishes to avoid showing any image location(s) to the user, then both this option and show_multiple_locations MUST be set to False. Possible values: ● True ● False Related options: ● show_multiple_locations ● location_strategy

Configuration option = Default value	Type	Description
show_multiple_locations = False	boolean value	Show all image locations when returning an image. This configuration option indicates whether to show all the image locations when returning image details to the user. When multiple image locations exist for an image, the locations are ordered based on the location strategy indicated by the configuration opt location_strategy. The image locations are shown under the image property locations. NOTES: ● Revealing image locations can present a GRAVE SECURITY RISK as image locations can sometimes include credentials. Hence, this is set to False by default. Set this to True with EXTREME CAUTION and ONLY IF you know what you are doing! ● See https://wiki.openstack.org/wiki/OSSN/OSSN-0065 for more information. ● If an operator wishes to avoid showing any image location(s) to the user, then both this option and show_image_direct_url MUST be set to False. Possible values: ● True ● False Related options: ● show_image_direct_url ● location_strategy Deprecated since: Newton *Reason:*Use of this option, deprecated since Newton, is a security risk and will be removed once we figure out a way to satisfy those use cases that currently require it. An earlier announcement that the same functionality can be achieved with greater granularity by using policies is incorrect. You cannot work around this option via policy configuration at the present time, though that is the direction we believe the fix will take. Please keep an eye on the Glance release notes to stay up to date on progress in addressing this issue.

Configuration option = Default value	Type	Description
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
tcp_keepidle = 600	integer value	Set the wait time before a connection recheck. Provide a positive integer value representing time in seconds which is set as the idle wait time before a TCP keep alive packet can be sent to the host. The default value is 600 seconds. Setting tcp_keepidle helps verify at regular intervals that a connection is intact and prevents frequent TCP connection reestablishment. Possible values: - Positive integer value representing time in seconds Related options: - None
transport_url = rabbit://	string value	The network address and optional user credentials for connecting to the messaging backend, in URL format. The expected format is: driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query Example: rabbit://rabbitmq:password@127.0.0.1:5672// For full details on the fields in the URL see the documentation of oslo_messaging.TransportURL at https://docs.openstack.org/oslo.messaging/latest/reference/transport.html
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
user_storage_quota = 0	string value	Maximum amount of image storage per tenant. This enforces an upper limit on the cumulative storage consumed by all images of a tenant across all stores. This is a per-tenant limit. The default unit for this configuration option is Bytes. However, storage units can be specified using case-sensitive literals B, KB, MB, GB and TB representing Bytes, KiloBytes, MegaBytes, GigaBytes and TeraBytes respectively. Note that there should not be any space between the value and unit. Value 0 signifies no quota enforcement. Negative values are invalid and result in errors. Possible values: • A string that is a valid concatenation of a non-negative integer representing the storage value and an optional string literal representing storage units as mentioned above. Related options: • None
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
worker_self_reference_url = None	string value	The URL to this worker. If this is set, other glance workers will know how to contact this one directly if needed. For image import, a single worker stages the image and other workers need to be able to proxy the import request to the right one. If unset, this will be considered to be public_endpoint, which normally would be set to the same value on all workers, effectively disabling the proxying behavior. Possible values: • A URL by which this worker is reachable from other workers Related options: • <code>public_endpoint</code>

Configuration option = Default value	Type	Description
workers = None	integer value	Number of Glance worker processes to start. Provide a non-negative integer value to set the number of child process workers to service requests. By default, the number of CPUs available is set as the value for workers limited to 8. For example if the processor count is 6, 6 workers will be used, if the processor count is 24 only 8 workers will be used. The limit will only apply to the default value, if 24 workers is configured, 24 is used. Each worker process is made to listen on the port set in the configuration file and contains a greenthread pool of size 1000.  NOTE Setting the number of workers to zero, triggers the creation of a single API process with a greenthread pool of size 1000. Possible values: • 0 • Positive integer value (typically equal to the number of CPUs) Related options: • None

3.1.2. cinder

The following table outlines the options available under the **[cinder]** group in the **/etc/glance/glance-api.conf** file.

Table 3.1. cinder

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
cinder_api_insecure = False	boolean value	Allow to perform insecure SSL requests to cinder. If this option is set to True, HTTPS endpoint connection is verified using the CA certificates file specified by cinder_ca_certificates_file option. Possible values: • True • False Related options: • cinder_ca_certificates_file
cinder_ca_certificates_file = None	string value	Location of a CA certificates file used for cinder client requests. The specified CA certificates file, if set, is used to verify cinder connections via HTTPS endpoint. If the endpoint is HTTP, this value is ignored. cinder_api_insecure must be set to True to enable the verification. Possible values: • Path to a ca certificates file Related options: • cinder_api_insecure

Configuration option = Default value	Type	Description
cinder_catalog_info = volumev3::publicURL	string value	Information to match when looking for cinder in the service catalog. When the cinder_endpoint_template is not set and any of cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, cinder_store_password is not set, cinder store uses this information to lookup cinder endpoint from the service catalog in the current context. cinder_os_region_name, if set, is taken into consideration to fetch the appropriate endpoint. The service catalog can be listed by the openstack catalog list command. Possible values: • A string of of the following form: <service_type>:<service_name>: <interface> At least service_type and interface should be specified. service_name can be omitted. Related options: • cinder_os_region_name • cinder_endpoint_template • cinder_store_auth_address • cinder_store_user_name • cinder_store_project_name • cinder_store_password

Configuration option = Default value	Type	Description
cinder_endpoint_template = None	string value	Override service catalog lookup with template for cinder endpoint. When this option is set, this value is used to generate cinder endpoint, instead of looking up from the service catalog. This value is ignored if cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, and cinder_store_password are specified. If this configuration option is set, cinder_catalog_info will be ignored. Possible values: ● URL template string for cinder endpoint, where %%(tenant)s is replaced with the current tenant (project) name. For example: http://cinder.openstack.example.org/v2/%%(tenant)s Related options: ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name ● cinder_store_password ● cinder_catalog_info
cinder_enforce_multipath = False	boolean value	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. Possible values: ● True or False Related options: ● cinder_use_multipath

Configuration option = Default value	Type	Description
cinder_http_retries = 3	integer value	Number of cinderclient retries on failed http calls. When a call failed by any errors, cinderclient will retry the call up to the specified times after sleeping a few seconds. Possible values: • A positive integer Related options: • None
cinder_mount_point_base = /var/lib/glance/mnt	string value	Directory where the NFS volume is mounted on the glance node. Possible values: • A string representing absolute path of mount point.
cinder_os_region_name = None	string value	Region name to lookup cinder service from the service catalog. This is used only when cinder_catalog_info is used for determining the endpoint. If set, the lookup for cinder endpoint by this node is filtered to the specified region. It is useful when multiple regions are listed in the catalog. If this is not set, the endpoint is looked up from every region. Possible values: • A string that is a valid region name. Related options: • cinder_catalog_info

Configuration option = Default value	Type	Description
cinder_state_transition_timeout = 300	integer value	Time period, in seconds, to wait for a cinder volume transition to complete. When the cinder volume is created, deleted, or attached to the glance node to read/write the volume data, the volume's state is changed. For example, the newly created volume status changes from creating to available after the creation process is completed. This specifies the maximum time to wait for the status change. If a timeout occurs while waiting, or the status is changed to an unexpected value (e.g. error), the image creation fails. Possible values: • A positive integer Related options: • None

Configuration option = Default value	Type	Description
cinder_store_auth_address = None	string value	The address where the cinder authentication service is listening. When all of cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, and cinder_store_password options are specified, the specified values are always used for the authentication. This is useful to hide the image volumes from users by storing them in a project/tenant specific to the image service. It also enables users to share the image volume among other projects under the control of glance's ACL. If either of these options are not set, the cinder endpoint is looked up from the service catalog, and current context's user and project are used. Possible values: • A valid authentication service address, for example: http://openstack.example.org/identity/v2.0 Related options: • cinder_store_user_name • cinder_store_password • cinder_store_project_name
cinder_store_password = None	string value	Password for the user authenticating against cinder. This must be used with all the following related options. If any of these are not specified, the user of the current context is used. Possible values: • A valid password for the user specified by cinder_store_user_name Related options: • cinder_store_auth_address • cinder_store_user_name • cinder_store_project_name

Configuration option = Default value	Type	Description
cinder_store_project_name = None	string value	Project name where the image volume is stored in cinder. If this configuration option is not set, the project in current context is used. This must be used with all the following related options. If any of these are not specified, the project of the current context is used. Possible values: • A valid project name Related options: • cinder_store_auth_address • cinder_store_user_name • cinder_store_password
cinder_store_user_name = None	string value	User name to authenticate against cinder. This must be used with all the following related options. If any of these are not specified, the user of the current context is used. Possible values: • A valid user name Related options: • cinder_store_auth_address • cinder_store_password • cinder_store_project_name
cinder_use_multipath = False	boolean value	Flag to identify mutipath is supported or not in the deployment. Set it to False if multipath is not supported. Possible values: • True or False Related options: • cinder_enforce_multipath

Configuration option = Default value	Type	Description
cinder_volume_type = None	string value	Volume type that will be used for volume creation in cinder. Some cinder backends can have several volume types to optimize storage usage. Adding this option allows an operator to choose a specific volume type in cinder that can be optimized for images. If this is not set, then the default volume type specified in the cinder configuration will be used for volume creation. Possible values: • A valid volume type from cinder Related options: • None  NOTE You cannot use an encrypted volume_type associated with an NFS backend. An encrypted volume stored on an NFS backend will raise an exception whenever glance_store tries to write or access image data stored in that volume. Consult your Cinder administrator to determine an appropriate volume_type.
rootwrap_config = /etc/glance/rootwrap.conf	string value	Path to the rootwrap configuration file to use for running commands as root. The cinder store requires root privileges to operate the image volumes (for connecting to iSCSI/FC volumes and reading/writing the volume data, etc.). The configuration file should allow the required commands by cinder store and os-brick library. Possible values: • Path to the rootwrap config file Related options: • None

3.1.3. cors

The following table outlines the options available under the **[cors]** group in the **/etc/glance/glance-api.conf** file.

Table 3.2. cors

Configuration option = Default value	Type	Description
allow_credentials = True	boolean value	Indicate that the actual request can include user credentials
allow_headers = ['Content-MD5', 'X-Image-Meta-Checksum', 'X-Storage-Token', 'Accept-Encoding', 'X-Auth-Token', 'X-Identity-Status', 'X-Roles', 'X-Service-Catalog', 'X-User-Id', 'X-Tenant-Id', 'X-OpenStack-Request-ID']	list value	Indicate which header field names may be used during the actual request.
allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH']	list value	Indicate which methods can be used during the actual request.
allowed_origin = None	list value	Indicate whether this resource may be shared with the domain received in the requests "origin" header. Format: "<protocol>://<host>[:<port>]", no trailing slash. Example: https://horizon.example.com
expose_headers = ['X-Image-Meta-Checksum', 'X-Auth-Token', 'X-Subject-Token', 'X-Service-Token', 'X-OpenStack-Request-ID']	list value	Indicate which headers are safe to expose to the API. Defaults to HTTP Simple Headers.
max_age = 3600	integer value	Maximum cache age of CORS preflight requests.

3.1.4. database

The following table outlines the options available under the **[database]** group in the **/etc/glance/glance-api.conf** file.

Table 3.3. database

Configuration option = Default value	Type	Description
backend = sqlalchemy	string value	The back end to use for the database.

Configuration option = Default value	Type	Description
connection = None	string value	The SQLAlchemy connection string to use to connect to the database.
connection_debug = 0	integer value	Verbosity of SQL debugging information: 0=None, 100=Everything.
<code>`connection_parameters = `</code>	string value	Optional URL parameters to append onto the connection URL at connect time; specify as param1=value1&param2=value2&...
connection_recycle_time = 3600	integer value	Connections which have been present in the connection pool longer than this number of seconds will be replaced with a new one the next time they are checked out from the pool.
connection_trace = False	boolean value	Add Python stack traces to SQL as comment strings.
db_inc_retry_interval = True	boolean value	If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = 20	integer value	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	integer value	If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = 1	integer value	Seconds between retries of a database transaction.
max_overflow = 50	integer value	If set, use this value for max_overflow with SQLAlchemy.
max_pool_size = 5	integer value	Maximum number of SQL connections to keep open in a pool. Setting a value of 0 indicates no limit.
max_retries = 10	integer value	Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.
mysql_enable_ndb = False	boolean value	If True, transparently enables support for handling MySQL Cluster (NDB).

Configuration option = Default value	Type	Description
mysql_sql_mode = TRADITIONAL	string value	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: <code>mysql_sql_mode=</code>
pool_timeout = None	integer value	If set, use this value for <code>pool_timeout</code> with SQLAlchemy.
retry_interval = 10	integer value	Interval between retries of opening a SQL connection.
slave_connection = None	string value	The SQLAlchemy connection string to use to connect to the slave database.
sqlite_synchronous = True	boolean value	If True, SQLite uses synchronous mode.
use_db_reconnect = False	boolean value	Enable the experimental use of database reconnect on connection lost.
use_tpool = False	boolean value	Enable the experimental use of thread pooling for all DB API calls

3.1.5. file

The following table outlines the options available under the **[file]** group in the `/etc/glance/glance-api.conf` file.

Table 3.4. file

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
filesystem_store_chunk_size = 65536	integer value	Chunk size, in bytes. The chunk size used when reading or writing image files. Raising this value may improve the throughput but it may also slightly increase the memory usage when handling a large number of requests. Possible Values: - Any positive integer value Related options: - None
filesystem_store_datadir = /var/lib/glance/images	string value	Directory to which the filesystem backend store writes images. Upon start up, Glance creates the directory if it doesn't already exist and verifies write access to the user under which glance-api runs. If the write access isn't available, a BadStoreConfiguration exception is raised and the filesystem store may not be available for adding new images.  NOTE This directory is used only when filesystem store is used as a storage backend. Either filesystem_store_datadir or filesystem_store_datadirs option must be specified in glance-api.conf. If both options are specified, a BadStoreConfiguration will be raised and the filesystem store may not be available for adding new images. Possible values: - A valid path to a directory Related options: filesystem_store_datadirs filesystem_store_file_perm

Configuration option = Default value	Type	Description
filesystem_store_datadirs = None	multi valued	List of directories and their priorities to which the filesystem backend store writes images. The filesystem store can be configured to store images in multiple directories as opposed to using a single directory specified by the filesystem_store_datadir configuration option. When using multiple directories, each directory can be given an optional priority to specify the preference order in which they should be used. Priority is an integer that is concatenated to the directory path with a colon where a higher value indicates higher priority. When two directories have the same priority, the directory with most free space is used. When no priority is specified, it defaults to zero. More information on configuring filesystem store with multiple store directories can be found at https://docs.openstack.org/glance/latest/configuration/configuring.html  NOTE This directory is used only when filesystem store is used as a storage backend. Either filesystem_store_datadir or filesystem_store_datadirs option must be specified in glance-api.conf. If both options are specified, a BadStoreConfiguration will be raised and the filesystem store may not be available for adding new images. Possible values: - List of strings of the following form: <a valid directory path>:<optional integer priority> Related options: filesystem_store_datadir filesystem_store_file_perm

Configuration option = Default value	Type	Description
filesystem_store_file_permission = 0	integer value	File access permissions for the image files. Set the intended file access permissions for image data. This provides a way to enable other services, e.g. Nova, to consume images directly from the filesystem store. The users running the services that are intended to be given access to could be made a member of the group that owns the files created. Assigning a value less than or equal to zero for this configuration option signifies that no changes be made to the default permissions. This value will be decoded as an octal digit. For more information, please refer the documentation at https://docs.openstack.org/glance/latest/configuration/configuring.html Possible values: • A valid file access permission • Zero • Any negative integer Related options: • None
filesystem_store_metadata_file = None	string value	Filesystem store metadata file. The path to a file which contains the metadata to be returned with any location associated with the filesystem store. Once this option is set, it is used for new images created afterward only - previously existing images are not affected. The file must contain a valid JSON object. The object should contain the keys id and mountpoint. The value for both keys should be a string. Possible values: • A valid path to the store metadata file Related options: • None

Configuration option = Default value	Type	Description
filesystem_thin_provisioning = False	boolean value	Enable or not thin provisioning in this backend. This configuration option enable the feature of not really write null byte sequences on the filesystem, the holes who can appear will automatically be interpreted by the filesystem as null bytes, and do not really consume your storage. Enabling this feature will also speed up image upload and save network traffic in addition to save space in the backend, as null bytes sequences are not sent over the network. Possible Values: • True • False Related options: • None

3.1.6. glance.store.http.store

The following table outlines the options available under the **[glance.store.http.store]** group in the **/etc/glance/glance-api.conf** file.

Table 3.5. glance.store.http.store

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
http_proxy_information = {}	dict value	The http/https proxy information to be used to connect to the remote server. This configuration option specifies the http/https proxy information that should be used to connect to the remote server. The proxy information should be a key value pair of the scheme and proxy, for example, http:10.0.0.1:3128. You can also specify proxies for multiple schemes by separating the key value pairs with a comma, for example, http:10.0.0.1:3128, https:10.0.0.1:1080. Possible values: • A comma separated list of scheme:proxy pairs as described above Related options: • None
https_ca_certificates_file = None	string value	Path to the CA bundle file. This configuration option enables the operator to use a custom Certificate Authority file to verify the remote server certificate. If this option is set, the https_insecure option will be ignored and the CA file specified will be used to authenticate the server certificate and establish a secure connection to the server. Possible values: • A valid path to a CA file Related options: • https_insecure

Configuration option = Default value	Type	Description
https_insecure = True	boolean value	Set verification of the remote server certificate. This configuration option takes in a boolean value to determine whether or not to verify the remote server certificate. If set to True, the remote server certificate is not verified. If the option is set to False, then the default CA truststore is used for verification. This option is ignored if https_ca_certificates_file is set. The remote server certificate will then be verified using the file specified using the https_ca_certificates_file option. Possible values: • True • False Related options: • https_ca_certificates_file

3.1.7. glance.store.rbd.store

The following table outlines the options available under the **[glance.store.rbd.store]** group in the **/etc/glance/glance-api.conf** file.

Table 3.6. glance.store.rbd.store

Configuration option = Default value	Type	Description
rados_connect_timeout = 0	integer value	Timeout value for connecting to Ceph cluster. This configuration option takes in the timeout value in seconds used when connecting to the Ceph cluster i.e. it sets the time to wait for glance-api before closing the connection. This prevents glance-api hangups during the connection to RBD. If the value for this option is set to less than or equal to 0, no timeout is set and the default librados value is used. Possible Values: • Any integer value Related options: • None

Configuration option = Default value	Type	Description
<code>`rbd_store_ceph_conf = `</code>	string value	Ceph configuration file path. This configuration option specifies the path to the Ceph configuration file to be used. If the value for this option is not set by the user or is set to the empty string, librados will read the standard ceph.conf file by searching the default Ceph configuration file locations in sequential order. See the Ceph documentation for details.  NOTE If using Cephx authentication, this file should include a reference to the right keyring in a client.<USER> section NOTE 2: If you leave this option empty (the default), the actual Ceph configuration file used may change depending on what version of librados is being used. If it is important for you to know exactly which configuration file is in effect, you may specify that file here using this option. Possible Values: ● A valid path to a configuration file Related options: ● <code>rbd_store_user</code>

Configuration option = Default value	Type	Description
rbd_store_chunk_size = 8	integer value	Size, in megabytes, to chunk RADOS images into. Provide an integer value representing the size in megabytes to chunk Glance images into. The default chunk size is 8 megabytes. For optimal performance, the value should be a power of two. When Ceph's RBD object storage system is used as the storage backend for storing Glance images, the images are chunked into objects of the size set using this option. These chunked objects are then stored across the distributed block data store to use for Glance. Possible Values: - Any positive integer value Related options: - None
rbd_store_pool = images	string value	RADOS pool in which images are stored. When RBD is used as the storage backend for storing Glance images, the images are stored by means of logical grouping of the objects (chunks of images) into a pool. Each pool is defined with the number of placement groups it can contain. The default pool that is used is <i>images</i>. More information on the RBD storage backend can be found here: http://ceph.com/planet/how-data-is-stored-in-ceph-cluster/ Possible Values: - A valid pool name Related options: - None

Configuration option = Default value	Type	Description
rbd_store_user = None	string value	RADOS user to authenticate as. This configuration option takes in the RADOS user to authenticate as. This is only needed when RADOS authentication is enabled and is applicable only if the user is using Cephx authentication. If the value for this option is not set by the user or is set to None, a default value will be chosen, which will be based on the client. section in rbd_store_ceph_conf. Possible Values: • A valid RADOS user Related options: • rbd_store_ceph_conf
rbd_thin_provisioning = False	boolean value	Enable or not thin provisioning in this backend. This configuration option enable the feature of not really write null byte sequences on the RBD backend, the holes who can appear will automatically be interpreted by Ceph as null bytes, and do not really consume your storage. Enabling this feature will also speed up image upload and save network traffic in addition to save space in the backend, as null bytes sequences are not sent over the network. Possible Values: • True • False Related options: • None

3.1.8. glance.store.s3.store

The following table outlines the options available under the **[glance.store.s3.store]** group in the **/etc/glance/glance-api.conf** file.

Table 3.7. glance.store.s3.store

Configuration option = Default value	Type	Description
s3_store_access_key = None	string value	The S3 query token access key. This configuration option takes the access key for authenticating with the Amazon S3 or S3 compatible storage server. This option is required when using the S3 storage backend. Possible values: - Any string value that is the access key for a user with appropriate privileges Related Options: - s3_store_host - s3_store_secret_key
s3_store_bucket = None	string value	The S3 bucket to be used to store the Glance data. This configuration option specifies where the glance images will be stored in the S3. If s3_store_create_bucket_on_put is set to true, it will be created automatically even if the bucket does not exist. Possible values: - Any string value Related Options: - s3_store_create_bucket_on_put - s3_store_bucket_url_format

Configuration option = Default value	Type	Description
s3_store_bucket_url_for mat = auto	string value	The S3 calling format used to determine the object. This configuration option takes access model that is used to specify the address of an object in an S3 bucket. NOTE: In path-style, the endpoint for the object looks like https://s3.amazonaws.com/bucket/example.img. And in virtual-style, the endpoint for the object looks like https://bucket.s3.amazonaws.com/example.img. If you do not follow the DNS naming convention in the bucket name, you can get objects in the path style, but not in the virtual style. Possible values: - Any string value of auto, virtual, or path Related Options: - s3_store_bucket
s3_store_create_bucket_ on_put = False	boolean value	Determine whether S3 should create a new bucket. This configuration option takes boolean value to indicate whether Glance should create a new bucket to S3 if it does not exist. Possible values: - Any Boolean value Related Options: - None

Configuration option = Default value	Type	Description
s3_store_host = None	string value	The host where the S3 server is listening. This configuration option sets the host of the S3 or S3 compatible storage Server. This option is required when using the S3 storage backend. The host can contain a DNS name (e.g. s3.amazonaws.com, my-object-storage.com) or an IP address (127.0.0.1). Possible values: • A valid DNS name • A valid IPv4 address Related Options: • s3_store_access_key • s3_store_secret_key
s3_store_large_object_chunk_size = 10	integer value	What multipart upload part size, in MB, should S3 use when uploading parts. This configuration option takes the image split size in MB for Multipart Upload. Note: You can only split up to 10,000 images. Possible values: • Any positive integer value (must be greater than or equal to 5M) Related Options: • s3_store_large_object_size • s3_store_thread_pools

Configuration option = Default value	Type	Description
s3_store_large_object_size = 100	integer value	What size, in MB, should S3 start chunking image files and do a multipart upload in S3. This configuration option takes a threshold in MB to determine whether to upload the image to S3 as is or to split it (Multipart Upload). Note: You can only split up to 10,000 images. Possible values: - Any positive integer value Related Options: - s3_store_large_object_chunk_size - s3_store_thread_pools
s3_store_secret_key = None	string value	The S3 query token secret key. This configuration option takes the secret key for authenticating with the Amazon S3 or S3 compatible storage server. This option is required when using the S3 storage backend. Possible values: - Any string value that is a secret key corresponding to the access key specified using the s3_store_host option Related Options: - s3_store_host - s3_store_access_key

Configuration option = Default value	Type	Description
s3_store_thread_pools = 10	integer value	The number of thread pools to perform a multipart upload in S3. This configuration option takes the number of thread pools when performing a Multipart Upload. Possible values: - Any positive integer value Related Options: - s3_store_large_object_size - s3_store_large_object_chunk_size

3.1.9. glance.store.swift.store

The following table outlines the options available under the **[glance.store.swift.store]** group in the **/etc/glance/glance-api.conf** file.

Table 3.8. glance.store.swift.store

Configuration option = Default value	Type	Description
default_swift_reference = ref1	string value	Reference to default Swift account/backing store parameters. Provide a string value representing a reference to the default set of parameters required for using swift account/backing store for image storage. The default reference value for this configuration option is <i>ref1</i>. This configuration option dereferences the parameters and facilitates image storage in Swift storage backend every time a new image is added. Possible values: - A valid string value Related options: - None

Configuration option = Default value	Type	Description
swift_buffer_on_upload = False	boolean value	Buffer image segments before upload to Swift. Provide a boolean value to indicate whether or not Glance should buffer image data to disk while uploading to swift. This enables Glance to resume uploads on error. NOTES: When enabling this option, one should take great care as this increases disk usage on the API node. Be aware that depending upon how the file system is configured, the disk space used for buffering may decrease the actual disk space available for the glance image cache. Disk utilization will cap according to the following equation: (swift_store_large_object_chunk_size * workers * 1000) Possible values: • True • False Related options: • swift_upload_buffer_dir
swift_store_admin_tenants = []	list value	List of tenants that will be granted admin access. This is a list of tenants that will be granted read/write access on all Swift containers created by Glance in multi-tenant mode. The default value is an empty list. Possible values: • A comma separated list of strings representing UUIDs of Keystone projects/tenants Related options: • None
swift_store_auth_address = None	string value	The address where the Swift authentication service is listening.

Configuration option = Default value	Type	Description
swift_store_auth_insecure = False	boolean value	Set verification of the server certificate. This boolean determines whether or not to verify the server certificate. If this option is set to True, swiftclient won't check for a valid SSL certificate when authenticating. If the option is set to False, then the default CA truststore is used for verification. Possible values: • True • False Related options: • swift_store_cacert
swift_store_auth_version = 2	string value	Version of the authentication service to use. Valid versions are 2 and 3 for keystone and 1 (deprecated) for swauth and rackspace.
swift_store_cacert = None	string value	Path to the CA bundle file. This configuration option enables the operator to specify the path to a custom Certificate Authority file for SSL verification when connecting to Swift. Possible values: • A valid path to a CA file Related options: • swift_store_auth_insecure

Configuration option = Default value	Type	Description
swift_store_config_file = None	string value	Absolute path to the file containing the swift account(s) configurations. Include a string value representing the path to a configuration file that has references for each of the configured Swift account(s)/backing stores. By default, no file path is specified and customized Swift referencing is disabled. Configuring this option is highly recommended while using Swift storage backend for image storage as it avoids storage of credentials in the database.  NOTE Please do not configure this option if you have set swift_store_multi_tenant to True. Possible values: ● String value representing an absolute path on the glance-api node Related options: ● swift_store_multi_tenant

Configuration option = Default value	Type	Description
swift_store_container = glance	string value	Name of single container to store images/name prefix for multiple containers When a single container is being used to store images, this configuration option indicates the container within the Glance account to be used for storing all images. When multiple containers are used to store images, this will be the name prefix for all containers. Usage of single/multiple containers can be controlled using the configuration option swift_store_multiple_containers_seed. When using multiple containers, the containers will be named after the value set for this configuration option with the first N chars of the image UUID as the suffix delimited by an underscore (where N is specified by swift_store_multiple_containers_seed). Example: if the seed is set to 3 and swift_store_container = glance, then an image with UUID fdae39a1-bac5-4238-aba4-69bcc726e848 would be placed in the container glance_fda. All dashes in the UUID are included when creating the container name but do not count toward the character limit, so when N=10 the container name would be glance_fdae39a1-ba. Possible values: ● If using single container, this configuration option can be any string that is a valid swift container name in Glance's Swift account ● If using multiple containers, this configuration option can be any string as long as it satisfies the container naming rules enforced by Swift. The value of swift_store_multiple_containers_seed should be taken into account as well. Related options: ● swift_store_multiple_containers_seed ● swift_store_multi_tenant ● swift_store_create_container_on_put

Configuration option = Default value	Type	Description
swift_store_create_container_on_put = False	boolean value	Create container, if it doesn't already exist, when uploading image. At the time of uploading an image, if the corresponding container doesn't exist, it will be created provided this configuration option is set to True. By default, it won't be created. This behavior is applicable for both single and multiple containers mode. Possible values: • True • False Related options: • None
swift_store_endpoint = None	string value	The URL endpoint to use for Swift backend storage. Provide a string value representing the URL endpoint to use for storing Glance images in Swift store. By default, an endpoint is not set and the storage URL returned by auth is used. Setting an endpoint with swift_store_endpoint overrides the storage URL and is used for Glance image storage.  NOTE The URL should include the path up to, but excluding the container. The location of an object is obtained by appending the container and object to the configured URL. Possible values: • String value representing a valid URL path up to a Swift container Related Options: • None

Configuration option = Default value	Type	Description
swift_store_endpoint_type = publicURL	string value	Endpoint Type of Swift service. This string value indicates the endpoint type to use to fetch the Swift endpoint. The endpoint type determines the actions the user will be allowed to perform, for instance, reading and writing to the Store. This setting is only used if <code>swift_store_auth_version</code> is greater than 1. Possible values: ● <code>publicURL</code> ● <code>adminURL</code> ● <code>internalURL</code> Related options: ● <code>swift_store_endpoint</code>
swift_store_expire_soon_interval = 60	integer value	Time in seconds defining the size of the window in which a new token may be requested before the current token is due to expire. Typically, the Swift storage driver fetches a new token upon the expiration of the current token to ensure continued access to Swift. However, some Swift transactions (like uploading image segments) may not recover well if the token expires on the fly. Hence, by fetching a new token before the current token expiration, we make sure that the token does not expire or is close to expiry before a transaction is attempted. By default, the Swift storage driver requests for a new token 60 seconds or less before the current token expiration. Possible values: ● Zero ● Positive integer value Related Options: ● None
swift_store_key = None	string value	Auth key for the user authenticating against the Swift authentication service.

Configuration option = Default value	Type	Description
swift_store_large_object_ chunk_size = 200	integer value	The maximum size, in MB, of the segments when image data is segmented. When image data is segmented to upload images that are larger than the limit enforced by the Swift cluster, image data is broken into segments that are no bigger than the size specified by this configuration option. Refer to swift_store_large_object_size for more detail. For example: if swift_store_large_object_size is 5GB and swift_store_large_object_chunk_size is 1GB, an image of size 6.2GB will be segmented into 7 segments where the first six segments will be 1GB in size and the seventh segment will be 0.2GB. Possible values: • A positive integer that is less than or equal to the large object limit enforced by Swift cluster in consideration. Related options: • swift_store_large_object_size

Configuration option = Default value	Type	Description
swift_store_large_object_size = 5120	integer value	The size threshold, in MB, after which Glance will start segmenting image data. Swift has an upper limit on the size of a single uploaded object. By default, this is 5GB. To upload objects bigger than this limit, objects are segmented into multiple smaller objects that are tied together with a manifest file. For more detail, refer to https://docs.openstack.org/swift/latest/overview_large_objects.html This configuration option specifies the size threshold over which the Swift driver will start segmenting image data into multiple smaller files. Currently, the Swift driver only supports creating Dynamic Large Objects.  NOTE This should be set by taking into account the large object limit enforced by the Swift cluster in consideration. Possible values: • A positive integer that is less than or equal to the large object limit enforced by the Swift cluster in consideration. Related options: • swift_store_large_object_chunk_size

Configuration option = Default value	Type	Description
swift_store_multi_tenant = False	boolean value	Store images in tenant's Swift account. This enables multi-tenant storage mode which causes Glance images to be stored in tenant specific Swift accounts. If this is disabled, Glance stores all images in its own account. More details multi-tenant store can be found at https://wiki.openstack.org/wiki/GlanceSwiftTenantSpecificStorage  NOTE If using multi-tenant swift store, please make sure that you do not set a swift configuration file with the <i>swift_store_config_file</i> option. Possible values: • True • False Related options: • <code>swift_store_config_file</code>

Configuration option = Default value	Type	Description
swift_store_multiple_containers_seed = 0	integer value	Seed indicating the number of containers to use for storing images. When using a single-tenant store, images can be stored in one or more than one containers. When set to 0, all images will be stored in one single container. When set to an integer value between 1 and 32, multiple containers will be used to store images. This configuration option will determine how many containers are created. 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. Please refer to swift_store_container for more detail on the naming convention. More detail about using multiple containers can be found at https://specs.openstack.org/openstack/glance-specs/specs/kilo/swift-store-multiple-containers.html  NOTE This is used only when swift_store_multi_tenant is disabled. Possible values: • A non-negative integer less than or equal to 32 Related options: • swift_store_container • swift_store_multi_tenant • swift_store_create_container_on_put

Configuration option = Default value	Type	Description
swift_store_region = None	string value	The region of Swift endpoint to use by Glance. Provide a string value representing a Swift region where Glance can connect to for image storage. By default, there is no region set. When Glance uses Swift as the storage backend to store images for a specific tenant that has multiple endpoints, setting of a Swift region with swift_store_region allows Glance to connect to Swift in the specified region as opposed to a single region connectivity. This option can be configured for both single-tenant and multi-tenant storage.  NOTE Setting the region with swift_store_region is tenant-specific and is necessary only if the tenant has multiple endpoints across different regions. Possible values: • A string value representing a valid Swift region. Related Options: • None

Configuration option = Default value	Type	Description
swift_store_retry_get_count = 0	integer value	The number of times a Swift download will be retried before the request fails. Provide an integer value representing the number of times an image download must be retried before erroring out. The default value is zero (no retry on a failed image download). When set to a positive integer value, swift_store_retry_get_count ensures that the download is attempted this many more times upon a download failure before sending an error message. Possible values: • Zero • Positive integer value Related Options: • None
swift_store_service_type = object-store	string value	Type of Swift service to use. Provide a string value representing the service type to use for storing images while using Swift backend storage. The default service type is set to object-store.  NOTE If swift_store_auth_version is set to 2, the value for this configuration option needs to be object-store. If using a higher version of Keystone or a different auth scheme, this option may be modified. Possible values: • A string representing a valid service type for Swift storage. Related Options: • None

Configuration option = Default value	Type	Description
swift_store_ssl_compression = True	boolean value	SSL layer compression for HTTPS Swift requests. Provide a boolean value to determine whether or not to compress HTTPS Swift requests for images at the SSL layer. By default, compression is enabled. When using Swift as the backend store for Glance image storage, SSL layer compression of HTTPS Swift requests can be set using this option. If set to False, SSL layer compression of HTTPS Swift requests is disabled. Disabling this option may improve performance for images which are already in a compressed format, for example, qcow2. Possible values: • True • False Related Options: • None

Configuration option = Default value	Type	Description
swift_store_use_trusts = True	boolean value	Use trusts for multi-tenant Swift store. This option instructs the Swift store to create a trust for each add/get request when the multi-tenant store is in use. Using trusts allows the Swift store to avoid problems that can be caused by an authentication token expiring during the upload or download of data. By default, swift_store_use_trusts is set to True(use of trusts is enabled). If set to False, a user token is used for the Swift connection instead, eliminating the overhead of trust creation.  NOTE This option is considered only when swift_store_multi_tenant is set to True Possible values: • True • False Related options: • swift_store_multi_tenant
swift_store_user = None	string value	The user to authenticate against the Swift authentication service.

Configuration option = Default value	Type	Description
swift_upload_buffer_dir = None	string value	Directory to buffer image segments before upload to Swift. Provide a string value representing the absolute path to the directory on the glance node where image segments will be buffered briefly before they are uploaded to swift. NOTES: • This is required only when the configuration option swift_buffer_on_upload is set to True. • This directory should be provisioned keeping in mind the swift_store_large_object_chunk_size and the maximum number of images that could be uploaded simultaneously by a given glance node. Possible values: • String value representing an absolute directory path Related options: • swift_buffer_on_upload • swift_store_large_object_chunk_size

3.1.10. glance.store.vmware_datastore.store

The following table outlines the options available under the **[glance.store.vmware_datastore.store]** group in the **/etc/glance/glance-api.conf** file.

Table 3.9. glance.store.vmware_datastore.store

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
vmware_api_retry_count = 10	integer value	The number of VMware API retries. This configuration option specifies the number of times the VMware ESX/VC server API must be retried upon connection related issues or server API call overload. It is not possible to specify <i>retry forever</i>. Possible Values: - Any positive integer value Related options: - None
vmware_ca_file = None	string value	Absolute path to the CA bundle file. This configuration option enables the operator to use a custom Certificate Authority File to verify the ESX/vCenter certificate. If this option is set, the "vmware_insecure" option will be ignored and the CA file specified will be used to authenticate the ESX/vCenter server certificate and establish a secure connection to the server. Possible Values: - Any string that is a valid absolute path to a CA file Related options: - vmware_insecure

Configuration option = Default value	Type	Description
vmware_datastores = None	multi valued	The datastores where the image can be stored. This configuration option specifies the datastores where the image can be stored in the VMWare store backend. This option may be specified multiple times for specifying multiple datastores. 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 ":" to specify the priority. 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. Possible Values: Any string of the format: <datacenter_path>:<datastore_name>:<optional_weight> Related options: * None

Configuration option = Default value	Type	Description
vmware_insecure = False	boolean value	Set verification of the ESX/vCenter server certificate. This configuration option takes a boolean value to determine whether or not to verify the ESX/vCenter server certificate. If this option is set to True, the ESX/vCenter server certificate is not verified. If this option is set to False, then the default CA truststore is used for verification. This option is ignored if the "vmware_ca_file" option is set. In that case, the ESX/vCenter server certificate will then be verified using the file specified using the "vmware_ca_file" option . Possible Values: • True • False Related options: • vmware_ca_file
vmware_server_host = None	host address value	Address of the ESX/ESXi or vCenter Server target system. This configuration option sets the address of the ESX/ESXi or vCenter Server target system. This option is required when using the VMware storage backend. The address can contain an IP address (127.0.0.1) or a DNS name (www.my-domain.com). Possible Values: • A valid IPv4 or IPv6 address • A valid DNS name Related options: • vmware_server_username • vmware_server_password

Configuration option = Default value	Type	Description
vmware_server_password = None	string value	Server password. This configuration option takes the password for authenticating with the VMware ESX/ESXi or vCenter Server. This option is required when using the VMware storage backend. Possible Values: - Any string that is a password corresponding to the username specified using the "vmware_server_username" option Related options: - vmware_server_host - vmware_server_username
vmware_server_username = None	string value	Server username. This configuration option takes the username for authenticating with the VMware ESX/ESXi or vCenter Server. This option is required when using the VMware storage backend. Possible Values: - Any string that is the username for a user with appropriate privileges Related options: - vmware_server_host - vmware_server_password

Configuration option = Default value	Type	Description
vmware_store_image_dir = /openstack_glance	string value	The directory where the glance images will be stored in the datastore. This configuration option specifies the path to the directory where the glance images will be stored in the VMware datastore. If this option is not set, the default directory where the glance images are stored is openstack_glance. Possible Values: - Any string that is a valid path to a directory Related options: - None
vmware_task_poll_interval = 5	integer value	Interval in seconds used for polling remote tasks invoked on VMware ESX/VC server. This configuration option takes in the sleep time in seconds for polling an on-going async task as part of the VMWare ESX/VC server API call. Possible Values: - Any positive integer value Related options: - None

3.1.11. glance_store

The following table outlines the options available under the **[glance_store]** group in the **/etc/glance/glance-api.conf** file.

Table 3.10. glance_store

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
cinder_api_insecure = False	boolean value	Allow to perform insecure SSL requests to cinder. If this option is set to True, HTTPS endpoint connection is verified using the CA certificates file specified by cinder_ca_certificates_file option. Possible values: • True • False Related options: • cinder_ca_certificates_file
cinder_ca_certificates_file = None	string value	Location of a CA certificates file used for cinder client requests. The specified CA certificates file, if set, is used to verify cinder connections via HTTPS endpoint. If the endpoint is HTTP, this value is ignored. cinder_api_insecure must be set to True to enable the verification. Possible values: • Path to a ca certificates file Related options: • cinder_api_insecure

Configuration option = Default value	Type	Description
cinder_catalog_info = volumev3::publicURL	string value	Information to match when looking for cinder in the service catalog. When the cinder_endpoint_template is not set and any of cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, cinder_store_password is not set, cinder store uses this information to lookup cinder endpoint from the service catalog in the current context. cinder_os_region_name, if set, is taken into consideration to fetch the appropriate endpoint. The service catalog can be listed by the openstack catalog list command. Possible values: • A string of of the following form: <service_type>:<service_name>:<interface> At least service_type and interface should be specified. service_name can be omitted. Related options: • cinder_os_region_name • cinder_endpoint_template • cinder_store_auth_address • cinder_store_user_name • cinder_store_project_name • cinder_store_password

Configuration option = Default value	Type	Description
cinder_endpoint_template = None	string value	Override service catalog lookup with template for cinder endpoint. When this option is set, this value is used to generate cinder endpoint, instead of looking up from the service catalog. This value is ignored if cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, and cinder_store_password are specified. If this configuration option is set, cinder_catalog_info will be ignored. Possible values: ● URL template string for cinder endpoint, where %%(tenant)s is replaced with the current tenant (project) name. For example: http://cinder.openstack.example.org/v2/%%(tenant)s Related options: ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name ● cinder_store_password ● cinder_catalog_info
cinder_enforce_multipath = False	boolean value	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. Possible values: ● True or False Related options: ● cinder_use_multipath

Configuration option = Default value	Type	Description
cinder_http_retries = 3	integer value	Number of cinderclient retries on failed http calls. When a call failed by any errors, cinderclient will retry the call up to the specified times after sleeping a few seconds. Possible values: • A positive integer Related options: • None
cinder_mount_point_base = /var/lib/glance/mnt	string value	Directory where the NFS volume is mounted on the glance node. Possible values: • A string representing absolute path of mount point.
cinder_os_region_name = None	string value	Region name to lookup cinder service from the service catalog. This is used only when cinder_catalog_info is used for determining the endpoint. If set, the lookup for cinder endpoint by this node is filtered to the specified region. It is useful when multiple regions are listed in the catalog. If this is not set, the endpoint is looked up from every region. Possible values: • A string that is a valid region name. Related options: • cinder_catalog_info

Configuration option = Default value	Type	Description
cinder_state_transition_timeout = 300	integer value	Time period, in seconds, to wait for a cinder volume transition to complete. When the cinder volume is created, deleted, or attached to the glance node to read/write the volume data, the volume's state is changed. For example, the newly created volume status changes from creating to available after the creation process is completed. This specifies the maximum time to wait for the status change. If a timeout occurs while waiting, or the status is changed to an unexpected value (e.g. error), the image creation fails. Possible values: • A positive integer Related options: • None

Configuration option = Default value	Type	Description
cinder_store_auth_address = None	string value	The address where the cinder authentication service is listening. When all of cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, and cinder_store_password options are specified, the specified values are always used for the authentication. This is useful to hide the image volumes from users by storing them in a project/tenant specific to the image service. It also enables users to share the image volume among other projects under the control of glance's ACL. If either of these options are not set, the cinder endpoint is looked up from the service catalog, and current context's user and project are used. Possible values: • A valid authentication service address, for example: http://openstack.example.org/identity/v2.0 Related options: • cinder_store_user_name • cinder_store_password • cinder_store_project_name
cinder_store_password = None	string value	Password for the user authenticating against cinder. This must be used with all the following related options. If any of these are not specified, the user of the current context is used. Possible values: • A valid password for the user specified by cinder_store_user_name Related options: • cinder_store_auth_address • cinder_store_user_name • cinder_store_project_name

Configuration option = Default value	Type	Description
cinder_store_project_name = None	string value	Project name where the image volume is stored in cinder. If this configuration option is not set, the project in current context is used. This must be used with all the following related options. If any of these are not specified, the project of the current context is used. Possible values: • A valid project name Related options: • cinder_store_auth_address • cinder_store_user_name • cinder_store_password
cinder_store_user_name = None	string value	User name to authenticate against cinder. This must be used with all the following related options. If any of these are not specified, the user of the current context is used. Possible values: • A valid user name Related options: • cinder_store_auth_address • cinder_store_password • cinder_store_project_name

Configuration option = Default value	Type	Description
cinder_use_multipath = False	boolean value	Flag to identify mutipath is supported or not in the deployment. Set it to False if multipath is not supported. Possible values: • True or False Related options: • cinder_enforce_multipath
cinder_volume_type = None	string value	Volume type that will be used for volume creation in cinder. Some cinder backends can have several volume types to optimize storage usage. Adding this option allows an operator to choose a specific volume type in cinder that can be optimized for images. If this is not set, then the default volume type specified in the cinder configuration will be used for volume creation. Possible values: • A valid volume type from cinder Related options: • None  NOTE You cannot use an encrypted volume_type associated with an NFS backend. An encrypted volume stored on an NFS backend will raise an exception whenever glance_store tries to write or access image data stored in that volume. Consult your Cinder administrator to determine an appropriate volume_type.

Configuration option = Default value	Type	Description
default_backend = None	string value	The store identifier for the default backend in which data will be stored. The value must be defined as one of the keys in the dict defined by the enabled_backends configuration option in the DEFAULT configuration group. If a value is not defined for this option: - the consuming service may refuse to start - store_add calls that do not specify a specific backend will raise a glance_store.exceptions.UnknownScheme exception Related Options: - enabled_backends

Configuration option = Default value	Type	Description
default_store = file	string value	The default scheme to use for storing images. Provide a string value representing the default scheme to use for storing images. If not set, Glance uses file as the default scheme to store images with the file store.  NOTE The value given for this configuration option must be a valid scheme for a store registered with the stores configuration option. Possible values: • file • filesystem • http • https • swift • swift+http • swift+https • swift+config • rbd • cinder • vsphere • s3 Related Options: • stores Deprecated since: Rocky Reason: This option is deprecated against new config option ``default_backend`` which acts similar to ``default_store`` config option. This option is scheduled for removal in the U development cycle.

Configuration option = Default value	Type	Description
default_swift_reference = ref1	string value	Reference to default Swift account/backing store parameters. Provide a string value representing a reference to the default set of parameters required for using swift account/backing store for image storage. The default reference value for this configuration option is <i>ref1</i>. This configuration option dereferences the parameters and facilitates image storage in Swift storage backend every time a new image is added. Possible values: • A valid string value Related options: • None
filesystem_store_chunk_size = 65536	integer value	Chunk size, in bytes. The chunk size used when reading or writing image files. Raising this value may improve the throughput but it may also slightly increase the memory usage when handling a large number of requests. Possible Values: • Any positive integer value Related options: • None

Configuration option = Default value	Type	Description
filesystem_store_datadir = /var/lib/glance/images	string value	Directory to which the filesystem backend store writes images. Upon start up, Glance creates the directory if it doesn't already exist and verifies write access to the user under which glance-api runs. If the write access isn't available, a BadStoreConfiguration exception is raised and the filesystem store may not be available for adding new images.  NOTE This directory is used only when filesystem store is used as a storage backend. Either filesystem_store_datadir or filesystem_store_datadirs option must be specified in glance-api.conf. If both options are specified, a BadStoreConfiguration will be raised and the filesystem store may not be available for adding new images. Possible values: ● A valid path to a directory Related options: ● filesystem_store_datadirs ● filesystem_store_file_perm

Configuration option = Default value	Type	Description
filesystem_store_datadirs = None	multi valued	List of directories and their priorities to which the filesystem backend store writes images. The filesystem store can be configured to store images in multiple directories as opposed to using a single directory specified by the filesystem_store_datadir configuration option. When using multiple directories, each directory can be given an optional priority to specify the preference order in which they should be used. Priority is an integer that is concatenated to the directory path with a colon where a higher value indicates higher priority. When two directories have the same priority, the directory with most free space is used. When no priority is specified, it defaults to zero. More information on configuring filesystem store with multiple store directories can be found at https://docs.openstack.org/glance/latest/configuration/configuring.html  NOTE This directory is used only when filesystem store is used as a storage backend. Either filesystem_store_datadir or filesystem_store_datadirs option must be specified in glance-api.conf. If both options are specified, a BadStoreConfiguration will be raised and the filesystem store may not be available for adding new images. Possible values: - List of strings of the following form: <a valid directory path>:<optional integer priority> Related options: filesystem_store_datadir filesystem_store_file_perm

Configuration option = Default value	Type	Description
filesystem_store_file_permission = 0	integer value	File access permissions for the image files. Set the intended file access permissions for image data. This provides a way to enable other services, e.g. Nova, to consume images directly from the filesystem store. The users running the services that are intended to be given access to could be made a member of the group that owns the files created. Assigning a value less than or equal to zero for this configuration option signifies that no changes be made to the default permissions. This value will be decoded as an octal digit. For more information, please refer the documentation at https://docs.openstack.org/glance/latest/configuration/configuring.html Possible values: ● A valid file access permission ● Zero ● Any negative integer Related options: ● None
filesystem_store_metadata_file = None	string value	Filesystem store metadata file. The path to a file which contains the metadata to be returned with any location associated with the filesystem store. Once this option is set, it is used for new images created afterward only – previously existing images are not affected. The file must contain a valid JSON object. The object should contain the keys id and mountpoint. The value for both keys should be a string. Possible values: ● A valid path to the store metadata file Related options: ● None

Configuration option = Default value	Type	Description
filesystem_thin_provisioning = False	boolean value	Enable or not thin provisioning in this backend. This configuration option enable the feature of not really write null byte sequences on the filesystem, the holes who can appear will automatically be interpreted by the filesystem as null bytes, and do not really consume your storage. Enabling this feature will also speed up image upload and save network traffic in addition to save space in the backend, as null bytes sequences are not sent over the network. Possible Values: • True • False Related options: • None
http_proxy_information = {}	dict value	The http/https proxy information to be used to connect to the remote server. This configuration option specifies the http/https proxy information that should be used to connect to the remote server. The proxy information should be a key value pair of the scheme and proxy, for example, http:10.0.0.1:3128. You can also specify proxies for multiple schemes by separating the key value pairs with a comma, for example, http:10.0.0.1:3128, https:10.0.0.1:1080. Possible values: • A comma separated list of scheme:proxy pairs as described above Related options: • None

Configuration option = Default value	Type	Description
https_ca_certificates_file = None	string value	Path to the CA bundle file. This configuration option enables the operator to use a custom Certificate Authority file to verify the remote server certificate. If this option is set, the https_insecure option will be ignored and the CA file specified will be used to authenticate the server certificate and establish a secure connection to the server. Possible values: • A valid path to a CA file Related options: • https_insecure
https_insecure = True	boolean value	Set verification of the remote server certificate. This configuration option takes in a boolean value to determine whether or not to verify the remote server certificate. If set to True, the remote server certificate is not verified. If the option is set to False, then the default CA truststore is used for verification. This option is ignored if https_ca_certificates_file is set. The remote server certificate will then be verified using the file specified using the https_ca_certificates_file option. Possible values: • True • False Related options: • https_ca_certificates_file

Configuration option = Default value	Type	Description
rados_connect_timeout = 0	integer value	Timeout value for connecting to Ceph cluster. This configuration option takes in the timeout value in seconds used when connecting to the Ceph cluster i.e. it sets the time to wait for glance-api before closing the connection. This prevents glance-api hangups during the connection to RBD. If the value for this option is set to less than or equal to 0, no timeout is set and the default librados value is used. Possible Values: - Any integer value Related options: - None
`rbd_store_ceph_conf = `	string value	Ceph configuration file path. This configuration option specifies the path to the Ceph configuration file to be used. If the value for this option is not set by the user or is set to the empty string, librados will read the standard ceph.conf file by searching the default Ceph configuration file locations in sequential order. See the Ceph documentation for details.  NOTE If using Cephx authentication, this file should include a reference to the right keyring in a client.<USER> section NOTE 2: If you leave this option empty (the default), the actual Ceph configuration file used may change depending on what version of librados is being used. If it is important for you to know exactly which configuration file is in effect, you may specify that file here using this option. Possible Values: - A valid path to a configuration file Related options: - rbd_store_user

Configuration option = Default value	Type	Description
rbd_store_chunk_size = 8	integer value	Size, in megabytes, to chunk RADOS images into. Provide an integer value representing the size in megabytes to chunk Glance images into. The default chunk size is 8 megabytes. For optimal performance, the value should be a power of two. When Ceph's RBD object storage system is used as the storage backend for storing Glance images, the images are chunked into objects of the size set using this option. These chunked objects are then stored across the distributed block data store to use for Glance. Possible Values: - Any positive integer value Related options: - None
rbd_store_pool = images	string value	RADOS pool in which images are stored. When RBD is used as the storage backend for storing Glance images, the images are stored by means of logical grouping of the objects (chunks of images) into a pool. Each pool is defined with the number of placement groups it can contain. The default pool that is used is <i>images</i>. More information on the RBD storage backend can be found here: http://ceph.com/planet/how-data-is-stored-in-ceph-cluster/ Possible Values: - A valid pool name Related options: - None

Configuration option = Default value	Type	Description
rbd_store_user = None	string value	RADOS user to authenticate as. This configuration option takes in the RADOS user to authenticate as. This is only needed when RADOS authentication is enabled and is applicable only if the user is using Cephx authentication. If the value for this option is not set by the user or is set to None, a default value will be chosen, which will be based on the client. section in rbd_store_ceph_conf. Possible Values: • A valid RADOS user Related options: • rbd_store_ceph_conf
rbd_thin_provisioning = False	boolean value	Enable or not thin provisioning in this backend. This configuration option enable the feature of not really write null byte sequences on the RBD backend, the holes who can appear will automatically be interpreted by Ceph as null bytes, and do not really consume your storage. Enabling this feature will also speed up image upload and save network traffic in addition to save space in the backend, as null bytes sequences are not sent over the network. Possible Values: • True • False Related options: • None

Configuration option = Default value	Type	Description
rootwrap_config = /etc/glance/rootwrap.conf	string value	Path to the rootwrap configuration file to use for running commands as root. The cinder store requires root privileges to operate the image volumes (for connecting to iSCSI/FC volumes and reading/writing the volume data, etc.). The configuration file should allow the required commands by cinder store and os-brick library. Possible values: ● Path to the rootwrap config file Related options: ● None
s3_store_access_key = None	string value	The S3 query token access key. This configuration option takes the access key for authenticating with the Amazon S3 or S3 compatible storage server. This option is required when using the S3 storage backend. Possible values: ● Any string value that is the access key for a user with appropriate privileges Related Options: ● s3_store_host ● s3_store_secret_key
s3_store_bucket = None	string value	The S3 bucket to be used to store the Glance data. This configuration option specifies where the glance images will be stored in the S3. If s3_store_create_bucket_on_put is set to true, it will be created automatically even if the bucket does not exist. Possible values: ● Any string value Related Options: ● s3_store_create_bucket_on_put ● s3_store_bucket_url_format

Configuration option = Default value	Type	Description
s3_store_bucket_url_for mat = auto	string value	The S3 calling format used to determine the object. This configuration option takes access model that is used to specify the address of an object in an S3 bucket. NOTE: In path-style, the endpoint for the object looks like https://s3.amazonaws.com/bucket/example.img. And in virtual-style, the endpoint for the object looks like https://bucket.s3.amazonaws.com/example.img. If you do not follow the DNS naming convention in the bucket name, you can get objects in the path style, but not in the virtual style. Possible values: - Any string value of auto, virtual, or path Related Options: - s3_store_bucket
s3_store_create_bucket_ on_put = False	boolean value	Determine whether S3 should create a new bucket. This configuration option takes boolean value to indicate whether Glance should create a new bucket to S3 if it does not exist. Possible values: - Any Boolean value Related Options: - None

Configuration option = Default value	Type	Description
s3_store_host = None	string value	The host where the S3 server is listening. This configuration option sets the host of the S3 or S3 compatible storage Server. This option is required when using the S3 storage backend. The host can contain a DNS name (e.g. s3.amazonaws.com, my-object-storage.com) or an IP address (127.0.0.1). Possible values: ● A valid DNS name ● A valid IPv4 address Related Options: ● s3_store_access_key ● s3_store_secret_key
s3_store_large_object_chunk_size = 10	integer value	What multipart upload part size, in MB, should S3 use when uploading parts. This configuration option takes the image split size in MB for Multipart Upload. Note: You can only split up to 10,000 images. Possible values: ● Any positive integer value (must be greater than or equal to 5M) Related Options: ● s3_store_large_object_size ● s3_store_thread_pools

Configuration option = Default value	Type	Description
s3_store_large_object_size = 100	integer value	What size, in MB, should S3 start chunking image files and do a multipart upload in S3. This configuration option takes a threshold in MB to determine whether to upload the image to S3 as is or to split it (Multipart Upload). Note: You can only split up to 10,000 images. Possible values: - Any positive integer value Related Options: - s3_store_large_object_chunk_size - s3_store_thread_pools
s3_store_secret_key = None	string value	The S3 query token secret key. This configuration option takes the secret key for authenticating with the Amazon S3 or S3 compatible storage server. This option is required when using the S3 storage backend. Possible values: - Any string value that is a secret key corresponding to the access key specified using the s3_store_host option Related Options: - s3_store_host - s3_store_access_key

Configuration option = Default value	Type	Description
s3_store_thread_pools = 10	integer value	The number of thread pools to perform a multipart upload in S3. This configuration option takes the number of thread pools when performing a Multipart Upload. Possible values: - Any positive integer value Related Options: - s3_store_large_object_size - s3_store_large_object_chunk_size
stores = ['file', 'http']	list value	List of enabled Glance stores. Register the storage backends to use for storing disk images as a comma separated list. The default stores enabled for storing disk images with Glance are file and http. Possible values: - A comma separated list that could include: - file - http - swift - rbd - cinder - vmware - s3 Related Options: - default_store Deprecated since: Rocky Reason: This option is deprecated against new config option ``enabled_backends`` which helps to configure multiple backend stores of different schemes. This option is scheduled for removal in the U development cycle.

Configuration option = Default value	Type	Description
swift_buffer_on_upload = False	boolean value	Buffer image segments before upload to Swift. Provide a boolean value to indicate whether or not Glance should buffer image data to disk while uploading to swift. This enables Glance to resume uploads on error. NOTES: When enabling this option, one should take great care as this increases disk usage on the API node. Be aware that depending upon how the file system is configured, the disk space used for buffering may decrease the actual disk space available for the glance image cache. Disk utilization will cap according to the following equation: (swift_store_large_object_chunk_size * workers * 1000) Possible values: • True • False Related options: • swift_upload_buffer_dir
swift_store_admin_tenants = []	list value	List of tenants that will be granted admin access. This is a list of tenants that will be granted read/write access on all Swift containers created by Glance in multi-tenant mode. The default value is an empty list. Possible values: • A comma separated list of strings representing UUIDs of Keystone projects/tenants Related options: • None
swift_store_auth_address = None	string value	The address where the Swift authentication service is listening.

Configuration option = Default value	Type	Description
swift_store_auth_insecure = False	boolean value	Set verification of the server certificate. This boolean determines whether or not to verify the server certificate. If this option is set to True, swiftclient won't check for a valid SSL certificate when authenticating. If the option is set to False, then the default CA truststore is used for verification. Possible values: • True • False Related options: • swift_store_cacert
swift_store_auth_version = 2	string value	Version of the authentication service to use. Valid versions are 2 and 3 for keystone and 1 (deprecated) for swauth and rackspace.
swift_store_cacert = None	string value	Path to the CA bundle file. This configuration option enables the operator to specify the path to a custom Certificate Authority file for SSL verification when connecting to Swift. Possible values: • A valid path to a CA file Related options: • swift_store_auth_insecure

Configuration option = Default value	Type	Description
swift_store_config_file = None	string value	Absolute path to the file containing the swift account(s) configurations. Include a string value representing the path to a configuration file that has references for each of the configured Swift account(s)/backing stores. By default, no file path is specified and customized Swift referencing is disabled. Configuring this option is highly recommended while using Swift storage backend for image storage as it avoids storage of credentials in the database.  NOTE Please do not configure this option if you have set swift_store_multi_tenant to True. Possible values: ● String value representing an absolute path on the glance-api node Related options: ● swift_store_multi_tenant

Configuration option = Default value	Type	Description
swift_store_container = glance	string value	Name of single container to store images/name prefix for multiple containers When a single container is being used to store images, this configuration option indicates the container within the Glance account to be used for storing all images. When multiple containers are used to store images, this will be the name prefix for all containers. Usage of single/multiple containers can be controlled using the configuration option swift_store_multiple_containers_seed. When using multiple containers, the containers will be named after the value set for this configuration option with the first N chars of the image UUID as the suffix delimited by an underscore (where N is specified by swift_store_multiple_containers_seed). Example: if the seed is set to 3 and swift_store_container = glance, then an image with UUID fdac39a1-bac5-4238-aba4-69bcc726e848 would be placed in the container glance_fdac39a1-bac5-4238-aba4-69bcc726e848. All dashes in the UUID are included when creating the container name but do not count toward the character limit, so when N=10 the container name would be glance_fdac39a1-bac5-4238-aba4-69bcc726e848. Possible values: ● If using single container, this configuration option can be any string that is a valid swift container name in Glance's Swift account ● If using multiple containers, this configuration option can be any string as long as it satisfies the container naming rules enforced by Swift. The value of swift_store_multiple_containers_seed should be taken into account as well. Related options: ● swift_store_multiple_containers_seed ● swift_store_multi_tenant ● swift_store_create_container_on_put

Configuration option = Default value	Type	Description
swift_store_create_container_on_put = False	boolean value	Create container, if it doesn't already exist, when uploading image. At the time of uploading an image, if the corresponding container doesn't exist, it will be created provided this configuration option is set to True. By default, it won't be created. This behavior is applicable for both single and multiple containers mode. Possible values: • True • False Related options: • None
swift_store_endpoint = None	string value	The URL endpoint to use for Swift backend storage. Provide a string value representing the URL endpoint to use for storing Glance images in Swift store. By default, an endpoint is not set and the storage URL returned by auth is used. Setting an endpoint with swift_store_endpoint overrides the storage URL and is used for Glance image storage.  NOTE The URL should include the path up to, but excluding the container. The location of an object is obtained by appending the container and object to the configured URL. Possible values: • String value representing a valid URL path up to a Swift container Related Options: • None

Configuration option = Default value	Type	Description
swift_store_endpoint_type = publicURL	string value	Endpoint Type of Swift service. This string value indicates the endpoint type to use to fetch the Swift endpoint. The endpoint type determines the actions the user will be allowed to perform, for instance, reading and writing to the Store. This setting is only used if <code>swift_store_auth_version</code> is greater than 1. Possible values: ● <code>publicURL</code> ● <code>adminURL</code> ● <code>internalURL</code> Related options: ● <code>swift_store_endpoint</code>
swift_store_expire_soon_interval = 60	integer value	Time in seconds defining the size of the window in which a new token may be requested before the current token is due to expire. Typically, the Swift storage driver fetches a new token upon the expiration of the current token to ensure continued access to Swift. However, some Swift transactions (like uploading image segments) may not recover well if the token expires on the fly. Hence, by fetching a new token before the current token expiration, we make sure that the token does not expire or is close to expiry before a transaction is attempted. By default, the Swift storage driver requests for a new token 60 seconds or less before the current token expiration. Possible values: ● Zero ● Positive integer value Related Options: ● None
swift_store_key = None	string value	Auth key for the user authenticating against the Swift authentication service.

Configuration option = Default value	Type	Description
swift_store_large_object_ chunk_size = 200	integer value	The maximum size, in MB, of the segments when image data is segmented. When image data is segmented to upload images that are larger than the limit enforced by the Swift cluster, image data is broken into segments that are no bigger than the size specified by this configuration option. Refer to swift_store_large_object_size for more detail. For example: if swift_store_large_object_size is 5GB and swift_store_large_object_chunk_size is 1GB, an image of size 6.2GB will be segmented into 7 segments where the first six segments will be 1GB in size and the seventh segment will be 0.2GB. Possible values: • A positive integer that is less than or equal to the large object limit enforced by Swift cluster in consideration. Related options: • swift_store_large_object_size

Configuration option = Default value	Type	Description
swift_store_large_object_size = 5120	integer value	The size threshold, in MB, after which Glance will start segmenting image data. Swift has an upper limit on the size of a single uploaded object. By default, this is 5GB. To upload objects bigger than this limit, objects are segmented into multiple smaller objects that are tied together with a manifest file. For more detail, refer to https://docs.openstack.org/swift/latest/overview_large_objects.html This configuration option specifies the size threshold over which the Swift driver will start segmenting image data into multiple smaller files. Currently, the Swift driver only supports creating Dynamic Large Objects. NOTE This should be set by taking into account the large object limit enforced by the Swift cluster in consideration. Possible values: ● A positive integer that is less than or equal to the large object limit enforced by the Swift cluster in consideration. Related options: ● swift_store_large_object_chunk_size

Configuration option = Default value	Type	Description
swift_store_multi_tenant = False	boolean value	Store images in tenant's Swift account. This enables multi-tenant storage mode which causes Glance images to be stored in tenant specific Swift accounts. If this is disabled, Glance stores all images in its own account. More details multi-tenant store can be found at https://wiki.openstack.org/wiki/GlanceSwiftTenantSpecificStorage NOTE If using multi-tenant swift store, please make sure that you do not set a swift configuration file with the <i>swift_store_config_file</i> option. Possible values: • True • False Related options: • <code>swift_store_config_file</code>

Configuration option = Default value	Type	Description
swift_store_multiple_containers_seed = 0	integer value	Seed indicating the number of containers to use for storing images. When using a single-tenant store, images can be stored in one or more than one containers. When set to 0, all images will be stored in one single container. When set to an integer value between 1 and 32, multiple containers will be used to store images. This configuration option will determine how many containers are created. 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. Please refer to swift_store_container for more detail on the naming convention. More detail about using multiple containers can be found at https://specs.openstack.org/openstack/glance-specs/specs/kilo/swift-store-multiple-containers.html  NOTE This is used only when swift_store_multi_tenant is disabled. Possible values: • A non-negative integer less than or equal to 32 Related options: • swift_store_container • swift_store_multi_tenant • swift_store_create_container_on_put

Configuration option = Default value	Type	Description
swift_store_region = None	string value	The region of Swift endpoint to use by Glance. Provide a string value representing a Swift region where Glance can connect to for image storage. By default, there is no region set. When Glance uses Swift as the storage backend to store images for a specific tenant that has multiple endpoints, setting of a Swift region with swift_store_region allows Glance to connect to Swift in the specified region as opposed to a single region connectivity. This option can be configured for both single-tenant and multi-tenant storage.  NOTE Setting the region with swift_store_region is tenant-specific and is necessary only if the tenant has multiple endpoints across different regions. Possible values: • A string value representing a valid Swift region. Related Options: • None

Configuration option = Default value	Type	Description
swift_store_retry_get_count = 0	integer value	The number of times a Swift download will be retried before the request fails. Provide an integer value representing the number of times an image download must be retried before erroring out. The default value is zero (no retry on a failed image download). When set to a positive integer value, swift_store_retry_get_count ensures that the download is attempted this many more times upon a download failure before sending an error message. Possible values: • Zero • Positive integer value Related Options: • None
swift_store_service_type = object-store	string value	Type of Swift service to use. Provide a string value representing the service type to use for storing images while using Swift backend storage. The default service type is set to object-store.  NOTE If swift_store_auth_version is set to 2, the value for this configuration option needs to be object-store. If using a higher version of Keystone or a different auth scheme, this option may be modified. Possible values: • A string representing a valid service type for Swift storage. Related Options: • None

Configuration option = Default value	Type	Description
swift_store_ssl_compression = True	boolean value	SSL layer compression for HTTPS Swift requests. Provide a boolean value to determine whether or not to compress HTTPS Swift requests for images at the SSL layer. By default, compression is enabled. When using Swift as the backend store for Glance image storage, SSL layer compression of HTTPS Swift requests can be set using this option. If set to False, SSL layer compression of HTTPS Swift requests is disabled. Disabling this option may improve performance for images which are already in a compressed format, for example, qcow2. Possible values: • True • False Related Options: • None

Configuration option = Default value	Type	Description
swift_store_use_trusts = True	boolean value	Use trusts for multi-tenant Swift store. This option instructs the Swift store to create a trust for each add/get request when the multi-tenant store is in use. Using trusts allows the Swift store to avoid problems that can be caused by an authentication token expiring during the upload or download of data. By default, swift_store_use_trusts is set to True(use of trusts is enabled). If set to False, a user token is used for the Swift connection instead, eliminating the overhead of trust creation.  NOTE This option is considered only when swift_store_multi_tenant is set to True Possible values: • True • False Related options: • swift_store_multi_tenant
swift_store_user = None	string value	The user to authenticate against the Swift authentication service.

Configuration option = Default value	Type	Description
swift_upload_buffer_dir = None	string value	Directory to buffer image segments before upload to Swift. Provide a string value representing the absolute path to the directory on the glance node where image segments will be buffered briefly before they are uploaded to swift. NOTES: • This is required only when the configuration option swift_buffer_on_upload is set to True. • This directory should be provisioned keeping in mind the swift_store_large_object_chunk_size and the maximum number of images that could be uploaded simultaneously by a given glance node. Possible values: • String value representing an absolute directory path Related options: • swift_buffer_on_upload • swift_store_large_object_chunk_size
vmware_api_retry_count = 10	integer value	The number of VMware API retries. This configuration option specifies the number of times the VMware ESX/VC server API must be retried upon connection related issues or server API call overload. It is not possible to specify <i>retry forever</i>. Possible Values: • Any positive integer value Related options: • None

Configuration option = Default value	Type	Description
vmware_ca_file = None	string value	Absolute path to the CA bundle file. This configuration option enables the operator to use a custom Certificate Authority File to verify the ESX/vCenter certificate. If this option is set, the "vmware_insecure" option will be ignored and the CA file specified will be used to authenticate the ESX/vCenter server certificate and establish a secure connection to the server. Possible Values: - Any string that is a valid absolute path to a CA file Related options: - vmware_insecure
vmware_datastores = None	multi valued	The datastores where the image can be stored. This configuration option specifies the datastores where the image can be stored in the VMWare store backend. This option may be specified multiple times for specifying multiple datastores. 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 ":" to specify the priority. 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. Possible Values: Any string of the format: <datacenter_path>:<datastore_name>:<optional_weight> Related options: * None

Configuration option = Default value	Type	Description
vmware_insecure = False	boolean value	Set verification of the ESX/vCenter server certificate. This configuration option takes a boolean value to determine whether or not to verify the ESX/vCenter server certificate. If this option is set to True, the ESX/vCenter server certificate is not verified. If this option is set to False, then the default CA truststore is used for verification. This option is ignored if the "vmware_ca_file" option is set. In that case, the ESX/vCenter server certificate will then be verified using the file specified using the "vmware_ca_file" option . Possible Values: • True • False Related options: • vmware_ca_file
vmware_server_host = None	host address value	Address of the ESX/ESXi or vCenter Server target system. This configuration option sets the address of the ESX/ESXi or vCenter Server target system. This option is required when using the VMware storage backend. The address can contain an IP address (127.0.0.1) or a DNS name (www.my-domain.com). Possible Values: • A valid IPv4 or IPv6 address • A valid DNS name Related options: • vmware_server_username • vmware_server_password

Configuration option = Default value	Type	Description
vmware_server_password = None	string value	Server password. This configuration option takes the password for authenticating with the VMware ESX/ESXi or vCenter Server. This option is required when using the VMware storage backend. Possible Values: - Any string that is a password corresponding to the username specified using the "vmware_server_username" option Related options: - vmware_server_host - vmware_server_username
vmware_server_username = None	string value	Server username. This configuration option takes the username for authenticating with the VMware ESX/ESXi or vCenter Server. This option is required when using the VMware storage backend. Possible Values: - Any string that is the username for a user with appropriate privileges Related options: - vmware_server_host - vmware_server_password
vmware_store_image_dir = /openstack_glance	string value	The directory where the glance images will be stored in the datastore. This configuration option specifies the path to the directory where the glance images will be stored in the VMware datastore. If this option is not set, the default directory where the glance images are stored is openstack_glance. Possible Values: - Any string that is a valid path to a directory Related options: - None

Configuration option = Default value	Type	Description
vmware_task_poll_interval = 5	integer value	Interval in seconds used for polling remote tasks invoked on VMware ESX/VC server. This configuration option takes in the sleep time in seconds for polling an on-going async task as part of the VMWare ESX/VC server API call. Possible Values: - Any positive integer value Related options: - None

3.1.12. image_format

The following table outlines the options available under the **[image_format]** group in the **/etc/glance/glance-api.conf** file.

Table 3.11. image_format

Configuration option = Default value	Type	Description
container_formats = ['ami', 'ari', 'aki', 'bare', 'ovf', 'ova', 'docker', 'compressed']	list value	Supported values for the <i>container_format</i> image attribute
disk_formats = ['ami', 'ari', 'aki', 'vhd', 'vhdx', 'vmdk', 'raw', 'qcow2', 'vdi', 'iso', 'ploop']	list value	Supported values for the <i>disk_format</i> image attribute
vmdk_allowed_types = ['streamOptimized', 'monolithicSparse']	list value	A list of strings describing allowed VMDK <i>create-type</i> subformats that will be allowed. This is recommended to only include single-file-with-sparse-header variants to avoid potential host file exposure due to processing named extents. If this list is empty, then no VMDK image types allowed. Note that this is currently only checked during image conversion (if enabled), and limits the types of VMDK images we will convert from.

3.1.13. keystone_authtoken

The following table outlines the options available under the **[keystone_authtoken]** group in the **/etc/glance/glance-api.conf** file.

Table 3.12. keystone_authtoken

Configuration option = Default value	Type	Description
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load
auth_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint. This option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release. Deprecated since: Queens <i>*Reason:</i>*The <code>auth_uri</code> option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release.
auth_version = None	string value	API version of the Identity API endpoint.
cache = None	string value	Request environment key where the Swift cache object is stored. When <code>auth_token</code> middleware is deployed with a Swift cache, use this option to have the middleware share a caching backend with swift. Otherwise, use the memcached_servers option instead.
cafile = None	string value	A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
certfile = None	string value	Required if identity server requires client certificate
delay_auth_decision = False	boolean value	Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components.

Configuration option = Default value	Type	Description
enforce_token_bind = permissive	string value	Used to control the use and type of token binding. Can be set to: "disabled" to not check token binding. "permissive" (default) to validate binding information if the bind type is of a form known to the server and ignore it if not. "strict" like "permissive" but if the bind type is unknown the token will be rejected. "required" any form of token binding is needed to be allowed. Finally the name of a binding method that must be present in tokens.
http_connect_timeout = None	integer value	Request timeout value for communicating with Identity API server.
http_request_max_retries = 3	integer value	How many times are we trying to reconnect when communicating with Identity API Server.
include_service_catalog = True	boolean value	(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	boolean value	Verify HTTPS connections.
interface = internal	string value	Interface to use for the Identity API endpoint. Valid values are "public", "internal" (default) or "admin".
keyfile = None	string value	Required if identity server requires client certificate
memcache_pool_conn_get_timeout = 10	integer value	(Optional) Number of seconds that an operation will wait to get a memcached client connection from the pool.
memcache_pool_dead_retry = 300	integer value	(Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = 10	integer value	(Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = 3	integer value	(Optional) Socket timeout in seconds for communicating with a memcached server.
memcache_pool_unused_timeout = 60	integer value	(Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
memcache_secret_key = None	string value	(Optional, mandatory if memcache_security_strategy is defined) This string is used for key derivation.

Configuration option = Default value	Type	Description
memcache_security_strategy = None	string value	(Optional) If defined, indicate whether token data should be authenticated or authenticated and encrypted. If MAC, token data is authenticated (with HMAC) in the cache. If ENCRYPT, token data is encrypted and authenticated in the cache. If the value is not one of these options or empty, auth_token will raise an exception on initialization.
memcache_use_advanced_pool = False	boolean value	(Optional) Use the advanced (eventlet safe) memcached client pool. The advanced pool will only work under python 2.x.
memcached_servers = None	list value	Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
region_name = None	string value	The region in which the identity server can be found.
service_token_roles = ['service']	list value	A choice of roles that must be present in a service token. Service tokens are allowed to request that an expired token can be used and so this check should tightly control that only actual services should be sending this token. Roles here are applied as an ANY check so any role in this list must be present. For backwards compatibility reasons this currently only affects the allow_expired check.
service_token_roles_required = False	boolean value	For backwards compatibility reasons we must let valid service tokens pass that don't pass the service_token_roles check as valid. Setting this true will become the default in a future release and should be enabled if possible.
service_type = None	string value	The name or type of the service as it appears in the service catalog. This is used to validate tokens that have restricted access rules.
token_cache_time = 300	integer value	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.

Configuration option = Default value	Type	Description
www_authenticate_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint.

3.1.14. oslo_concurrency

The following table outlines the options available under the **[oslo_concurrency]** group in the `/etc/glance/glance-api.conf` file.

Table 3.13. oslo_concurrency

Configuration option = Default value	Type	Description
disable_process_locking = False	boolean value	Enables or disables inter-process locks.
lock_path = None	string value	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 <code>OSLO_LOCK_PATH</code> . If external locks are used, a lock path must be set.

3.1.15. oslo_messaging_amqp

The following table outlines the options available under the **[oslo_messaging_amqp]** group in the `/etc/glance/glance-api.conf` file.

Table 3.14. oslo_messaging_amqp

Configuration option = Default value	Type	Description
addressing_mode = dynamic	string value	Indicates the addressing mode used by the driver. Permitted values: <i>legacy</i> - use legacy non-routable addressing <i>routable</i> - use routable addresses <i>dynamic</i> - use legacy addresses if the message bus does not support routing otherwise use routable addressing

Configuration option = Default value	Type	Description
anycast_address = anycast	string value	Appended to the address prefix when sending to a group of consumers. Used by the message bus to identify messages that should be delivered in a round-robin fashion across consumers.
broadcast_prefix = broadcast	string value	address prefix used when broadcasting to all servers
connection_retry_backoff = 2	integer value	Increase the connection_retry_interval by this many seconds after each unsuccessful failover attempt.
connection_retry_interval = 1	integer value	Seconds to pause before attempting to re-connect.
connection_retry_interval _max = 30	integer value	Maximum limit for connection_retry_interval + connection_retry_backoff
container_name = None	string value	Name for the AMQP container. must be globally unique. Defaults to a generated UUID
default_notification_exch ange = None	string value	Exchange name used in notification addresses. Exchange name resolution precedence: Target.exchange if set else default_notification_exchange if set else control_exchange if set else <i>notify</i>
default_notify_timeout = 30	integer value	The deadline for a sent notification message delivery. Only used when caller does not provide a timeout expiry.
default_reply_retry = 0	integer value	The maximum number of attempts to re-send a reply message which failed due to a recoverable error.
default_reply_timeout = 30	integer value	The deadline for an rpc reply message delivery.
default_rpc_exchange = None	string value	Exchange name used in RPC addresses. Exchange name resolution precedence: Target.exchange if set else default_rpc_exchange if set else control_exchange if set else <i>rpc</i>
default_send_timeout = 30	integer value	The deadline for an rpc cast or call message delivery. Only used when caller does not provide a timeout expiry.
default_sender_link_time out = 600	integer value	The duration to schedule a purge of idle sender links. Detach link after expiry.

Configuration option = Default value	Type	Description
group_request_prefix = unicast	string value	address prefix when sending to any server in group
idle_timeout = 0	integer value	Timeout for inactive connections (in seconds)
link_retry_delay = 10	integer value	Time to pause between re-connecting an AMQP 1.0 link that failed due to a recoverable error.
multicast_address = multicast	string value	Appended to the address prefix when sending a fanout message. Used by the message bus to identify fanout messages.
notify_address_prefix = openstack.org/om/notify	string value	Address prefix for all generated Notification addresses
notify_server_credit = 100	integer value	Window size for incoming Notification messages
pre_settled = ['rpc-cast', 'rpc-reply']	multi valued	Send messages of this type pre-settled. Pre-settled messages will not receive acknowledgement from the peer. Note well: pre-settled messages may be silently discarded if the delivery fails. Permitted values: <i>rpc-call</i> - send RPC Calls pre-settled <i>rpc-reply</i> - send RPC Replies pre-settled <i>rpc-cast</i> - Send RPC Casts pre-settled <i>notify</i> - Send Notifications pre-settled
pseudo_vhost = True	boolean value	Enable virtual host support for those message buses that do not natively support virtual hosting (such as qpid). When set to true the virtual host name will be added to all message bus addresses, effectively creating a private <i>subnet</i> per virtual host. Set to False if the message bus supports virtual hosting using the <i>hostname</i> field in the AMQP 1.0 Open performative as the name of the virtual host.
reply_link_credit = 200	integer value	Window size for incoming RPC Reply messages.
rpc_address_prefix = openstack.org/om/rpc	string value	Address prefix for all generated RPC addresses
rpc_server_credit = 100	integer value	Window size for incoming RPC Request messages
`sasldb_config_dir = `	string value	Path to directory that contains the SASL configuration
`sasldb_config_name = `	string value	Name of configuration file (without .conf suffix)

Configuration option = Default value	Type	Description
<code>`sasl_default_realm = `</code>	string value	SASL realm to use if no realm present in username
<code>`sasl_mechanisms = `</code>	string value	Space separated list of acceptable SASL mechanisms
server_request_prefix = exclusive	string value	address prefix used when sending to a specific server
ssl = False	boolean value	Attempt to connect via SSL. If no other ssl-related parameters are given, it will use the system's CA-bundle to verify the server's certificate.
<code>`ssl_ca_file = `</code>	string value	CA certificate PEM file used to verify the server's certificate
<code>`ssl_cert_file = `</code>	string value	Self-identifying certificate PEM file for client authentication
<code>`ssl_key_file = `</code>	string value	Private key PEM file used to sign <code>ssl_cert_file</code> certificate (optional)
ssl_key_password = None	string value	Password for decrypting <code>ssl_key_file</code> (if encrypted)
ssl_verify_vhost = False	boolean value	By default SSL checks that the name in the server's certificate matches the hostname in the <code>transport_url</code> . In some configurations it may be preferable to use the virtual hostname instead, for example if the server uses the Server Name Indication TLS extension (rfc6066) to provide a certificate per virtual host. Set <code>ssl_verify_vhost</code> to True if the server's SSL certificate uses the virtual host name instead of the DNS name.
trace = False	boolean value	Debug: dump AMQP frames to stdout
unicast_address = unicast	string value	Appended to the address prefix when sending to a particular RPC/Notification server. Used by the message bus to identify messages sent to a single destination.

3.1.16. oslo_messaging_kafka

The following table outlines the options available under the **[oslo_messaging_kafka]** group in the `/etc/glance/glance-api.conf` file.

Table 3.15. oslo_messaging_kafka

Configuration option = Default value	Type	Description
compression_codec = none	string value	The compression codec for all data generated by the producer. If not set, compression will not be used. Note that the allowed values of this depend on the kafka version
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
consumer_group = oslo_messaging_consumer	string value	Group id for Kafka consumer. Consumers in one group will coordinate message consumption
enable_auto_commit = False	boolean value	Enable asynchronous consumer commits
kafka_consumer_timeout = 1.0	floating point value	Default timeout(s) for Kafka consumers
kafka_max_fetch_bytes = 1048576	integer value	Max fetch bytes of Kafka consumer
max_poll_records = 500	integer value	The maximum number of records returned in a poll call
pool_size = 10	integer value	Pool Size for Kafka Consumers
producer_batch_size = 16384	integer value	Size of batch for the producer async send
producer_batch_timeout = 0.0	floating point value	Upper bound on the delay for KafkaProducer batching in seconds
sasl_mechanism = PLAIN	string value	Mechanism when security protocol is SASL
security_protocol = PLAINTEXT	string value	Protocol used to communicate with brokers
<code>`ssl_cafile = `</code>	string value	CA certificate PEM file used to verify the server certificate
<code>`ssl_client_cert_file = `</code>	string value	Client certificate PEM file used for authentication.
<code>`ssl_client_key_file = `</code>	string value	Client key PEM file used for authentication.

Configuration option = Default value	Type	Description
<code>`ssl_client_key_password = `</code>	string value	Client key password file used for authentication.

3.1.17. oslo_messaging_notifications

The following table outlines the options available under the **[oslo_messaging_notifications]** group in the `/etc/glance/glance-api.conf` file.

Table 3.16. oslo_messaging_notifications

Configuration option = Default value	Type	Description
driver = []	multi valued	The Drivers(s) to handle sending notifications. Possible values are messaging, messagingv2, routing, log, test, noop
retry = -1	integer value	The maximum number of attempts to re-send a notification message which failed to be delivered due to a recoverable error. 0 - No retry, -1 - indefinite
topics = ['notifications']	list value	AMQP topic used for OpenStack notifications.
transport_url = None	string value	A URL representing the messaging driver to use for notifications. If not set, we fall back to the same configuration used for RPC.

3.1.18. oslo_messaging_rabbit

The following table outlines the options available under the **[oslo_messaging_rabbit]** group in the `/etc/glance/glance-api.conf` file.

Table 3.17. oslo_messaging_rabbit

Configuration option = Default value	Type	Description
amqp_auto_delete = False	boolean value	Auto-delete queues in AMQP.
amqp_durable_queues = False	boolean value	Use durable queues in AMQP.

Configuration option = Default value	Type	Description
direct_mandatory_flag = True	boolean value	(DEPRECATED) Enable/Disable the RabbitMQ mandatory flag for direct send. The direct send is used as reply, so the MessageUndeliverable exception is raised in case the client queue does not exist. MessageUndeliverable exception will be used to loop for a timeout to let a chance to sender to recover. This flag is deprecated and it will not be possible to deactivate this functionality anymore
enable_cancel_on_failover = False	boolean value	Enable x-cancel-on-ha-failover flag so that rabbitmq server will cancel and notify consumers when queue is down
heartbeat_in_pthread = True	boolean value	Run the health check heartbeat thread through a native python thread by default. If this option is equal to False then the health check heartbeat will inherit the execution model from the parent process. For example if the parent process has monkey patched the stdlib by using eventlet/greenlet then the heartbeat will be run through a green thread.
heartbeat_rate = 2	integer value	How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = 60	integer value	Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables heartbeat).
kombu_compression = None	string value	EXPERIMENTAL: Possible values are: gzip, bz2. If not set compression will not be used. This option may not be available in future versions.
kombu_failover_strategy = round-robin	string value	Determines how the next RabbitMQ node is chosen in case the one we are currently connected to becomes unavailable. Takes effect only if more than one RabbitMQ node is provided in config.
kombu_missing_consumer_retry_timeout = 60	integer value	How long to wait a missing client before abandoning to send it its replies. This value should not be longer than rpc_response_timeout.
kombu_reconnect_delay = 1.0	floating point value	How long to wait before reconnecting in response to an AMQP consumer cancel notification.

Configuration option = Default value	Type	Description
rabbit_ha_queues = False	boolean value	Try to use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database. In RabbitMQ 3.0, queue mirroring is no longer controlled by the x-ha-policy argument when declaring a queue. If you just want to make sure that all queues (except those with auto-generated names) are mirrored across all nodes, run: "rabbitmqctl set_policy HA ^(?!amq\.).* {"ha-mode": "all"}"
rabbit_interval_max = 30	integer value	Maximum interval of RabbitMQ connection retries. Default is 30 seconds.
rabbit_login_method = AMQPLAIN	string value	The RabbitMQ login method.
rabbit_qos_prefetch_count = 0	integer value	Specifies the number of messages to prefetch. Setting to zero allows unlimited messages.
rabbit_retry_backoff = 2	integer value	How long to backoff for between retries when connecting to RabbitMQ.
rabbit_retry_interval = 1	integer value	How frequently to retry connecting with RabbitMQ.
rabbit_transient_queues_ttl = 1800	integer value	Positive integer representing duration in seconds for queue TTL (x-expires). Queues which are unused for the duration of the TTL are automatically deleted. The parameter affects only reply and fanout queues.
ssl = False	boolean value	Connect over SSL.
<code>`ssl_ca_file = `</code>	string value	SSL certification authority file (valid only if SSL enabled).
<code>`ssl_cert_file = `</code>	string value	SSL cert file (valid only if SSL enabled).
<code>`ssl_key_file = `</code>	string value	SSL key file (valid only if SSL enabled).
<code>`ssl_version = `</code>	string value	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.

3.1.19. oslo_middleware

The following table outlines the options available under the **[oslo_middleware]** group in the **/etc/glance/glance-api.conf** file.

Table 3.18. oslo_middleware

Configuration option = Default value	Type	Description
enable_proxy_headers_parsing = False	boolean value	Whether the application is behind a proxy or not. This determines if the middleware should parse the headers or not.

3.1.20. oslo_policy

The following table outlines the options available under the **[oslo_policy]** group in the **/etc/glance/glance-api.conf** file.

Table 3.19. oslo_policy

Configuration option = Default value	Type	Description
enforce_new_defaults = False	boolean value	This option controls whether or not to use old deprecated defaults when evaluating policies. If True , the old deprecated defaults are not going to be evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be disallowed. It is encouraged to enable this flag along with the enforce_scope flag so that you can get the benefits of new defaults and scope_type together
enforce_scope = False	boolean value	This option controls whether or not to enforce scope when evaluating policies. If True , the scope of the token used in the request is compared to the scope_types of the policy being enforced. If the scopes do not match, an InvalidScope exception will be raised. If False , a message will be logged informing operators that policies are being invoked with mismatching scope.
policy_default_rule = default	string value	Default rule. Enforced when a requested rule is not found.
policy_dirs = ['policy.d']	multi valued	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.

Configuration option = Default value	Type	Description
policy_file = policy.yaml	string value	The relative or absolute path of a file that maps roles to permissions for a given service. Relative paths must be specified in relation to the configuration file setting this option.
remote_content_type = application/x-www-form-urlencoded	string value	Content Type to send and receive data for REST based policy check
remote_ssl_ca_cert_file = None	string value	Absolute path to ca cert file for REST based policy check
remote_ssl_client_cert_file = None	string value	Absolute path to client cert for REST based policy check
remote_ssl_client_key_file = None	string value	Absolute path client key file REST based policy check
remote_ssl_verify_server_cert = False	boolean value	server identity verification for REST based policy check

3.1.21. paste_deploy

The following table outlines the options available under the **[paste_deploy]** group in the **/etc/glance/glance-api.conf** file.

Table 3.20. paste_deploy

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
config_file = None	string value	Name of the paste configuration file. Provide a string value representing the name of the paste configuration file to use for configuring pipelines for server application deployments. NOTES: ● Provide the name or the path relative to the glance directory for the paste configuration file and not the absolute path. ● The sample paste configuration file shipped with Glance need not be edited in most cases as it comes with ready-made pipelines for all common deployment flavors. If no value is specified for this option, the paste.ini file with the prefix of the corresponding Glance service's configuration file name will be searched for in the known configuration directories. (For example, if this option is missing from or has no value set in glance-api.conf, the service will look for a file named glance-api-paste.ini.) If the paste configuration file is not found, the service will not start. Possible values: ● A string value representing the name of the paste configuration file. Related Options: ● flavor

Configuration option = Default value	Type	Description
flavor = None	string value	Deployment flavor to use in the server application pipeline. Provide a string value representing the appropriate deployment flavor used in the server application pipeline. This is typically the partial name of a pipeline in the paste configuration file with the service name removed. For example, if your paste section name in the paste configuration file is [pipeline:glance-api-keystone], set flavor to keystone. Possible values: - String value representing a partial pipeline name. Related Options: - config_file

3.1.22. profiler

The following table outlines the options available under the **[profiler]** group in the **/etc/glance/glance-api.conf** file.

Table 3.21. profiler

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
connection_string = messaging://	string value	Connection string for a notifier backend. Default value is messaging:// which sets the notifier to oslo_messaging. Examples of possible values: ● messaging:// - use oslo_messaging driver for sending spans. ● redis://127.0.0.1:6379 - use redis driver for sending spans. ● mongodb://127.0.0.1:27017 - use mongodb driver for sending spans. ● elasticsearch://127.0.0.1:9200 - use elasticsearch driver for sending spans. ● jaeger://127.0.0.1:6831 - use jaeger tracing as driver for sending spans.
enabled = False	boolean value	Enable the profiling for all services on this node. Default value is False (fully disable the profiling feature). Possible values: ● True: Enables the feature ● False: Disables the feature. The profiling cannot be started via this project operations. If the profiling is triggered by another project, this project part will be empty.
es_doc_type = notification	string value	Document type for notification indexing in elasticsearch.
es_scroll_size = 10000	integer value	Elasticsearch splits large requests in batches. This parameter defines maximum size of each batch (for example: es_scroll_size=10000).
es_scroll_time = 2m	string value	This parameter is a time value parameter (for example: es_scroll_time=2m), indicating for how long the nodes that participate in the search will maintain relevant resources in order to continue and support it.

Configuration option = Default value	Type	Description
filter_error_trace = False	boolean value	Enable filter traces that contain error/exception to a separated place. Default value is set to False. Possible values: • True: Enable filter traces that contain error/exception. • False: Disable the filter.
hmac_keys = SECRET_KEY	string value	Secret key(s) to use for encrypting context data for performance profiling. This string value should have the following format: <key1>[,<key2>,...<keyn>], where each key is some random string. A user who triggers the profiling via the REST API has to set one of these keys in the headers of the REST API call to include profiling results of this node for this particular project. Both "enabled" flag and "hmac_keys" config options should be set to enable profiling. Also, to generate correct profiling information across all services at least one key needs to be consistent between OpenStack projects. This ensures it can be used from client side to generate the trace, containing information from all possible resources.
sentinel_service_name = mymaster	string value	Redis sentinel uses a service name to identify a master redis service. This parameter defines the name (for example: sentinal_service_name=mymaster).
socket_timeout = 0.1	floating point value	Redis sentinel provides a timeout option on the connections. This parameter defines that timeout (for example: socket_timeout=0.1).

Configuration option = Default value	Type	Description
trace_sqlalchemy = False	boolean value	Enable SQL requests profiling in services. Default value is False (SQL requests won't be traced). Possible values: • True: Enables SQL requests profiling. Each SQL query will be part of the trace and can be analyzed by how much time was spent for that. • False: Disables SQL requests profiling. The spent time is only shown on a higher level of operations. Single SQL queries cannot be analyzed this way.

3.1.23. store_type_location_strategy

The following table outlines the options available under the **[store_type_location_strategy]** group in the **/etc/glance/glance-api.conf** file.

Table 3.22. store_type_location_strategy

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
store_type_preference = []	list value	Preference order of storage backends. Provide a comma separated list of store names in the order in which images should be retrieved from storage backends. These store names must be registered with the stores configuration option.  NOTE The store_type_preference configuration option is applied only if store_type is chosen as a value for the location_strategy configuration option. An empty list will not change the location order. Possible values: ● Empty list ● Comma separated list of registered store names. Legal values are: ○ file ○ http ○ rbd ○ swift ○ cinder ○ vmware Related options: ● location_strategy ● stores

3.1.24. task

The following table outlines the options available under the **[task]** group in the **/etc/glance/glance-api.conf** file.

Table 3.23. task

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
task_executor = taskflow	string value	Task executor to be used to run task scripts. Provide a string value representing the executor to use for task executions. By default, TaskFlow executor is used. TaskFlow helps make task executions easy, consistent, scalable and reliable. It also enables creation of lightweight task objects and/or functions that are combined together into flows in a declarative manner. Possible values: ● taskflow Related Options: ● None
task_time_to_live = 48	integer value	Time in hours for which a task lives after, either succeeding or failing

Configuration option = Default value	Type	Description
work_dir = None	string value	Absolute path to the work directory to use for asynchronous task operations. The directory set here will be used to operate over images - normally before they are imported in the destination store.  NOTE When providing a value for work_dir, please make sure that 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 with 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. Possible values: - String value representing the absolute path to the working directory Related Options: - None

3.1.25. taskflow_executor

The following table outlines the options available under the **[taskflow_executor]** group in the **/etc/glance/glance-api.conf** file.

Table 3.24. taskflow_executor

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
conversion_format = None	string value	Set the desired image conversion format. Provide a valid image format to which you want images to be converted before they are stored for consumption by Glance. Appropriate image format conversions are desirable for specific storage backends in order to facilitate efficient handling of bandwidth and usage of the storage infrastructure. By default, conversion_format is not set and must be set explicitly in the configuration file. The allowed values for this option are raw, qcow2 and vmdk. The raw format is the unstructured disk format and should be chosen when RBD or Ceph storage backends are used for image storage. qcow2 is supported by the QEMU emulator that expands dynamically and supports Copy on Write. The vmdk is another common disk format supported by many common virtual machine monitors like VMWare Workstation. Possible values: ● qcow2 ● raw ● vmdk Related options: ● disk_formats

Configuration option = Default value	Type	Description
engine_mode = parallel	string value	Set the taskflow engine mode. Provide a string type value to set the mode in which the taskflow engine would schedule tasks to the workers on the hosts. Based on this mode, the engine executes tasks either in single or multiple threads. The possible values for this configuration option are: serial and parallel. When set to serial, the engine runs all the tasks in a single thread which results in serial execution of tasks. Setting this to parallel makes the engine run tasks in multiple threads. This results in parallel execution of tasks. Possible values: ● serial ● parallel Related options: ● max_workers
max_workers = 10	integer value	Set the number of engine executable tasks. Provide an integer value to limit the number of workers that can be instantiated on the hosts. In other words, this number defines the number of parallel tasks that can be executed at the same time by the taskflow engine. This value can be greater than one when the engine mode is set to parallel. Possible values: ● Integer value greater than or equal to 1 Related options: ● engine_mode

3.1.26. wsgi

The following table outlines the options available under the **[wsgi]** group in the **/etc/glance/glance-api.conf** file.

Table 3.25. wsgi

Configuration option = Default value	Type	Description
python_interpreter = /usr/bin/python3	string value	Path to the python interpreter to use when spawning external processes. By default this is sys.executable, which should be the same interpreter running Glance itself. However, in some situations (i.e. uwsgi) this may not actually point to a python interpreter itself.
task_pool_threads = 16	integer value	The number of threads (per worker process) in the pool for processing asynchronous tasks. This controls how many asynchronous tasks (i.e. for image interoperable import) each worker can run at a time. If this is too large, you may have increased memory footprint per worker and/or you may overwhelm other system resources such as disk or outbound network bandwidth. If this is too small, image import requests will have to wait until a thread becomes available to begin processing.

3.2. GLANCE-SCRUBBER.CONF

This section contains options for the `/etc/glance/glance-scrubber.conf` file.

3.2.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/glance/glance-scrubber.conf` file.

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
allow_additional_image_properties = True	boolean value	Allow users to add additional/custom properties to images. Glance defines a standard set of properties (in its schema) that appear on every image. These properties are also known as base properties. In addition to these properties, Glance allows users to add custom properties to images. These are known as additional properties. By default, this configuration option is set to True and users are allowed to add additional properties. The number of additional properties that can be added to an image can be controlled via image_property_quota configuration option. Possible values: • True • False Related options: • image_property_quota Deprecated since: Ussuri Reason: This option is redundant. Control custom image property usage via the <i>image_property_quota</i> configuration option. This option is scheduled to be removed during the Victoria development cycle.

Configuration option = Default value	Type	Description
api_limit_max = 1000	integer value	Maximum number of results that could be returned by a request. As described in the help text of limit_param_default, some requests may return multiple results. The number of results to be returned are governed either by the limit parameter in the request or the limit_param_default configuration option. The value in either case, can't be greater than the absolute maximum defined by this configuration option. Anything greater than this value is trimmed down to the maximum value defined here. NOTE Setting this to a very large value may slow down database queries and increase response times. Setting this to a very low value may result in poor user experience. Possible values: • Any positive integer Related options: • <code>limit_param_default</code>

Configuration option = Default value	Type	Description
daemon = False	boolean value	Run scrubber as a daemon. This boolean configuration option indicates whether scrubber should run as a long-running process that wakes up at regular intervals to scrub images. The wake up interval can be specified using the configuration option wakeup_time. If this configuration option is set to False, which is the default value, scrubber runs once to scrub images and exits. In this case, if the operator wishes to implement continuous scrubbing of images, scrubber needs to be scheduled as a cron job. Possible values: • True • False Related options: • wakeup_time
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.

Configuration option = Default value	Type	Description
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
delayed_delete = False	boolean value	Turn on/off delayed delete. Typically when an image is deleted, the glance-api service puts the image into deleted state and deletes its data at the same time. Delayed delete is a feature in Glance that delays the actual deletion of image data until a later point in time (as determined by the configuration option scrub_time). When delayed delete is turned on, the glance-api service puts the image into pending_delete state upon deletion and leaves the image data in the storage backend for the image scrubber to delete at a later time. The image scrubber will move the image into deleted state upon successful deletion of image data.  NOTE When delayed delete is turned on, image scrubber MUST be running as a periodic task to prevent the backend storage from filling up with undesired usage. Possible values: • True • False Related options: • scrub_time • wakeup_time • scrub_pool_size

Configuration option = Default value	Type	Description
digest_algorithm = sha256	string value	Digest algorithm to use for digital signature. Provide a string value representing the digest algorithm to use for generating digital signatures. By default, sha256 is used. To get a list of the available algorithms supported by the version of OpenSSL on your platform, run the command: openssl list-message-digest-algorithms. Examples are <i>sha1</i>, <i>sha256</i>, and <i>sha512</i>.  NOTE digest_algorithm is not related to Glance's image signing and verification. It is only used to sign the universally unique identifier (UUID) as a part of the certificate file and key file validation. Possible values: ● An OpenSSL message digest algorithm identifier Relation options: ● None
enabled_import_methods = ['glance-direct', 'web-download', 'copy-image']	list value	List of enabled Image Import Methods  'glance-direct', 'copy-image' and 'web-download' are enabled by default.  Related options: ** [DEFAULT]/node_staging_uri

Configuration option = Default value	Type	Description
enforce_secure_rbac = False	boolean value	Enforce API access based on common persona definitions used across OpenStack. Enabling this option formalizes project-specific read/write operations, like creating private images or updating the status of shared image, behind the member role. It also formalizes a read-only variant useful for project-specific API operations, like listing private images in a project, behind the reader role. Operators should take an opportunity to understand glance's new image policies, audit assignments in their deployment, and update permissions using the default roles in keystone (e.g., admin, member, and reader). Related options: • [oslo_policy]/enforce_new_defaults Deprecated since: Wallaby Reason: This option has been introduced to require operators to opt into enforcing authorization based on common RBAC personas, which is EXPERIMENTAL as of the Wallaby release. This behavior will be the default and STABLE in a future release, allowing this option to be removed.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.

Configuration option = Default value	Type	Description
hashing_algorithm = sha512	string value	Secure hashing algorithm used for computing the <i>os_hash_value</i> property. This option configures the Glance "multihash", which consists of two image properties: the <i>os_hash_algo</i> and the <i>os_hash_value</i>. The <i>os_hash_algo</i> will be populated by the value of this configuration option, and the <i>os_hash_value</i> will be populated by the hexdigest computed when the algorithm is applied to the uploaded or imported image data. The value must be a valid secure hash algorithm name recognized by the python <i>hashlib</i> library. You can determine what these are by examining the <i>hashlib.algorithms_available</i> data member of the version of the library being used in your Glance installation. For interoperability purposes, however, we recommend that you use the set of secure hash names supplied by the <i>hashlib.algorithms_guaranteed</i> data member because those algorithms are guaranteed to be supported by the <i>hashlib</i> library on all platforms. Thus, any image consumer using <i>hashlib</i> locally should be able to verify the <i>os_hash_value</i> of the image. The default value of <i>sha512</i> is a performant secure hash algorithm. If this option is misconfigured, any attempts to store image data will fail. For that reason, we recommend using the default value. Possible values: - Any secure hash algorithm name recognized by the Python <i>hashlib</i> library Related options: - None
image_location_quota = 10	integer value	Maximum number of locations allowed on an image. Any negative value is interpreted as unlimited. Related options: None

Configuration option = Default value	Type	Description
image_member_quota = 128	integer value	Maximum number of image members per image. This limits the maximum of users an image can be shared with. Any negative value is interpreted as unlimited. Related options: • None
image_property_quota = 128	integer value	Maximum number of properties allowed on an image. This enforces an upper limit on the number of additional properties an image can have. Any negative value is interpreted as unlimited.  NOTE This won't have any impact if additional properties are disabled. Please refer to allow_additional_image_properties. Related options: • allow_additional_image_properties

Configuration option = Default value	Type	Description
image_size_cap = 1099511627776	integer value	Maximum size of image a user can upload in bytes. An image upload greater than the size mentioned here would result in an image creation failure. This configuration option defaults to 1099511627776 bytes (1 TiB). NOTES: • This value should only be increased after careful consideration and must be set less than or equal to 8 EiB (9223372036854775808). • This value must be set with careful consideration of the backend storage capacity. Setting this to a very low value may result in a large number of image failures. And, setting this to a very large value may result in faster consumption of storage. Hence, this must be set according to the nature of images created and storage capacity available. Possible values: • Any positive number less than or equal to 9223372036854775808
image_tag_quota = 128	integer value	Maximum number of tags allowed on an image. Any negative value is interpreted as unlimited. Related options: • None
<code>`instance_format = [instance: %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.
<code>`instance_uuid_format = [instance: %(uuid)s] `</code>	string value	The format for an instance UUID that is passed with the log message.

Configuration option = Default value	Type	Description
limit_param_default = 25	integer value	The default number of results to return for a request. Responses to certain API requests, like list images, may return multiple items. The number of results returned can be explicitly controlled by specifying the limit parameter in the API request. However, if a limit parameter is not specified, this configuration value will be used as the default number of results to be returned for any API request. NOTES: • The value of this configuration option may not be greater than the value specified by api_limit_max. • Setting this to a very large value may slow down database queries and increase response times. Setting this to a very low value may result in poor user experience. Possible values: • Any positive integer Related options: • <code>api_limit_max</code>
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, <code>log-date-format</code>).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code>. This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code>. This option is ignored if <code>log_config_append</code> is set.

Configuration option = Default value	Type	Description
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless log_rotation_type is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	string value	Format string to use for log messages with context. Used by oslo_log.formatters.ContextFormatter
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is DEBUG. Used by oslo_log.formatters.ContextFormatter
logging_default_format_s tring = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s	string value	Format string to use for log messages when context is undefined. Used by oslo_log.formatters.ContextFormatter
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d ERROR %(name)s % (instance)s	string value	Prefix each line of exception output with this format. Used by oslo_log.formatters.ContextFormatter
logging_user_identity_for mat = %(user)s % (tenant)s %(domain)s % (user_domain)s % (project_domain)s	string value	Defines the format string for %(user_identity)s that is used in logging_context_format_string. Used by oslo_log.formatters.ContextFormatter
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".

Configuration option = Default value	Type	Description
metadata_encryption_key = None	string value	AES key for encrypting store location metadata. Provide a string value representing the AES cipher to use for encrypting Glance store metadata. NOTE The AES key to use must be set to a random string of length 16, 24 or 32 bytes. Possible values: • String value representing a valid AES key Related options: • None

Configuration option = Default value	Type	Description
node_staging_uri = file:///tmp/staging/	string value	The URL provides location where the temporary data will be stored This option is for Glance internal use only. Glance will save the image data uploaded by the user to <i>staging</i> endpoint during the image import process. This option does not change the <i>staging</i> API endpoint by any means.  NOTE It is discouraged to use same path as [task]/work_dir  NOTE <i>file://<absolute-directory-path></i> is the only option api_image_import flow will support for now.  NOTE The staging path must be on shared filesystem available to all Glance API nodes. Possible values: ● String starting with <i>file://</i> followed by absolute FS path Related options: ● [task]/work_dir
publish_errors = False	boolean value	Enables or disables publication of error events.

Configuration option = Default value	Type	Description
pydev_worker_debug_host = None	host address value	Host address of the pydev server. Provide a string value representing the hostname or IP of the pydev server to use for debugging. The pydev server listens for debug connections on this address, facilitating remote debugging in Glance. Possible values: - Valid hostname - Valid IP address Related options: - None
pydev_worker_debug_port = 5678	port value	Port number that the pydev server will listen on. Provide a port number to bind the pydev server to. The pydev process accepts debug connections on this port and facilitates remote debugging in Glance. Possible values: - A valid port number Related options: - None
rate_limit_burst = 0	integer value	Maximum number of logged messages per rate_limit_interval.
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.

Configuration option = Default value	Type	Description
restore = None	string value	Restore the image status from <i>pending_delete</i> to <i>active</i>. This option is used by administrator to reset the image's status from <i>pending_delete</i> to <i>active</i> when the image is deleted by mistake and <i>pending delete</i> feature is enabled in Glance. Please make sure the glance-scrubber daemon is stopped before restoring the image to avoid image data inconsistency. Possible values: ● image's uuid
scrub_pool_size = 1	integer value	The size of thread pool to be used for scrubbing images. When there are a large number of images to scrub, it is beneficial to scrub images in parallel so that the scrub queue stays in control and the backend storage is reclaimed in a timely fashion. This configuration option denotes the maximum number of images to be scrubbed in parallel. The default value is one, which signifies serial scrubbing. Any value above one indicates parallel scrubbing. Possible values: ● Any non-zero positive integer Related options: ● delayed_delete

Configuration option = Default value	Type	Description
scrub_time = 0	integer value	The amount of time, in seconds, to delay image scrubbing. When delayed delete is turned on, an image is put into pending_delete state upon deletion until the scrubber deletes its image data. Typically, soon after the image is put into pending_delete state, it is available for scrubbing. However, scrubbing can be delayed until a later point using this configuration option. This option denotes the time period an image spends in pending_delete state before it is available for scrubbing. It is important to realize that this has storage implications. The larger the scrub_time, the longer the time to reclaim backend storage from deleted images. Possible values: - Any non-negative integer Related options: delayed_delete

Configuration option = Default value	Type	Description
show_image_direct_url = False	boolean value	Show direct image location when returning an image. This configuration option indicates whether to show the direct image location when returning image details to the user. The direct image location is where the image data is stored in backend storage. This image location is shown under the image property direct_url. When multiple image locations exist for an image, the best location is displayed based on the location strategy indicated by the configuration option location_strategy. NOTES: ● Revealing image locations can present a GRAVE SECURITY RISK as image locations can sometimes include credentials. Hence, this is set to False by default. Set this to True with EXTREME CAUTION and ONLY IF you know what you are doing! ● If an operator wishes to avoid showing any image location(s) to the user, then both this option and show_multiple_locations MUST be set to False. Possible values: ● True ● False Related options: ● show_multiple_locations ● location_strategy

Configuration option = Default value	Type	Description
show_multiple_locations = False	boolean value	Show all image locations when returning an image. This configuration option indicates whether to show all the image locations when returning image details to the user. When multiple image locations exist for an image, the locations are ordered based on the location strategy indicated by the configuration opt location_strategy. The image locations are shown under the image property locations. NOTES: ● Revealing image locations can present a GRAVE SECURITY RISK as image locations can sometimes include credentials. Hence, this is set to False by default. Set this to True with EXTREME CAUTION and ONLY IF you know what you are doing! ● See https://wiki.openstack.org/wiki/OSSN/OSSN-0065 for more information. ● If an operator wishes to avoid showing any image location(s) to the user, then both this option and show_image_direct_url MUST be set to False. Possible values: ● True ● False Related options: ● show_image_direct_url ● location_strategy Deprecated since: Newton *Reason:*Use of this option, deprecated since Newton, is a security risk and will be removed once we figure out a way to satisfy those use cases that currently require it. An earlier announcement that the same functionality can be achieved with greater granularity by using policies is incorrect. You cannot work around this option via policy configuration at the present time, though that is the direction we believe the fix will take. Please keep an eye on the Glance release notes to stay up to date on progress in addressing this issue.

Configuration option = Default value	Type	Description
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
user_storage_quota = 0	string value	Maximum amount of image storage per tenant. This enforces an upper limit on the cumulative storage consumed by all images of a tenant across all stores. This is a per-tenant limit. The default unit for this configuration option is Bytes. However, storage units can be specified using case-sensitive literals B, KB, MB, GB and TB representing Bytes, KiloBytes, MegaBytes, GigaBytes and TeraBytes respectively. Note that there should not be any space between the value and unit. Value 0 signifies no quota enforcement. Negative values are invalid and result in errors. Possible values: • A string that is a valid concatenation of a non-negative integer representing the storage value and an optional string literal representing storage units as mentioned above. Related options: • None

Configuration option = Default value	Type	Description
wakeup_time = 300	integer value	Time interval, in seconds, between scrubber runs in daemon mode. Scrubber can be run either as a cron job or daemon. When run as a daemon, this configuration time specifies the time period between two runs. When the scrubber wakes up, it fetches and scrubs all pending_delete images that are available for scrubbing after taking scrub_time into consideration. If the wakeup time is set to a large number, there may be a large number of images to be scrubbed for each run. Also, this impacts how quickly the backend storage is reclaimed. Possible values: - Any non-negative integer Related options: daemon delayed_delete
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
worker_self_reference_url = None	string value	The URL to this worker. If this is set, other glance workers will know how to contact this one directly if needed. For image import, a single worker stages the image and other workers need to be able to proxy the import request to the right one. If unset, this will be considered to be public_endpoint, which normally would be set to the same value on all workers, effectively disabling the proxying behavior. Possible values: • A URL by which this worker is reachable from other workers Related options: • <code>public_endpoint</code>

3.2.2. database

The following table outlines the options available under the **[database]** group in the `/etc/glance/glance-scrubber.conf` file.

Table 3.26. database

Configuration option = Default value	Type	Description
backend = sqlalchemy	string value	The back end to use for the database.
connection = None	string value	The SQLAlchemy connection string to use to connect to the database.
connection_debug = 0	integer value	Verbosity of SQL debugging information: 0=None, 100=Everything.
<code>`connection_parameters = `</code>	string value	Optional URL parameters to append onto the connection URL at connect time; specify as <code>param1=value1&param2=value2&...</code>
connection_recycle_time = 3600	integer value	Connections which have been present in the connection pool longer than this number of seconds will be replaced with a new one the next time they are checked out from the pool.

Configuration option = Default value	Type	Description
connection_trace = False	boolean value	Add Python stack traces to SQL as comment strings.
db_inc_retry_interval = True	boolean value	If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = 20	integer value	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	integer value	If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = 1	integer value	Seconds between retries of a database transaction.
max_overflow = 50	integer value	If set, use this value for max_overflow with SQLAlchemy.
max_pool_size = 5	integer value	Maximum number of SQL connections to keep open in a pool. Setting a value of 0 indicates no limit.
max_retries = 10	integer value	Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.
mysql_enable_ndb = False	boolean value	If True, transparently enables support for handling MySQL Cluster (NDB).
mysql_sql_mode = TRADITIONAL	string value	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	integer value	If set, use this value for pool_timeout with SQLAlchemy.
retry_interval = 10	integer value	Interval between retries of opening a SQL connection.
slave_connection = None	string value	The SQLAlchemy connection string to use to connect to the slave database.
sqlite_synchronous = True	boolean value	If True, SQLite uses synchronous mode.

Configuration option = Default value	Type	Description
use_db_reconnect = False	boolean value	Enable the experimental use of database reconnect on connection lost.
use_tpool = False	boolean value	Enable the experimental use of thread pooling for all DB API calls

3.2.3. glance_store

The following table outlines the options available under the **[glance_store]** group in the **/etc/glance/glance-scrubber.conf** file.

Table 3.27. glance_store

Configuration option = Default value	Type	Description
cinder_api_insecure = False	boolean value	Allow to perform insecure SSL requests to cinder. If this option is set to True, HTTPS endpoint connection is verified using the CA certificates file specified by cinder_ca_certificates_file option. Possible values: • True • False Related options: • cinder_ca_certificates_file
cinder_ca_certificates_file = None	string value	Location of a CA certificates file used for cinder client requests. The specified CA certificates file, if set, is used to verify cinder connections via HTTPS endpoint. If the endpoint is HTTP, this value is ignored. cinder_api_insecure must be set to True to enable the verification. Possible values: • Path to a ca certificates file Related options: • cinder_api_insecure

Configuration option = Default value	Type	Description
cinder_catalog_info = volumev3::publicURL	string value	Information to match when looking for cinder in the service catalog. When the cinder_endpoint_template is not set and any of cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, cinder_store_password is not set, cinder store uses this information to lookup cinder endpoint from the service catalog in the current context. cinder_os_region_name, if set, is taken into consideration to fetch the appropriate endpoint. The service catalog can be listed by the openstack catalog list command. Possible values: ● A string of of the following form: <service_type>:<service_name>: <interface> At least service_type and interface should be specified. service_name can be omitted. Related options: ● cinder_os_region_name ● cinder_endpoint_template ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name ● cinder_store_password

Configuration option = Default value	Type	Description
cinder_endpoint_template = None	string value	Override service catalog lookup with template for cinder endpoint. When this option is set, this value is used to generate cinder endpoint, instead of looking up from the service catalog. This value is ignored if cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, and cinder_store_password are specified. If this configuration option is set, cinder_catalog_info will be ignored. Possible values: ● URL template string for cinder endpoint, where %%(tenant)s is replaced with the current tenant (project) name. For example: http://cinder.openstack.example.org/v2/%%(tenant)s Related options: ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name ● cinder_store_password ● cinder_catalog_info
cinder_enforce_multipath = False	boolean value	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. Possible values: ● True or False Related options: ● cinder_use_multipath

Configuration option = Default value	Type	Description
cinder_http_retries = 3	integer value	Number of cinderclient retries on failed http calls. When a call failed by any errors, cinderclient will retry the call up to the specified times after sleeping a few seconds. Possible values: • A positive integer Related options: • None
cinder_mount_point_base = /var/lib/glance/mnt	string value	Directory where the NFS volume is mounted on the glance node. Possible values: • A string representing absolute path of mount point.
cinder_os_region_name = None	string value	Region name to lookup cinder service from the service catalog. This is used only when cinder_catalog_info is used for determining the endpoint. If set, the lookup for cinder endpoint by this node is filtered to the specified region. It is useful when multiple regions are listed in the catalog. If this is not set, the endpoint is looked up from every region. Possible values: • A string that is a valid region name. Related options: • cinder_catalog_info

Configuration option = Default value	Type	Description
cinder_state_transition_timeout = 300	integer value	Time period, in seconds, to wait for a cinder volume transition to complete. When the cinder volume is created, deleted, or attached to the glance node to read/write the volume data, the volume's state is changed. For example, the newly created volume status changes from creating to available after the creation process is completed. This specifies the maximum time to wait for the status change. If a timeout occurs while waiting, or the status is changed to an unexpected value (e.g. error), the image creation fails. Possible values: • A positive integer Related options: • None

Configuration option = Default value	Type	Description
cinder_store_auth_address = None	string value	The address where the cinder authentication service is listening. When all of cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, and cinder_store_password options are specified, the specified values are always used for the authentication. This is useful to hide the image volumes from users by storing them in a project/tenant specific to the image service. It also enables users to share the image volume among other projects under the control of glance's ACL. If either of these options are not set, the cinder endpoint is looked up from the service catalog, and current context's user and project are used. Possible values: • A valid authentication service address, for example: http://openstack.example.org/identity/v2.0 Related options: • cinder_store_user_name • cinder_store_password • cinder_store_project_name
cinder_store_password = None	string value	Password for the user authenticating against cinder. This must be used with all the following related options. If any of these are not specified, the user of the current context is used. Possible values: • A valid password for the user specified by cinder_store_user_name Related options: • cinder_store_auth_address • cinder_store_user_name • cinder_store_project_name

Configuration option = Default value	Type	Description
cinder_store_project_name = None	string value	Project name where the image volume is stored in cinder. If this configuration option is not set, the project in current context is used. This must be used with all the following related options. If any of these are not specified, the project of the current context is used. Possible values: • A valid project name Related options: • cinder_store_auth_address • cinder_store_user_name • cinder_store_password
cinder_store_user_name = None	string value	User name to authenticate against cinder. This must be used with all the following related options. If any of these are not specified, the user of the current context is used. Possible values: • A valid user name Related options: • cinder_store_auth_address • cinder_store_password • cinder_store_project_name
cinder_use_multipath = False	boolean value	Flag to identify mutipath is supported or not in the deployment. Set it to False if multipath is not supported. Possible values: • True or False Related options: • cinder_enforce_multipath

Configuration option = Default value	Type	Description
cinder_volume_type = None	string value	Volume type that will be used for volume creation in cinder. Some cinder backends can have several volume types to optimize storage usage. Adding this option allows an operator to choose a specific volume type in cinder that can be optimized for images. If this is not set, then the default volume type specified in the cinder configuration will be used for volume creation. Possible values: ● A valid volume type from cinder Related options: ● None NOTE You cannot use an encrypted volume_type associated with an NFS backend. An encrypted volume stored on an NFS backend will raise an exception whenever glance_store tries to write or access image data stored in that volume. Consult your Cinder administrator to determine an appropriate volume_type.

Configuration option = Default value	Type	Description
default_store = file	string value	The default scheme to use for storing images. Provide a string value representing the default scheme to use for storing images. If not set, Glance uses file as the default scheme to store images with the file store.  NOTE The value given for this configuration option must be a valid scheme for a store registered with the stores configuration option. Possible values: ● file ● filesystem ● http ● https ● swift ● swift+http ● swift+https ● swift+config ● rbd ● cinder ● vsphere ● s3 Related Options: ● stores Deprecated since: Rocky Reason: This option is deprecated against new config option ``default_backend`` which acts similar to ``default_store`` config option. This option is scheduled for removal in the U development cycle.

Configuration option = Default value	Type	Description
default_swift_reference = ref1	string value	Reference to default Swift account/backing store parameters. Provide a string value representing a reference to the default set of parameters required for using swift account/backing store for image storage. The default reference value for this configuration option is <i>ref1</i>. This configuration option dereferences the parameters and facilitates image storage in Swift storage backend every time a new image is added. Possible values: • A valid string value Related options: • None
filesystem_store_chunk_size = 65536	integer value	Chunk size, in bytes. The chunk size used when reading or writing image files. Raising this value may improve the throughput but it may also slightly increase the memory usage when handling a large number of requests. Possible Values: • Any positive integer value Related options: • None

Configuration option = Default value	Type	Description
filesystem_store_datadir = /var/lib/glance/images	string value	Directory to which the filesystem backend store writes images. Upon start up, Glance creates the directory if it doesn't already exist and verifies write access to the user under which glance-api runs. If the write access isn't available, a BadStoreConfiguration exception is raised and the filesystem store may not be available for adding new images.  NOTE This directory is used only when filesystem store is used as a storage backend. Either filesystem_store_datadir or filesystem_store_datadirs option must be specified in glance-api.conf. If both options are specified, a BadStoreConfiguration will be raised and the filesystem store may not be available for adding new images. Possible values: ● A valid path to a directory Related options: ● filesystem_store_datadirs ● filesystem_store_file_perm

Configuration option = Default value	Type	Description
filesystem_store_datadirs = None	multi valued	List of directories and their priorities to which the filesystem backend store writes images. The filesystem store can be configured to store images in multiple directories as opposed to using a single directory specified by the filesystem_store_datadir configuration option. When using multiple directories, each directory can be given an optional priority to specify the preference order in which they should be used. Priority is an integer that is concatenated to the directory path with a colon where a higher value indicates higher priority. When two directories have the same priority, the directory with most free space is used. When no priority is specified, it defaults to zero. More information on configuring filesystem store with multiple store directories can be found at https://docs.openstack.org/glance/latest/configuration/configuring.html  NOTE This directory is used only when filesystem store is used as a storage backend. Either filesystem_store_datadir or filesystem_store_datadirs option must be specified in glance-api.conf. If both options are specified, a BadStoreConfiguration will be raised and the filesystem store may not be available for adding new images. Possible values: - List of strings of the following form: <a valid directory path>:<optional integer priority> Related options: filesystem_store_datadir filesystem_store_file_perm

Configuration option = Default value	Type	Description
filesystem_store_file_permissions = 0	integer value	File access permissions for the image files. Set the intended file access permissions for image data. This provides a way to enable other services, e.g. Nova, to consume images directly from the filesystem store. The users running the services that are intended to be given access to could be made a member of the group that owns the files created. Assigning a value less than or equal to zero for this configuration option signifies that no changes be made to the default permissions. This value will be decoded as an octal digit. For more information, please refer the documentation at https://docs.openstack.org/glance/latest/configuration/configuring.html Possible values: ● A valid file access permission ● Zero ● Any negative integer Related options: ● None
filesystem_store_metadata_file = None	string value	Filesystem store metadata file. The path to a file which contains the metadata to be returned with any location associated with the filesystem store. Once this option is set, it is used for new images created afterward only - previously existing images are not affected. The file must contain a valid JSON object. The object should contain the keys id and mountpoint. The value for both keys should be a string. Possible values: ● A valid path to the store metadata file Related options: ● None

Configuration option = Default value	Type	Description
filesystem_thin_provisioning = False	boolean value	Enable or not thin provisioning in this backend. This configuration option enable the feature of not really write null byte sequences on the filesystem, the holes who can appear will automatically be interpreted by the filesystem as null bytes, and do not really consume your storage. Enabling this feature will also speed up image upload and save network traffic in addition to save space in the backend, as null bytes sequences are not sent over the network. Possible Values: • True • False Related options: • None
http_proxy_information = {}	dict value	The http/https proxy information to be used to connect to the remote server. This configuration option specifies the http/https proxy information that should be used to connect to the remote server. The proxy information should be a key value pair of the scheme and proxy, for example, http:10.0.0.1:3128. You can also specify proxies for multiple schemes by separating the key value pairs with a comma, for example, http:10.0.0.1:3128, https:10.0.0.1:1080. Possible values: • A comma separated list of scheme:proxy pairs as described above Related options: • None

Configuration option = Default value	Type	Description
https_ca_certificates_file = None	string value	Path to the CA bundle file. This configuration option enables the operator to use a custom Certificate Authority file to verify the remote server certificate. If this option is set, the https_insecure option will be ignored and the CA file specified will be used to authenticate the server certificate and establish a secure connection to the server. Possible values: • A valid path to a CA file Related options: • https_insecure
https_insecure = True	boolean value	Set verification of the remote server certificate. This configuration option takes in a boolean value to determine whether or not to verify the remote server certificate. If set to True, the remote server certificate is not verified. If the option is set to False, then the default CA truststore is used for verification. This option is ignored if https_ca_certificates_file is set. The remote server certificate will then be verified using the file specified using the https_ca_certificates_file option. Possible values: • True • False Related options: • https_ca_certificates_file

Configuration option = Default value	Type	Description
rados_connect_timeout = 0	integer value	Timeout value for connecting to Ceph cluster. This configuration option takes in the timeout value in seconds used when connecting to the Ceph cluster i.e. it sets the time to wait for glance-api before closing the connection. This prevents glance-api hangups during the connection to RBD. If the value for this option is set to less than or equal to 0, no timeout is set and the default librados value is used. Possible Values: - Any integer value Related options: - None
`rbd_store_ceph_conf = `	string value	Ceph configuration file path. This configuration option specifies the path to the Ceph configuration file to be used. If the value for this option is not set by the user or is set to the empty string, librados will read the standard ceph.conf file by searching the default Ceph configuration file locations in sequential order. See the Ceph documentation for details.  NOTE If using Cephx authentication, this file should include a reference to the right keyring in a client.<USER> section NOTE 2: If you leave this option empty (the default), the actual Ceph configuration file used may change depending on what version of librados is being used. If it is important for you to know exactly which configuration file is in effect, you may specify that file here using this option. Possible Values: - A valid path to a configuration file Related options: - rbd_store_user

Configuration option = Default value	Type	Description
rbd_store_chunk_size = 8	integer value	Size, in megabytes, to chunk RADOS images into. Provide an integer value representing the size in megabytes to chunk Glance images into. The default chunk size is 8 megabytes. For optimal performance, the value should be a power of two. When Ceph's RBD object storage system is used as the storage backend for storing Glance images, the images are chunked into objects of the size set using this option. These chunked objects are then stored across the distributed block data store to use for Glance. Possible Values: - Any positive integer value Related options: - None
rbd_store_pool = images	string value	RADOS pool in which images are stored. When RBD is used as the storage backend for storing Glance images, the images are stored by means of logical grouping of the objects (chunks of images) into a pool. Each pool is defined with the number of placement groups it can contain. The default pool that is used is <i>images</i>. More information on the RBD storage backend can be found here: http://ceph.com/planet/how-data-is-stored-in-ceph-cluster/ Possible Values: - A valid pool name Related options: - None

Configuration option = Default value	Type	Description
rbd_store_user = None	string value	RADOS user to authenticate as. This configuration option takes in the RADOS user to authenticate as. This is only needed when RADOS authentication is enabled and is applicable only if the user is using Cephx authentication. If the value for this option is not set by the user or is set to None, a default value will be chosen, which will be based on the client. section in rbd_store_ceph_conf. Possible Values: • A valid RADOS user Related options: • rbd_store_ceph_conf
rbd_thin_provisioning = False	boolean value	Enable or not thin provisioning in this backend. This configuration option enable the feature of not really write null byte sequences on the RBD backend, the holes who can appear will automatically be interpreted by Ceph as null bytes, and do not really consume your storage. Enabling this feature will also speed up image upload and save network traffic in addition to save space in the backend, as null bytes sequences are not sent over the network. Possible Values: • True • False Related options: • None

Configuration option = Default value	Type	Description
rootwrap_config = /etc/glance/rootwrap.conf	string value	Path to the rootwrap configuration file to use for running commands as root. The cinder store requires root privileges to operate the image volumes (for connecting to iSCSI/FC volumes and reading/writing the volume data, etc.). The configuration file should allow the required commands by cinder store and os-brick library. Possible values: ● Path to the rootwrap config file Related options: ● None
s3_store_access_key = None	string value	The S3 query token access key. This configuration option takes the access key for authenticating with the Amazon S3 or S3 compatible storage server. This option is required when using the S3 storage backend. Possible values: ● Any string value that is the access key for a user with appropriate privileges Related Options: ● s3_store_host ● s3_store_secret_key
s3_store_bucket = None	string value	The S3 bucket to be used to store the Glance data. This configuration option specifies where the glance images will be stored in the S3. If s3_store_create_bucket_on_put is set to true, it will be created automatically even if the bucket does not exist. Possible values: ● Any string value Related Options: ● s3_store_create_bucket_on_put ● s3_store_bucket_url_format

Configuration option = Default value	Type	Description
s3_store_bucket_url_for mat = auto	string value	The S3 calling format used to determine the object. This configuration option takes access model that is used to specify the address of an object in an S3 bucket. NOTE: In path-style, the endpoint for the object looks like https://s3.amazonaws.com/bucket/example.img. And in virtual-style, the endpoint for the object looks like https://bucket.s3.amazonaws.com/example.img. If you do not follow the DNS naming convention in the bucket name, you can get objects in the path style, but not in the virtual style. Possible values: - Any string value of auto, virtual, or path Related Options: - s3_store_bucket
s3_store_create_bucket_ on_put = False	boolean value	Determine whether S3 should create a new bucket. This configuration option takes boolean value to indicate whether Glance should create a new bucket to S3 if it does not exist. Possible values: - Any Boolean value Related Options: - None

Configuration option = Default value	Type	Description
s3_store_host = None	string value	The host where the S3 server is listening. This configuration option sets the host of the S3 or S3 compatible storage Server. This option is required when using the S3 storage backend. The host can contain a DNS name (e.g. s3.amazonaws.com, my-object-storage.com) or an IP address (127.0.0.1). Possible values: ● A valid DNS name ● A valid IPv4 address Related Options: ● s3_store_access_key ● s3_store_secret_key
s3_store_large_object_chunk_size = 10	integer value	What multipart upload part size, in MB, should S3 use when uploading parts. This configuration option takes the image split size in MB for Multipart Upload. Note: You can only split up to 10,000 images. Possible values: ● Any positive integer value (must be greater than or equal to 5M) Related Options: ● s3_store_large_object_size ● s3_store_thread_pools

Configuration option = Default value	Type	Description
s3_store_large_object_size = 100	integer value	What size, in MB, should S3 start chunking image files and do a multipart upload in S3. This configuration option takes a threshold in MB to determine whether to upload the image to S3 as is or to split it (Multipart Upload). Note: You can only split up to 10,000 images. Possible values: - Any positive integer value Related Options: - s3_store_large_object_chunk_size - s3_store_thread_pools
s3_store_secret_key = None	string value	The S3 query token secret key. This configuration option takes the secret key for authenticating with the Amazon S3 or S3 compatible storage server. This option is required when using the S3 storage backend. Possible values: - Any string value that is a secret key corresponding to the access key specified using the s3_store_host option Related Options: - s3_store_host - s3_store_access_key
s3_store_thread_pools = 10	integer value	The number of thread pools to perform a multipart upload in S3. This configuration option takes the number of thread pools when performing a Multipart Upload. Possible values: - Any positive integer value Related Options: - s3_store_large_object_size - s3_store_large_object_chunk_size

Configuration option = Default value	Type	Description
stores = ['file', 'http']	list value	List of enabled Glance stores. Register the storage backends to use for storing disk images as a comma separated list. The default stores enabled for storing disk images with Glance are file and http. Possible values: ● A comma separated list that could include: ○ file ○ http ○ swift ○ rbd ○ cinder ○ vmware ○ s3 Related Options: ● default_store Deprecated since: Rocky Reason: This option is deprecated against new config option ``enabled_backends`` which helps to configure multiple backend stores of different schemes. This option is scheduled for removal in the U development cycle.

Configuration option = Default value	Type	Description
swift_buffer_on_upload = False	boolean value	Buffer image segments before upload to Swift. Provide a boolean value to indicate whether or not Glance should buffer image data to disk while uploading to swift. This enables Glance to resume uploads on error. NOTES: When enabling this option, one should take great care as this increases disk usage on the API node. Be aware that depending upon how the file system is configured, the disk space used for buffering may decrease the actual disk space available for the glance image cache. Disk utilization will cap according to the following equation: (swift_store_large_object_chunk_size * workers * 1000) Possible values: • True • False Related options: • swift_upload_buffer_dir
swift_store_admin_tenants = []	list value	List of tenants that will be granted admin access. This is a list of tenants that will be granted read/write access on all Swift containers created by Glance in multi-tenant mode. The default value is an empty list. Possible values: • A comma separated list of strings representing UUIDs of Keystone projects/tenants Related options: • None
swift_store_auth_address = None	string value	The address where the Swift authentication service is listening.

Configuration option = Default value	Type	Description
swift_store_auth_insecure = False	boolean value	Set verification of the server certificate. This boolean determines whether or not to verify the server certificate. If this option is set to True, swiftclient won't check for a valid SSL certificate when authenticating. If the option is set to False, then the default CA truststore is used for verification. Possible values: • True • False Related options: • swift_store_cacert
swift_store_auth_version = 2	string value	Version of the authentication service to use. Valid versions are 2 and 3 for keystone and 1 (deprecated) for swauth and rackspace.
swift_store_cacert = None	string value	Path to the CA bundle file. This configuration option enables the operator to specify the path to a custom Certificate Authority file for SSL verification when connecting to Swift. Possible values: • A valid path to a CA file Related options: • swift_store_auth_insecure

Configuration option = Default value	Type	Description
swift_store_config_file = None	string value	Absolute path to the file containing the swift account(s) configurations. Include a string value representing the path to a configuration file that has references for each of the configured Swift account(s)/backing stores. By default, no file path is specified and customized Swift referencing is disabled. Configuring this option is highly recommended while using Swift storage backend for image storage as it avoids storage of credentials in the database. NOTE Please do not configure this option if you have set swift_store_multi_tenant to True. Possible values: ● String value representing an absolute path on the glance-api node Related options: ● swift_store_multi_tenant

Configuration option = Default value	Type	Description
swift_store_container = glance	string value	Name of single container to store images/name prefix for multiple containers When a single container is being used to store images, this configuration option indicates the container within the Glance account to be used for storing all images. When multiple containers are used to store images, this will be the name prefix for all containers. Usage of single/multiple containers can be controlled using the configuration option swift_store_multiple_containers_seed. When using multiple containers, the containers will be named after the value set for this configuration option with the first N chars of the image UUID as the suffix delimited by an underscore (where N is specified by swift_store_multiple_containers_seed). Example: if the seed is set to 3 and swift_store_container = glance, then an image with UUID fdae39a1-bac5-4238-aba4-69bcc726e848 would be placed in the container glance_fda. All dashes in the UUID are included when creating the container name but do not count toward the character limit, so when N=10 the container name would be glance_fdae39a1-ba. Possible values: ● If using single container, this configuration option can be any string that is a valid swift container name in Glance's Swift account ● If using multiple containers, this configuration option can be any string as long as it satisfies the container naming rules enforced by Swift. The value of swift_store_multiple_containers_seed should be taken into account as well. Related options: ● swift_store_multiple_containers_seed ● swift_store_multi_tenant ● swift_store_create_container_on_put

Configuration option = Default value	Type	Description
swift_store_create_container_on_put = False	boolean value	Create container, if it doesn't already exist, when uploading image. At the time of uploading an image, if the corresponding container doesn't exist, it will be created provided this configuration option is set to True. By default, it won't be created. This behavior is applicable for both single and multiple containers mode. Possible values: • True • False Related options: • None
swift_store_endpoint = None	string value	The URL endpoint to use for Swift backend storage. Provide a string value representing the URL endpoint to use for storing Glance images in Swift store. By default, an endpoint is not set and the storage URL returned by auth is used. Setting an endpoint with swift_store_endpoint overrides the storage URL and is used for Glance image storage.  NOTE The URL should include the path up to, but excluding the container. The location of an object is obtained by appending the container and object to the configured URL. Possible values: • String value representing a valid URL path up to a Swift container Related Options: • None

Configuration option = Default value	Type	Description
swift_store_endpoint_type = publicURL	string value	Endpoint Type of Swift service. This string value indicates the endpoint type to use to fetch the Swift endpoint. The endpoint type determines the actions the user will be allowed to perform, for instance, reading and writing to the Store. This setting is only used if swift_store_auth_version is greater than 1. Possible values: ● publicURL ● adminURL ● internalURL Related options: ● swift_store_endpoint
swift_store_expire_soon_interval = 60	integer value	Time in seconds defining the size of the window in which a new token may be requested before the current token is due to expire. Typically, the Swift storage driver fetches a new token upon the expiration of the current token to ensure continued access to Swift. However, some Swift transactions (like uploading image segments) may not recover well if the token expires on the fly. Hence, by fetching a new token before the current token expiration, we make sure that the token does not expire or is close to expiry before a transaction is attempted. By default, the Swift storage driver requests for a new token 60 seconds or less before the current token expiration. Possible values: ● Zero ● Positive integer value Related Options: ● None
swift_store_key = None	string value	Auth key for the user authenticating against the Swift authentication service.

Configuration option = Default value	Type	Description
swift_store_large_object_chunk_size = 200	integer value	The maximum size, in MB, of the segments when image data is segmented. When image data is segmented to upload images that are larger than the limit enforced by the Swift cluster, image data is broken into segments that are no bigger than the size specified by this configuration option. Refer to swift_store_large_object_size for more detail. For example: if swift_store_large_object_size is 5GB and swift_store_large_object_chunk_size is 1GB, an image of size 6.2GB will be segmented into 7 segments where the first six segments will be 1GB in size and the seventh segment will be 0.2GB. Possible values: • A positive integer that is less than or equal to the large object limit enforced by Swift cluster in consideration. Related options: • swift_store_large_object_size

Configuration option = Default value	Type	Description
swift_store_large_object_size = 5120	integer value	The size threshold, in MB, after which Glance will start segmenting image data. Swift has an upper limit on the size of a single uploaded object. By default, this is 5GB. To upload objects bigger than this limit, objects are segmented into multiple smaller objects that are tied together with a manifest file. For more detail, refer to https://docs.openstack.org/swift/latest/overview_large_objects.html This configuration option specifies the size threshold over which the Swift driver will start segmenting image data into multiple smaller files. Currently, the Swift driver only supports creating Dynamic Large Objects.  NOTE This should be set by taking into account the large object limit enforced by the Swift cluster in consideration. Possible values: • A positive integer that is less than or equal to the large object limit enforced by the Swift cluster in consideration. Related options: • swift_store_large_object_chunk_size

Configuration option = Default value	Type	Description
swift_store_multi_tenant = False	boolean value	Store images in tenant's Swift account. This enables multi-tenant storage mode which causes Glance images to be stored in tenant specific Swift accounts. If this is disabled, Glance stores all images in its own account. More details multi-tenant store can be found at https://wiki.openstack.org/wiki/GlanceSwiftTenantSpecificStorage NOTE If using multi-tenant swift store, please make sure that you do not set a swift configuration file with the <i>swift_store_config_file</i> option. Possible values: • True • False Related options: • <code>swift_store_config_file</code>

Configuration option = Default value	Type	Description
swift_store_multiple_containers_seed = 0	integer value	Seed indicating the number of containers to use for storing images. When using a single-tenant store, images can be stored in one or more than one containers. When set to 0, all images will be stored in one single container. When set to an integer value between 1 and 32, multiple containers will be used to store images. This configuration option will determine how many containers are created. 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. Please refer to swift_store_container for more detail on the naming convention. More detail about using multiple containers can be found at https://specs.openstack.org/openstack/glance-specs/specs/kilo/swift-store-multiple-containers.html  NOTE This is used only when <code>swift_store_multi_tenant</code> is disabled. Possible values: • A non-negative integer less than or equal to 32 Related options: • swift_store_container • swift_store_multi_tenant • swift_store_create_container_on_put

Configuration option = Default value	Type	Description
swift_store_region = None	string value	The region of Swift endpoint to use by Glance. Provide a string value representing a Swift region where Glance can connect to for image storage. By default, there is no region set. When Glance uses Swift as the storage backend to store images for a specific tenant that has multiple endpoints, setting of a Swift region with swift_store_region allows Glance to connect to Swift in the specified region as opposed to a single region connectivity. This option can be configured for both single-tenant and multi-tenant storage.  NOTE Setting the region with swift_store_region is tenant-specific and is necessary only if the tenant has multiple endpoints across different regions. Possible values: • A string value representing a valid Swift region. Related Options: • None

Configuration option = Default value	Type	Description
swift_store_retry_get_count = 0	integer value	The number of times a Swift download will be retried before the request fails. Provide an integer value representing the number of times an image download must be retried before erroring out. The default value is zero (no retry on a failed image download). When set to a positive integer value, swift_store_retry_get_count ensures that the download is attempted this many more times upon a download failure before sending an error message. Possible values: • Zero • Positive integer value Related Options: • None
swift_store_service_type = object-store	string value	Type of Swift service to use. Provide a string value representing the service type to use for storing images while using Swift backend storage. The default service type is set to object-store.  NOTE If swift_store_auth_version is set to 2, the value for this configuration option needs to be object-store. If using a higher version of Keystone or a different auth scheme, this option may be modified. Possible values: • A string representing a valid service type for Swift storage. Related Options: • None

Configuration option = Default value	Type	Description
swift_store_ssl_compression = True	boolean value	SSL layer compression for HTTPS Swift requests. Provide a boolean value to determine whether or not to compress HTTPS Swift requests for images at the SSL layer. By default, compression is enabled. When using Swift as the backend store for Glance image storage, SSL layer compression of HTTPS Swift requests can be set using this option. If set to False, SSL layer compression of HTTPS Swift requests is disabled. Disabling this option may improve performance for images which are already in a compressed format, for example, qcow2. Possible values: • True • False Related Options: • None

Configuration option = Default value	Type	Description
swift_store_use_trusts = True	boolean value	Use trusts for multi-tenant Swift store. This option instructs the Swift store to create a trust for each add/get request when the multi-tenant store is in use. Using trusts allows the Swift store to avoid problems that can be caused by an authentication token expiring during the upload or download of data. By default, swift_store_use_trusts is set to True(use of trusts is enabled). If set to False, a user token is used for the Swift connection instead, eliminating the overhead of trust creation.  NOTE This option is considered only when swift_store_multi_tenant is set to True Possible values: • True • False Related options: • swift_store_multi_tenant
swift_store_user = None	string value	The user to authenticate against the Swift authentication service.

Configuration option = Default value	Type	Description
swift_upload_buffer_dir = None	string value	Directory to buffer image segments before upload to Swift. Provide a string value representing the absolute path to the directory on the glance node where image segments will be buffered briefly before they are uploaded to swift. NOTES: • This is required only when the configuration option swift_buffer_on_upload is set to True. • This directory should be provisioned keeping in mind the swift_store_large_object_chunk_size and the maximum number of images that could be uploaded simultaneously by a given glance node. Possible values: • String value representing an absolute directory path Related options: • swift_buffer_on_upload • swift_store_large_object_chunk_size
vmware_api_retry_count = 10	integer value	The number of VMware API retries. This configuration option specifies the number of times the VMware ESX/VC server API must be retried upon connection related issues or server API call overload. It is not possible to specify <i>retry forever</i>. Possible Values: • Any positive integer value Related options: • None

Configuration option = Default value	Type	Description
vmware_ca_file = None	string value	Absolute path to the CA bundle file. This configuration option enables the operator to use a custom Certificate Authority File to verify the ESX/vCenter certificate. If this option is set, the "vmware_insecure" option will be ignored and the CA file specified will be used to authenticate the ESX/vCenter server certificate and establish a secure connection to the server. Possible Values: - Any string that is a valid absolute path to a CA file Related options: - vmware_insecure
vmware_datastores = None	multi valued	The datastores where the image can be stored. This configuration option specifies the datastores where the image can be stored in the VMWare store backend. This option may be specified multiple times for specifying multiple datastores. 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 ":" to specify the priority. 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. Possible Values: Any string of the format: <datacenter_path>:<datastore_name>:<optional_weight> Related options: * None

Configuration option = Default value	Type	Description
vmware_insecure = False	boolean value	Set verification of the ESX/vCenter server certificate. This configuration option takes a boolean value to determine whether or not to verify the ESX/vCenter server certificate. If this option is set to True, the ESX/vCenter server certificate is not verified. If this option is set to False, then the default CA truststore is used for verification. This option is ignored if the "vmware_ca_file" option is set. In that case, the ESX/vCenter server certificate will then be verified using the file specified using the "vmware_ca_file" option . Possible Values: • True • False Related options: • vmware_ca_file
vmware_server_host = None	host address value	Address of the ESX/ESXi or vCenter Server target system. This configuration option sets the address of the ESX/ESXi or vCenter Server target system. This option is required when using the VMware storage backend. The address can contain an IP address (127.0.0.1) or a DNS name (www.my-domain.com). Possible Values: • A valid IPv4 or IPv6 address • A valid DNS name Related options: • vmware_server_username • vmware_server_password

Configuration option = Default value	Type	Description
vmware_server_password = None	string value	Server password. This configuration option takes the password for authenticating with the VMware ESX/ESXi or vCenter Server. This option is required when using the VMware storage backend. Possible Values: - Any string that is a password corresponding to the username specified using the "vmware_server_username" option Related options: - vmware_server_host - vmware_server_username
vmware_server_username = None	string value	Server username. This configuration option takes the username for authenticating with the VMware ESX/ESXi or vCenter Server. This option is required when using the VMware storage backend. Possible Values: - Any string that is the username for a user with appropriate privileges Related options: - vmware_server_host - vmware_server_password

Configuration option = Default value	Type	Description
vmware_store_image_dir = /openstack_glance	string value	The directory where the glance images will be stored in the datastore. This configuration option specifies the path to the directory where the glance images will be stored in the VMware datastore. If this option is not set, the default directory where the glance images are stored is openstack_glance. Possible Values: - Any string that is a valid path to a directory Related options: - None
vmware_task_poll_interval = 5	integer value	Interval in seconds used for polling remote tasks invoked on VMware ESX/VC server. This configuration option takes in the sleep time in seconds for polling an on-going async task as part of the VMWare ESX/VC server API call. Possible Values: - Any positive integer value Related options: - None

3.2.4. oslo_concurrency

The following table outlines the options available under the **[oslo_concurrency]** group in the **/etc/glance/glance-scrubber.conf** file.

Table 3.28. oslo_concurrency

Configuration option = Default value	Type	Description
disable_process_locking = False	boolean value	Enables or disables inter-process locks.

Configuration option = Default value	Type	Description
lock_path = None	string value	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.

3.2.5. oslo_policy

The following table outlines the options available under the **[oslo_policy]** group in the `/etc/glance/glance-scrubber.conf` file.

Table 3.29. oslo_policy

Configuration option = Default value	Type	Description
enforce_new_defaults = False	boolean value	This option controls whether or not to use old deprecated defaults when evaluating policies. If True , the old deprecated defaults are not going to be evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be disallowed. It is encouraged to enable this flag along with the enforce_scope flag so that you can get the benefits of new defaults and scope_type together
enforce_scope = False	boolean value	This option controls whether or not to enforce scope when evaluating policies. If True , the scope of the token used in the request is compared to the scope_types of the policy being enforced. If the scopes do not match, an InvalidScope exception will be raised. If False , a message will be logged informing operators that policies are being invoked with mismatching scope.
policy_default_rule = default	string value	Default rule. Enforced when a requested rule is not found.
policy_dirs = ['policy.d']	multi valued	Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the <code>config_dir</code> option, or absolute paths. The file defined by <code>policy_file</code> must exist for these directories to be searched. Missing or empty directories are ignored.

Configuration option = Default value	Type	Description
policy_file = policy.yaml	string value	The relative or absolute path of a file that maps roles to permissions for a given service. Relative paths must be specified in relation to the configuration file setting this option.
remote_content_type = application/x-www-form-urlencoded	string value	Content Type to send and receive data for REST based policy check
remote_ssl_ca_cert_file = None	string value	Absolute path to ca cert file for REST based policy check
remote_ssl_client_cert_file = None	string value	Absolute path to client cert for REST based policy check
remote_ssl_client_key_file = None	string value	Absolute path client key file REST based policy check
remote_ssl_verify_server_cert = False	boolean value	server identity verification for REST based policy check

3.3. GLANCE-CACHE.CONF

This section contains options for the **/etc/glance/glance-cache.conf** file.

3.3.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the **/etc/glance/glance-cache.conf** file.

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
allow_additional_image_properties = True	boolean value	Allow users to add additional/custom properties to images. Glance defines a standard set of properties (in its schema) that appear on every image. These properties are also known as base properties. In addition to these properties, Glance allows users to add custom properties to images. These are known as additional properties. By default, this configuration option is set to True and users are allowed to add additional properties. The number of additional properties that can be added to an image can be controlled via image_property_quota configuration option. Possible values: • True • False Related options: • image_property_quota Deprecated since: Ussuri Reason: This option is redundant. Control custom image property usage via the <i>image_property_quota</i> configuration option. This option is scheduled to be removed during the Victoria development cycle.

Configuration option = Default value	Type	Description
api_limit_max = 1000	integer value	Maximum number of results that could be returned by a request. As described in the help text of limit_param_default, some requests may return multiple results. The number of results to be returned are governed either by the limit parameter in the request or the limit_param_default configuration option. The value in either case, can't be greater than the absolute maximum defined by this configuration option. Anything greater than this value is trimmed down to the maximum value defined here.  NOTE Setting this to a very large value may slow down database queries and increase response times. Setting this to a very low value may result in poor user experience. Possible values: - Any positive integer Related options: - limit_param_default
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.

Configuration option = Default value	Type	Description
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
digest_algorithm = sha256	string value	Digest algorithm to use for digital signature. Provide a string value representing the digest algorithm to use for generating digital signatures. By default, sha256 is used. To get a list of the available algorithms supported by the version of OpenSSL on your platform, run the command: openssl list-message-digest-algorithms. Examples are <i>sha1</i>, <i>sha256</i>, and <i>sha512</i>.  NOTE digest_algorithm is not related to Glance's image signing and verification. It is only used to sign the universally unique identifier (UUID) as a part of the certificate file and key file validation. Possible values: • An OpenSSL message digest algorithm identifier Relation options: • None
enabled_import_methods = ['glance-direct', 'web-download', 'copy-image']	list value	List of enabled Image Import Methods 'glance-direct', 'copy-image' and 'web-download' are enabled by default. Related options: ** [DEFAULT]/node_staging_uri

Configuration option = Default value	Type	Description
enforce_secure_rbac = False	boolean value	Enforce API access based on common persona definitions used across OpenStack. Enabling this option formalizes project-specific read/write operations, like creating private images or updating the status of shared image, behind the member role. It also formalizes a read-only variant useful for project-specific API operations, like listing private images in a project, behind the reader role. Operators should take an opportunity to understand glance's new image policies, audit assignments in their deployment, and update permissions using the default roles in keystone (e.g., admin, member, and reader). Related options: • [oslo_policy]/enforce_new_defaults Deprecated since: Wallaby Reason: This option has been introduced to require operators to opt into enforcing authorization based on common RBAC personas, which is EXPERIMENTAL as of the Wallaby release. This behavior will be the default and STABLE in a future release, allowing this option to be removed.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.

Configuration option = Default value	Type	Description
hashing_algorithm = sha512	string value	Secure hashing algorithm used for computing the <i>os_hash_value</i> property. This option configures the Glance "multihash", which consists of two image properties: the <i>os_hash_algo</i> and the <i>os_hash_value</i>. The <i>os_hash_algo</i> will be populated by the value of this configuration option, and the <i>os_hash_value</i> will be populated by the hexdigest computed when the algorithm is applied to the uploaded or imported image data. The value must be a valid secure hash algorithm name recognized by the python <i>hashlib</i> library. You can determine what these are by examining the <i>hashlib.algorithms_available</i> data member of the version of the library being used in your Glance installation. For interoperability purposes, however, we recommend that you use the set of secure hash names supplied by the <i>hashlib.algorithms_guaranteed</i> data member because those algorithms are guaranteed to be supported by the <i>hashlib</i> library on all platforms. Thus, any image consumer using <i>hashlib</i> locally should be able to verify the <i>os_hash_value</i> of the image. The default value of <i>sha512</i> is a performant secure hash algorithm. If this option is misconfigured, any attempts to store image data will fail. For that reason, we recommend using the default value. Possible values: - Any secure hash algorithm name recognized by the Python <i>hashlib</i> library Related options: - None

Configuration option = Default value	Type	Description
image_cache_dir = None	string value	Base directory for image cache. This is the location where image data is cached and served out of. All cached images are stored directly under this directory. This directory also contains three subdirectories, namely, incomplete, invalid and queue. The incomplete subdirectory is the staging area for downloading images. An image is first downloaded to this directory. When the image download is successful it is moved to the base directory. However, if the download fails, the partially downloaded image file is moved to the invalid subdirectory. The queue subdirectory is used for queuing images for download. This is used primarily by the cache-prefetcher, which can be scheduled as a periodic task like cache-pruner and cache-cleaner, to cache images ahead of their usage. Upon receiving the request to cache an image, Glance touches a file in the queue directory with the image id as the file name. The cache-prefetcher, when running, polls for the files in queue directory and starts downloading them in the order they were created. When the download is successful, the zero-sized file is deleted from the queue directory. If the download fails, the zero-sized file remains and it'll be retried the next time cache-prefetcher runs. Possible values: • A valid path Related options: • image_cache_sqlite_db

Configuration option = Default value	Type	Description
image_cache_driver = sqlite	string value	The driver to use for image cache management. This configuration option provides the flexibility to choose between the different image-cache drivers available. An image-cache driver is responsible for providing the essential functions of image-cache like write images to/read images from cache, track age and usage of cached images, provide a list of cached images, fetch size of the cache, queue images for caching and clean up the cache, etc. The essential functions of a driver are defined in the base class glance.image_cache.drivers.base.Driver. All image-cache drivers (existing and prospective) must implement this interface. Currently available drivers are sqlite and xattr. These drivers primarily differ in the way they store the information about cached images: • The sqlite driver uses a sqlite database (which sits on every glance node locally) to track the usage of cached images. • The xattr driver uses the extended attributes of files to store this information. It also requires a filesystem that sets atime on the files when accessed. Possible values: • sqlite • xattr Related options: • None

Configuration option = Default value	Type	Description
image_cache_max_size = 10737418240	integer value	The upper limit on cache size, in bytes, after which the cache-pruner cleans up the image cache.  NOTE This is just a threshold for cache-pruner to act upon. It is NOT a hard limit beyond which the image cache would never grow. In fact, depending on how often the cache-pruner runs and how quickly the cache fills, the image cache can far exceed the size specified here very easily. Hence, care must be taken to appropriately schedule the cache-pruner and in setting this limit. Glance caches an image when it is downloaded. Consequently, the size of the image cache grows over time as the number of downloads increases. To keep the cache size from becoming unmanageable, it is recommended to run the cache-pruner as a periodic task. When the cache pruner is kicked off, it compares the current size of image cache and triggers a cleanup if the image cache grew beyond the size specified here. After the cleanup, the size of cache is less than or equal to size specified here. Possible values: - Any non-negative integer Related options: - None

Configuration option = Default value	Type	Description
image_cache_sqlite_db = cache.db	string value	The relative path to sqlite file database that will be used for image cache management. This is a relative path to the sqlite file database that tracks the age and usage statistics of image cache. The path is relative to image cache base directory, specified by the configuration option image_cache_dir. This is a lightweight database with just one table. Possible values: • A valid relative path to sqlite file database Related options: • image_cache_dir
image_cache_stall_time = 86400	integer value	The amount of time, in seconds, an incomplete image remains in the cache. Incomplete images are images for which download is in progress. Please see the description of configuration option image_cache_dir for more detail. Sometimes, due to various reasons, it is possible the download may hang and the incompletely downloaded image remains in the incomplete directory. This configuration option sets a time limit on how long the incomplete images should remain in the incomplete directory before they are cleaned up. Once an incomplete image spends more time than is specified here, it'll be removed by cache-cleaner on its next run. It is recommended to run cache-cleaner as a periodic task on the Glance API nodes to keep the incomplete images from occupying disk space. Possible values: • Any non-negative integer Related options: • None

Configuration option = Default value	Type	Description
image_location_quota = 10	integer value	Maximum number of locations allowed on an image. Any negative value is interpreted as unlimited. Related options: • None
image_member_quota = 128	integer value	Maximum number of image members per image. This limits the maximum of users an image can be shared with. Any negative value is interpreted as unlimited. Related options: • None
image_property_quota = 128	integer value	Maximum number of properties allowed on an image. This enforces an upper limit on the number of additional properties an image can have. Any negative value is interpreted as unlimited.  NOTE This won't have any impact if additional properties are disabled. Please refer to allow_additional_image_properties. Related options: • allow_additional_image_properties

Configuration option = Default value	Type	Description
image_size_cap = 1099511627776	integer value	Maximum size of image a user can upload in bytes. An image upload greater than the size mentioned here would result in an image creation failure. This configuration option defaults to 1099511627776 bytes (1 TiB). NOTES: • This value should only be increased after careful consideration and must be set less than or equal to 8 EiB (9223372036854775808). • This value must be set with careful consideration of the backend storage capacity. Setting this to a very low value may result in a large number of image failures. And, setting this to a very large value may result in faster consumption of storage. Hence, this must be set according to the nature of images created and storage capacity available. Possible values: • Any positive number less than or equal to 9223372036854775808
image_tag_quota = 128	integer value	Maximum number of tags allowed on an image. Any negative value is interpreted as unlimited. Related options: • None
<code>`instance_format = [instance: %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.
<code>`instance_uuid_format = [instance: %(uuid)s] `</code>	string value	The format for an instance UUID that is passed with the log message.

Configuration option = Default value	Type	Description
limit_param_default = 25	integer value	The default number of results to return for a request. Responses to certain API requests, like list images, may return multiple items. The number of results returned can be explicitly controlled by specifying the limit parameter in the API request. However, if a limit parameter is not specified, this configuration value will be used as the default number of results to be returned for any API request. NOTES: • The value of this configuration option may not be greater than the value specified by api_limit_max. • Setting this to a very large value may slow down database queries and increase response times. Setting this to a very low value may result in poor user experience. Possible values: • Any positive integer Related options: • <code>api_limit_max</code>
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, <code>log-date-format</code>).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code>. This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code>. This option is ignored if <code>log_config_append</code> is set.

Configuration option = Default value	Type	Description
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless log_rotation_type is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	string value	Format string to use for log messages with context. Used by oslo_log.formatters.ContextFormatter
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is DEBUG. Used by oslo_log.formatters.ContextFormatter
logging_default_format_s tring = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s	string value	Format string to use for log messages when context is undefined. Used by oslo_log.formatters.ContextFormatter
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d ERROR %(name)s % (instance)s	string value	Prefix each line of exception output with this format. Used by oslo_log.formatters.ContextFormatter
logging_user_identity_for mat = %(user)s % (tenant)s %(domain)s % (user_domain)s % (project_domain)s	string value	Defines the format string for %(user_identity)s that is used in logging_context_format_string. Used by oslo_log.formatters.ContextFormatter
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".

Configuration option = Default value	Type	Description
metadata_encryption_key = None	string value	AES key for encrypting store location metadata. Provide a string value representing the AES cipher to use for encrypting Glance store metadata. NOTE The AES key to use must be set to a random string of length 16, 24 or 32 bytes. Possible values: • String value representing a valid AES key Related options: • None

Configuration option = Default value	Type	Description
node_staging_uri = file:///tmp/staging/	string value	The URL provides location where the temporary data will be stored This option is for Glance internal use only. Glance will save the image data uploaded by the user to <i>staging</i> endpoint during the image import process. This option does not change the <i>staging</i> API endpoint by any means.  NOTE It is discouraged to use same path as [task]/work_dir  NOTE <i>file://<absolute-directory-path></i> is the only option api_image_import flow will support for now.  NOTE The staging path must be on shared filesystem available to all Glance API nodes. Possible values: ● String starting with <i>file://</i> followed by absolute FS path Related options: ● [task]/work_dir
publish_errors = False	boolean value	Enables or disables publication of error events.

Configuration option = Default value	Type	Description
pydev_worker_debug_host = None	host address value	Host address of the pydev server. Provide a string value representing the hostname or IP of the pydev server to use for debugging. The pydev server listens for debug connections on this address, facilitating remote debugging in Glance. Possible values: - Valid hostname - Valid IP address Related options: - None
pydev_worker_debug_port = 5678	port value	Port number that the pydev server will listen on. Provide a port number to bind the pydev server to. The pydev process accepts debug connections on this port and facilitates remote debugging in Glance. Possible values: - A valid port number Related options: - None
rate_limit_burst = 0	integer value	Maximum number of logged messages per rate_limit_interval.
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.

Configuration option = Default value	Type	Description
show_image_direct_url = False	boolean value	Show direct image location when returning an image. This configuration option indicates whether to show the direct image location when returning image details to the user. The direct image location is where the image data is stored in backend storage. This image location is shown under the image property direct_url. When multiple image locations exist for an image, the best location is displayed based on the location strategy indicated by the configuration option location_strategy. NOTES: ● Revealing image locations can present a GRAVE SECURITY RISK as image locations can sometimes include credentials. Hence, this is set to False by default. Set this to True with EXTREME CAUTION and ONLY IF you know what you are doing! ● If an operator wishes to avoid showing any image location(s) to the user, then both this option and show_multiple_locations MUST be set to False. Possible values: ● True ● False Related options: ● show_multiple_locations ● location_strategy

Configuration option = Default value	Type	Description
show_multiple_locations = False	boolean value	Show all image locations when returning an image. This configuration option indicates whether to show all the image locations when returning image details to the user. When multiple image locations exist for an image, the locations are ordered based on the location strategy indicated by the configuration opt location_strategy. The image locations are shown under the image property locations. NOTES: ● Revealing image locations can present a GRAVE SECURITY RISK as image locations can sometimes include credentials. Hence, this is set to False by default. Set this to True with EXTREME CAUTION and ONLY IF you know what you are doing! ● See https://wiki.openstack.org/wiki/OSSN/OSSN-0065 for more information. ● If an operator wishes to avoid showing any image location(s) to the user, then both this option and show_image_direct_url MUST be set to False. Possible values: ● True ● False Related options: ● show_image_direct_url ● location_strategy Deprecated since: Newton <i>*Reason:</i>*Use of this option, deprecated since Newton, is a security risk and will be removed once we figure out a way to satisfy those use cases that currently require it. An earlier announcement that the same functionality can be achieved with greater granularity by using policies is incorrect. You cannot work around this option via policy configuration at the present time, though that is the direction we believe the fix will take. Please keep an eye on the Glance release notes to stay up to date on progress in addressing this issue.

Configuration option = Default value	Type	Description
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
user_storage_quota = 0	string value	Maximum amount of image storage per tenant. This enforces an upper limit on the cumulative storage consumed by all images of a tenant across all stores. This is a per-tenant limit. The default unit for this configuration option is Bytes. However, storage units can be specified using case-sensitive literals B, KB, MB, GB and TB representing Bytes, KiloBytes, MegaBytes, GigaBytes and TeraBytes respectively. Note that there should not be any space between the value and unit. Value 0 signifies no quota enforcement. Negative values are invalid and result in errors. Possible values: • A string that is a valid concatenation of a non-negative integer representing the storage value and an optional string literal representing storage units as mentioned above. Related options: • None

Configuration option = Default value	Type	Description
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.
worker_self_reference_url = None	string value	The URL to this worker. If this is set, other glance workers will know how to contact this one directly if needed. For image import, a single worker stages the image and other workers need to be able to proxy the import request to the right one. If unset, this will be considered to be public_endpoint, which normally would be set to the same value on all workers, effectively disabling the proxying behavior. Possible values: ● A URL by which this worker is reachable from other workers Related options: ● public_endpoint

3.3.2. glance_store

The following table outlines the options available under the **[glance_store]** group in the **/etc/glance/glance-cache.conf** file.

Table 3.30. glance_store

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
cinder_api_insecure = False	boolean value	Allow to perform insecure SSL requests to cinder. If this option is set to True, HTTPS endpoint connection is verified using the CA certificates file specified by cinder_ca_certificates_file option. Possible values: • True • False Related options: • cinder_ca_certificates_file
cinder_ca_certificates_file = None	string value	Location of a CA certificates file used for cinder client requests. The specified CA certificates file, if set, is used to verify cinder connections via HTTPS endpoint. If the endpoint is HTTP, this value is ignored. cinder_api_insecure must be set to True to enable the verification. Possible values: • Path to a ca certificates file Related options: • cinder_api_insecure

Configuration option = Default value	Type	Description
cinder_catalog_info = volumev3::publicURL	string value	Information to match when looking for cinder in the service catalog. When the cinder_endpoint_template is not set and any of cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, cinder_store_password is not set, cinder store uses this information to lookup cinder endpoint from the service catalog in the current context. cinder_os_region_name, if set, is taken into consideration to fetch the appropriate endpoint. The service catalog can be listed by the openstack catalog list command. Possible values: • A string of of the following form: <service_type>:<service_name>: <interface> At least service_type and interface should be specified. service_name can be omitted. Related options: • cinder_os_region_name • cinder_endpoint_template • cinder_store_auth_address • cinder_store_user_name • cinder_store_project_name • cinder_store_password

Configuration option = Default value	Type	Description
cinder_endpoint_template = None	string value	Override service catalog lookup with template for cinder endpoint. When this option is set, this value is used to generate cinder endpoint, instead of looking up from the service catalog. This value is ignored if cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, and cinder_store_password are specified. If this configuration option is set, cinder_catalog_info will be ignored. Possible values: ● URL template string for cinder endpoint, where %%(tenant)s is replaced with the current tenant (project) name. For example: http://cinder.openstack.example.org/v2/%%(tenant)s Related options: ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name ● cinder_store_password ● cinder_catalog_info
cinder_enforce_multipath = False	boolean value	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. Possible values: ● True or False Related options: ● cinder_use_multipath

Configuration option = Default value	Type	Description
cinder_http_retries = 3	integer value	Number of cinderclient retries on failed http calls. When a call failed by any errors, cinderclient will retry the call up to the specified times after sleeping a few seconds. Possible values: • A positive integer Related options: • None
cinder_mount_point_base = /var/lib/glance/mnt	string value	Directory where the NFS volume is mounted on the glance node. Possible values: • A string representing absolute path of mount point.
cinder_os_region_name = None	string value	Region name to lookup cinder service from the service catalog. This is used only when cinder_catalog_info is used for determining the endpoint. If set, the lookup for cinder endpoint by this node is filtered to the specified region. It is useful when multiple regions are listed in the catalog. If this is not set, the endpoint is looked up from every region. Possible values: • A string that is a valid region name. Related options: • cinder_catalog_info

Configuration option = Default value	Type	Description
cinder_state_transition_timeout = 300	integer value	Time period, in seconds, to wait for a cinder volume transition to complete. When the cinder volume is created, deleted, or attached to the glance node to read/write the volume data, the volume's state is changed. For example, the newly created volume status changes from creating to available after the creation process is completed. This specifies the maximum time to wait for the status change. If a timeout occurs while waiting, or the status is changed to an unexpected value (e.g. error), the image creation fails. Possible values: • A positive integer Related options: • None

Configuration option = Default value	Type	Description
cinder_store_auth_address = None	string value	The address where the cinder authentication service is listening. When all of cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, and cinder_store_password options are specified, the specified values are always used for the authentication. This is useful to hide the image volumes from users by storing them in a project/tenant specific to the image service. It also enables users to share the image volume among other projects under the control of glance's ACL. If either of these options are not set, the cinder endpoint is looked up from the service catalog, and current context's user and project are used. Possible values: • A valid authentication service address, for example: http://openstack.example.org/identity/v2.0 Related options: • cinder_store_user_name • cinder_store_password • cinder_store_project_name
cinder_store_password = None	string value	Password for the user authenticating against cinder. This must be used with all the following related options. If any of these are not specified, the user of the current context is used. Possible values: • A valid password for the user specified by cinder_store_user_name Related options: • cinder_store_auth_address • cinder_store_user_name • cinder_store_project_name

Configuration option = Default value	Type	Description
cinder_store_project_name = None	string value	Project name where the image volume is stored in cinder. If this configuration option is not set, the project in current context is used. This must be used with all the following related options. If any of these are not specified, the project of the current context is used. Possible values: • A valid project name Related options: • cinder_store_auth_address • cinder_store_user_name • cinder_store_password
cinder_store_user_name = None	string value	User name to authenticate against cinder. This must be used with all the following related options. If any of these are not specified, the user of the current context is used. Possible values: • A valid user name Related options: • cinder_store_auth_address • cinder_store_password • cinder_store_project_name
cinder_use_multipath = False	boolean value	Flag to identify mutipath is supported or not in the deployment. Set it to False if multipath is not supported. Possible values: • True or False Related options: • cinder_enforce_multipath

Configuration option = Default value	Type	Description
cinder_volume_type = None	string value	Volume type that will be used for volume creation in cinder. Some cinder backends can have several volume types to optimize storage usage. Adding this option allows an operator to choose a specific volume type in cinder that can be optimized for images. If this is not set, then the default volume type specified in the cinder configuration will be used for volume creation. Possible values: • A valid volume type from cinder Related options: • None  NOTE You cannot use an encrypted volume_type associated with an NFS backend. An encrypted volume stored on an NFS backend will raise an exception whenever glance_store tries to write or access image data stored in that volume. Consult your Cinder administrator to determine an appropriate volume_type.

Configuration option = Default value	Type	Description
default_store = file	string value	The default scheme to use for storing images. Provide a string value representing the default scheme to use for storing images. If not set, Glance uses file as the default scheme to store images with the file store.  NOTE The value given for this configuration option must be a valid scheme for a store registered with the stores configuration option. Possible values: ● file ● filesystem ● http ● https ● swift ● swift+http ● swift+https ● swift+config ● rbd ● cinder ● vsphere ● s3 Related Options: ● stores Deprecated since: Rocky Reason: This option is deprecated against new config option ``default_backend`` which acts similar to ``default_store`` config option. This option is scheduled for removal in the U development cycle.

Configuration option = Default value	Type	Description
default_swift_reference = ref1	string value	Reference to default Swift account/backing store parameters. Provide a string value representing a reference to the default set of parameters required for using swift account/backing store for image storage. The default reference value for this configuration option is <i>ref1</i>. This configuration option dereferences the parameters and facilitates image storage in Swift storage backend every time a new image is added. Possible values: • A valid string value Related options: • None
filesystem_store_chunk_size = 65536	integer value	Chunk size, in bytes. The chunk size used when reading or writing image files. Raising this value may improve the throughput but it may also slightly increase the memory usage when handling a large number of requests. Possible Values: • Any positive integer value Related options: • None

Configuration option = Default value	Type	Description
filesystem_store_datadir = /var/lib/glance/images	string value	Directory to which the filesystem backend store writes images. Upon start up, Glance creates the directory if it doesn't already exist and verifies write access to the user under which glance-api runs. If the write access isn't available, a BadStoreConfiguration exception is raised and the filesystem store may not be available for adding new images.  NOTE This directory is used only when filesystem store is used as a storage backend. Either filesystem_store_datadir or filesystem_store_datadirs option must be specified in glance-api.conf. If both options are specified, a BadStoreConfiguration will be raised and the filesystem store may not be available for adding new images. Possible values: ● A valid path to a directory Related options: ● filesystem_store_datadirs ● filesystem_store_file_perm

Configuration option = Default value	Type	Description
filesystem_store_datadirs = None	multi valued	List of directories and their priorities to which the filesystem backend store writes images. The filesystem store can be configured to store images in multiple directories as opposed to using a single directory specified by the filesystem_store_datadir configuration option. When using multiple directories, each directory can be given an optional priority to specify the preference order in which they should be used. Priority is an integer that is concatenated to the directory path with a colon where a higher value indicates higher priority. When two directories have the same priority, the directory with most free space is used. When no priority is specified, it defaults to zero. More information on configuring filesystem store with multiple store directories can be found at https://docs.openstack.org/glance/latest/configuration/configuring.html  NOTE This directory is used only when filesystem store is used as a storage backend. Either filesystem_store_datadir or filesystem_store_datadirs option must be specified in glance-api.conf. If both options are specified, a BadStoreConfiguration will be raised and the filesystem store may not be available for adding new images. Possible values: - List of strings of the following form: <a valid directory path>:<optional integer priority> Related options: filesystem_store_datadir filesystem_store_file_perm

Configuration option = Default value	Type	Description
filesystem_store_file_permission = 0	integer value	File access permissions for the image files. Set the intended file access permissions for image data. This provides a way to enable other services, e.g. Nova, to consume images directly from the filesystem store. The users running the services that are intended to be given access to could be made a member of the group that owns the files created. Assigning a value less than or equal to zero for this configuration option signifies that no changes be made to the default permissions. This value will be decoded as an octal digit. For more information, please refer the documentation at https://docs.openstack.org/glance/latest/configuration/configuring.html Possible values: • A valid file access permission • Zero • Any negative integer Related options: • None
filesystem_store_metadata_file = None	string value	Filesystem store metadata file. The path to a file which contains the metadata to be returned with any location associated with the filesystem store. Once this option is set, it is used for new images created afterward only - previously existing images are not affected. The file must contain a valid JSON object. The object should contain the keys id and mountpoint. The value for both keys should be a string. Possible values: • A valid path to the store metadata file Related options: • None

Configuration option = Default value	Type	Description
filesystem_thin_provisioning = False	boolean value	Enable or not thin provisioning in this backend. This configuration option enable the feature of not really write null byte sequences on the filesystem, the holes who can appear will automatically be interpreted by the filesystem as null bytes, and do not really consume your storage. Enabling this feature will also speed up image upload and save network traffic in addition to save space in the backend, as null bytes sequences are not sent over the network. Possible Values: • True • False Related options: • None
http_proxy_information = {}	dict value	The http/https proxy information to be used to connect to the remote server. This configuration option specifies the http/https proxy information that should be used to connect to the remote server. The proxy information should be a key value pair of the scheme and proxy, for example, http:10.0.0.1:3128. You can also specify proxies for multiple schemes by separating the key value pairs with a comma, for example, http:10.0.0.1:3128, https:10.0.0.1:1080. Possible values: • A comma separated list of scheme:proxy pairs as described above Related options: • None

Configuration option = Default value	Type	Description
https_ca_certificates_file = None	string value	Path to the CA bundle file. This configuration option enables the operator to use a custom Certificate Authority file to verify the remote server certificate. If this option is set, the https_insecure option will be ignored and the CA file specified will be used to authenticate the server certificate and establish a secure connection to the server. Possible values: • A valid path to a CA file Related options: • https_insecure
https_insecure = True	boolean value	Set verification of the remote server certificate. This configuration option takes in a boolean value to determine whether or not to verify the remote server certificate. If set to True, the remote server certificate is not verified. If the option is set to False, then the default CA truststore is used for verification. This option is ignored if https_ca_certificates_file is set. The remote server certificate will then be verified using the file specified using the https_ca_certificates_file option. Possible values: • True • False Related options: • https_ca_certificates_file

Configuration option = Default value	Type	Description
rados_connect_timeout = 0	integer value	Timeout value for connecting to Ceph cluster. This configuration option takes in the timeout value in seconds used when connecting to the Ceph cluster i.e. it sets the time to wait for glance-api before closing the connection. This prevents glance-api hangups during the connection to RBD. If the value for this option is set to less than or equal to 0, no timeout is set and the default librados value is used. Possible Values: - Any integer value Related options: - None
`rbd_store_ceph_conf = `	string value	Ceph configuration file path. This configuration option specifies the path to the Ceph configuration file to be used. If the value for this option is not set by the user or is set to the empty string, librados will read the standard ceph.conf file by searching the default Ceph configuration file locations in sequential order. See the Ceph documentation for details.  NOTE If using Cephx authentication, this file should include a reference to the right keyring in a client.<USER> section NOTE 2: If you leave this option empty (the default), the actual Ceph configuration file used may change depending on what version of librados is being used. If it is important for you to know exactly which configuration file is in effect, you may specify that file here using this option. Possible Values: - A valid path to a configuration file Related options: - rbd_store_user

Configuration option = Default value	Type	Description
rbd_store_chunk_size = 8	integer value	Size, in megabytes, to chunk RADOS images into. Provide an integer value representing the size in megabytes to chunk Glance images into. The default chunk size is 8 megabytes. For optimal performance, the value should be a power of two. When Ceph's RBD object storage system is used as the storage backend for storing Glance images, the images are chunked into objects of the size set using this option. These chunked objects are then stored across the distributed block data store to use for Glance. Possible Values: - Any positive integer value Related options: - None
rbd_store_pool = images	string value	RADOS pool in which images are stored. When RBD is used as the storage backend for storing Glance images, the images are stored by means of logical grouping of the objects (chunks of images) into a pool. Each pool is defined with the number of placement groups it can contain. The default pool that is used is <i>images</i>. More information on the RBD storage backend can be found here: http://ceph.com/planet/how-data-is-stored-in-ceph-cluster/ Possible Values: - A valid pool name Related options: - None

Configuration option = Default value	Type	Description
rbd_store_user = None	string value	RADOS user to authenticate as. This configuration option takes in the RADOS user to authenticate as. This is only needed when RADOS authentication is enabled and is applicable only if the user is using Cephx authentication. If the value for this option is not set by the user or is set to None, a default value will be chosen, which will be based on the client. section in rbd_store_ceph_conf. Possible Values: • A valid RADOS user Related options: • rbd_store_ceph_conf
rbd_thin_provisioning = False	boolean value	Enable or not thin provisioning in this backend. This configuration option enable the feature of not really write null byte sequences on the RBD backend, the holes who can appear will automatically be interpreted by Ceph as null bytes, and do not really consume your storage. Enabling this feature will also speed up image upload and save network traffic in addition to save space in the backend, as null bytes sequences are not sent over the network. Possible Values: • True • False Related options: • None

Configuration option = Default value	Type	Description
rootwrap_config = /etc/glance/rootwrap.conf	string value	Path to the rootwrap configuration file to use for running commands as root. The cinder store requires root privileges to operate the image volumes (for connecting to iSCSI/FC volumes and reading/writing the volume data, etc.). The configuration file should allow the required commands by cinder store and os-brick library. Possible values: ● Path to the rootwrap config file Related options: ● None
s3_store_access_key = None	string value	The S3 query token access key. This configuration option takes the access key for authenticating with the Amazon S3 or S3 compatible storage server. This option is required when using the S3 storage backend. Possible values: ● Any string value that is the access key for a user with appropriate privileges Related Options: ● s3_store_host ● s3_store_secret_key
s3_store_bucket = None	string value	The S3 bucket to be used to store the Glance data. This configuration option specifies where the glance images will be stored in the S3. If s3_store_create_bucket_on_put is set to true, it will be created automatically even if the bucket does not exist. Possible values: ● Any string value Related Options: ● s3_store_create_bucket_on_put ● s3_store_bucket_url_format

Configuration option = Default value	Type	Description
s3_store_bucket_url_for mat = auto	string value	The S3 calling format used to determine the object. This configuration option takes access model that is used to specify the address of an object in an S3 bucket. NOTE: In path-style, the endpoint for the object looks like https://s3.amazonaws.com/bucket/example.img. And in virtual-style, the endpoint for the object looks like https://bucket.s3.amazonaws.com/example.img. If you do not follow the DNS naming convention in the bucket name, you can get objects in the path style, but not in the virtual style. Possible values: - Any string value of auto, virtual, or path Related Options: - s3_store_bucket
s3_store_create_bucket_ on_put = False	boolean value	Determine whether S3 should create a new bucket. This configuration option takes boolean value to indicate whether Glance should create a new bucket to S3 if it does not exist. Possible values: - Any Boolean value Related Options: - None

Configuration option = Default value	Type	Description
s3_store_host = None	string value	The host where the S3 server is listening. This configuration option sets the host of the S3 or S3 compatible storage Server. This option is required when using the S3 storage backend. The host can contain a DNS name (e.g. s3.amazonaws.com, my-object-storage.com) or an IP address (127.0.0.1). Possible values: ● A valid DNS name ● A valid IPv4 address Related Options: ● s3_store_access_key ● s3_store_secret_key
s3_store_large_object_chunk_size = 10	integer value	What multipart upload part size, in MB, should S3 use when uploading parts. This configuration option takes the image split size in MB for Multipart Upload. Note: You can only split up to 10,000 images. Possible values: ● Any positive integer value (must be greater than or equal to 5M) Related Options: ● s3_store_large_object_size ● s3_store_thread_pools

Configuration option = Default value	Type	Description
s3_store_large_object_size = 100	integer value	What size, in MB, should S3 start chunking image files and do a multipart upload in S3. This configuration option takes a threshold in MB to determine whether to upload the image to S3 as is or to split it (Multipart Upload). Note: You can only split up to 10,000 images. Possible values: - Any positive integer value Related Options: - s3_store_large_object_chunk_size - s3_store_thread_pools
s3_store_secret_key = None	string value	The S3 query token secret key. This configuration option takes the secret key for authenticating with the Amazon S3 or S3 compatible storage server. This option is required when using the S3 storage backend. Possible values: - Any string value that is a secret key corresponding to the access key specified using the s3_store_host option Related Options: - s3_store_host - s3_store_access_key

Configuration option = Default value	Type	Description
s3_store_thread_pools = 10	integer value	The number of thread pools to perform a multipart upload in S3. This configuration option takes the number of thread pools when performing a Multipart Upload. Possible values: - Any positive integer value Related Options: - s3_store_large_object_size - s3_store_large_object_chunk_size
stores = ['file', 'http']	list value	List of enabled Glance stores. Register the storage backends to use for storing disk images as a comma separated list. The default stores enabled for storing disk images with Glance are file and http. Possible values: - A comma separated list that could include: - file - http - swift - rbd - cinder - vmware - s3 Related Options: - default_store Deprecated since: Rocky Reason: This option is deprecated against new config option ``enabled_backends`` which helps to configure multiple backend stores of different schemes. This option is scheduled for removal in the U development cycle.

Configuration option = Default value	Type	Description
swift_buffer_on_upload = False	boolean value	Buffer image segments before upload to Swift. Provide a boolean value to indicate whether or not Glance should buffer image data to disk while uploading to swift. This enables Glance to resume uploads on error. NOTES: When enabling this option, one should take great care as this increases disk usage on the API node. Be aware that depending upon how the file system is configured, the disk space used for buffering may decrease the actual disk space available for the glance image cache. Disk utilization will cap according to the following equation: (swift_store_large_object_chunk_size * workers * 1000) Possible values: • True • False Related options: • swift_upload_buffer_dir
swift_store_admin_tenants = []	list value	List of tenants that will be granted admin access. This is a list of tenants that will be granted read/write access on all Swift containers created by Glance in multi-tenant mode. The default value is an empty list. Possible values: • A comma separated list of strings representing UUIDs of Keystone projects/tenants Related options: • None
swift_store_auth_address = None	string value	The address where the Swift authentication service is listening.

Configuration option = Default value	Type	Description
swift_store_auth_insecure = False	boolean value	Set verification of the server certificate. This boolean determines whether or not to verify the server certificate. If this option is set to True, swiftclient won't check for a valid SSL certificate when authenticating. If the option is set to False, then the default CA truststore is used for verification. Possible values: • True • False Related options: • swift_store_cacert
swift_store_auth_version = 2	string value	Version of the authentication service to use. Valid versions are 2 and 3 for keystone and 1 (deprecated) for swauth and rackspace.
swift_store_cacert = None	string value	Path to the CA bundle file. This configuration option enables the operator to specify the path to a custom Certificate Authority file for SSL verification when connecting to Swift. Possible values: • A valid path to a CA file Related options: • swift_store_auth_insecure

Configuration option = Default value	Type	Description
swift_store_config_file = None	string value	Absolute path to the file containing the swift account(s) configurations. Include a string value representing the path to a configuration file that has references for each of the configured Swift account(s)/backing stores. By default, no file path is specified and customized Swift referencing is disabled. Configuring this option is highly recommended while using Swift storage backend for image storage as it avoids storage of credentials in the database.  NOTE Please do not configure this option if you have set swift_store_multi_tenant to True. Possible values: ● String value representing an absolute path on the glance-api node Related options: ● swift_store_multi_tenant

Configuration option = Default value	Type	Description
swift_store_container = glance	string value	Name of single container to store images/name prefix for multiple containers When a single container is being used to store images, this configuration option indicates the container within the Glance account to be used for storing all images. When multiple containers are used to store images, this will be the name prefix for all containers. Usage of single/multiple containers can be controlled using the configuration option swift_store_multiple_containers_seed. When using multiple containers, the containers will be named after the value set for this configuration option with the first N chars of the image UUID as the suffix delimited by an underscore (where N is specified by swift_store_multiple_containers_seed). Example: if the seed is set to 3 and swift_store_container = glance, then an image with UUID fdae39a1-bac5-4238-aba4-69bcc726e848 would be placed in the container glance_fda. All dashes in the UUID are included when creating the container name but do not count toward the character limit, so when N=10 the container name would be glance_fdae39a1-ba. Possible values: ● If using single container, this configuration option can be any string that is a valid swift container name in Glance's Swift account ● If using multiple containers, this configuration option can be any string as long as it satisfies the container naming rules enforced by Swift. The value of swift_store_multiple_containers_seed should be taken into account as well. Related options: ● swift_store_multiple_containers_seed ● swift_store_multi_tenant ● swift_store_create_container_on_put

Configuration option = Default value	Type	Description
swift_store_create_container_on_put = False	boolean value	Create container, if it doesn't already exist, when uploading image. At the time of uploading an image, if the corresponding container doesn't exist, it will be created provided this configuration option is set to True. By default, it won't be created. This behavior is applicable for both single and multiple containers mode. Possible values: • True • False Related options: • None
swift_store_endpoint = None	string value	The URL endpoint to use for Swift backend storage. Provide a string value representing the URL endpoint to use for storing Glance images in Swift store. By default, an endpoint is not set and the storage URL returned by auth is used. Setting an endpoint with swift_store_endpoint overrides the storage URL and is used for Glance image storage.  NOTE The URL should include the path up to, but excluding the container. The location of an object is obtained by appending the container and object to the configured URL. Possible values: • String value representing a valid URL path up to a Swift container Related Options: • None

Configuration option = Default value	Type	Description
swift_store_endpoint_type = publicURL	string value	Endpoint Type of Swift service. This string value indicates the endpoint type to use to fetch the Swift endpoint. The endpoint type determines the actions the user will be allowed to perform, for instance, reading and writing to the Store. This setting is only used if <code>swift_store_auth_version</code> is greater than 1. Possible values: ● <code>publicURL</code> ● <code>adminURL</code> ● <code>internalURL</code> Related options: ● <code>swift_store_endpoint</code>
swift_store_expire_soon_interval = 60	integer value	Time in seconds defining the size of the window in which a new token may be requested before the current token is due to expire. Typically, the Swift storage driver fetches a new token upon the expiration of the current token to ensure continued access to Swift. However, some Swift transactions (like uploading image segments) may not recover well if the token expires on the fly. Hence, by fetching a new token before the current token expiration, we make sure that the token does not expire or is close to expiry before a transaction is attempted. By default, the Swift storage driver requests for a new token 60 seconds or less before the current token expiration. Possible values: ● Zero ● Positive integer value Related Options: ● None
swift_store_key = None	string value	Auth key for the user authenticating against the Swift authentication service.

Configuration option = Default value	Type	Description
swift_store_large_object_ chunk_size = 200	integer value	The maximum size, in MB, of the segments when image data is segmented. When image data is segmented to upload images that are larger than the limit enforced by the Swift cluster, image data is broken into segments that are no bigger than the size specified by this configuration option. Refer to swift_store_large_object_size for more detail. For example: if swift_store_large_object_size is 5GB and swift_store_large_object_chunk_size is 1GB, an image of size 6.2GB will be segmented into 7 segments where the first six segments will be 1GB in size and the seventh segment will be 0.2GB. Possible values: • A positive integer that is less than or equal to the large object limit enforced by Swift cluster in consideration. Related options: • swift_store_large_object_size

Configuration option = Default value	Type	Description
swift_store_large_object_size = 5120	integer value	The size threshold, in MB, after which Glance will start segmenting image data. Swift has an upper limit on the size of a single uploaded object. By default, this is 5GB. To upload objects bigger than this limit, objects are segmented into multiple smaller objects that are tied together with a manifest file. For more detail, refer to https://docs.openstack.org/swift/latest/overview_large_objects.html This configuration option specifies the size threshold over which the Swift driver will start segmenting image data into multiple smaller files. Currently, the Swift driver only supports creating Dynamic Large Objects.  NOTE This should be set by taking into account the large object limit enforced by the Swift cluster in consideration. Possible values: ● A positive integer that is less than or equal to the large object limit enforced by the Swift cluster in consideration. Related options: ● swift_store_large_object_chunk_size

Configuration option = Default value	Type	Description
swift_store_multi_tenant = False	boolean value	Store images in tenant's Swift account. This enables multi-tenant storage mode which causes Glance images to be stored in tenant specific Swift accounts. If this is disabled, Glance stores all images in its own account. More details multi-tenant store can be found at https://wiki.openstack.org/wiki/GlanceSwiftTenantSpecificStorage  NOTE If using multi-tenant swift store, please make sure that you do not set a swift configuration file with the <i>swift_store_config_file</i> option. Possible values: • True • False Related options: • <i>swift_store_config_file</i>

Configuration option = Default value	Type	Description
swift_store_multiple_containers_seed = 0	integer value	Seed indicating the number of containers to use for storing images. When using a single-tenant store, images can be stored in one or more than one containers. When set to 0, all images will be stored in one single container. When set to an integer value between 1 and 32, multiple containers will be used to store images. This configuration option will determine how many containers are created. 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. Please refer to swift_store_container for more detail on the naming convention. More detail about using multiple containers can be found at https://specs.openstack.org/openstack/glance-specs/specs/kilo/swift-store-multiple-containers.html  NOTE This is used only when swift_store_multi_tenant is disabled. Possible values: • A non-negative integer less than or equal to 32 Related options: • swift_store_container • swift_store_multi_tenant • swift_store_create_container_on_put

Configuration option = Default value	Type	Description
swift_store_region = None	string value	The region of Swift endpoint to use by Glance. Provide a string value representing a Swift region where Glance can connect to for image storage. By default, there is no region set. When Glance uses Swift as the storage backend to store images for a specific tenant that has multiple endpoints, setting of a Swift region with swift_store_region allows Glance to connect to Swift in the specified region as opposed to a single region connectivity. This option can be configured for both single-tenant and multi-tenant storage.  NOTE Setting the region with swift_store_region is tenant-specific and is necessary only if the tenant has multiple endpoints across different regions. Possible values: • A string value representing a valid Swift region. Related Options: • None

Configuration option = Default value	Type	Description
swift_store_retry_get_count = 0	integer value	The number of times a Swift download will be retried before the request fails. Provide an integer value representing the number of times an image download must be retried before erroring out. The default value is zero (no retry on a failed image download). When set to a positive integer value, swift_store_retry_get_count ensures that the download is attempted this many more times upon a download failure before sending an error message. Possible values: • Zero • Positive integer value Related Options: • None
swift_store_service_type = object-store	string value	Type of Swift service to use. Provide a string value representing the service type to use for storing images while using Swift backend storage. The default service type is set to object-store.  NOTE If swift_store_auth_version is set to 2, the value for this configuration option needs to be object-store. If using a higher version of Keystone or a different auth scheme, this option may be modified. Possible values: • A string representing a valid service type for Swift storage. Related Options: • None

Configuration option = Default value	Type	Description
swift_store_ssl_compression = True	boolean value	SSL layer compression for HTTPS Swift requests. Provide a boolean value to determine whether or not to compress HTTPS Swift requests for images at the SSL layer. By default, compression is enabled. When using Swift as the backend store for Glance image storage, SSL layer compression of HTTPS Swift requests can be set using this option. If set to False, SSL layer compression of HTTPS Swift requests is disabled. Disabling this option may improve performance for images which are already in a compressed format, for example, qcow2. Possible values: • True • False Related Options: • None

Configuration option = Default value	Type	Description
swift_store_use_trusts = True	boolean value	Use trusts for multi-tenant Swift store. This option instructs the Swift store to create a trust for each add/get request when the multi-tenant store is in use. Using trusts allows the Swift store to avoid problems that can be caused by an authentication token expiring during the upload or download of data. By default, swift_store_use_trusts is set to True(use of trusts is enabled). If set to False, a user token is used for the Swift connection instead, eliminating the overhead of trust creation.  NOTE This option is considered only when swift_store_multi_tenant is set to True Possible values: • True • False Related options: • swift_store_multi_tenant
swift_store_user = None	string value	The user to authenticate against the Swift authentication service.

Configuration option = Default value	Type	Description
swift_upload_buffer_dir = None	string value	Directory to buffer image segments before upload to Swift. Provide a string value representing the absolute path to the directory on the glance node where image segments will be buffered briefly before they are uploaded to swift. NOTES: • This is required only when the configuration option swift_buffer_on_upload is set to True. • This directory should be provisioned keeping in mind the swift_store_large_object_chunk_size and the maximum number of images that could be uploaded simultaneously by a given glance node. Possible values: • String value representing an absolute directory path Related options: • swift_buffer_on_upload • swift_store_large_object_chunk_size
vmware_api_retry_count = 10	integer value	The number of VMware API retries. This configuration option specifies the number of times the VMware ESX/VC server API must be retried upon connection related issues or server API call overload. It is not possible to specify <i>retry forever</i>. Possible Values: • Any positive integer value Related options: • None

Configuration option = Default value	Type	Description
vmware_ca_file = None	string value	Absolute path to the CA bundle file. This configuration option enables the operator to use a custom Certificate Authority File to verify the ESX/vCenter certificate. If this option is set, the "vmware_insecure" option will be ignored and the CA file specified will be used to authenticate the ESX/vCenter server certificate and establish a secure connection to the server. Possible Values: - Any string that is a valid absolute path to a CA file Related options: - vmware_insecure
vmware_datastores = None	multi valued	The datastores where the image can be stored. This configuration option specifies the datastores where the image can be stored in the VMWare store backend. This option may be specified multiple times for specifying multiple datastores. 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 ":" to specify the priority. 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. Possible Values: Any string of the format: <datacenter_path>:<datastore_name>:<optional_weight> Related options: * None

Configuration option = Default value	Type	Description
vmware_insecure = False	boolean value	Set verification of the ESX/vCenter server certificate. This configuration option takes a boolean value to determine whether or not to verify the ESX/vCenter server certificate. If this option is set to True, the ESX/vCenter server certificate is not verified. If this option is set to False, then the default CA truststore is used for verification. This option is ignored if the "vmware_ca_file" option is set. In that case, the ESX/vCenter server certificate will then be verified using the file specified using the "vmware_ca_file" option . Possible Values: • True • False Related options: • vmware_ca_file
vmware_server_host = None	host address value	Address of the ESX/ESXi or vCenter Server target system. This configuration option sets the address of the ESX/ESXi or vCenter Server target system. This option is required when using the VMware storage backend. The address can contain an IP address (127.0.0.1) or a DNS name (www.my-domain.com). Possible Values: • A valid IPv4 or IPv6 address • A valid DNS name Related options: • vmware_server_username • vmware_server_password

Configuration option = Default value	Type	Description
vmware_server_password = None	string value	Server password. This configuration option takes the password for authenticating with the VMware ESX/ESXi or vCenter Server. This option is required when using the VMware storage backend. Possible Values: - Any string that is a password corresponding to the username specified using the "vmware_server_username" option Related options: - vmware_server_host - vmware_server_username
vmware_server_username = None	string value	Server username. This configuration option takes the username for authenticating with the VMware ESX/ESXi or vCenter Server. This option is required when using the VMware storage backend. Possible Values: - Any string that is the username for a user with appropriate privileges Related options: - vmware_server_host - vmware_server_password

Configuration option = Default value	Type	Description
vmware_store_image_dir = /openstack_glance	string value	The directory where the glance images will be stored in the datastore. This configuration option specifies the path to the directory where the glance images will be stored in the VMware datastore. If this option is not set, the default directory where the glance images are stored is openstack_glance. Possible Values: - Any string that is a valid path to a directory Related options: - None
vmware_task_poll_interval = 5	integer value	Interval in seconds used for polling remote tasks invoked on VMware ESX/VC server. This configuration option takes in the sleep time in seconds for polling an on-going async task as part of the VMWare ESX/VC server API call. Possible Values: - Any positive integer value Related options: - None

3.3.3. oslo_policy

The following table outlines the options available under the **[oslo_policy]** group in the **/etc/glance/glance-cache.conf** file.

Table 3.31. oslo_policy

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
enforce_new_defaults = False	boolean value	This option controls whether or not to use old deprecated defaults when evaluating policies. If True , the old deprecated defaults are not going to be evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be disallowed. It is encouraged to enable this flag along with the enforce_scope flag so that you can get the benefits of new defaults and scope_type together
enforce_scope = False	boolean value	This option controls whether or not to enforce scope when evaluating policies. If True , the scope of the token used in the request is compared to the scope_types of the policy being enforced. If the scopes do not match, an InvalidScope exception will be raised. If False , a message will be logged informing operators that policies are being invoked with mismatching scope.
policy_default_rule = default	string value	Default rule. Enforced when a requested rule is not found.
policy_dirs = ['policy.d']	multi valued	Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the <code>config_dir</code> option, or absolute paths. The file defined by <code>policy_file</code> must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = policy.yaml	string value	The relative or absolute path of a file that maps roles to permissions for a given service. Relative paths must be specified in relation to the configuration file setting this option.
remote_content_type = application/x-www-form-urlencoded	string value	Content Type to send and receive data for REST based policy check
remote_ssl_ca_cert_file = None	string value	Absolute path to ca cert file for REST based policy check
remote_ssl_client_cert_file = None	string value	Absolute path to client cert for REST based policy check
remote_ssl_client_key_file = None	string value	Absolute path client key file REST based policy check

Configuration option = Default value	Type	Description
remote_ssl_verify_server _cert = False	boolean value	server identity verification for REST based policy check

CHAPTER 4. HEAT

The following chapter contains information about the configuration options in the **heat** service.

4.1. HEAT.CONF

This section contains options for the `/etc/heat/heat.conf` file.

4.1.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/heat/heat.conf` file.

Configuration option = Default value	Type	Description
action_retry_limit = 5	integer value	Number of times to retry to bring a resource to a non-error state. Set to 0 to disable retries.
allow_trusts_redelegation = False	boolean value	Create trusts with redelegation enabled. This option is only used when <code>reauthentication_auth_method</code> is set to "trusts". Note that enabling this option does have security implications as all trusts created by Heat will use both impersonation and redelegation enabled. Enable it only when there are other services that need to create trusts from tokens Heat uses to access them, examples are Aodh and Heat in another region when configured to use trusts too.
auth_encryption_key = notgood but just long enough i t	string value	Key used to encrypt authentication info in the database. Length of this key must be 32 characters.
backdoor_port = None	string value	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.
backdoor_socket = None	string value	Enable eventlet backdoor, using the provided path as a unix socket that can receive connections. This option is mutually exclusive with <code>backdoor_port</code> in that only one should be provided. If both are provided then the existence of this option overrides the usage of that option. Inside the path {pid} will be replaced with the PID of the current process.

Configuration option = Default value	Type	Description
client_retry_limit = 2	integer value	Number of times to retry when a client encounters an expected intermittent error. Set to 0 to disable retries.
cloud_backend = heat.engine.clients.OpenStackClients	string value	Fully qualified class name to use as a client backend.
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
control_exchange = openstack	string value	The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.
convergence_engine = True	boolean value	Enables engine with convergence architecture. All stacks with this option will be created using convergence engine.
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.
default_deployment_signal_transport = CFN_SIGNAL	string value	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. ZAQAR_SIGNAL will create a dedicated zaqar queue to be signaled using the provided keystone credentials.

Configuration option = Default value	Type	Description
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
default_notification_level = INFO	string value	Default notification level for outgoing notifications.
default_publisher_id = None	string value	Default publisher_id for outgoing notifications.

Configuration option = Default value	Type	Description
default_software_config_transport = POLL_SERVER_CFN	string value	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). ZAQAR_MESSAGE will create a dedicated zaqar queue and post the metadata for polling.
default_user_data_format = HEAT_CFNTOOLS	string value	Template default for how the user_data should be formatted for the server. For HEAT_CFNTOOLS, the user_data is bundled as part of the heat-cfntools cloud-init boot configuration data. For RAW the user_data is passed to Nova unmodified. For SOFTWARE_CONFIG user_data is bundled as part of the software config data, and metadata is derived from any associated SoftwareDeployment resources.
deferred_auth_method = trusts	string value	Select deferred auth method, stored password or trusts. Deprecated since: 9.0.0 *Reason:*Stored password based deferred auth is broken when used with keystone v3 and is not supported.
enable_cloud_watch_lite = False	boolean value	Enable the legacy OS::Heat::CWLiteAlarm resource. Deprecated since: 10.0.0 *Reason:*Heat CloudWatch Service has been removed.
enable_stack_abandon = False	boolean value	Enable the preview Stack Abandon feature.
enable_stack_adopt = False	boolean value	Enable the preview Stack Adopt feature.
encrypt_parameters_and_properties = False	boolean value	Encrypt template parameters that were marked as hidden and also all the resource properties before storing them in database.

Configuration option = Default value	Type	Description
engine_life_check_timeout = 2	integer value	RPC timeout for the engine liveness check that is used for stack locking.
environment_dir = /etc/heat/environment.d	string value	The directory to search for environment files.
error_wait_time = 240	integer value	The amount of time in seconds after an error has occurred that tasks may continue to run before being cancelled.
event_purge_batch_size = 200	integer value	Controls how many events will be pruned whenever a stack's events are purged. Set this lower to keep more events at the expense of more frequent purges.
executor_thread_pool_size = 64	integer value	Size of executor thread pool when executor is threading or eventlet.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
graceful_shutdown_timeout = 60	integer value	Specify a timeout after which a gracefully shutdown server will exit. Zero value means endless wait.
heat_metadata_server_url = None	string value	URL of the Heat metadata server. NOTE: Setting this is only needed if you require instances to use a different endpoint than in the keystone catalog
heat_stack_user_role = heat_stack_user	string value	Keystone role for heat template-defined users.
heat_waitcondition_server_url = None	string value	URL of the Heat waitcondition server.
heat_watch_server_url = ``	string value	URL of the Heat CloudWatch server. Deprecated since: 10.0.0 *Reason:*Heat CloudWatch Service has been removed.
hidden_stack_tags = ['data-processing-cluster']	list value	Stacks containing these tag names will be hidden. Multiple tags should be given in a comma-delimited list (eg. hidden_stack_tags=hide_me,me_too).
host = <based on operating system>	string value	Name of the engine node. This can be an opaque identifier. It is not necessarily a hostname, FQDN, or IP address.

Configuration option = Default value	Type	Description
instance_connection_https_validate_certificates = 1	string value	Instance connection to CFN/CW API validate certs if SSL is used.
instance_connection_is_secure = 0	string value	Instance connection to CFN/CW API via https.
<code>`instance_format = [instance: %(uuid)s]`</code>	string value	The format for an instance that is passed with the log message.
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	string value	The format for an instance UUID that is passed with the log message.
keystone_backend = heat.engine.clients.os.keystone. heat_keystoneclient.KsClientWrapper	string value	Fully qualified class name to use as a keystone backend.
loadbalancer_template = None	string value	Custom template for the built-in loadbalancer nested stack.
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code> . This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code> . This option is ignored if <code>log_config_append</code> is set.
log_options = True	boolean value	Enables or disables logging values of all registered options when starting a service (at DEBUG level).

Configuration option = Default value	Type	Description
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless log_rotation_type is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	string value	Format string to use for log messages with context. Used by oslo_log.formatters.ContextFormatter
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is DEBUG. Used by oslo_log.formatters.ContextFormatter
logging_default_format_s tring = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s	string value	Format string to use for log messages when context is undefined. Used by oslo_log.formatters.ContextFormatter
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s	string value	Prefix each line of exception output with this format. Used by oslo_log.formatters.ContextFormatter
logging_user_identity_for mat = %(user)s % (tenant)s %(domain)s % (user_domain)s % (project_domain)s	string value	Defines the format string for %(user_identity)s that is used in logging_context_format_string. Used by oslo_log.formatters.ContextFormatter

Configuration option = Default value	Type	Description
max_events_per_stack = 1000	integer value	Rough number of maximum events that will be available per stack. Actual number of events can be a bit higher since purge checks take place randomly 200/event_purge_batch_size percent of the time. Older events are deleted when events are purged. Set to 0 for unlimited events per stack.
max_interface_check_attempts = 10	integer value	Number of times to check whether an interface has been attached or detached.
max_ironic_api_microversion = None	floating point value	Maximum ironic API version for client plugin. With this limitation, any ironic feature supported with microversion number above max_ironic_api_microversion will not be available.
max_json_body_size = 1048576	integer value	Maximum raw byte size of JSON request body. Should be larger than max_template_size.
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
max_nested_stack_depth = 5	integer value	Maximum depth allowed when using nested stacks.
max_nova_api_microversion = None	floating point value	Maximum nova API version for client plugin. With this limitation, any nova feature supported with microversion number above max_nova_api_microversion will not be available.
max_resources_per_stack = 1000	integer value	Maximum resources allowed per top-level stack. -1 stands for unlimited.
max_server_name_length = 53	integer value	Maximum length of a server name to be used in nova.
max_stacks_per_tenant = 512	integer value	Maximum number of stacks any one tenant may have active at one time. -1 stands for unlimited.
max_template_size = 524288	integer value	Maximum raw byte size of any template.
num_engine_workers = None	integer value	Number of heat-engine processes to fork and run. Will default to either to 4 or number of CPUs on the host, whichever is greater.

Configuration option = Default value	Type	Description
observe_on_update = False	boolean value	On update, enables heat to collect existing resource properties from reality and converge to updated template.
onready = None	string value	Deprecated.
periodic_interval = 60	integer value	Seconds between running periodic tasks.
plugin_dirs = ['/usr/lib64/heat', '/usr/lib/heat', '/usr/local/lib/heat', '/usr/local/lib64/heat']	list value	List of directories to search for plug-ins.
publish_errors = False	boolean value	Enables or disables publication of error events.
rate_limit_burst = 0	integer value	Maximum number of logged messages per rate_limit_interval.
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
reauthentication_auth_method = `	string value	Allow reauthentication on token expiry, such that long-running tasks may complete. Note this defeats the expiry of any provided user tokens.
region_name_for_services = None	string value	Default region name used to get services endpoints.
region_name_for_shared_services = None	string value	Region name for shared services endpoints.
rpc_conn_pool_size = 30	integer value	Size of RPC connection pool.
rpc_ping_enabled = False	boolean value	Add an endpoint to answer to ping calls. Endpoint is named oslo_rpc_server_ping
rpc_response_timeout = 60	integer value	Seconds to wait for a response from a call.
run_external_periodic_tasks = True	boolean value	Some periodic tasks can be run in a separate process. Should we run them here?

Configuration option = Default value	Type	Description
<code>server_keystone_endpoint_type =`</code>	string value	If set, is used to control which authentication endpoint is used by user-controlled servers to make calls back to Heat. If unset <code>www_authenticate_uri</code> is used.
shared_services_types = <code>['image', 'volume', 'volume2']</code>	list value	The shared services located in the other region. Needs <code>region_name_for_shared_services</code> option to be set for this to take effect.
stack_action_timeout = <code>3600</code>	integer value	Timeout in seconds for stack action (ie. create or update).
stack_domain_admin = <code>None</code>	string value	Keystone username, a user with roles sufficient to manage users and projects in the <code>stack_user_domain</code> .
stack_domain_admin_password = <code>None</code>	string value	Keystone password for <code>stack_domain_admin</code> user.
stack_scheduler_hints = <code>False</code>	boolean value	When this feature is enabled, scheduler hints identifying the heat stack context of a server or volume resource are passed to the configured schedulers in nova and cinder, for creates done using heat resource types <code>OS::Cinder::Volume</code> , <code>OS::Nova::Server</code> , and <code>AWS::EC2::Instance</code> . <code>heat_root_stack_id</code> will be set to the id of the root stack of the resource, <code>heat_stack_id</code> will be set to the id of the resource's parent stack, <code>heat_stack_name</code> will be set to the name of the resource's parent stack, <code>heat_path_in_stack</code> will be set to a list of comma delimited strings of <code>stackresource name</code> and <code>stackname</code> with <code>list[0]</code> being <code>rootstackname</code> , <code>heat_resource_name</code> will be set to the resource's name, and <code>heat_resource_uuid</code> will be set to the resource's orchestration id.
stack_user_domain_id = <code>None</code>	string value	Keystone domain ID which contains heat template-defined users. If this option is set, <code>stack_user_domain_name</code> option will be ignored.
stack_user_domain_name = <code>None</code>	string value	Keystone domain name which contains heat template-defined users. If <code>stack_user_domain_id</code> option is set, this option is ignored.
syslog-log-facility = <code>LOG_USER</code>	string value	Syslog facility to receive log lines. This option is ignored if <code>log_config_append</code> is set.

Configuration option = Default value	Type	Description
template_dir = /etc/heat/templates	string value	The directory to search for template files.
transport_url = rabbit://	string value	The network address and optional user credentials for connecting to the messaging backend, in URL format. The expected format is: <pre>driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query</pre> Example: rabbit://rabbitmq:password@127.0.0.1:5672// For full details on the fields in the URL see the documentation of oslo_messaging.TransportURL at https://docs.openstack.org/oslo.messaging/latest/reference/transport.html
trusts_delegated_roles = []	list value	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.
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

4.1.2. auth_password

The following table outlines the options available under the **[auth_password]** group in the **/etc/heat/heat.conf** file.

Table 4.1. auth_password

Configuration option = Default value	Type	Description
allowed_auth_uris = []	list value	Allowed keystone endpoints for auth_uri when multi_cloud is enabled. At least one endpoint needs to be specified.
multi_cloud = False	boolean value	Allow orchestration of multiple clouds.

4.1.3. clients

The following table outlines the options available under the **[clients]** group in the **/etc/heat/heat.conf** file.

Table 4.2. clients

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = publicURL	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = False	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.4. clients_aodh

The following table outlines the options available under the **[clients_aodh]** group in the **/etc/heat/heat.conf** file.

Table 4.3. clients_aodh

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.5. clients_barbican

The following table outlines the options available under the **[clients_barbican]** group in the **/etc/heat/heat.conf** file.

Table 4.4. clients_barbican

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.6. clients_cinder

The following table outlines the options available under the **[clients_cinder]** group in the **/etc/heat/heat.conf** file.

Table 4.5. clients_cinder

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
http_log_debug = False	boolean value	Allow client's debug log output.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.7. clients_designate

The following table outlines the options available under the **[clients_designate]** group in the **/etc/heat/heat.conf** file.

Table 4.6. clients_designate

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.8. clients_glance

The following table outlines the options available under the **[clients_glance]** group in the **/etc/heat/heat.conf** file.

Table 4.7. clients_glance

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.9. clients_heat

The following table outlines the options available under the **[clients_heat]** group in the **/etc/heat/heat.conf** file.

Table 4.8. clients_heat

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.
url = `	string value	Optional heat url in format like http://0.0.0.0:8004/v1/(tenant_id)s .

4.1.10. clients_keystone

The following table outlines the options available under the **[clients_keystone]** group in the **/etc/heat/heat.conf** file.

Table 4.9. clients_keystone

Configuration option = Default value	Type	Description
<code>`auth_uri = `</code>	string value	Unversioned keystone url in format like http://0.0.0.0:5000 .
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.11. clients_magnum

The following table outlines the options available under the **[clients_magnum]** group in the `/etc/heat/heat.conf` file.

Table 4.10. clients_magnum

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.12. clients_manila

The following table outlines the options available under the **[clients_manila]** group in the `/etc/heat/heat.conf` file.

Table 4.11. clients_manila

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.13. clients_mistral

The following table outlines the options available under the **[clients_mistral]** group in the **/etc/heat/heat.conf** file.

Table 4.12. clients_mistral

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.14. clients_monasca

The following table outlines the options available under the **[clients_monasca]** group in the **/etc/heat/heat.conf** file.

Table 4.13. clients_monasca

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.

Configuration option = Default value	Type	Description
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.15. clients_neutron

The following table outlines the options available under the **[clients_neutron]** group in the **/etc/heat/heat.conf** file.

Table 4.14. clients_neutron

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.16. clients_nova

The following table outlines the options available under the **[clients_nova]** group in the **/etc/heat/heat.conf** file.

Table 4.15. clients_nova

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.

Configuration option = Default value	Type	Description
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
http_log_debug = False	boolean value	Allow client's debug log output.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.17. clients_octavia

The following table outlines the options available under the **[clients_octavia]** group in the **/etc/heat/heat.conf** file.

Table 4.16. clients_octavia

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.18. clients_sahara

The following table outlines the options available under the **[clients_sahara]** group in the **/etc/heat/heat.conf** file.

Table 4.17. clients_sahara

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.

Configuration option = Default value	Type	Description
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.19. clients_senlin

The following table outlines the options available under the **[clients_senlin]** group in the **/etc/heat/heat.conf** file.

Table 4.18. clients_senlin

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.20. clients_swift

The following table outlines the options available under the **[clients_swift]** group in the **/etc/heat/heat.conf** file.

Table 4.19. clients_swift

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.

Configuration option = Default value	Type	Description
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.21. clients_trove

The following table outlines the options available under the **[clients_trove]** group in the **/etc/heat/heat.conf** file.

Table 4.20. clients_trove

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.22. clients_vitrage

The following table outlines the options available under the **[clients_vitrage]** group in the **/etc/heat/heat.conf** file.

Table 4.21. clients_vitrage

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.

Configuration option = Default value	Type	Description
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.23. clients_zaqar

The following table outlines the options available under the **[clients_zaqar]** group in the `/etc/heat/heat.conf` file.

Table 4.22. clients_zaqar

Configuration option = Default value	Type	Description
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
endpoint_type = None	string value	Type of endpoint in Identity service catalog to use for communication with the OpenStack service.
insecure = None	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.

4.1.24. cors

The following table outlines the options available under the **[cors]** group in the `/etc/heat/heat.conf` file.

Table 4.23. cors

Configuration option = Default value	Type	Description
allow_credentials = True	boolean value	Indicate that the actual request can include user credentials

Configuration option = Default value	Type	Description
allow_headers = ['X-Auth-Token', 'X-Identity-Status', 'X-Roles', 'X-Service-Catalog', 'X-User-Id', 'X-Tenant-Id', 'X-OpenStack-Request-ID']	list value	Indicate which header field names may be used during the actual request.
allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH']	list value	Indicate which methods can be used during the actual request.
allowed_origin = None	list value	Indicate whether this resource may be shared with the domain received in the requests "origin" header. Format: "<protocol>://<host>[:<port>]", no trailing slash. Example: https://horizon.example.com
expose_headers = ['X-Auth-Token', 'X-Subject-Token', 'X-Service-Token', 'X-OpenStack-Request-ID']	list value	Indicate which headers are safe to expose to the API. Defaults to HTTP Simple Headers.
max_age = 3600	integer value	Maximum cache age of CORS preflight requests.

4.1.25. database

The following table outlines the options available under the **[database]** group in the `/etc/heat/heat.conf` file.

Table 4.24. database

Configuration option = Default value	Type	Description
backend = sqlalchemy	string value	The back end to use for the database.
connection = None	string value	The SQLAlchemy connection string to use to connect to the database.
connection_debug = 0	integer value	Verbosity of SQL debugging information: 0=None, 100=Everything.
<code>`connection_parameters</code> = ``	string value	Optional URL parameters to append onto the connection URL at connect time; specify as param1=value1&param2=value2&...

Configuration option = Default value	Type	Description
connection_recycle_time = 3600	integer value	Connections which have been present in the connection pool longer than this number of seconds will be replaced with a new one the next time they are checked out from the pool.
connection_trace = False	boolean value	Add Python stack traces to SQL as comment strings.
db_inc_retry_interval = True	boolean value	If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = 20	integer value	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	integer value	If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = 1	integer value	Seconds between retries of a database transaction.
max_overflow = 50	integer value	If set, use this value for max_overflow with SQLAlchemy.
max_pool_size = 5	integer value	Maximum number of SQL connections to keep open in a pool. Setting a value of 0 indicates no limit.
max_retries = 10	integer value	Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.
mysql_enable_ndb = False	boolean value	If True, transparently enables support for handling MySQL Cluster (NDB).
mysql_sql_mode = TRADITIONAL	string value	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	integer value	If set, use this value for pool_timeout with SQLAlchemy.
retry_interval = 10	integer value	Interval between retries of opening a SQL connection.
slave_connection = None	string value	The SQLAlchemy connection string to use to connect to the slave database.

Configuration option = Default value	Type	Description
sqlite_synchronous = True	boolean value	If True, SQLite uses synchronous mode.
use_db_reconnect = False	boolean value	Enable the experimental use of database reconnect on connection lost.

4.1.26. ec2authtoken

The following table outlines the options available under the **[ec2authtoken]** group in the **/etc/heat/heat.conf** file.

Table 4.25. ec2authtoken

Configuration option = Default value	Type	Description
allowed_auth_uris = []	list value	Allowed keystone endpoints for auth_uri when multi_cloud is enabled. At least one endpoint needs to be specified.
auth_uri = None	string value	Authentication Endpoint URI.
ca_file = None	string value	Optional CA cert file to use in SSL connections.
cert_file = None	string value	Optional PEM-formatted certificate chain file.
insecure = False	boolean value	If set, then the server's certificate will not be verified.
key_file = None	string value	Optional PEM-formatted file that contains the private key.
multi_cloud = False	boolean value	Allow orchestration of multiple clouds.

4.1.27. eventlet_opts

The following table outlines the options available under the **[eventlet_opts]** group in the **/etc/heat/heat.conf** file.

Table 4.26. eventlet_opts

Configuration option = Default value	Type	Description
client_socket_timeout = 900	integer value	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.
wsgi_keep_alive = True	boolean value	If False, closes the client socket connection explicitly.

4.1.28. healthcheck

The following table outlines the options available under the **[healthcheck]** group in the **/etc/heat/heat.conf** file.

Table 4.27. healthcheck

Configuration option = Default value	Type	Description
backends = []	list value	Additional backends that can perform health checks and report that information back as part of a request.
detailed = False	boolean value	Show more detailed information as part of the response. Security note: Enabling this option may expose sensitive details about the service being monitored. Be sure to verify that it will not violate your security policies.
disable_by_file_path = None	string value	Check the presence of a file to determine if an application is running on a port. Used by DisableByFileHealthcheck plugin.
disable_by_file_paths = []	list value	Check the presence of a file based on a port to determine if an application is running on a port. Expects a "port:path" list of strings. Used by DisableByFilesPortsHealthcheck plugin.
path = /healthcheck	string value	The path to respond to healthcheck requests on.

4.1.29. heat_api

The following table outlines the options available under the **[heat_api]** group in the **/etc/heat/heat.conf** file.

Table 4.28. heat_api

Configuration option = Default value	Type	Description
backlog = 4096	integer value	Number of backlog requests to configure the socket with.
bind_host = 0.0.0.0	IP address value	Address to bind the server. Useful when selecting a particular network interface.
bind_port = 8004	port value	The port on which the server will listen.
cert_file = None	string value	Location of the SSL certificate file to use for SSL mode.
key_file = None	string value	Location of the SSL key file to use for enabling SSL mode.
max_header_line = 16384	integer value	Maximum line size of message headers to be accepted. max_header_line may need to be increased when using large tokens (typically those generated by the Keystone v3 API with big service catalogs).
tcp_keepidle = 600	integer value	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.
workers = 0	integer value	Number of workers for Heat service. Default value 0 means, that service will start number of workers equal number of cores on server.

4.1.30. heat_api_cfn

The following table outlines the options available under the **[heat_api_cfn]** group in the **/etc/heat/heat.conf** file.

Table 4.29. heat_api_cfn

Configuration option = Default value	Type	Description
backlog = 4096	integer value	Number of backlog requests to configure the socket with.
bind_host = 0.0.0.0	IP address value	Address to bind the server. Useful when selecting a particular network interface.
bind_port = 8000	port value	The port on which the server will listen.

Configuration option = Default value	Type	Description
cert_file = None	string value	Location of the SSL certificate file to use for SSL mode.
key_file = None	string value	Location of the SSL key file to use for enabling SSL mode.
max_header_line = 16384	integer value	Maximum line size of message headers to be accepted. max_header_line may need to be increased when using large tokens (typically those generated by the Keystone v3 API with big service catalogs).
tcp_keepidle = 600	integer value	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.
workers = 1	integer value	Number of workers for Heat service.

4.1.31. heat_api_cloudwatch

The following table outlines the options available under the **[heat_api_cloudwatch]** group in the **/etc/heat/heat.conf** file.

Table 4.30. heat_api_cloudwatch

Configuration option = Default value	Type	Description
backlog = 4096	integer value	Number of backlog requests to configure the socket with. Deprecated since: 10.0.0 *Reason:*Heat CloudWatch API has been removed.
bind_host = 0.0.0.0	IP address value	Address to bind the server. Useful when selecting a particular network interface. Deprecated since: 10.0.0 *Reason:*Heat CloudWatch API has been removed.
bind_port = 8003	port value	The port on which the server will listen. Deprecated since: 10.0.0 *Reason:*Heat CloudWatch API has been removed.

Configuration option = Default value	Type	Description
cert_file = None	string value	Location of the SSL certificate file to use for SSL mode. Deprecated since: 10.0.0 *Reason:*Heat CloudWatch API has been Removed.
key_file = None	string value	Location of the SSL key file to use for enabling SSL mode. Deprecated since: 10.0.0 *Reason:*Heat CloudWatch API has been Removed.
max_header_line = 16384	integer value	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.) Deprecated since: 10.0.0 *Reason:*Heat CloudWatch API has been Removed.
tcp_keepidle = 600	integer value	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. Deprecated since: 10.0.0 *Reason:*Heat CloudWatch API has been Removed.
workers = 1	integer value	Number of workers for Heat service. Deprecated since: 10.0.0 *Reason:*Heat CloudWatch API has been Removed.

4.1.32. keystone_authtoken

The following table outlines the options available under the **[keystone_authtoken]** group in the **/etc/heat/heat.conf** file.

Table 4.31. keystone_authtoken

Configuration option = Default value	Type	Description
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load

Configuration option = Default value	Type	Description
auth_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint. This option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release. Deprecated since: Queens <i>*Reason:</i>*The <code>auth_uri</code> option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release.
auth_version = None	string value	API version of the Identity API endpoint.
cache = None	string value	Request environment key where the Swift cache object is stored. When <code>auth_token</code> middleware is deployed with a Swift cache, use this option to have the middleware share a caching backend with swift. Otherwise, use the memcached_servers option instead.
cafile = None	string value	A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
certfile = None	string value	Required if identity server requires client certificate
delay_auth_decision = False	boolean value	Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components.
enforce_token_bind = permissive	string value	Used to control the use and type of token binding. Can be set to: "disabled" to not check token binding. "permissive" (default) to validate binding information if the bind type is of a form known to the server and ignore it if not. "strict" like "permissive" but if the bind type is unknown the token will be rejected. "required" any form of token binding is needed to be allowed. Finally the name of a binding method that must be present in tokens.
http_connect_timeout = None	integer value	Request timeout value for communicating with Identity API server.

Configuration option = Default value	Type	Description
http_request_max_retries = 3	integer value	How many times are we trying to reconnect when communicating with Identity API Server.
include_service_catalog = True	boolean value	(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	boolean value	Verify HTTPS connections.
interface = internal	string value	Interface to use for the Identity API endpoint. Valid values are "public", "internal" (default) or "admin".
keyfile = None	string value	Required if identity server requires client certificate
memcache_pool_conn_get_timeout = 10	integer value	(Optional) Number of seconds that an operation will wait to get a memcached client connection from the pool.
memcache_pool_dead_retry = 300	integer value	(Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = 10	integer value	(Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = 3	integer value	(Optional) Socket timeout in seconds for communicating with a memcached server.
memcache_pool_unused_timeout = 60	integer value	(Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
memcache_secret_key = None	string value	(Optional, mandatory if memcache_security_strategy is defined) This string is used for key derivation.
memcache_security_strategy = None	string value	(Optional) If defined, indicate whether token data should be authenticated or authenticated and encrypted. If MAC, token data is authenticated (with HMAC) in the cache. If ENCRYPT, token data is encrypted and authenticated in the cache. If the value is not one of these options or empty, auth_token will raise an exception on initialization.
memcache_use_advanced_pool = False	boolean value	(Optional) Use the advanced (eventlet safe) memcached client pool. The advanced pool will only work under python 2.x.

Configuration option = Default value	Type	Description
memcached_servers = None	list value	Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
region_name = None	string value	The region in which the identity server can be found.
service_token_roles = ['service']	list value	A choice of roles that must be present in a service token. Service tokens are allowed to request that an expired token can be used and so this check should tightly control that only actual services should be sending this token. Roles here are applied as an ANY check so any role in this list must be present. For backwards compatibility reasons this currently only affects the allow_expired check.
service_token_roles_required = False	boolean value	For backwards compatibility reasons we must let valid service tokens pass that don't pass the service_token_roles check as valid. Setting this true will become the default in a future release and should be enabled if possible.
service_type = None	string value	The name or type of the service as it appears in the service catalog. This is used to validate tokens that have restricted access rules.
token_cache_time = 300	integer value	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.
www_authenticate_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint.

4.1.33. noauth

The following table outlines the options available under the **[noauth]** group in the `/etc/heat/heat.conf` file.

Table 4.32. noauth

Configuration option = Default value	Type	Description
<code>`token_response = `</code>	string value	JSON file containing the content returned by the noauth middleware.

4.1.34. oslo_messaging_amqp

The following table outlines the options available under the **[oslo_messaging_amqp]** group in the `/etc/heat/heat.conf` file.

Table 4.33. oslo_messaging_amqp

Configuration option = Default value	Type	Description
addressing_mode = dynamic	string value	Indicates the addressing mode used by the driver. Permitted values: <i>legacy</i> - use legacy non-routable addressing <i>routable</i> - use routable addresses <i>dynamic</i> - use legacy addresses if the message bus does not support routing otherwise use routable addressing
anycast_address = anycast	string value	Appended to the address prefix when sending to a group of consumers. Used by the message bus to identify messages that should be delivered in a round-robin fashion across consumers.
broadcast_prefix = broadcast	string value	address prefix used when broadcasting to all servers
connection_retry_backoff = 2	integer value	Increase the connection_retry_interval by this many seconds after each unsuccessful failover attempt.
connection_retry_interval = 1	integer value	Seconds to pause before attempting to re-connect.
connection_retry_interval_max = 30	integer value	Maximum limit for connection_retry_interval + connection_retry_backoff
container_name = None	string value	Name for the AMQP container. must be globally unique. Defaults to a generated UUID
default_notification_exchange = None	string value	Exchange name used in notification addresses. Exchange name resolution precedence: Target.exchange if set else default_notification_exchange if set else control_exchange if set else <i>notify</i>

Configuration option = Default value	Type	Description
default_notify_timeout = 30	integer value	The deadline for a sent notification message delivery. Only used when caller does not provide a timeout expiry.
default_reply_retry = 0	integer value	The maximum number of attempts to re-send a reply message which failed due to a recoverable error.
default_reply_timeout = 30	integer value	The deadline for an rpc reply message delivery.
default_rpc_exchange = None	string value	Exchange name used in RPC addresses. Exchange name resolution precedence: Target.exchange if set else default_rpc_exchange if set else control_exchange if set else <i>rpc</i>
default_send_timeout = 30	integer value	The deadline for an rpc cast or call message delivery. Only used when caller does not provide a timeout expiry.
default_sender_link_timeout = 600	integer value	The duration to schedule a purge of idle sender links. Detach link after expiry.
group_request_prefix = unicast	string value	address prefix when sending to any server in group
idle_timeout = 0	integer value	Timeout for inactive connections (in seconds)
link_retry_delay = 10	integer value	Time to pause between re-connecting an AMQP 1.0 link that failed due to a recoverable error.
multicast_address = multicast	string value	Appended to the address prefix when sending a fanout message. Used by the message bus to identify fanout messages.
notify_address_prefix = openstack.org/om/notify	string value	Address prefix for all generated Notification addresses
notify_server_credit = 100	integer value	Window size for incoming Notification messages
pre_settled = ['rpc-cast', 'rpc-reply']	multi valued	Send messages of this type pre-settled. Pre-settled messages will not receive acknowledgement from the peer. Note well: pre-settled messages may be silently discarded if the delivery fails. Permitted values: <i>rpc-call</i> - send RPC Calls pre-settled <i>rpc-reply</i> - send RPC Replies pre-settled <i>rpc-cast</i> - Send RPC Casts pre-settled <i>notify</i> - Send Notifications pre-settled

Configuration option = Default value	Type	Description
pseudo_vhost = True	boolean value	Enable virtual host support for those message buses that do not natively support virtual hosting (such as <code>qpidd</code>). When set to true the virtual host name will be added to all message bus addresses, effectively creating a private <i>subnet</i> per virtual host. Set to False if the message bus supports virtual hosting using the <i>hostname</i> field in the AMQP 1.0 Open performative as the name of the virtual host.
reply_link_credit = 200	integer value	Window size for incoming RPC Reply messages.
rpc_address_prefix = openstack.org/om/rpc	string value	Address prefix for all generated RPC addresses
rpc_server_credit = 100	integer value	Window size for incoming RPC Request messages
<code>`sasldb_config_dir = `</code>	string value	Path to directory that contains the SASL configuration
<code>`sasldb_config_name = `</code>	string value	Name of configuration file (without <code>.conf</code> suffix)
<code>`sasldb_default_realm = `</code>	string value	SASL realm to use if no realm present in username
<code>`sasldb_mechanisms = `</code>	string value	Space separated list of acceptable SASL mechanisms
server_request_prefix = exclusive	string value	address prefix used when sending to a specific server
ssl = False	boolean value	Attempt to connect via SSL. If no other ssl-related parameters are given, it will use the system's CA-bundle to verify the server's certificate.
<code>`ssl_ca_file = `</code>	string value	CA certificate PEM file used to verify the server's certificate
<code>`ssl_cert_file = `</code>	string value	Self-identifying certificate PEM file for client authentication
<code>`ssl_key_file = `</code>	string value	Private key PEM file used to sign <code>ssl_cert_file</code> certificate (optional)
ssl_key_password = None	string value	Password for decrypting <code>ssl_key_file</code> (if encrypted)

Configuration option = Default value	Type	Description
ssl_verify_vhost = False	boolean value	By default SSL checks that the name in the server's certificate matches the hostname in the transport_url. In some configurations it may be preferable to use the virtual hostname instead, for example if the server uses the Server Name Indication TLS extension (rfc6066) to provide a certificate per virtual host. Set ssl_verify_vhost to True if the server's SSL certificate uses the virtual host name instead of the DNS name.
trace = False	boolean value	Debug: dump AMQP frames to stdout
unicast_address = unicast	string value	Appended to the address prefix when sending to a particular RPC/Notification server. Used by the message bus to identify messages sent to a single destination.

4.1.35. oslo_messaging_kafka

The following table outlines the options available under the **[oslo_messaging_kafka]** group in the **/etc/heat/heat.conf** file.

Table 4.34. oslo_messaging_kafka

Configuration option = Default value	Type	Description
compression_codec = none	string value	The compression codec for all data generated by the producer. If not set, compression will not be used. Note that the allowed values of this depend on the kafka version
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
consumer_group = oslo_messaging_consumer	string value	Group id for Kafka consumer. Consumers in one group will coordinate message consumption
enable_auto_commit = False	boolean value	Enable asynchronous consumer commits
kafka_consumer_timeout = 1.0	floating point value	Default timeout(s) for Kafka consumers

Configuration option = Default value	Type	Description
kafka_max_fetch_bytes = 1048576	integer value	Max fetch bytes of Kafka consumer
max_poll_records = 500	integer value	The maximum number of records returned in a poll call
pool_size = 10	integer value	Pool Size for Kafka Consumers
producer_batch_size = 16384	integer value	Size of batch for the producer async send
producer_batch_timeout = 0.0	floating point value	Upper bound on the delay for KafkaProducer batching in seconds
sasl_mechanism = PLAIN	string value	Mechanism when security protocol is SASL
security_protocol = PLAINTEXT	string value	Protocol used to communicate with brokers
<code>`ssl_cafile = `</code>	string value	CA certificate PEM file used to verify the server certificate
<code>`ssl_client_cert_file = `</code>	string value	Client certificate PEM file used for authentication.
<code>`ssl_client_key_file = `</code>	string value	Client key PEM file used for authentication.
<code>`ssl_client_key_password = `</code>	string value	Client key password file used for authentication.

4.1.36. oslo_messaging_notifications

The following table outlines the options available under the **[oslo_messaging_notifications]** group in the **/etc/heat/heat.conf** file.

Table 4.35. oslo_messaging_notifications

Configuration option = Default value	Type	Description
driver = []	multi valued	The Drivers(s) to handle sending notifications. Possible values are messaging, messagingv2, routing, log, test, noop
retry = -1	integer value	The maximum number of attempts to re-send a notification message which failed to be delivered due to a recoverable error. 0 - No retry, -1 - indefinite

Configuration option = Default value	Type	Description
topics = ['notifications']	list value	AMQP topic used for OpenStack notifications.
transport_url = None	string value	A URL representing the messaging driver to use for notifications. If not set, we fall back to the same configuration used for RPC.

4.1.37. oslo_messaging_rabbit

The following table outlines the options available under the **[oslo_messaging_rabbit]** group in the **/etc/heat/heat.conf** file.

Table 4.36. oslo_messaging_rabbit

Configuration option = Default value	Type	Description
amqp_auto_delete = False	boolean value	Auto-delete queues in AMQP.
amqp_durable_queues = False	boolean value	Use durable queues in AMQP.
direct_mandatory_flag = True	boolean value	(DEPRECATED) Enable/Disable the RabbitMQ mandatory flag for direct send. The direct send is used as reply, so the MessageUndeliverable exception is raised in case the client queue does not exist. MessageUndeliverable exception will be used to loop for a timeout to let a chance to sender to recover. This flag is deprecated and it will not be possible to deactivate this functionality anymore
enable_cancel_on_failover = False	boolean value	Enable x-cancel-on-ha-failover flag so that rabbitmq server will cancel and notify consumers when queue is down
heartbeat_in_pthread = True	boolean value	Run the health check heartbeat thread through a native python thread by default. If this option is equal to False then the health check heartbeat will inherit the execution model from the parent process. For example if the parent process has monkey patched the stdlib by using eventlet/greenlet then the heartbeat will be run through a green thread.
heartbeat_rate = 2	integer value	How often times during the heartbeat_timeout_threshold we check the heartbeat.

Configuration option = Default value	Type	Description
heartbeat_timeout_thresh old = 60	integer value	Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables heartbeat).
kombu_compression = None	string value	EXPERIMENTAL: Possible values are: gzip, bz2. If not set compression will not be used. This option may not be available in future versions.
kombu_failover_strategy = round-robin	string value	Determines how the next RabbitMQ node is chosen in case the one we are currently connected to becomes unavailable. Takes effect only if more than one RabbitMQ node is provided in config.
kombu_missing_consumer_retry_timeout = 60	integer value	How long to wait a missing client before abandoning to send it its replies. This value should not be longer than <code>rpc_response_timeout</code> .
kombu_reconnect_delay = 1.0	floating point value	How long to wait before reconnecting in response to an AMQP consumer cancel notification.
rabbit_ha_queues = False	boolean value	Try to use HA queues in RabbitMQ (<code>x-ha-policy: all</code>). If you change this option, you must wipe the RabbitMQ database. In RabbitMQ 3.0, queue mirroring is no longer controlled by the <code>x-ha-policy</code> argument when declaring a queue. If you just want to make sure that all queues (except those with auto-generated names) are mirrored across all nodes, run: <code>"rabbitmqctl set_policy HA ^(?!amq\.).* {\"ha-mode\": \"all\"}"</code>
rabbit_interval_max = 30	integer value	Maximum interval of RabbitMQ connection retries. Default is 30 seconds.
rabbit_login_method = AMQPLAIN	string value	The RabbitMQ login method.
rabbit_qos_prefetch_count = 0	integer value	Specifies the number of messages to prefetch. Setting to zero allows unlimited messages.
rabbit_retry_backoff = 2	integer value	How long to backoff for between retries when connecting to RabbitMQ.
rabbit_retry_interval = 1	integer value	How frequently to retry connecting with RabbitMQ.

Configuration option = Default value	Type	Description
rabbit_transient_queues_ttl = 1800	integer value	Positive integer representing duration in seconds for queue TTL (x-expire). Queues which are unused for the duration of the TTL are automatically deleted. The parameter affects only reply and fanout queues.
ssl = False	boolean value	Connect over SSL.
<code>`ssl_ca_file = `</code>	string value	SSL certification authority file (valid only if SSL enabled).
<code>`ssl_cert_file = `</code>	string value	SSL cert file (valid only if SSL enabled).
<code>`ssl_key_file = `</code>	string value	SSL key file (valid only if SSL enabled).
<code>`ssl_version = `</code>	string value	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.

4.1.38. oslo_middleware

The following table outlines the options available under the **[oslo_middleware]** group in the `/etc/heat/heat.conf` file.

Table 4.37. oslo_middleware

Configuration option = Default value	Type	Description
enable_proxy_headers_parsing = False	boolean value	Whether the application is behind a proxy or not. This determines if the middleware should parse the headers or not.
max_request_body_size = 114688	integer value	The maximum body size for each request, in bytes.
secure_proxy_ssl_header = X-Forwarded-Proto	string value	The HTTP Header that will be used to determine what the original request protocol scheme was, even if it was hidden by a SSL termination proxy.

4.1.39. oslo_policy

The following table outlines the options available under the **[oslo_policy]** group in the `/etc/heat/heat.conf` file.

Table 4.38. oslo_policy

Configuration option = Default value	Type	Description
enforce_new_defaults = False	boolean value	This option controls whether or not to use old deprecated defaults when evaluating policies. If True , the old deprecated defaults are not going to be evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be disallowed. It is encouraged to enable this flag along with the enforce_scope flag so that you can get the benefits of new defaults and scope_type together
enforce_scope = False	boolean value	This option controls whether or not to enforce scope when evaluating policies. If True , the scope of the token used in the request is compared to the scope_types of the policy being enforced. If the scopes do not match, an InvalidScope exception will be raised. If False , a message will be logged informing operators that policies are being invoked with mismatching scope.
policy_default_rule = default	string value	Default rule. Enforced when a requested rule is not found.
policy_dirs = ['policy.d']	multi valued	Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the <code>config_dir</code> option, or absolute paths. The file defined by <code>policy_file</code> must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = policy.json	string value	The relative or absolute path of a file that maps roles to permissions for a given service. Relative paths must be specified in relation to the configuration file setting this option.
remote_content_type = application/x-www-form-urlencoded	string value	Content Type to send and receive data for REST based policy check
remote_ssl_ca_cert_file = None	string value	Absolute path to ca cert file for REST based policy check
remote_ssl_client_cert_file = None	string value	Absolute path to client cert for REST based policy check
remote_ssl_client_key_file = None	string value	Absolute path client key file REST based policy check

Configuration option = Default value	Type	Description
remote_ssl_verify_server _cert = False	boolean value	server identity verification for REST based policy check

4.1.40. paste_deploy

The following table outlines the options available under the **[paste_deploy]** group in the **/etc/heat/heat.conf** file.

Table 4.39. paste_deploy

Configuration option = Default value	Type	Description
api_paste_config = api-paste.ini	string value	The API paste config file to use.
flavor = None	string value	The flavor to use.

4.1.41. profiler

The following table outlines the options available under the **[profiler]** group in the **/etc/heat/heat.conf** file.

Table 4.40. profiler

Configuration option = Default value	Type	Description
connection_string = messaging://	string value	Connection string for a notifier backend. Default value is messaging:// which sets the notifier to oslo_messaging. Examples of possible values: ● messaging:// - use oslo_messaging driver for sending spans. ● redis://127.0.0.1:6379 - use redis driver for sending spans. ● mongodb://127.0.0.1:27017 - use mongodb driver for sending spans. ● elasticsearch://127.0.0.1:9200 - use elasticsearch driver for sending spans. ● jaeger://127.0.0.1:6831 - use jaeger tracing as driver for sending spans.

Configuration option = Default value	Type	Description
enabled = False	boolean value	Enable the profiling for all services on this node. Default value is False (fully disable the profiling feature). Possible values: • True: Enables the feature • False: Disables the feature. The profiling cannot be started via this project operations. If the profiling is triggered by another project, this project part will be empty.
es_doc_type = notification	string value	Document type for notification indexing in elasticsearch.
es_scroll_size = 10000	integer value	Elasticsearch splits large requests in batches. This parameter defines maximum size of each batch (for example: es_scroll_size=10000).
es_scroll_time = 2m	string value	This parameter is a time value parameter (for example: es_scroll_time=2m), indicating for how long the nodes that participate in the search will maintain relevant resources in order to continue and support it.
filter_error_trace = False	boolean value	Enable filter traces that contain error/exception to a separated place. Default value is set to False. Possible values: • True: Enable filter traces that contain error/exception. • False: Disable the filter.

Configuration option = Default value	Type	Description
hmac_keys = SECRET_KEY	string value	Secret key(s) to use for encrypting context data for performance profiling. This string value should have the following format: <key1>[,<key2>,...<keyn>], where each key is some random string. A user who triggers the profiling via the REST API has to set one of these keys in the headers of the REST API call to include profiling results of this node for this particular project. Both "enabled" flag and "hmac_keys" config options should be set to enable profiling. Also, to generate correct profiling information across all services at least one key needs to be consistent between OpenStack projects. This ensures it can be used from client side to generate the trace, containing information from all possible resources.
sentinel_service_name = mymaster	string value	Redis sentinel uses a service name to identify a master redis service. This parameter defines the name (for example: sentinal_service_name=mymaster).
socket_timeout = 0.1	floating point value	Redis sentinel provides a timeout option on the connections. This parameter defines that timeout (for example: socket_timeout=0.1).
trace_sqlalchemy = False	boolean value	Enable SQL requests profiling in services. Default value is False (SQL requests won't be traced). Possible values: • True: Enables SQL requests profiling. Each SQL query will be part of the trace and can be analyzed by how much time was spent for that. • False: Disables SQL requests profiling. The spent time is only shown on a higher level of operations. Single SQL queries cannot be analyzed this way.

4.1.42. revision

The following table outlines the options available under the **[revision]** group in the `/etc/heat/heat.conf` file.

Table 4.41. revision

Configuration option = Default value	Type	Description
heat_revision = unknown	string value	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.

4.1.43. ssl

The following table outlines the options available under the **[ssl]** group in the `/etc/heat/heat.conf` file.

Table 4.42. ssl

Configuration option = Default value	Type	Description
ca_file = None	string value	CA certificate file to use to verify connecting clients.
cert_file = None	string value	Certificate file to use when starting the server securely.
ciphers = None	string value	Sets the list of available ciphers. value should be a string in the OpenSSL cipher list format.
key_file = None	string value	Private key file to use when starting the server securely.
version = None	string value	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.

4.1.44. trustee

The following table outlines the options available under the **[trustee]** group in the `/etc/heat/heat.conf` file.

Table 4.43. trustee

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_section = None	string value	Config Section from which to load plugin specific options

Configuration option = Default value	Type	Description
auth_type = None	string value	Authentication type to load
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
system-scope = None	string value	Scope for system operations
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User id
username = None	string value	Username

4.1.45. volumes

The following table outlines the options available under the **[volumes]** group in the `/etc/heat/heat.conf` file.

Table 4.44. volumes

Configuration option = Default value	Type	Description
backups_enabled = True	boolean value	Indicate if cinder-backup service is enabled. This is a temporary workaround until cinder-backup service becomes discoverable, see LP#1334856.

CHAPTER 5. IRONIC

The following chapter contains information about the configuration options in the **ironic** service.

5.1. IRONIC.CONF

This section contains options for the **/etc/ironic/ironic.conf** file.

5.1.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the **/etc/ironic/ironic.conf** file.

Configuration option = Default value	Type	Description
auth_strategy = keystone	string value	Authentication strategy used by ironic-api. "noauth" should not be used in a production environment because all authentication will be disabled.
backdoor_port = None	string value	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.
backdoor_socket = None	string value	Enable eventlet backdoor, using the provided path as a unix socket that can receive connections. This option is mutually exclusive with <i>backdoor_port</i> in that only one should be provided. If both are provided then the existence of this option overrides the usage of that option. Inside the path {pid} will be replaced with the PID of the current process.
bindir = \$pybasedir/bin	string value	Directory where ironic binaries are installed.
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
control_exchange = openstack	string value	The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.

Configuration option = Default value	Type	Description
debug_tracebacks_in_api = False	boolean value	Return server tracebacks in the API response for any error responses. WARNING: this is insecure and should not be used in a production environment.
default_bios_interface = None	string value	Default bios interface to be used for nodes that do not have bios_interface field set. A complete list of bios interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.bios" endpoint.
default_boot_interface = None	string value	Default boot interface to be used for nodes that do not have boot_interface field set. A complete list of boot interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.boot" endpoint.
default_console_interface = None	string value	Default console interface to be used for nodes that do not have console_interface field set. A complete list of console interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.console" endpoint.
default_deploy_interface = None	string value	Default deploy interface to be used for nodes that do not have deploy_interface field set. A complete list of deploy interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.deploy" endpoint.
default_inspect_interface = None	string value	Default inspect interface to be used for nodes that do not have inspect_interface field set. A complete list of inspect interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.inspect" endpoint.

Configuration option = Default value	Type	Description
default_log_levels = ['amqp=WARNING', 'amqpplib=WARNING', 'qpid.messaging=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'sqlalchemy=WARNING', 'stevedore=INFO', 'eventlet.wsgi.server=INFO', 'iso8601=WARNING', 'requests=WARNING', 'glanceclient=WARNING', 'urllib3.connectionpool=WARNING', 'keystonemiddleware.auth_token=INFO', 'keystoneauth.session=INFO', 'openstack=WARNING']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
default_management_interface = None	string value	Default management interface to be used for nodes that do not have management_interface field set. A complete list of management interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.management" endpoint.
default_network_interface = None	string value	Default network interface to be used for nodes that do not have network_interface field set. A complete list of network interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.network" endpoint.
default_portgroup_mode = active-backup	string value	Default mode for portgroups. Allowed values can be found in the linux kernel documentation on bonding: https://www.kernel.org/doc/Documentation/networking/bonding.txt .
default_power_interface = None	string value	Default power interface to be used for nodes that do not have power_interface field set. A complete list of power interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.power" endpoint.
default_raid_interface = None	string value	Default raid interface to be used for nodes that do not have raid_interface field set. A complete list of raid interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.raid" endpoint.

Configuration option = Default value	Type	Description
default_rescue_interface = None	string value	Default rescue interface to be used for nodes that do not have rescue_interface field set. A complete list of rescue interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.rescue" endpoint.
default_resource_class = None	string value	Resource class to use for new nodes when no resource class is provided in the creation request.
default_storage_interface = noop	string value	Default storage interface to be used for nodes that do not have storage_interface field set. A complete list of storage interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.storage" endpoint.
default_vendor_interface = None	string value	Default vendor interface to be used for nodes that do not have vendor_interface field set. A complete list of vendor interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.vendor" endpoint.
enabled_bios_interfaces = ['no-bios']	list value	Specify the list of bios interfaces to load during service initialization. Missing bios interfaces, or bios interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one bios interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented bios interfaces. A complete list of bios interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.bios" endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled bios interfaces on every ironic-conductor service.

Configuration option = Default value	Type	Description
enabled_boot_interfaces = ['pxe']	list value	Specify the list of boot interfaces to load during service initialization. Missing boot interfaces, or boot interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one boot interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented boot interfaces. A complete list of boot interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.boot" entrypoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled boot interfaces on every ironic-conductor service.
enabled_console_interfaces = ['no-console']	list value	Specify the list of console interfaces to load during service initialization. Missing console interfaces, or console interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one console interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented console interfaces. A complete list of console interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.console" entrypoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled console interfaces on every ironic-conductor service.
enabled_deploy_interfaces = ['direct']	list value	Specify the list of deploy interfaces to load during service initialization. Missing deploy interfaces, or deploy interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one deploy interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented deploy interfaces. A complete list of deploy interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.deploy" entrypoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled deploy interfaces on every ironic-conductor service.

Configuration option = Default value	Type	Description
enabled_hardware_types = ['ipmi']	list value	Specify the list of hardware types to load during service initialization. Missing hardware types, or hardware types which fail to initialize, will prevent the conductor service from starting. This option defaults to a recommended set of production-oriented hardware types. A complete list of hardware types present on your system may be found by enumerating the "ironic.hardware.types" endpoint.
enabled_inspect_interfaces = ['no-inspect']	list value	Specify the list of inspect interfaces to load during service initialization. Missing inspect interfaces, or inspect interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one inspect interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented inspect interfaces. A complete list of inspect interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.inspect" endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled inspect interfaces on every ironic-conductor service.
enabled_management_interfaces = ['ipmitool']	list value	Specify the list of management interfaces to load during service initialization. Missing management interfaces, or management interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one management interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented management interfaces. A complete list of management interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.management" endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled management interfaces on every ironic-conductor service.

Configuration option = Default value	Type	Description
enabled_network_interfaces = ['flat', 'noop']	list value	Specify the list of network interfaces to load during service initialization. Missing network interfaces, or network interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one network interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented network interfaces. A complete list of network interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.network" endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled network interfaces on every ironic-conductor service.
enabled_power_interfaces = ['ipmitool']	list value	Specify the list of power interfaces to load during service initialization. Missing power interfaces, or power interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one power interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented power interfaces. A complete list of power interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.power" endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled power interfaces on every ironic-conductor service.
enabled_raid_interfaces = ['agent', 'no-raid']	list value	Specify the list of raid interfaces to load during service initialization. Missing raid interfaces, or raid interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one raid interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented raid interfaces. A complete list of raid interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.raid" endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled raid interfaces on every ironic-conductor service.

Configuration option = Default value	Type	Description
enabled_rescue_interfaces = ['no-rescue']	list value	Specify the list of rescue interfaces to load during service initialization. Missing rescue interfaces, or rescue interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one rescue interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented rescue interfaces. A complete list of rescue interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.rescue" endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled rescue interfaces on every ironic-conductor service.
enabled_storage_interfaces = ['cinder', 'noop']	list value	Specify the list of storage interfaces to load during service initialization. Missing storage interfaces, or storage interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one storage interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented storage interfaces. A complete list of storage interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.storage" endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled storage interfaces on every ironic-conductor service.
enabled_vendor_interfaces = ['ipmitool', 'no-vendor']	list value	Specify the list of vendor interfaces to load during service initialization. Missing vendor interfaces, or vendor interfaces which fail to initialize, will prevent the ironic-conductor service from starting. At least one vendor interface that is supported by each enabled hardware type must be enabled here, or the ironic-conductor service will not start. Must not be an empty list. The default value is a recommended set of production-oriented vendor interfaces. A complete list of vendor interfaces present on your system may be found by enumerating the "ironic.hardware.interfaces.vendor" endpoint. When setting this value, please make sure that every enabled hardware type will have the same set of enabled vendor interfaces on every ironic-conductor service.

Configuration option = Default value	Type	Description
esp_image = None	string value	Path to EFI System Partition image file. This file is recommended for creating UEFI bootable ISO images efficiently. ESP image should contain a FAT12/16/32-formatted file system holding EFI boot loaders (e.g. GRUB2) for each hardware architecture ironic needs to boot. This option is only used when neither ESP nor ISO deploy image is configured to the node being deployed in which case ironic will attempt to fetch ESP image from the configured location or extract ESP image from UEFI-bootable deploy ISO image.
executor_thread_pool_size = 64	integer value	Size of executor thread pool when executor is threading or eventlet.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
force_raw_images = True	boolean value	If True, convert backing images to "raw" disk image format.
graceful_shutdown_timeout = 60	integer value	Specify a timeout after which a gracefully shutdown server will exit. Zero value means endless wait.
grub_config_path = /boot/grub/grub.cfg	string value	GRUB2 configuration file location on the UEFI ISO images produced by ironic. The default value is usually incorrect and should not be relied on. If you use a GRUB2 image from a certain distribution, use a distribution-specific path here, e.g. EFI/ubuntu/grub.cfg
grub_config_template = \$pybasedir/common/grub_conf.template	string value	Template file for grub configuration file.
hash_partition_exponent = 5	integer value	Exponent to determine number of hash partitions to use when distributing load across conductors. Larger values will result in more even distribution of load and less load when rebalancing the ring, but more memory usage. Number of partitions per conductor is ($2^{\text{hash\_partition\_exponent}}$). This determines the granularity of rebalancing: given 10 hosts, and an exponent of the 2, there are 40 partitions in the ring. A few thousand partitions should make rebalancing smooth in most cases. The default is suitable for up to a few hundred conductors. Configuring for too many partitions has a negative impact on CPU usage.

Configuration option = Default value	Type	Description
hash_ring_algorithm = md5	string value	Hash function to use when building the hash ring. If running on a FIPS system, do not use md5. WARNING: all ironic services in a cluster MUST use the same algorithm at all times. Changing the algorithm requires an offline update.
hash_ring_reset_interval = 15	integer value	Time (in seconds) after which the hash ring is considered outdated and is refreshed on the next access.
host = <based on operating system>	string value	Name of this node. This can be an opaque identifier. It is not necessarily a hostname, FQDN, or IP address. However, the node name must be valid within an AMQP key, and if using ZeroMQ (will be removed in the Stein release), a valid hostname, FQDN, or IP address.
http_basic_auth_user_file = /etc/ironic/htpasswd	string value	Path to Apache format user authentication file used when auth_strategy=http_basic
<code>`instance_format = [instance: %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.
<code>`instance_uuid_format = [instance: %(uuid)s] `</code>	string value	The format for an instance UUID that is passed with the log message.
isolinux_bin = /usr/lib/syslinux/isolinux.bin	string value	Path to isolinux binary file.
isolinux_config_template = \$pybasedir/common/isolinux_config.template	string value	Template file for isolinux configuration file.
ldlinux_c32 = None	string value	Path to ldlinux.c32 file. This file is required for syslinux 5.0 or later. If not specified, the file is looked for in "/usr/lib/syslinux/modules/bios/ldlinux.c32" and "/usr/share/syslinux/ldlinux.c32".
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).

Configuration option = Default value	Type	Description
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code> . This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code> . This option is ignored if <code>log_config_append</code> is set.
log_in_db_max_size = 4096	integer value	Max number of characters of any node <code>last_error/maintenance_reason</code> pushed to database.
log_options = True	boolean value	Enables or disables logging values of all registered options when starting a service (at <code>DEBUG</code> level).
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless <code>log_rotation_type</code> is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	string value	Format string to use for log messages with context. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is <code>DEBUG</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_default_format_s tring = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s	string value	Format string to use for log messages when context is undefined. Used by <code>oslo_log.formatters.ContextFormatter</code>

Configuration option = Default value	Type	Description
logging_exception_prefix = <code>%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s</code>	string value	Prefix each line of exception output with this format. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_user_identity_format = <code>%(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s</code>	string value	Defines the format string for <code>%(user_identity)s</code> that is used in <code>logging_context_format_string</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
minimum_memory_wait_retries = 6	integer value	Number of retries to hold onto the worker before failing or returning the thread to the pool if the conductor can automatically retry.
minimum_memory_wait_time = 15	integer value	Seconds to wait between retries for free memory before launching the process. This, combined with memory_wait_retries allows the conductor to determine how long we should attempt to directly retry.
minimum_memory_warning_only = False	boolean value	Setting to govern if Ironic should only warn instead of attempting to hold back the request in order to prevent the exhaustion of system memory.
minimum_required_memory = 1024	integer value	Minimum memory in MiB for the system to have available prior to starting a memory intensive process on the conductor.
my_ip = <based on operating system>	string value	IPv4 address of this host. If unset, will determine the IP programmatically. If unable to do so, will use "127.0.0.1". NOTE: This field does accept an IPv6 address as an override for templates and URLs, however it is recommended that [DEFAULT]my_ip_v6 is used along with DNS names for service URLs for dual-stack environments.
my_ip_v6 = None	string value	IP address of this host using IPv6. This value must be supplied via the configuration and cannot be adequately programmatically determined like the [DEFAULT]my_ip parameter for IPv4.

Configuration option = Default value	Type	Description
notification_level = None	string value	Specifies the minimum level for which to send notifications. If not set, no notifications will be sent. The default is for this option to be unset.
parallel_image_downloads = False	boolean value	Run image downloads and raw format conversions in parallel.
pecan_debug = False	boolean value	Enable pecan debug mode. WARNING: this is insecure and should not be used in a production environment.
pin_release_version = None	string value	Used for rolling upgrades. Setting this option downgrades (or pins) the Bare Metal API, the internal ironic RPC communication, and the database objects to their respective versions, so they are compatible with older services. When doing a rolling upgrade from version N to version N+1, set (to pin) this to N. To unpin (default), leave it unset and the latest versions will be used.
publish_errors = False	boolean value	Enables or disables publication of error events.
pybasedir = /usr/lib/python3.9/site-packages/ironic	string value	Directory where the ironic python module is installed.
rate_limit_burst = 0	integer value	Maximum number of logged messages per rate_limit_interval.
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
raw_image_growth_factor = 2.0	floating point value	The scale factor used for estimating the size of a raw image converted from compact image formats such as QCOW2. Default is 2.0, must be greater than 1.0.
rootwrap_config = /etc/ironic/rootwrap.conf	string value	Path to the rootwrap configuration file to use for running commands as root.
rpc_conn_pool_size = 30	integer value	Size of RPC connection pool.

Configuration option = Default value	Type	Description
rpc_ping_enabled = False	boolean value	Add an endpoint to answer to ping calls. Endpoint is named oslo_rpc_server_ping
rpc_response_timeout = 60	integer value	Seconds to wait for a response from a call.
rpc_transport = oslo	string value	Which RPC transport implementation to use between conductor and API services
run_external_periodic_tasks = True	boolean value	Some periodic tasks can be run in a separate process. Should we run them here?
state_path = \$pybasedir	string value	Top-level directory for maintaining ironic's state.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
tempdir = /tmp	string value	Temporary working directory, default is Python temp dir.
transport_url = rabbit://	string value	The network address and optional user credentials for connecting to the messaging backend, in URL format. The expected format is: <pre>driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query</pre> Example: rabbit://rabbitmq:password@127.0.0.1:5672// For full details on the fields in the URL see the documentation of oslo_messaging.TransportURL at https://docs.openstack.org/oslo.messaging/latest/reference/transport.html
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
versioned_notifications_topics = ['ironic_versioned_notifications']	list value	Specifies the topics for the versioned notifications issued by Ironic. The default value is fine for most deployments and rarely needs to be changed. However, if you have a third-party service that consumes versioned notifications, it might be worth getting a topic for that service. Ironic will send a message containing a versioned notification payload to each topic queue in this list. The list of versioned notifications is visible in https://docs.openstack.org/ironic/latest/admin/notifications.html
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.
webserver_connection_timeout = 60	integer value	Connection timeout when accessing remote web servers with images.
webserver_verify_ca = True	string value	CA certificates to be used for certificate verification. This can be either a Boolean value or a path to a CA_BUNDLE file.If set to True, the certificates present in the standard path are used to verify the host certificates.If set to False, the conductor will ignore verifying the SSL certificate presented by the host.If it's a path, conductor uses the specified certificate for SSL verification. If the path does not exist, the behavior is same as when this value is set to True i.e the certificates present in the standard path are used for SSL verification.Defaults to True.

5.1.2. agent

The following table outlines the options available under the **[agent]** group in the `/etc/ironic/ironic.conf` file.

Table 5.1. agent

Configuration option = Default value	Type	Description
agent_api_version = v1	string value	API version to use for communicating with the ramdisk agent.
api_ca_file = None	string value	Path to the TLS CA that is used to start the bare metal API. In some boot methods this file can be passed to the ramdisk.
certificates_path = /var/lib/ironic/certificates	string value	Path to store auto-generated TLS certificates used to validate connections to the ramdisk.
command_timeout = 60	integer value	Timeout (in seconds) for IPA commands.
command_wait_attempts = 100	integer value	Number of attempts to check for asynchronous commands completion before timing out.
command_wait_interval = 6	integer value	Number of seconds to wait for between checks for asynchronous commands completion.
deploy_logs_collect = on_failure	string value	Whether Ironic should collect the deployment logs on deployment failure (on_failure), always or never.
deploy_logs_local_path = /var/log/ironic/deploy	string value	The path to the directory where the logs should be stored, used when the deploy_logs_storage_backend is configured to "local".
deploy_logs_storage_backend = local	string value	The name of the storage backend where the logs will be stored.
deploy_logs_swift_container = ironic_deploy_logs_container	string value	The name of the Swift container to store the logs, used when the deploy_logs_storage_backend is configured to "swift".
deploy_logs_swift_days_to_expire = 30	integer value	Number of days before a log object is marked as expired in Swift. If None, the logs will be kept forever or until manually deleted. Used when the deploy_logs_storage_backend is configured to "swift".

Configuration option = Default value	Type	Description
image_download_source = http	string value	Specifies whether direct deploy interface should try to use the image source directly or if ironic should cache the image on the conductor and serve it from ironic's own http server.
manage_agent_boot = True	boolean value	Whether Ironic will manage booting of the agent ramdisk. If set to False, you will need to configure your mechanism to allow booting the agent ramdisk.
max_command_attempts = 3	integer value	This is the maximum number of attempts that will be done for IPA commands that fails due to network problems.
memory_consumed_by_agent = 0	integer value	The memory size in MiB consumed by agent when it is booted on a bare metal node. This is used for checking if the image can be downloaded and deployed on the bare metal node after booting agent ramdisk. This may be set according to the memory consumed by the agent ramdisk image.
neutron_agent_max_attempts = 100	integer value	Max number of attempts to validate a Neutron agent status before raising network error for a dead agent.
neutron_agent_poll_interval = 2	integer value	The number of seconds Neutron agent will wait between polling for device changes. This value should be the same as CONF.AGENT.polling_interval in Neutron configuration.
neutron_agent_status_retry_interval = 10	integer value	Wait time in seconds between attempts for validating Neutron agent status.
post_deploy_get_power_state_retries = 6	integer value	Number of times to retry getting power state to check if bare metal node has been powered off after a soft power off.
post_deploy_get_power_state_retry_interval = 5	integer value	Amount of time (in seconds) to wait between polling power state after trigger soft poweroff.
require_tls = False	boolean value	If set to True, callback URLs without https:// will be rejected by the conductor.

Configuration option = Default value	Type	Description
stream_raw_images = True	boolean value	Whether the agent ramdisk should stream raw images directly onto the disk or not. By streaming raw images directly onto the disk the agent ramdisk will not spend time copying the image to a tmpfs partition (therefore consuming less memory) prior to writing it to the disk. Unless the disk where the image will be copied to is really slow, this option should be set to True. Defaults to True.
verify_ca = True	string value	Path to the TLS CA to validate connection to the ramdisk. Set to True to use the system default CA storage. Set to False to disable validation. Ignored when automatic TLS setup is used.

5.1.3. anaconda

The following table outlines the options available under the **[anaconda]** group in the **/etc/ironic/ironic.conf** file.

Table 5.2. anaconda

Configuration option = Default value	Type	Description
default_ks_template = \$pybasedir/drivers/modules/ks.cfg.template	string value	kickstart template to use when no kickstart template is specified in the instance_info or the glance OS image.

5.1.4. ansible

The following table outlines the options available under the **[ansible]** group in the **/etc/ironic/ironic.conf** file.

Table 5.3. ansible

Configuration option = Default value	Type	Description
ansible_extra_args = None	string value	Extra arguments to pass on every invocation of Ansible.
ansible_playbook_script = ansible-playbook	string value	Path to "ansible-playbook" script. Default will search the \$PATH configured for user running ironic-conductor process. Provide the full path when ansible-playbook is not in \$PATH or installed in not default location.

Configuration option = Default value	Type	Description
config_file_path = \$pybasedir/drivers/modules/ansible/playbooks/ansible.cfg	string value	Path to ansible configuration file. If set to empty, system default will be used.
default_clean_playbook = clean.yaml	string value	Path (relative to \$playbooks_path or absolute) to the default playbook used for node cleaning. It may be overridden by per-node <i>ansible_clean_playbook</i> option in node's <i>driver_info</i> field.
default_clean_steps_config = clean_steps.yaml	string value	Path (relative to \$playbooks_path or absolute) to the default auxiliary cleaning steps file used during the node cleaning. It may be overridden by per-node <i>ansible_clean_steps_config</i> option in node's <i>driver_info</i> field.
default_deploy_playbook = deploy.yaml	string value	Path (relative to \$playbooks_path or absolute) to the default playbook used for deployment. It may be overridden by per-node <i>ansible_deploy_playbook</i> option in node's <i>driver_info</i> field.
default_key_file = None	string value	Absolute path to the private SSH key file to use by Ansible by default when connecting to the ramdisk over SSH. Default is to use default SSH keys configured for the user running the ironic-conductor service. Private keys with password must be pre-loaded into <i>ssh-agent</i> . It may be overridden by per-node <i>ansible_key_file</i> option in node's <i>driver_info</i> field.
default_python_interpreter = None	string value	Absolute path to the python interpreter on the managed machines. It may be overridden by per-node <i>ansible_python_interpreter</i> option in node's <i>driver_info</i> field. By default, ansible uses <code>/usr/bin/python</code>
default_shutdown_playbook = shutdown.yaml	string value	Path (relative to \$playbooks_path or absolute) to the default playbook used for graceful in-band shutdown of the node. It may be overridden by per-node <i>ansible_shutdown_playbook</i> option in node's <i>driver_info</i> field.
default_username = ansible	string value	Name of the user to use for Ansible when connecting to the ramdisk over SSH. It may be overridden by per-node <i>ansible_username</i> option in node's <i>driver_info</i> field.

Configuration option = Default value	Type	Description
extra_memory = 10	integer value	Extra amount of memory in MiB expected to be consumed by Ansible-related processes on the node. Affects decision whether image will fit into RAM.
image_store_cafile = None	string value	Specific CA bundle to use for validating SSL connections to the image store. If not specified, CA available in the ramdisk will be used. Is not used by default playbooks included with the driver. Suitable for environments that use self-signed certificates.
image_store_certfile = None	string value	Client cert to use for SSL connections to image store. Is not used by default playbooks included with the driver.
image_store_insecure = False	boolean value	Skip verifying SSL connections to the image store when downloading the image. Setting it to "True" is only recommended for testing environments that use self-signed certificates.
image_store_keyfile = None	string value	Client key to use for SSL connections to image store. Is not used by default playbooks included with the driver.
playbooks_path = \$pybasedir/drivers/modules/ansible/playbooks	string value	Path to directory with playbooks, roles and local inventory.
post_deploy_get_power_state_retries = 6	integer value	Number of times to retry getting power state to check if bare metal node has been powered off after a soft power off. Value of 0 means do not retry on failure.
post_deploy_get_power_state_retry_interval = 5	integer value	Amount of time (in seconds) to wait between polling power state after trigger soft poweroff.
verbosity = None	integer value	Set ansible verbosity level requested when invoking "ansible-playbook" command. 4 includes detailed SSH session logging. Default is 4 when global debug is enabled and 0 otherwise.

5.1.5. api

The following table outlines the options available under the **[api]** group in the **/etc/ironic/ironic.conf** file.

Table 5.4. api

Configuration option = Default value	Type	Description
api_workers = None	integer value	Number of workers for OpenStack Ironic API service. The default is equal to the number of CPUs available, but not more than 4. One worker is used if the CPU number cannot be detected.
enable_ssl_api = False	boolean value	Enable the integrated stand-alone API to service requests via HTTPS instead of HTTP. If there is a front-end service performing HTTPS offloading from the service, this option should be False; note, you will want to enable proxy headers parsing with [oslo_middleware]enable_proxy_headers_parsing option or configure [api]public_endpoint option to set URLs in responses to the SSL terminated one.
host_ip = 0.0.0.0	host address value	The IP address or hostname on which ironic-api listens.
max_limit = 1000	integer value	The maximum number of items returned in a single response from a collection resource.
network_data_schema = \$pybasedir/api/controllers/v1/network-data-schema.json	string value	Schema for network data used by this deployment.
port = 6385	port value	The TCP port on which ironic-api listens.
public_endpoint = None	string value	Public URL to use when building the links to the API resources (for example, "https://ironic.rocks:6384"). If None the links will be built using the request's host URL. If the API is operating behind a proxy, you will want to change this to represent the proxy's URL. Defaults to None. Ignored when proxy headers parsing is enabled via [oslo_middleware]enable_proxy_headers_parsing option.
ramdisk_heartbeat_timeout = 300	integer value	Maximum interval (in seconds) for agent heartbeats.
restrict_lookup = True	boolean value	Whether to restrict the lookup API to only nodes in certain states.

5.1.6. audit

The following table outlines the options available under the **[audit]** group in the **/etc/ironic/ironic.conf** file.

Table 5.5. audit

Configuration option = Default value	Type	Description
audit_map_file = /etc/ironic/api_audit_map. conf	string value	Path to audit map file for ironic-api service. Used only when API audit is enabled.
enabled = False	boolean value	Enable auditing of API requests (for ironic-api service).
ignore_req_list = `	string value	Comma separated list of Ironic REST API HTTP methods to be ignored during audit logging. For example: auditing will not be done on any GET or POST requests if this is set to "GET,POST". It is used only when API audit is enabled.

5.1.7. cinder

The following table outlines the options available under the **[cinder]** group in the **/etc/ironic/ironic.conf** file.

Table 5.6. cinder

Configuration option = Default value	Type	Description
action_retries = 3	integer value	Number of retries in the case of a failed action (currently only used when detaching volumes).
action_retry_interval = 5	integer value	Retry interval in seconds in the case of a failed action (only specific actions are retried).
auth-url = None	string value	Authentication URL
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.

Configuration option = Default value	Type	Description
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
max-version = None	string value	The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.
min-version = None	string value	The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is "latest".
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to

Configuration option = Default value	Type	Description
project-name = None	string value	Project name to scope to
region-name = None	string value	The default region_name for endpoint URL discovery.
retries = 3	integer value	Client retries in the case of a failed request connection.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = volumev3	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrievable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User id
username = None	string value	Username
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.

Configuration option = Default value	Type	Description
version = None	string value	Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

5.1.8. conductor

The following table outlines the options available under the **[conductor]** group in the **/etc/ironic/ironic.conf** file.

Table 5.7. conductor

Configuration option = Default value	Type	Description
allow_deleting_available_nodes = True	boolean value	Allow deleting nodes which are in state <i>available</i> . Defaults to True.
allow_provisioning_in_maintenance = True	boolean value	Whether to allow nodes to enter or undergo deploy or cleaning when in maintenance mode. If this option is set to False, and a node enters maintenance during deploy or cleaning, the process will be aborted after the next heartbeat. Automated cleaning or making a node available will also fail. If True (the default), the process will begin and will pause after the node starts heartbeating. Moving it from maintenance will make the process continue.
automated_clean = True	boolean value	Enables or disables automated cleaning. Automated cleaning is a configurable set of steps, such as erasing disk drives, that are performed on the node to ensure it is in a baseline state and ready to be deployed to. This is done after instance deletion as well as during the transition from a "manageable" to "available" state. When enabled, the particular steps performed to clean a node depend on which driver that node is managed by; see the individual 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.
bootloader = None	string value	Glance ID, http:// or file:// URL of the EFI system partition image containing EFI boot loader. This image will be used by ironic when building UEFI-bootable ISO out of kernel and ramdisk. Required for UEFI boot from partition images.

Configuration option = Default value	Type	Description
check_allocations_interval = 60	integer value	Interval between checks of orphaned allocations, in seconds. Set to 0 to disable checks.
check_provision_state_interval = 60	integer value	Interval between checks of provision timeouts, in seconds. Set to 0 to disable checks.
check_rescue_state_interval = 60	integer value	Interval (seconds) between checks of rescue timeouts.
clean_callback_timeout = 1800	integer value	Timeout (seconds) to wait for a callback from the ramdisk doing the cleaning. If the timeout is reached the node will be put in the "clean failed" provision state. Set to 0 to disable timeout.
clean_step_priority_override = {}	dict value	Priority to run automated clean steps for both in-band and out of band clean steps, provided in interface.step_name:priority format, e.g. deploy.erase_devices_metadata:123. The option can be specified multiple times to define priorities for multiple steps. If set to 0, this specific step will not run during cleaning. If unset for an inband clean step, will use the priority set in the ramdisk.
conductor_group =	string value	Name of the conductor group to join. Can be up to 255 characters and is case insensitive. This conductor will only manage nodes with a matching "conductor_group" field set on the node.
configdrive_swift_container = ironic_configdrive_container	string value	Name of the Swift container to store config drive data. Used when configdrive_use_object_store is True.
configdrive_swift_temp_url_duration = None	integer value	The timeout (in seconds) after which a configdrive temporary URL becomes invalid. Defaults to deploy_callback_timeout if it is set, otherwise to 1800 seconds. Used when configdrive_use_object_store is True.
deploy_callback_timeout = 1800	integer value	Timeout (seconds) to wait for a callback from a deploy ramdisk. Set to 0 to disable timeout.
deploy_kernel = None	string value	Glance ID, http:// or file:// URL of the kernel of the default deploy image.

Configuration option = Default value	Type	Description
deploy_ramdisk = None	string value	Glance ID, http:// or file:// URL of the initramfs of the default deploy image.
enable_mdns = False	boolean value	Whether to enable publishing the baremetal API endpoint via multicast DNS.
force_power_state_during_sync = True	boolean value	During sync_power_state, should the hardware power state be set to the state recorded in the database (True) or should the database be updated based on the hardware state (False).
heartbeat_interval = 10	integer value	Seconds between conductor heart beats.
heartbeat_timeout = 60	integer value	Maximum time (in seconds) since the last check-in of a conductor. A conductor is considered inactive when this time has been exceeded.
inspect_wait_timeout = 1800	integer value	Timeout (seconds) for waiting for node inspection. 0 - unlimited.
node_locked_retry_attempts = 3	integer value	Number of attempts to grab a node lock.
node_locked_retry_interval = 1	integer value	Seconds to sleep between node lock attempts.
periodic_max_workers = 8	integer value	Maximum number of worker threads that can be started simultaneously by a periodic task. Should be less than RPC thread pool size.
power_failure_recovery_interval = 300	integer value	Interval (in seconds) between checking the power state for nodes previously put into maintenance mode due to power synchronization failure. A node is automatically moved out of maintenance mode once its power state is retrieved successfully. Set to 0 to disable this check.
power_state_change_timeout = 60	integer value	Number of seconds to wait for power operations to complete, i.e., so that a baremetal node is in the desired power state. If timed out, the power operation is considered a failure.
power_state_sync_max_retries = 3	integer value	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

Configuration option = Default value	Type	Description
require_rescue_password_hashed = False	boolean value	Option to cause the conductor to not fallback to an un-hashed version of the rescue password, permitting rescue with older ironic-python-agent ramdisks.
rescue_callback_timeout = 1800	integer value	Timeout (seconds) to wait for a callback from the rescue ramdisk. If the timeout is reached the node will be put in the "rescue failed" provision state. Set to 0 to disable timeout.
rescue_kernel = None	string value	Glance ID, http:// or file:// URL of the kernel of the default rescue image.
rescue_password_hash_algorithm = sha256	string value	Password hash algorithm to be used for the rescue password.
rescue_ramdisk = None	string value	Glance ID, http:// or file:// URL of the initramfs of the default rescue image.
send_sensor_data = False	boolean value	Enable sending sensor data message via the notification bus
send_sensor_data_for_undeployed_nodes = False	boolean value	The default for sensor data collection is to only collect data for machines that are deployed, however operators may desire to know if there are failures in hardware that is not presently in use. When set to true, the conductor will collect sensor information from all nodes when sensor data collection is enabled via the send_sensor_data setting.
send_sensor_data_interval = 600	integer value	Seconds between conductor sending sensor data message to ceilometer via the notification bus.
send_sensor_data_types = ['ALL']	list value	List of comma separated meter types which need to be sent to Ceilometer. The default value, "ALL", is a special value meaning send all the sensor data.
send_sensor_data_wait_timeout = 300	integer value	The time in seconds to wait for send sensors data periodic task to be finished before allowing periodic call to happen again. Should be less than send_sensor_data_interval value.
send_sensor_data_workers = 4	integer value	The maximum number of workers that can be started simultaneously for send data from sensors periodic task.
soft_power_off_timeout = 600	integer value	Timeout (in seconds) of soft reboot and soft power off operation. This value always has to be positive.

Configuration option = Default value	Type	Description
sync_local_state_interval = 180	integer value	When conductors join or leave the cluster, existing conductors may need to update any persistent local state as nodes are moved around the cluster. This option controls how often, in seconds, each conductor will check for nodes that it should "take over". Set it to 0 (or a negative value) to disable the check entirely.
sync_power_state_interval = 60	integer value	Interval between syncing the node power state to the database, in seconds. Set to 0 to disable syncing.
sync_power_state_workers = 8	integer value	The maximum number of worker threads that can be started simultaneously to sync nodes power states from the periodic task.
workers_pool_size = 100	integer value	The size of the workers greenthread pool. Note that 2 threads will be reserved by the conductor itself for handling heart beats and periodic tasks. On top of that, sync_power_state_workers will take up to 7 green threads with the default value of 8.

5.1.9. console

The following table outlines the options available under the **[console]** group in the **/etc/ironic/ironic.conf** file.

Table 5.8. console

Configuration option = Default value	Type	Description
kill_timeout = 1	integer value	Time (in seconds) to wait for the console subprocess to exit before sending SIGKILL signal.
port_range = None	string value	A range of ports available to be used for the console proxy service running on the host of ironic conductor, in the form of <start>:<stop>. This option is used by both Shellinabox and Socat console
socat_address = \$my_ip	IP address value	IP address of Socat service running on the host of ironic conductor. Used only by Socat console.
subprocess_checking_interval = 1	integer value	Time interval (in seconds) for checking the status of console subprocess.
subprocess_timeout = 10	integer value	Time (in seconds) to wait for the console subprocess to start.

Configuration option = Default value	Type	Description
terminal = shellinaboxd	string value	Path to serial console terminal program. Used only by Shell In A Box console.
terminal_cert_dir = None	string value	Directory containing the terminal SSL cert (PEM) for serial console access. Used only by Shell In A Box console.
terminal_pid_dir = None	string value	Directory for holding terminal pid files. If not specified, the temporary directory will be used.
terminal_timeout = 600	integer value	Timeout (in seconds) for the terminal session to be closed on inactivity. Set to 0 to disable timeout. Used only by Socat console.

5.1.10. cors

The following table outlines the options available under the **[cors]** group in the `/etc/ironic/ironic.conf` file.

Table 5.9. cors

Configuration option = Default value	Type	Description
allow_credentials = True	boolean value	Indicate that the actual request can include user credentials
allow_headers = []	list value	Indicate which header field names may be used during the actual request.
allow_methods = ['OPTIONS', 'GET', 'HEAD', 'POST', 'PUT', 'DELETE', 'TRACE', 'PATCH']	list value	Indicate which methods can be used during the actual request.
allowed_origin = None	list value	Indicate whether this resource may be shared with the domain received in the requests "origin" header. Format: "<protocol>://<host>[:<port>]", no trailing slash. Example: https://horizon.example.com
expose_headers = []	list value	Indicate which headers are safe to expose to the API. Defaults to HTTP Simple Headers.
max_age = 3600	integer value	Maximum cache age of CORS preflight requests.

5.1.11. database

The following table outlines the options available under the **[database]** group in the `/etc/ironic/ironic.conf` file.

Table 5.10. database

Configuration option = Default value	Type	Description
backend = sqlalchemy	string value	The back end to use for the database.
connection = None	string value	The SQLAlchemy connection string to use to connect to the database.
connection_debug = 0	integer value	Verbosity of SQL debugging information: 0=None, 100=Everything.
<code>`connection_parameters = `</code>	string value	Optional URL parameters to append onto the connection URL at connect time; specify as param1=value1&param2=value2&...
connection_recycle_time = 3600	integer value	Connections which have been present in the connection pool longer than this number of seconds will be replaced with a new one the next time they are checked out from the pool.
connection_trace = False	boolean value	Add Python stack traces to SQL as comment strings.
db_inc_retry_interval = True	boolean value	If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = 20	integer value	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	integer value	If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = 1	integer value	Seconds between retries of a database transaction.
max_overflow = 50	integer value	If set, use this value for max_overflow with SQLAlchemy.
max_pool_size = 5	integer value	Maximum number of SQL connections to keep open in a pool. Setting a value of 0 indicates no limit.
max_retries = 10	integer value	Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.

Configuration option = Default value	Type	Description
mysql_enable_ndb = False	boolean value	If True, transparently enables support for handling MySQL Cluster (NDB).
mysql_engine = InnoDB	string value	MySQL engine to use.
mysql_sql_mode = TRADITIONAL	string value	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: <code>mysql_sql_mode=</code>
pool_timeout = None	integer value	If set, use this value for pool_timeout with SQLAlchemy.
retry_interval = 10	integer value	Interval between retries of opening a SQL connection.
slave_connection = None	string value	The SQLAlchemy connection string to use to connect to the slave database.
sqlite_synchronous = True	boolean value	If True, SQLite uses synchronous mode.
use_db_reconnect = False	boolean value	Enable the experimental use of database reconnect on connection lost.

5.1.12. deploy

The following table outlines the options available under the **[deploy]** group in the `/etc/ironic/ironic.conf` file.

Table 5.11. deploy

Configuration option = Default value	Type	Description
configdrive_use_object_store = False	boolean value	Whether to upload the config drive to object store. Set this option to True to store config drive in a swift endpoint.
continue_if_disk_secure_erase_fails = False	boolean value	Defines what to do if a secure erase operation (NVMe or ATA) fails during cleaning in the Ironic Python Agent. If False, the cleaning operation will fail and the node will be put in clean failed state. If True, shred will be invoked and cleaning will continue.

Configuration option = Default value	Type	Description
create_configuration_priority = None	integer value	Priority to run in-band clean step that creates RAID configuration from devices, via the IroniC Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 0 for the GenericHardwareManager). If set to 0, will not run during cleaning.
default_boot_mode = bios	string value	Default boot mode to use when no boot mode is requested in node's driver_info, capabilities or in the instance_info configuration. Currently the default boot mode is "bios", but it will be changed to "uefi" in the future. It is recommended to set an explicit value for this option. This option only has effect when management interface supports boot mode management
default_boot_option = local	string value	Default boot option to use when no boot option is requested in node's driver_info. Defaults to "local". Prior to the Ussuri release, the default was "netboot".
delete_configuration_priority = None	integer value	Priority to run in-band clean step that erases RAID configuration from devices, via the IroniC Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 0 for the GenericHardwareManager). If set to 0, will not run during cleaning.
disk_erasure_concurrency = 1	integer value	Defines the target pool size used by IroniC Python Agent ramdisk to erase disk devices. The number of threads created to erase disks will not exceed this value or the number of disks to be erased.
enable_ata_secure_erase = True	boolean value	Whether to support the use of ATA Secure Erase during the cleaning process. Defaults to True.
enable_nvme_secure_erase = True	boolean value	Whether to support the use of NVMe Secure Erase during the cleaning process. Currently nvme-cli format command is supported with user-data and crypto modes, depending on device capabilities. Defaults to True.
erase_devices_metadata_priority = None	integer value	Priority to run in-band clean step that erases metadata from devices, via the IroniC Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 99 for the GenericHardwareManager). If set to 0, will not run during cleaning.

Configuration option = Default value	Type	Description
erase_devices_priority = None	integer value	Priority to run in-band erase devices via the Ironic Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 10 for the GenericHardwareManager). If set to 0, will not run during cleaning.
erase_skip_read_only = False	boolean value	If the ironic-python-agent should skip read-only devices when running the "erase_devices" clean step where block devices are zeroed out. This requires ironic-python-agent 6.0.0 or greater. By default a read-only device will cause non-metadata based cleaning operations to fail due to the possible operational security risk of data being retained between deployments of the bare metal node.
external_callback_url = None	string value	Agent callback URL of the bare metal API for boot methods such as virtual media, where images could be served outside of the provisioning network. Defaults to the configuration from [service_catalog].
external_http_url = None	string value	URL of the ironic-conductor node's HTTP server for boot methods such as virtual media, where images could be served outside of the provisioning network. Does not apply when Swift is used. Defaults to http_url.
fast_track = False	boolean value	Whether to allow deployment agents to perform lookup, heartbeat operations during initial states of a machine lifecycle and by-pass the normal setup procedures for a ramdisk. This feature also enables power operations which are part of deployment processes to be bypassed if the ramdisk has performed a heartbeat operation using the fast_track_timeout setting.
fast_track_timeout = 300	integer value	Seconds for which the last heartbeat event is to be considered valid for the purpose of a fast track sequence. This setting should generally be less than the number of seconds for "Power-On Self Test" and typical ramdisk start-up. This value should not exceed the [api]ramdisk_heartbeat_timeout setting.
http_image_subdir = agent_images	string value	The name of subdirectory under ironic-conductor node's HTTP root path which is used to place instance images for the direct deploy interface, when local HTTP service is incorporated to provide instance image instead of swift tempurls.

Configuration option = Default value	Type	Description
http_root = /httpboot	string value	ironic-conductor node's HTTP root path.
http_url = None	string value	ironic-conductor node's HTTP server URL. Example: http://192.1.2.3:8080
power_off_after_deploy_failure = True	boolean value	Whether to power off a node after deploy failure. Defaults to True.
ramdisk_image_download_source = local	string value	Specifies whether a boot iso image should be served from its own original location using the image source url directly, or if ironic should cache the image on the conductor and serve it from ironic's own http server.
shred_final_overwrite_with_zeros = True	boolean value	Whether to write zeros to a node's block devices after writing random data. This will write zeros to the device even when <code>deploy.shred_random_overwrite_iterations</code> is 0. This option is only used if a device could not be ATA Secure Erased. Defaults to True.
shred_random_overwrite_iterations = 1	integer value	During shred, overwrite all block devices N times with random data. This is only used if a device could not be ATA Secure Erased. Defaults to 1.

5.1.13. dhcp

The following table outlines the options available under the **[dhcp]** group in the `/etc/ironic/ironic.conf` file.

Table 5.12. dhcp

Configuration option = Default value	Type	Description
dhcp_provider = neutron	string value	DHCP provider to use. "neutron" uses Neutron, and "none" uses a no-op provider.

5.1.14. disk_partitioner

The following table outlines the options available under the **[disk_partitioner]** group in the `/etc/ironic/ironic.conf` file.

Table 5.13. disk_partitioner

Configuration option = Default value	Type	Description
check_device_interval = 1	integer value	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	integer value	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.

5.1.15. disk_utils

The following table outlines the options available under the **[disk_utils]** group in the **/etc/ironic/ironic.conf** file.

Table 5.14. disk_utils

Configuration option = Default value	Type	Description
bios_boot_partition_size = 1	integer value	Size of BIOS Boot partition in MiB when configuring GPT partitioned systems for local boot in BIOS.
dd_block_size = 1M	string value	Block size to use when writing to the nodes disk.
efi_system_partition_size = 200	integer value	Size of EFI system partition in MiB when configuring UEFI systems for local boot.
image_convert_attempts = 3	integer value	Number of attempts to convert an image.
image_convert_memory_limit = 2048	integer value	Memory limit for "qemu-img convert" in MiB. Implemented via the address space resource limit.
partition_detection_attempts = 3	integer value	Maximum attempts to detect a newly created partition.
partprobe_attempts = 10	integer value	Maximum number of attempts to try to read the partition.

5.1.16. drac

The following table outlines the options available under the **[drac]** group in the **/etc/ironic/ironic.conf** file.

Table 5.15. drac

Configuration option = Default value	Type	Description
bios_factory_reset_timeout = 600	integer value	Maximum time (in seconds) to wait for factory reset of BIOS settings to complete.
boot_device_job_status_timeout = 30	integer value	Maximum amount of time (in seconds) to wait for the boot device configuration job to transition to the correct state to allow a reboot or power on to complete.
config_job_max_retries = 240	integer value	Maximum number of retries for the configuration job to complete successfully.
query_import_config_job_status_interval = 60	integer value	Number of seconds to wait between checking for completed import configuration task
query_raid_config_job_status_interval = 120	integer value	Interval (in seconds) between periodic RAID job status checks to determine whether the asynchronous RAID configuration was successfully finished or not.

5.1.17. glance

The following table outlines the options available under the **[glance]** group in the `/etc/ironic/ironic.conf` file.

Table 5.16. glance

Configuration option = Default value	Type	Description
allowed_direct_url_schemes = []	list value	A list of URL schemes that can be downloaded directly via the direct_url. Currently supported schemes: [file].
auth-url = None	string value	Authentication URL
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.

Configuration option = Default value	Type	Description
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
max-version = None	string value	The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.
min-version = None	string value	The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is "latest".
num_retries = 0	integer value	Number of retries when downloading an image from glance.
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project

Configuration option = Default value	Type	Description
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
region-name = None	string value	The default region_name for endpoint URL discovery.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = image	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrieable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
swift_account = None	string value	The account that Glance uses to communicate with Swift. The format is "AUTH_uuid". "uuid" is the UUID for the account configured in the glance-api.conf. For example: "AUTH_a422b2-91f3-2f46-74b7-d7c9e8958f5d30". If not set, the default value is calculated based on the ID of the project used to access Swift (as set in the [swift] section). Swift temporary URL format: "endpoint_url/api_version/account/container/object_id"
swift_api_version = v1	string value	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	string value	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"

Configuration option = Default value	Type	Description
swift_endpoint_url = None	string value	The "endpoint" (scheme, hostname, optional port) for the Swift URL of the form "endpoint_url/api_version/account/container/object_id". Do not include trailing "/". For example, use "https://swift.example.com". If using RADOS Gateway, endpoint may also contain /swift path; if it does not, it will be appended. Used for temporary URLs, will be fetched from the service catalog, if not provided.
swift_store_multiple_containers_seed = 0	integer value	This should match a config by the same name in the Glance configuration file. When set to 0, a single-tenant store will only use one container to store all images. When set to an integer value between 1 and 32, a single-tenant store will use multiple containers to store images, and this value will determine how many containers are created.
swift_temp_url_cache_enabled = False	boolean value	Whether to cache generated Swift temporary URLs. Setting it to true is only useful when an image caching proxy is used. Defaults to False.
swift_temp_url_duration = 1200	integer value	The length of time in seconds that the temporary URL will be valid for. Defaults to 20 minutes. If some deploys get a 401 response code when trying to download from the temporary URL, try raising this duration. This value must be greater than or equal to the value for swift_temp_url_expected_download_start_delay
swift_temp_url_expected_download_start_delay = 0	integer value	This is the delay (in seconds) from the time of the deploy request (when the Swift temporary URL is generated) to when the IPA ramdisk starts up and URL is used for the image download. This value is used to check if the Swift temporary URL duration is large enough to let the image download begin. Also if temporary URL caching is enabled this will determine if a cached entry will still be valid when the download starts. swift_temp_url_duration value must be greater than or equal to this option's value. Defaults to 0.
swift_temp_url_key = None	string value	The secret token given to Swift to allow temporary URL downloads. Required for temporary URLs. For the Swift backend, the key on the service project (as set in the [swift] section) is used by default.
system-scope = None	string value	Scope for system operations

Configuration option = Default value	Type	Description
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User id
username = None	string value	Username
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.
version = None	string value	Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

5.1.18. healthcheck

The following table outlines the options available under the **[healthcheck]** group in the **/etc/ironic/ironic.conf** file.

Table 5.17. healthcheck

Configuration option = Default value	Type	Description
backends = []	list value	Additional backends that can perform health checks and report that information back as part of a request.
detailed = False	boolean value	Show more detailed information as part of the response. Security note: Enabling this option may expose sensitive details about the service being monitored. Be sure to verify that it will not violate your security policies.
disable_by_file_path = None	string value	Check the presence of a file to determine if an application is running on a port. Used by DisableByFileHealthcheck plugin.

Configuration option = Default value	Type	Description
disable_by_file_paths = []	list value	Check the presence of a file based on a port to determine if an application is running on a port. Expects a "port:path" list of strings. Used by DisableByFilesPortsHealthcheck plugin.
enabled = False	boolean value	Enable the health check endpoint at /healthcheck. Note that this is unauthenticated. More information is available at https://docs.openstack.org/oslo.middleware/latest/reference/healthcheck_plugins.html .
path = /healthcheck	string value	The path to respond to healthcheck requests on.

5.1.19. ilo

The following table outlines the options available under the **[ilo]** group in the **/etc/ironic/ironic.conf** file.

Table 5.18. ilo

Configuration option = Default value	Type	Description
ca_file = None	string value	CA certificate file to validate iLO.
clean_priority_clear_secure_boot_keys = 0	integer value	Priority for clear_secure_boot_keys clean step. This step is not enabled by default. It can be enabled to clear all secure boot keys enrolled with iLO.
clean_priority_reset_bios_to_default = 10	integer value	Priority for reset_bios_to_default clean step.
clean_priority_reset_ilo = 0	integer value	Priority for reset_ilo clean step.
clean_priority_reset_ilo_credential = 30	integer value	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	integer value	Priority for reset_secure_boot_keys clean step. This step will reset the secure boot keys to manufacturing defaults.
client_port = 443	port value	Port to be used for iLO operations
client_timeout = 60	integer value	Timeout (in seconds) for iLO operations

Configuration option = Default value	Type	Description
default_boot_mode = auto	string value	Default boot mode to be used in provisioning when "boot_mode" capability is not provided in the "properties/capabilities" of the node. The default is "auto" for backward compatibility. When "auto" is specified, default boot mode will be selected based on boot mode settings on the system.
file_permission = 420	integer value	File permission for swift-less image hosting with the octal permission representation of file access permissions. This setting defaults to 644 , or as the octal number 0o644 in Python. This setting must be set to the octal number representation, meaning starting with 0o .
kernel_append_params = nofb nomodeset vga=normal	string value	Additional kernel parameters to pass down to the instance kernel. These parameters can be consumed by the kernel or by the applications by reading /proc/cmdline. Mind severe cmdline size limit! Can be overridden by instance_info/kernel_append_params property.
oob_erase_devices_job_status_interval = 300	integer value	Interval (in seconds) between periodic erase-devices status checks to determine whether the asynchronous out-of-band erase-devices was successfully finished or not. On an average, a 300GB HDD with default pattern "overwrite" would take approximately 9 hours and 300GB SSD with default pattern "block" would take approx. 30 seconds to complete sanitize disk erase.
power_wait = 2	integer value	Amount of time in seconds to wait in between power operations
swift_ilo_container = ironic_ilo_container	string value	The Swift iLO container to store data.
swift_object_expiry_time_out = 900	integer value	Amount of time in seconds for Swift objects to auto-expire.
use_web_server_for_images = False	boolean value	Set this to True to use http web server to host floppy images and generated boot ISO. This requires http_root and http_url to be configured in the [deploy] section of the config file. If this is set to False, then Ironic will use Swift to host the floppy images and generated boot_iso.

Configuration option = Default value	Type	Description
verify_ca = True	string value	CA certificate to validate iLO. This can be either a Boolean value, a path to a CA_BUNDLE file or directory with certificates of trusted CAs. If set to True the driver will verify the host certificates; if False the driver will ignore verifying the SSL certificate. If it's a path the driver will use the specified certificate or one of the certificates in the directory. Defaults to True.

5.1.20. inspector

The following table outlines the options available under the **[inspector]** group in the **/etc/ironic/ironic.conf** file.

Table 5.19. inspector

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
callback_endpoint_override = None	string value	endpoint to use as a callback for posting back introspection data when boot is managed by ironic. Standard keystoneauth options are used by default.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

Configuration option = Default value	Type	Description
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
<code>`extra_kernel_params = `</code>	string value	extra kernel parameters to pass to the inspection ramdisk when boot is managed by ironic (not ironic-inspector). Pairs key=value separated by spaces.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
max-version = None	string value	The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.
min-version = None	string value	The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is "latest".
password = None	string value	User's password
power_off = True	boolean value	whether to power off a node after inspection finishes
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project

Configuration option = Default value	Type	Description
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
region-name = None	string value	The default region_name for endpoint URL discovery.
require_managed_boot = False	boolean value	require that the in-band inspection boot is fully managed by ironic. Set this to True if your installation of ironic-inspector does not have a separate PXE boot environment.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = baremetal-introspection	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrievable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
status_check_period = 60	integer value	period (in seconds) to check status of nodes on inspection
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name

Configuration option = Default value	Type	Description
user-id = None	string value	User id
username = None	string value	Username
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.
version = None	string value	Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

5.1.21. ipmi

The following table outlines the options available under the **[ipmi]** group in the **/etc/ironic/ironic.conf** file.

Table 5.20. ipmi

Configuration option = Default value	Type	Description
additional_retryable_ipmi_errors = []	multi valued	Additional errors ipmitool may encounter, specific to the environment it is run in.
cipher_suite_versions = []	list value	List of possible cipher suites versions that can be supported by the hardware in case the field cipher_suite is not set for the node.
command_retry_timeout = 60	integer value	Maximum time in seconds to retry retryable IPMI operations. (An operation is retryable, for example, if the requested operation fails because the BMC is busy.) Setting this too high can cause the sync power state periodic task to hang when there are slow or unresponsive BMCs.
debug = False	boolean value	Enables all ipmi commands to be executed with an additional debugging output. This is a separate option as ipmitool can log a substantial amount of misleading text when in this mode.
disable_boot_timeout = True	boolean value	Default timeout behavior whether ironic sends a raw IPMI command to disable the 60 second timeout for booting. Setting this option to False will NOT send that command, the default value is True. It may be overridden by per-node <i>ipmi_disable_boot_timeout</i> option in node's <i>driver_info</i> field.

Configuration option = Default value	Type	Description
kill_on_timeout = True	boolean value	Kill ipmitool process invoked by ironic to read node power state if ipmitool process does not exit after command_retry_timeout timeout expires. Recommended setting is True
min_command_interval = 5	integer value	Minimum time, in seconds, between IPMI operations sent to a server. There is a risk with some hardware that setting this too low may cause the BMC to crash. Recommended setting is 5 seconds.
use_ipmitool_retries = False	boolean value	When set to True and the parameters are supported by ipmitool, the number of retries and the retry interval are passed to ipmitool as parameters, and ipmitool will do the retries. When set to False, ironic will retry the ipmitool commands. Recommended setting is False

5.1.22. irmc

The following table outlines the options available under the **[irmc]** group in the **/etc/ironic/ironic.conf** file.

Table 5.21. irmc

Configuration option = Default value	Type	Description
auth_method = basic	string value	Authentication method to be used for iRMC operations
clean_priority_restore_irmc_bios_config = 0	integer value	Priority for restore_irmc_bios_config clean step.
client_timeout = 60	integer value	Timeout (in seconds) for iRMC operations
fpga_ids = []	list value	List of vendor IDs and device IDs for CPU FPGA to inspect. List items are in format vendorID/deviceID and separated by commas. CPU inspection will use this value to find existence of CPU FPGA in a node. If this option is not defined, then leave out CUSTOM_CPU_FPGA in node traits. Sample fpga_ids value: 0x1000/0x0079,0x2100/0x0080

Configuration option = Default value	Type	Description
gpu_ids = []	list value	List of vendor IDs and device IDs for GPU device to inspect. List items are in format vendorID/deviceID and separated by commas. GPU inspection will use this value to count the number of GPU device in a node. If this option is not defined, then leave out pci_gpu_devices in capabilities property. Sample gpu_ids value: 0x1000/0x0079,0x2100/0x0080
port = 443	port value	Port to be used for iRMC operations
query_raid_config_fgi_status_interval = 300	integer value	Interval (in seconds) between periodic RAID status checks to determine whether the asynchronous RAID configuration was successfully finished or not. Foreground Initialization (FGI) will start 5 minutes after creating virtual drives.
remote_image_server = None	string value	IP of remote image server
remote_image_share_name = share	string value	share name of remote_image_server
remote_image_share_root = /remote_image_share_root	string value	Ironic conductor node's "NFS" or "CIFS" root path
remote_image_share_type = CIFS	string value	Share type of virtual media
remote_image_user_domain = `	string value	Domain name of remote_image_user_name
remote_image_user_name = None	string value	User name of remote_image_server
remote_image_user_password = None	string value	Password of remote_image_user_name
sensor_method = ipmitool	string value	Sensor data retrieval method.
snmp_community = public	string value	SNMP community. Required for versions "v1" and "v2c"
snmp_polling_interval = 10	integer value	SNMP polling interval in seconds

Configuration option = Default value	Type	Description
snmp_port = 161	port value	SNMP port
snmp_security = None	string value	SNMP security name. Required for version "v3"
snmp_version = v2c	string value	SNMP protocol version

5.1.23. ironic_lib

The following table outlines the options available under the **[ironic_lib]** group in the **/etc/ironic/ironic.conf** file.

Table 5.22. ironic_lib

Configuration option = Default value	Type	Description
fatal_exception_format_errors = False	boolean value	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.
root_helper = sudo ironic-rootwrap /etc/ironic/rootwrap.conf	string value	Command that is prefixed to commands that are run as root. If not specified, no commands are run as root.

5.1.24. iscsi

The following table outlines the options available under the **[iscsi]** group in the **/etc/ironic/ironic.conf** file.

Table 5.23. iscsi

Configuration option = Default value	Type	Description
conv_flags = None	string value	Flags that need to be sent to the dd command, to control the conversion of the original file when copying to the host. It can contain several options separated by commas.
portal_port = 3260	port value	The port number on which the iSCSI portal listens for incoming connections.

Configuration option = Default value	Type	Description
verify_attempts = 3	integer value	Maximum attempts to verify an iSCSI connection is active, sleeping 1 second between attempts. Defaults to 3.

5.1.25. json_rpc

The following table outlines the options available under the **[json_rpc]** group in the **/etc/ironic/ironic.conf** file.

Table 5.24. json_rpc

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_strategy = None	string value	Authentication strategy used by JSON RPC. Defaults to the global auth_strategy setting.
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
host_ip = ::	host address value	The IP address or hostname on which JSON RPC will listen.

Configuration option = Default value	Type	Description
http_basic_auth_user_file = /etc/ironic/htpasswd- json-rpc	string value	Path to Apache format user authentication file used when auth_strategy=http_basic
http_basic_password = None	string value	Password to use for HTTP Basic authentication client requests.
http_basic_username = None	string value	Name of the user to use for HTTP Basic authentication client requests.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
password = None	string value	User's password
port = 8089	port value	The port to use for JSON RPC
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
split-loggers = False	boolean value	Log requests to multiple loggers.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
use_ssl = False	boolean value	Whether to use TLS for JSON RPC
user-domain-id = None	string value	User's domain id

Configuration option = Default value	Type	Description
user-domain-name = None	string value	User's domain name
user-id = None	string value	User id
username = None	string value	Username

5.1.26. keystone_authtoken

The following table outlines the options available under the **[keystone_authtoken]** group in the **/etc/ironic/ironic.conf** file.

Table 5.25. keystone_authtoken

Configuration option = Default value	Type	Description
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load
auth_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint. This option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release. Deprecated since: Queens <i>*Reason:</i>*The <code>auth_uri</code> option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release.
auth_version = None	string value	API version of the Identity API endpoint.
cache = None	string value	Request environment key where the Swift cache object is stored. When <code>auth_token</code> middleware is deployed with a Swift cache, use this option to have the middleware share a caching backend with swift. Otherwise, use the memcached_servers option instead.

Configuration option = Default value	Type	Description
cafile = None	string value	A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
certfile = None	string value	Required if identity server requires client certificate
delay_auth_decision = False	boolean value	Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components.
enforce_token_bind = permissive	string value	Used to control the use and type of token binding. Can be set to: "disabled" to not check token binding. "permissive" (default) to validate binding information if the bind type is of a form known to the server and ignore it if not. "strict" like "permissive" but if the bind type is unknown the token will be rejected. "required" any form of token binding is needed to be allowed. Finally the name of a binding method that must be present in tokens.
http_connect_timeout = None	integer value	Request timeout value for communicating with Identity API server.
http_request_max_retries = 3	integer value	How many times are we trying to reconnect when communicating with Identity API Server.
include_service_catalog = True	boolean value	(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	boolean value	Verify HTTPS connections.
interface = internal	string value	Interface to use for the Identity API endpoint. Valid values are "public", "internal" (default) or "admin".
keyfile = None	string value	Required if identity server requires client certificate
memcache_pool_conn_get_timeout = 10	integer value	(Optional) Number of seconds that an operation will wait to get a memcached client connection from the pool.
memcache_pool_dead_retry = 300	integer value	(Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = 10	integer value	(Optional) Maximum total number of open connections to every memcached server.

Configuration option = Default value	Type	Description
memcache_pool_socket_timeout = 3	integer value	(Optional) Socket timeout in seconds for communicating with a memcached server.
memcache_pool_unused_timeout = 60	integer value	(Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
memcache_secret_key = None	string value	(Optional, mandatory if memcache_security_strategy is defined) This string is used for key derivation.
memcache_security_strategy = None	string value	(Optional) If defined, indicate whether token data should be authenticated or authenticated and encrypted. If MAC, token data is authenticated (with HMAC) in the cache. If ENCRYPT, token data is encrypted and authenticated in the cache. If the value is not one of these options or empty, auth_token will raise an exception on initialization.
memcache_use_advanced_pool = False	boolean value	(Optional) Use the advanced (eventlet safe) memcached client pool. The advanced pool will only work under python 2.x.
memcached_servers = None	list value	Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
region_name = None	string value	The region in which the identity server can be found.
service_token_roles = ['service']	list value	A choice of roles that must be present in a service token. Service tokens are allowed to request that an expired token can be used and so this check should tightly control that only actual services should be sending this token. Roles here are applied as an ANY check so any role in this list must be present. For backwards compatibility reasons this currently only affects the allow_expired check.
service_token_roles_required = False	boolean value	For backwards compatibility reasons we must let valid service tokens pass that don't pass the service_token_roles check as valid. Setting this true will become the default in a future release and should be enabled if possible.
service_type = None	string value	The name or type of the service as it appears in the service catalog. This is used to validate tokens that have restricted access rules.

Configuration option = Default value	Type	Description
token_cache_time = 300	integer value	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.
www_authenticate_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint.

5.1.27. mdns

The following table outlines the options available under the **[mdns]** group in the `/etc/ironic/ironic.conf` file.

Table 5.26. mdns

Configuration option = Default value	Type	Description
interfaces = None	list value	List of IP addresses of interfaces to use for mDNS. Defaults to all interfaces on the system.
lookup_attempts = 3	integer value	Number of attempts to lookup a service.
params = {}	dict value	Additional parameters to pass for the registered service.
registration_attempts = 5	integer value	Number of attempts to register a service. Currently has to be larger than 1 because of race conditions in the zeroconf library.

5.1.28. metrics

The following table outlines the options available under the **[metrics]** group in the `/etc/ironic/ironic.conf` file.

Table 5.27. metrics

Configuration option = Default value	Type	Description
agent_backend = noop	string value	Backend for the agent ramdisk to use for metrics. Default possible backends are "noop" and "statsd".
agent_global_prefix = None	string value	Prefix all metric names sent by the agent ramdisk with this value. The format of metric names is [global_prefix.][uuid.] [host_name.]prefix.metric_name.
agent_prepend_host = False	boolean value	Prepend the hostname to all metric names sent by the agent ramdisk. The format of metric names is [global_prefix.][uuid.] [host_name.]prefix.metric_name.
agent_prepend_host_reverse = True	boolean value	Split the prepended host value by "." and reverse it for metrics sent by the agent ramdisk (to better match the reverse hierarchical form of domain names).
agent_prepend_uuid = False	boolean value	Prepend the node's Ironic uuid to all metric names sent by the agent ramdisk. The format of metric names is [global_prefix.][uuid.] [host_name.]prefix.metric_name.
backend = noop	string value	Backend to use for the metrics system.
global_prefix = None	string value	Prefix all metric names with this value. By default, there is no global prefix. The format of metric names is [global_prefix.][host_name.]prefix.metric_name.
prepend_host = False	boolean value	Prepend the hostname to all metric names. The format of metric names is [global_prefix.] [host_name.]prefix.metric_name.
prepend_host_reverse = True	boolean value	Split the prepended host value by "." and reverse it (to better match the reverse hierarchical form of domain names).

5.1.29. metrics_statsd

The following table outlines the options available under the **[metrics_statsd]** group in the **/etc/ironic/ironic.conf** file.

Table 5.28. metrics_statsd

Configuration option = Default value	Type	Description
agent_statsd_host = localhost	string value	Host for the agent ramdisk to use with the statsd backend. This must be accessible from networks the agent is booted on.
agent_statsd_port = 8125	port value	Port for the agent ramdisk to use with the statsd backend.
statsd_host = localhost	string value	Host for use with the statsd backend.
statsd_port = 8125	port value	Port to use with the statsd backend.

5.1.30. molds

The following table outlines the options available under the **[molds]** group in the **/etc/ironic/ironic.conf** file.

Table 5.29. molds

Configuration option = Default value	Type	Description
password = None	string value	Password for "http" Basic auth. By default set empty.
retry_attempts = 3	integer value	Retry attempts for saving or getting configuration molds.
retry_interval = 3	integer value	Retry interval for saving or getting configuration molds.
storage = swift	string value	Configuration mold storage location. Supports "swift" and "http". By default "swift".
user = None	string value	User for "http" Basic auth. By default set empty.

5.1.31. neutron

The following table outlines the options available under the **[neutron]** group in the **/etc/ironic/ironic.conf** file.

Table 5.30. neutron

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
add_all_ports = False	boolean value	Option to enable transmission of all ports to neutron when creating ports for provisioning, cleaning, or rescue. This is done without IP addresses assigned to the port, and may be useful in some bonded network configurations.
auth-url = None	string value	Authentication URL
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
cleaning_network = None	string value	Neutron network UUID or name for the ramdisk to be booted into for cleaning nodes. Required for "neutron" network interface. It is also required if cleaning nodes when using "flat" network interface or "neutron" DHCP provider. If a name is provided, it must be unique among all networks or cleaning will fail.
cleaning_network_security_groups = []	list value	List of Neutron Security Group UUIDs to be applied during cleaning of the nodes. Optional for the "neutron" network interface and not used for the "flat" or "noop" network interfaces. If not specified, default security group is used.
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

Configuration option = Default value	Type	Description
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
dhcpv6_stateful_address_count = 4	integer value	Number of IPv6 addresses to allocate for ports created for provisioning, cleaning, rescue or inspection on DHCPv6-stateful networks. Different stages of the chain-loading process will request addresses with different CLID/IAID. Due to non-identical identifiers multiple addresses must be reserved for the host to ensure each step of the boot process can successfully lease addresses.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
inspection_network = None	string value	Neutron network UUID or name for the ramdisk to be booted into for in-band inspection of nodes. If a name is provided, it must be unique among all networks or inspection will fail.
inspection_network_security_groups = []	list value	List of Neutron Security Group UUIDs to be applied during the node inspection process. Optional for the "neutron" network interface and not used for the "flat" or "noop" network interfaces. If not specified, the default security group is used.
keyfile = None	string value	PEM encoded client certificate key file
max-version = None	string value	The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.
min-version = None	string value	The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is "latest".

Configuration option = Default value	Type	Description
password = None	string value	User's password
port_setup_delay = 0	integer value	Delay value to wait for Neutron agents to setup sufficient DHCP configuration for port.
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
provisioning_network = None	string value	Neutron network UUID or name for the ramdisk to be booted into for provisioning nodes. Required for "neutron" network interface. If a name is provided, it must be unique among all networks or deploy will fail.
provisioning_network_security_groups = []	list value	List of Neutron Security Group UUIDs to be applied during provisioning of the nodes. Optional for the "neutron" network interface and not used for the "flat" or "noop" network interfaces. If not specified, default security group is used.
region-name = None	string value	The default region_name for endpoint URL discovery.
request_timeout = 45	integer value	Timeout for request processing when interacting with Neutron. This value should be increased if neutron port action timeouts are observed as neutron performs pre-commit validation prior returning to the API client which can take longer than normal client/server interactions.
rescuing_network = None	string value	Neutron network UUID or name for booting the ramdisk for rescue mode. This is not the network that the rescue ramdisk will use post-boot – the tenant network is used for that. Required for "neutron" network interface, if rescue mode will be used. It is not used for the "flat" or "noop" network interfaces. If a name is provided, it must be unique among all networks or rescue will fail.

Configuration option = Default value	Type	Description
rescuing_network_security_groups = []	list value	List of Neutron Security Group UUIDs to be applied during the node rescue process. Optional for the "neutron" network interface and not used for the "flat" or "noop" network interfaces. If not specified, the default security group is used.
retries = 3	integer value	DEPRECATED: Client retries in the case of a failed request.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = network	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrievable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User id
username = None	string value	Username
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.

Configuration option = Default value	Type	Description
version = None	string value	Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

5.1.32. nova

The following table outlines the options available under the **[nova]** group in the **/etc/ironic/ironic.conf** file.

Table 5.31. nova

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to

Configuration option = Default value	Type	Description
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
max-version = None	string value	The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.
min-version = None	string value	The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is "latest".
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
region-name = None	string value	The default region_name for endpoint URL discovery.
send_power_notifications = True	boolean value	When set to True, it will enable the support for power state change callbacks to nova. This option should be set to False in deployments that do not have the openstack compute service.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = compute	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.

Configuration option = Default value	Type	Description
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrieable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User id
username = None	string value	Username
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.
version = None	string value	Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

5.1.33. oslo_concurrency

The following table outlines the options available under the **[oslo_concurrency]** group in the **/etc/ironic/ironic.conf** file.

Table 5.32. oslo_concurrency

Configuration option = Default value	Type	Description
disable_process_locking = False	boolean value	Enables or disables inter-process locks.
lock_path = None	string value	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.

5.1.34. oslo_messaging_amqp

The following table outlines the options available under the **[oslo_messaging_amqp]** group in the **/etc/ironic/ironic.conf** file.

Table 5.33. oslo_messaging_amqp

Configuration option = Default value	Type	Description
addressing_mode = dynamic	string value	Indicates the addressing mode used by the driver. Permitted values: <i>legacy</i> - use legacy non-routable addressing <i>routable</i> - use routable addresses <i>dynamic</i> - use legacy addresses if the message bus does not support routing otherwise use routable addressing
anycast_address = anycast	string value	Appended to the address prefix when sending to a group of consumers. Used by the message bus to identify messages that should be delivered in a round-robin fashion across consumers.
broadcast_prefix = broadcast	string value	address prefix used when broadcasting to all servers
connection_retry_backoff = 2	integer value	Increase the connection_retry_interval by this many seconds after each unsuccessful failover attempt.
connection_retry_interval = 1	integer value	Seconds to pause before attempting to re-connect.
connection_retry_interval_max = 30	integer value	Maximum limit for connection_retry_interval + connection_retry_backoff
container_name = None	string value	Name for the AMQP container. must be globally unique. Defaults to a generated UUID

Configuration option = Default value	Type	Description
default_notification_exchange = None	string value	Exchange name used in notification addresses. Exchange name resolution precedence: Target.exchange if set else default_notification_exchange if set else control_exchange if set else <i>notify</i>
default_notify_timeout = 30	integer value	The deadline for a sent notification message delivery. Only used when caller does not provide a timeout expiry.
default_reply_retry = 0	integer value	The maximum number of attempts to re-send a reply message which failed due to a recoverable error.
default_reply_timeout = 30	integer value	The deadline for an rpc reply message delivery.
default_rpc_exchange = None	string value	Exchange name used in RPC addresses. Exchange name resolution precedence: Target.exchange if set else default_rpc_exchange if set else control_exchange if set else <i>rpc</i>
default_send_timeout = 30	integer value	The deadline for an rpc cast or call message delivery. Only used when caller does not provide a timeout expiry.
default_sender_link_timeout = 600	integer value	The duration to schedule a purge of idle sender links. Detach link after expiry.
group_request_prefix = unicast	string value	address prefix when sending to any server in group
idle_timeout = 0	integer value	Timeout for inactive connections (in seconds)
link_retry_delay = 10	integer value	Time to pause between re-connecting an AMQP 1.0 link that failed due to a recoverable error.
multicast_address = multicast	string value	Appended to the address prefix when sending a fanout message. Used by the message bus to identify fanout messages.
notify_address_prefix = openstack.org/om/notify	string value	Address prefix for all generated Notification addresses
notify_server_credit = 100	integer value	Window size for incoming Notification messages

Configuration option = Default value	Type	Description
pre_settled = ['rpc-cast', 'rpc-reply']	multi valued	Send messages of this type pre-settled. Pre-settled messages will not receive acknowledgement from the peer. Note well: pre-settled messages may be silently discarded if the delivery fails. Permitted values: <i>rpc-call</i> - send RPC Calls pre-settled <i>rpc-reply</i> - send RPC Replies pre-settled <i>rpc-cast</i> - Send RPC Casts pre-settled <i>notify</i> - Send Notifications pre-settled
pseudo_vhost = True	boolean value	Enable virtual host support for those message buses that do not natively support virtual hosting (such as qpid). When set to true the virtual host name will be added to all message bus addresses, effectively creating a private <i>subnet</i> per virtual host. Set to False if the message bus supports virtual hosting using the <i>hostname</i> field in the AMQP 1.0 Open performative as the name of the virtual host.
reply_link_credit = 200	integer value	Window size for incoming RPC Reply messages.
rpc_address_prefix = openstack.org/om/rpc	string value	Address prefix for all generated RPC addresses
rpc_server_credit = 100	integer value	Window size for incoming RPC Request messages
<code>`sasldb_dir = `</code>	string value	Path to directory that contains the SASL configuration
<code>`sasldb_name = `</code>	string value	Name of configuration file (without .conf suffix)
<code>`sasldb_realm = `</code>	string value	SASL realm to use if no realm present in username
<code>`sasldb_mechanisms = `</code>	string value	Space separated list of acceptable SASL mechanisms
server_request_prefix = exclusive	string value	address prefix used when sending to a specific server
ssl = False	boolean value	Attempt to connect via SSL. If no other ssl-related parameters are given, it will use the system's CA-bundle to verify the server's certificate.
<code>`ssl_ca_file = `</code>	string value	CA certificate PEM file used to verify the server's certificate
<code>`ssl_cert_file = `</code>	string value	Self-identifying certificate PEM file for client authentication

Configuration option = Default value	Type	Description
<code>`ssl_key_file = `</code>	string value	Private key PEM file used to sign <code>ssl_cert_file</code> certificate (optional)
<code>ssl_key_password = None</code>	string value	Password for decrypting <code>ssl_key_file</code> (if encrypted)
<code>ssl_verify_vhost = False</code>	boolean value	By default SSL checks that the name in the server's certificate matches the hostname in the <code>transport_url</code> . In some configurations it may be preferable to use the virtual hostname instead, for example if the server uses the Server Name Indication TLS extension (rfc6066) to provide a certificate per virtual host. Set <code>ssl_verify_vhost</code> to True if the server's SSL certificate uses the virtual host name instead of the DNS name.
<code>trace = False</code>	boolean value	Debug: dump AMQP frames to stdout
<code>unicast_address = unicast</code>	string value	Appended to the address prefix when sending to a particular RPC/Notification server. Used by the message bus to identify messages sent to a single destination.

5.1.35. oslo_messaging_kafka

The following table outlines the options available under the **[oslo_messaging_kafka]** group in the `/etc/ironic/ironic.conf` file.

Table 5.34. oslo_messaging_kafka

Configuration option = Default value	Type	Description
<code>compression_codec = none</code>	string value	The compression codec for all data generated by the producer. If not set, compression will not be used. Note that the allowed values of this depend on the kafka version
<code>conn_pool_min_size = 2</code>	integer value	The pool size limit for connections expiration policy
<code>conn_pool_ttl = 1200</code>	integer value	The time-to-live in sec of idle connections in the pool
<code>consumer_group = oslo_messaging_consumer</code>	string value	Group id for Kafka consumer. Consumers in one group will coordinate message consumption

Configuration option = Default value	Type	Description
enable_auto_commit = False	boolean value	Enable asynchronous consumer commits
kafka_consumer_timeout = 1.0	floating point value	Default timeout(s) for Kafka consumers
kafka_max_fetch_bytes = 1048576	integer value	Max fetch bytes of Kafka consumer
max_poll_records = 500	integer value	The maximum number of records returned in a poll call
pool_size = 10	integer value	Pool Size for Kafka Consumers
producer_batch_size = 16384	integer value	Size of batch for the producer async send
producer_batch_timeout = 0.0	floating point value	Upper bound on the delay for KafkaProducer batching in seconds
sasl_mechanism = PLAIN	string value	Mechanism when security protocol is SASL
security_protocol = PLAINTEXT	string value	Protocol used to communicate with brokers
<code>`ssl_cafile = `</code>	string value	CA certificate PEM file used to verify the server certificate
<code>`ssl_client_cert_file = `</code>	string value	Client certificate PEM file used for authentication.
<code>`ssl_client_key_file = `</code>	string value	Client key PEM file used for authentication.
<code>`ssl_client_key_password = `</code>	string value	Client key password file used for authentication.

5.1.36. oslo_messaging_notifications

The following table outlines the options available under the **[oslo_messaging_notifications]** group in the **/etc/ironic/ironic.conf** file.

Table 5.35. oslo_messaging_notifications

Configuration option = Default value	Type	Description
driver = []	multi valued	The Drivers(s) to handle sending notifications. Possible values are messaging, messagingv2, routing, log, test, noop
retry = -1	integer value	The maximum number of attempts to re-send a notification message which failed to be delivered due to a recoverable error. 0 - No retry, -1 - indefinite
topics = ['notifications']	list value	AMQP topic used for OpenStack notifications.
transport_url = None	string value	A URL representing the messaging driver to use for notifications. If not set, we fall back to the same configuration used for RPC.

5.1.37. oslo_messaging_rabbit

The following table outlines the options available under the **[oslo_messaging_rabbit]** group in the **/etc/ironic/ironic.conf** file.

Table 5.36. oslo_messaging_rabbit

Configuration option = Default value	Type	Description
amqp_auto_delete = False	boolean value	Auto-delete queues in AMQP.
amqp_durable_queues = False	boolean value	Use durable queues in AMQP.
direct_mandatory_flag = True	boolean value	(DEPRECATED) Enable/Disable the RabbitMQ mandatory flag for direct send. The direct send is used as reply, so the MessageUndeliverable exception is raised in case the client queue does not exist. MessageUndeliverable exception will be used to loop for a timeout to let a chance to sender to recover. This flag is deprecated and it will not be possible to deactivate this functionality anymore
enable_cancel_on_failover = False	boolean value	Enable x-cancel-on-ha-failover flag so that rabbitmq server will cancel and notify consumers when queue is down

Configuration option = Default value	Type	Description
heartbeat_in_pthread = True	boolean value	Run the health check heartbeat thread through a native python thread by default. If this option is equal to False then the health check heartbeat will inherit the execution model from the parent process. For example if the parent process has monkey patched the stdlib by using eventlet/greenlet then the heartbeat will be run through a green thread.
heartbeat_rate = 2	integer value	How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = 60	integer value	Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables heartbeat).
kombu_compression = None	string value	EXPERIMENTAL: Possible values are: gzip, bz2. If not set compression will not be used. This option may not be available in future versions.
kombu_failover_strategy = round-robin	string value	Determines how the next RabbitMQ node is chosen in case the one we are currently connected to becomes unavailable. Takes effect only if more than one RabbitMQ node is provided in config.
kombu_missing_consumer_retry_timeout = 60	integer value	How long to wait a missing client before abandoning to send it its replies. This value should not be longer than rpc_response_timeout.
kombu_reconnect_delay = 1.0	floating point value	How long to wait before reconnecting in response to an AMQP consumer cancel notification.
rabbit_ha_queues = False	boolean value	Try to use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database. In RabbitMQ 3.0, queue mirroring is no longer controlled by the x-ha-policy argument when declaring a queue. If you just want to make sure that all queues (except those with auto-generated names) are mirrored across all nodes, run: "rabbitmqctl set_policy HA ^(?!amq\.).* {\"ha-mode\": \"all\"} \"
rabbit_interval_max = 30	integer value	Maximum interval of RabbitMQ connection retries. Default is 30 seconds.
rabbit_login_method = AMQPLAIN	string value	The RabbitMQ login method.

Configuration option = Default value	Type	Description
rabbit_qos_prefetch_count = 0	integer value	Specifies the number of messages to prefetch. Setting to zero allows unlimited messages.
rabbit_retry_backoff = 2	integer value	How long to backoff for between retries when connecting to RabbitMQ.
rabbit_retry_interval = 1	integer value	How frequently to retry connecting with RabbitMQ.
rabbit_transient_queues_ttl = 1800	integer value	Positive integer representing duration in seconds for queue TTL (x-expires). Queues which are unused for the duration of the TTL are automatically deleted. The parameter affects only reply and fanout queues.
ssl = False	boolean value	Connect over SSL.
<code>`ssl_ca_file = `</code>	string value	SSL certification authority file (valid only if SSL enabled).
<code>`ssl_cert_file = `</code>	string value	SSL cert file (valid only if SSL enabled).
<code>`ssl_key_file = `</code>	string value	SSL key file (valid only if SSL enabled).
<code>`ssl_version = `</code>	string value	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.

5.1.38. oslo_middleware

The following table outlines the options available under the **[oslo_middleware]** group in the `/etc/ironic/ironic.conf` file.

Table 5.37. oslo_middleware

Configuration option = Default value	Type	Description
enable_proxy_headers_parsing = False	boolean value	Whether the application is behind a proxy or not. This determines if the middleware should parse the headers or not.

5.1.39. oslo_policy

The following table outlines the options available under the **[oslo_policy]** group in the `/etc/ironic/ironic.conf` file.

Table 5.38. oslo_policy

Configuration option = Default value	Type	Description
enforce_new_defaults = False	boolean value	This option controls whether or not to use old deprecated defaults when evaluating policies. If True , the old deprecated defaults are not going to be evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be disallowed. It is encouraged to enable this flag along with the enforce_scope flag so that you can get the benefits of new defaults and scope_type together
enforce_scope = False	boolean value	This option controls whether or not to enforce scope when evaluating policies. If True , the scope of the token used in the request is compared to the scope_types of the policy being enforced. If the scopes do not match, an InvalidScope exception will be raised. If False , a message will be logged informing operators that policies are being invoked with mismatching scope.
policy_default_rule = default	string value	Default rule. Enforced when a requested rule is not found.
policy_dirs = ['policy.d']	multi valued	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	string value	The relative or absolute path of a file that maps roles to permissions for a given service. Relative paths must be specified in relation to the configuration file setting this option.
remote_content_type = application/x-www-form-urlencoded	string value	Content Type to send and receive data for REST based policy check
remote_ssl_ca_cert_file = None	string value	Absolute path to ca cert file for REST based policy check
remote_ssl_client_cert_file = None	string value	Absolute path to client cert for REST based policy check
remote_ssl_client_key_file = None	string value	Absolute path client key file REST based policy check

Configuration option = Default value	Type	Description
remote_ssl_verify_server _cert = False	boolean value	server identity verification for REST based policy check

5.1.40. oslo_reports

The following table outlines the options available under the **[oslo_reports]** group in the **/etc/ironic/ironic.conf** file.

Table 5.39. oslo_reports

Configuration option = Default value	Type	Description
file_event_handler = None	string value	The path to a file to watch for changes to trigger the reports, instead of signals. Setting this option disables the signal trigger for the reports. If application is running as a WSGI application it is recommended to use this instead of signals.
file_event_handler_interval = 1	integer value	How many seconds to wait between polls when file_event_handler is set
log_dir = None	string value	Path to a log directory where to create a file

5.1.41. profiler

The following table outlines the options available under the **[profiler]** group in the **/etc/ironic/ironic.conf** file.

Table 5.40. profiler

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
connection_string = messaging://	string value	Connection string for a notifier backend. Default value is messaging:// which sets the notifier to oslo_messaging. Examples of possible values: ● messaging:// - use oslo_messaging driver for sending spans. ● redis://127.0.0.1:6379 - use redis driver for sending spans. ● mongodb://127.0.0.1:27017 - use mongodb driver for sending spans. ● elasticsearch://127.0.0.1:9200 - use elasticsearch driver for sending spans. ● jaeger://127.0.0.1:6831 - use jaeger tracing as driver for sending spans.
enabled = False	boolean value	Enable the profiling for all services on this node. Default value is False (fully disable the profiling feature). Possible values: ● True: Enables the feature ● False: Disables the feature. The profiling cannot be started via this project operations. If the profiling is triggered by another project, this project part will be empty.
es_doc_type = notification	string value	Document type for notification indexing in elasticsearch.
es_scroll_size = 10000	integer value	Elasticsearch splits large requests in batches. This parameter defines maximum size of each batch (for example: es_scroll_size=10000).
es_scroll_time = 2m	string value	This parameter is a time value parameter (for example: es_scroll_time=2m), indicating for how long the nodes that participate in the search will maintain relevant resources in order to continue and support it.

Configuration option = Default value	Type	Description
filter_error_trace = False	boolean value	Enable filter traces that contain error/exception to a separated place. Default value is set to False. Possible values: • True: Enable filter traces that contain error/exception. • False: Disable the filter.
hmac_keys = SECRET_KEY	string value	Secret key(s) to use for encrypting context data for performance profiling. This string value should have the following format: <key1>[,<key2>,...<keyn>], where each key is some random string. A user who triggers the profiling via the REST API has to set one of these keys in the headers of the REST API call to include profiling results of this node for this particular project. Both "enabled" flag and "hmac_keys" config options should be set to enable profiling. Also, to generate correct profiling information across all services at least one key needs to be consistent between OpenStack projects. This ensures it can be used from client side to generate the trace, containing information from all possible resources.
sentinel_service_name = mymaster	string value	Redis sentinel uses a service name to identify a master redis service. This parameter defines the name (for example: sentinal_service_name=mymaster).
socket_timeout = 0.1	floating point value	Redis sentinel provides a timeout option on the connections. This parameter defines that timeout (for example: socket_timeout=0.1).

Configuration option = Default value	Type	Description
trace_sqlalchemy = False	boolean value	Enable SQL requests profiling in services. Default value is False (SQL requests won't be traced). Possible values: • True: Enables SQL requests profiling. Each SQL query will be part of the trace and can be analyzed by how much time was spent for that. • False: Disables SQL requests profiling. The spent time is only shown on a higher level of operations. Single SQL queries cannot be analyzed this way.

5.1.42. pxe

The following table outlines the options available under the **[pxe]** group in the **/etc/ironic/ironic.conf** file.

Table 5.41. pxe

Configuration option = Default value	Type	Description
boot_retry_check_interval = 90	integer value	Interval (in seconds) between periodic checks on PXE boot retry. Has no effect if <code>boot_retry_timeout</code> is not set.
boot_retry_timeout = None	integer value	Timeout (in seconds) after which PXE boot should be retried. Must be less than <code>[conductor]deploy_callback_timeout</code> . Disabled by default.
default_ephemeral_format = ext4	string value	Default file system format for ephemeral partition, if one is created.

Configuration option = Default value	Type	Description
dir_permission = None	integer value	The permission that will be applied to the TFTP folders upon creation. This should be set to the permission such that the tftpsrv has access to read the contents of the configured TFTP folder. This setting is only required when the operating system's umask is restrictive such that ironic-conductor is creating files that cannot be read by the TFTP server. Setting to <None> will result in the operating system's umask to be utilized for the creation of new tftp folders. It is recommended that an octal representation is specified. For example: 0o755
enable_netboot_fallback = False	boolean value	If True, generate a PXE environment even for nodes that use local boot. This is useful when the driver cannot switch nodes to local boot, e.g. with SNMP or with Redfish on machines that cannot do persistent boot. Mostly useful for standalone ironic since Neutron will prevent incorrect PXE boot.
image_cache_size = 20480	integer value	Maximum size (in MiB) of cache for master images, including those in use.
image_cache_ttl = 10080	integer value	Maximum TTL (in minutes) for old master images in cache.
images_path = /var/lib/ironic/images/	string value	On the ironic-conductor node, directory where images are stored on disk.
instance_master_path = /var/lib/ironic/master_images	string value	On the ironic-conductor node, directory where master instance images are stored on disk. Setting to the empty string disables image caching.
ip_version = 4	string value	The IP version that will be used for PXE booting. Defaults to 4. EXPERIMENTAL
ipxe_boot_script = \$pybasedir/drivers/modules/boot.ipxe	string value	On ironic-conductor node, the path to the main iPXE script file.
ipxe_bootfile_name = undionly.kpxe	string value	Bootfile DHCP parameter.
ipxe_bootfile_name_by_arch = {}	dict value	Bootfile DHCP parameter per node architecture. For example: aarch64:ipxe_aa64.efi

Configuration option = Default value	Type	Description
ipxe_config_template = \$pybasedir/drivers/modules/ipxe_config.template	string value	On ironic-conductor node, template file for iPXE operations.
ipxe_timeout = 0	integer value	Timeout value (in seconds) for downloading an image via iPXE. Defaults to 0 (no timeout)
ipxe_use_swift = False	boolean value	Download deploy and rescue images directly from swift using temporary URLs. If set to false (default), images are downloaded to the ironic-conductor node and served over its local HTTP server. Applicable only when <i>ipxe</i> compatible boot interface is used.
pxe_append_params = nofb nomodeset vga=normal	string value	Additional append parameters for baremetal PXE boot.
pxe_bootfile_name = pxelinux.0	string value	Bootfile DHCP parameter.
pxe_bootfile_name_by_arch = {}	dict value	Bootfile DHCP parameter per node architecture. For example: aarch64:grubaa64.efi
pxe_config_subdir = pxelinux.cfg	string value	Directory in which to create symbolic links which represent the MAC or IP address of the ports on a node and allow boot loaders to load the PXE file for the node. This directory name is relative to the PXE or iPXE folders.
pxe_config_template = \$pybasedir/drivers/modules/pxe_config.template	string value	On ironic-conductor node, template file for PXE loader configuration.
pxe_config_template_by_arch = {}	dict value	On ironic-conductor node, template file for PXE configuration per node architecture. For example: aarch64:/opt/share/grubaa64_pxe_config.template
tftp_master_path = /tftpboot/master_images	string value	On ironic-conductor node, directory where master TFTP images are stored on disk. Setting to the empty string disables image caching.
tftp_root = /tftpboot	string value	ironic-conductor node's TFTP root path. The ironic-conductor must have read/write access to this path.
tftp_server = \$my_ip	string value	IP address of ironic-conductor node's TFTP server.

Configuration option = Default value	Type	Description
uefi_ipxe_bootfile_name = ipxe.efi	string value	Bootfile DHCP parameter for UEFI boot mode. If you experience problems with booting using it, try snponly.efi.
uefi_pxe_bootfile_name = bootx64.efi	string value	Bootfile DHCP parameter for UEFI boot mode.
uefi_pxe_config_template = \$pybasedir/drivers/modules/pxe_grub_config.template	string value	On ironic-conductor node, template file for PXE configuration for UEFI boot loader. Generally this is used for GRUB specific templates.

5.1.43. redfish

The following table outlines the options available under the **[redfish]** group in the **/etc/ironic/ironic.conf** file.

Table 5.42. redfish

Configuration option = Default value	Type	Description
auth_type = auto	string value	Redfish HTTP client authentication method.
connection_attempts = 5	integer value	Maximum number of attempts to try to connect to Redfish
connection_cache_size = 1000	integer value	Maximum Redfish client connection cache size. Redfish driver would strive to reuse authenticated BMC connections (obtained through Redfish Session Service). This option caps the maximum number of connections to maintain. The value of 0 disables client connection caching completely.
connection_retry_interval = 4	integer value	Number of seconds to wait between attempts to connect to Redfish
file_permission = 420	integer value	File permission for swift-less image hosting with the octal permission representation of file access permissions. This setting defaults to 644 , or as the octal number 0o644 in Python. This setting must be set to the octal number representation, meaning starting with 0o .
firmware_update_fail_interval = 60	integer value	Number of seconds to wait between checking for failed firmware update tasks

Configuration option = Default value	Type	Description
firmware_update_status_interval = 60	integer value	Number of seconds to wait between checking for completed firmware update tasks
kernel_append_params = nofb nomodeset vga=normal	string value	Additional kernel parameters to pass down to the instance kernel. These parameters can be consumed by the kernel or by the applications by reading <code>/proc/cmdline</code> . Mind severe cmdline size limit! Can be overridden by instance_info/kernel_append_params property.
raid_config_fail_interval = 60	integer value	Number of seconds to wait between checking for failed raid config tasks
raid_config_status_interval = 60	integer value	Number of seconds to wait between checking for completed raid config tasks
swift_container = ironic_redfish_container	string value	The Swift container to store Redfish driver data. Applies only when use_swift is enabled.
swift_object_expiry_timeout = 900	integer value	Amount of time in seconds for Swift objects to auto-expire. Applies only when use_swift is enabled.
use_swift = True	boolean value	Upload generated ISO images for virtual media boot to Swift, then pass temporary URL to BMC for booting the node. If set to false, images are placed on the ironic-conductor node and served over its local HTTP server.

5.1.44. service_catalog

The following table outlines the options available under the **[service_catalog]** group in the `/etc/ironic/ironic.conf` file.

Table 5.43. service_catalog

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.

Configuration option = Default value	Type	Description
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
max-version = None	string value	The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.
min-version = None	string value	The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is "latest".
password = None	string value	User's password

Configuration option = Default value	Type	Description
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
region-name = None	string value	The default region_name for endpoint URL discovery.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = baremetal	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrieable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User id
username = None	string value	Username

Configuration option = Default value	Type	Description
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.
version = None	string value	Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

5.1.45. snmp

The following table outlines the options available under the **[snmp]** group in the `/etc/ironic/ironic.conf` file.

Table 5.44. snmp

Configuration option = Default value	Type	Description
power_timeout = 10	integer value	Seconds to wait for power action to be completed
reboot_delay = 0	integer value	Time (in seconds) to sleep between when rebooting (powering off and on again)
udp_transport_retries = 5	integer value	Maximum number of UDP request retries, 0 means no retries.
udp_transport_timeout = 1.0	floating point value	Response timeout in seconds used for UDP transport. Timeout should be a multiple of 0.5 seconds and is applicable to each retry.

5.1.46. ssl

The following table outlines the options available under the **[ssl]** group in the `/etc/ironic/ironic.conf` file.

Table 5.45. ssl

Configuration option = Default value	Type	Description
ca_file = None	string value	CA certificate file to use to verify connecting clients.
cert_file = None	string value	Certificate file to use when starting the server securely.
ciphers = None	string value	Sets the list of available ciphers. value should be a string in the OpenSSL cipher list format.

Configuration option = Default value	Type	Description
key_file = None	string value	Private key file to use when starting the server securely.
version = None	string value	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.

5.1.47. swift

The following table outlines the options available under the **[swift]** group in the `/etc/ironic/ironic.conf` file.

Table 5.46. swift

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

Configuration option = Default value	Type	Description
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
max-version = None	string value	The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.
min-version = None	string value	The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is "latest".
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
region-name = None	string value	The default region_name for endpoint URL discovery.
service-name = None	string value	The default service_name for endpoint URL discovery.

Configuration option = Default value	Type	Description
service-type = object-store	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrievable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
swift_max_retries = 2	integer value	Maximum number of times to retry a Swift request, before failing.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User id
username = None	string value	Username
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.
version = None	string value	Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

5.1.48. xclarity

The following table outlines the options available under the **[xclarity]** group in the **/etc/ironic/ironic.conf** file.

Table 5.47. xclarity

Configuration option = Default value	Type	Description
manager_ip = None	string value	IP address of the XClarity Controller. Configuration here is deprecated and will be removed in the Stein release. Please update the driver_info field to use "xclarity_manager_ip" instead
password = None	string value	Password for XClarity Controller username. Configuration here is deprecated and will be removed in the Stein release. Please update the driver_info field to use "xclarity_password" instead
port = 443	port value	Port to be used for XClarity Controller connection.
username = None	string value	Username for the XClarity Controller. Configuration here is deprecated and will be removed in the Stein release. Please update the driver_info field to use "xclarity_username" instead

CHAPTER 6. IRONIC-INSPECTOR

The following chapter contains information about the configuration options in the **ironic-inspector** service.

6.1. INSPECTOR.CONF

This section contains options for the **/etc/ironic-inspector/inspector.conf** file.

6.1.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the **/etc/ironic-inspector/inspector.conf** file.

Configuration option = Default value	Type	Description
api_max_limit = 1000	integer value	Limit the number of elements an API list-call returns
auth_strategy = keystone	string value	Authentication method used on the ironic-inspector API. "noauth", "keystone" or "http_basic" are valid options. "noauth" will disable all authentication.
can_manage_boot = True	boolean value	Whether the current installation of ironic-inspector can manage PXE booting of nodes. If set to False, the API will reject introspection requests with manage_boot missing or set to True.
clean_up_period = 60	integer value	Amount of time in seconds, after which repeat clean up of timed out nodes and old nodes status information. WARNING: If set to a value of 0, then the periodic task is disabled and inspector will not sync with ironic to complete the internal clean-up process. Not advisable if the deployment uses a PXE filter, and will result in the ironic-inspector ceasing periodic cleanup activities.
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.

Configuration option = Default value	Type	Description
default_log_levels = ['sqlalchemy=WARNING', 'iso8601=WARNING', 'requests=WARNING', 'urllib3.connectionpool= WARNING', 'keystonemiddleware=WA RNING', 'keystoneauth=WARNING' , 'ironicclient=WARNING', 'amqp=WARNING', 'amqpplib=WARNING', 'oslo.messaging=WARNI NG', 'oslo_messaging=WARNI NG']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
enable_mdns = False	boolean value	Whether to enable publishing the ironic-inspector API endpoint via multicast DNS.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
host = <based on operating system>	string value	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.
http_basic_auth_user_file = /etc/ironic-inspector/htpasswd	string value	Path to Apache format user authentication file used when auth_strategy=http_basic
<code>`instance_format = [instance: %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.
<code>`instance_uuid_format = [instance: %(uuid)s] `</code>	string value	The format for an instance UUID that is passed with the log message.
introspection_delay = 5	integer value	Delay (in seconds) between two introspections. Only applies when boot is managed by ironic-inspector (i.e. manage_boot==True).
ipmi_address_fields = ['redfish_address', 'ilo_address', 'drac_host', 'drac_address', 'ibmc_address']	list value	Ironic driver_info fields that are equivalent to ipmi_address.

Configuration option = Default value	Type	Description
leader_election_interval = 10	integer value	Interval (in seconds) between leader elections.
listen_address = ::	string value	IP to listen on.
listen_port = 5050	port value	Port to listen on.
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code> . This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code> . This option is ignored if <code>log_config_append</code> is set.
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless <code>log_rotation_type</code> is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	string value	Format string to use for log messages with context. Used by <code>oslo_log.formatters.ContextFormatter</code>

Configuration option = Default value	Type	Description
logging_debug_format_suffix = %(funcName)s %(pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is DEBUG. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_default_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s	string value	Format string to use for log messages when context is undefined. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s	string value	Prefix each line of exception output with this format. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_user_identity_format = %(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s	string value	Defines the format string for <code>%(user_identity)s</code> that is used in <code>logging_context_format_string</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
max_concurrency = 1000	integer value	The green thread pool size.
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
publish_errors = False	boolean value	Enables or disables publication of error events.
rate_limit_burst = 0	integer value	Maximum number of logged messages per <code>rate_limit_interval</code> .
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to <code>rate_limit_except_level</code> are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
rootwrap_config = /etc/ironic-inspector/rootwrap.conf	string value	Path to the rootwrap configuration file to use for running commands as root

Configuration option = Default value	Type	Description
standalone = True	boolean value	Whether to run ironic-inspector as a standalone service. It's EXPERIMENTAL to set to False.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
timeout = 3600	integer value	Timeout after which introspection is considered failed, set to 0 to disable.
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_ssl = False	boolean value	SSL Enabled/Disabled
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

6.1.2. capabilities

The following table outlines the options available under the **[capabilities]** group in the **/etc/ironic-inspector/inspector.conf** file.

Table 6.1. capabilities

Configuration option = Default value	Type	Description
boot_mode = False	boolean value	Whether to store the boot mode (BIOS or UEFI).
cpu_flags = {'aes': 'cpu_aes', 'pdpe1gb': 'cpu_hugepages_1g', 'pse': 'cpu_hugepages', 'smx': 'cpu_txt', 'svm': 'cpu_vt', 'vmx': 'cpu_vt'}	dict value	Mapping between a CPU flag and a capability to set if this flag is present.

6.1.3. coordination

The following table outlines the options available under the **[coordination]** group in the **/etc/ironic-inspector/inspector.conf** file.

Table 6.2. coordination

Configuration option = Default value	Type	Description
backend_url = memcached://localhost:1 1211	string value	The backend URL to use for distributed coordination. EXPERIMENTAL.

6.1.4. cors

The following table outlines the options available under the **[cors]** group in the **/etc/ironic-inspector/inspector.conf** file.

Table 6.3. cors

Configuration option = Default value	Type	Description
allow_credentials = True	boolean value	Indicate that the actual request can include user credentials
allow_headers = ['X-Auth-Token', 'X-OpenStack-Ironic-Inspector-API-Minimum-Version', 'X-OpenStack-Ironic-Inspector-API-Maximum-Version', 'X-OpenStack-Ironic-Inspector-API-Version']	list value	Indicate which header field names may be used during the actual request.

Configuration option = Default value	Type	Description
allow_methods = ['GET', 'POST', 'PUT', 'HEAD', 'PATCH', 'DELETE', 'OPTIONS']	list value	Indicate which methods can be used during the actual request.
allowed_origin = None	list value	Indicate whether this resource may be shared with the domain received in the requests "origin" header. Format: "<protocol>://<host>[:<port>]", no trailing slash. Example: https://horizon.example.com
expose_headers = []	list value	Indicate which headers are safe to expose to the API. Defaults to HTTP Simple Headers.
max_age = 3600	integer value	Maximum cache age of CORS preflight requests.

6.1.5. database

The following table outlines the options available under the **[database]** group in the `/etc/ironic-inspector/inspector.conf` file.

Table 6.4. database

Configuration option = Default value	Type	Description
backend = sqlalchemy	string value	The back end to use for the database.
connection = None	string value	The SQLAlchemy connection string to use to connect to the database.
connection_debug = 0	integer value	Verbosity of SQL debugging information: 0=None, 100=Everything.
connection_parameters = ``	string value	Optional URL parameters to append onto the connection URL at connect time; specify as param1=value1&param2=value2&...
connection_recycle_time = 3600	integer value	Connections which have been present in the connection pool longer than this number of seconds will be replaced with a new one the next time they are checked out from the pool.
connection_trace = False	boolean value	Add Python stack traces to SQL as comment strings.

Configuration option = Default value	Type	Description
db_inc_retry_interval = True	boolean value	If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = 20	integer value	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	integer value	If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = 1	integer value	Seconds between retries of a database transaction.
max_overflow = 50	integer value	If set, use this value for max_overflow with SQLAlchemy.
max_pool_size = 5	integer value	Maximum number of SQL connections to keep open in a pool. Setting a value of 0 indicates no limit.
max_retries = 10	integer value	Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.
mysql_enable_ndb = False	boolean value	If True, transparently enables support for handling MySQL Cluster (NDB).
mysql_sql_mode = TRADITIONAL	string value	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	integer value	If set, use this value for pool_timeout with SQLAlchemy.
retry_interval = 10	integer value	Interval between retries of opening a SQL connection.
slave_connection = None	string value	The SQLAlchemy connection string to use to connect to the slave database.
sqlite_synchronous = True	boolean value	If True, SQLite uses synchronous mode.
use_db_reconnect = False	boolean value	Enable the experimental use of database reconnect on connection lost.

6.1.6. discovery

The following table outlines the options available under the **[discovery]** group in the **/etc/ironic-inspector/inspector.conf** file.

Table 6.5. discovery

Configuration option = Default value	Type	Description
enabled_bmc_address_version = ['4', '6']	list value	IP version of BMC address that will be used when enrolling a new node in Ironic. Defaults to "4,6". Could be "4" (use v4 address only), "4,6" (v4 address have higher priority and if both addresses found v6 version is ignored), "6,4" (v6 is desired but fall back to v4 address for BMCs having v4 address, opposite to "4,6"), "6" (use v6 address only and ignore v4 version).
enroll_node_driver = fake-hardware	string value	The name of the Ironic driver used by the enroll hook when creating a new node in Ironic.
enroll_node_fields = {}	dict value	Additional fields to set on newly discovered nodes.

6.1.7. dnsmasq_pxe_filter

The following table outlines the options available under the **[dnsmasq_pxe_filter]** group in the **/etc/ironic-inspector/inspector.conf** file.

Table 6.6. dnsmasq_pxe_filter

Configuration option = Default value	Type	Description
dhcp_hostsdire = /var/lib/ironic-inspector/dhcp-hostsdire	string value	The MAC address cache directory, exposed to dnsmasq. This directory is expected to be in exclusive control of the driver.
dnsmasq_start_command = `	string value	A (shell) command line to start the dnsmasq service upon filter initialization. Default: don't start.
dnsmasq_stop_command = `	string value	A (shell) command line to stop the dnsmasq service upon inspector (error) exit. Default: don't stop.
purge_dhcp_hostsdire = True	boolean value	Purge the hostsdire upon driver initialization. Setting to false should only be performed when the deployment of inspector is such that there are multiple processes executing inside of the same host and namespace. In this case, the Operator is responsible for setting up a custom cleaning facility.

6.1.8. extra_hardware

The following table outlines the options available under the **[extra_hardware]** group in the `/etc/ironic-inspector/inspector.conf` file.

Table 6.7. extra_hardware

Configuration option = Default value	Type	Description
strict = False	boolean value	If True, refuse to parse extra data if at least one record is too short. Additionally, remove the incoming "data" even if parsing failed.

6.1.9. iptables

The following table outlines the options available under the **[iptables]** group in the `/etc/ironic-inspector/inspector.conf` file.

Table 6.8. iptables

Configuration option = Default value	Type	Description
dnsmasq_interface = br-ctlplane	string value	Interface on which dnsmasq listens, the default is for VM's.
ethoib_interfaces = []	list value	List of Ethernet Over InfiniBand interfaces on the Inspector host which are used for physical access to the DHCP network. Multiple interfaces would be attached to a bond or bridge specified in <code>dnsmasq_interface</code> . The MACs of the InfiniBand nodes which are not in desired state are going to be blacklisted based on the list of neighbor MACs on these interfaces.
firewall_chain = ironic-inspector	string value	iptables chain name to use.
ip_version = 4	string value	The IP version that will be used for iptables filter. Defaults to 4.

6.1.10. ironic

The following table outlines the options available under the **[ironic]** group in the `/etc/ironic-inspector/inspector.conf` file.

Table 6.9. ironic

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
max-version = None	string value	The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.

Configuration option = Default value	Type	Description
max_retries = 30	integer value	Maximum number of retries in case of conflict error (HTTP 409).
min-version = None	string value	The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is "latest".
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
region-name = None	string value	The default region_name for endpoint URL discovery.
retry_interval = 2	integer value	Interval between retries in case of conflict error (HTTP 409).
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = baremetal	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrieable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name

Configuration option = Default value	Type	Description
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User id
username = None	string value	Username
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.
version = None	string value	Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

6.1.11. keystone_authtoken

The following table outlines the options available under the **[keystone_authtoken]** group in the **/etc/ironic-inspector/inspector.conf** file.

Table 6.10. keystone_authtoken

Configuration option = Default value	Type	Description
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load

Configuration option = Default value	Type	Description
auth_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint. This option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release. Deprecated since: Queens <i>*Reason:</i>*The <code>auth_uri</code> option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release.
auth_version = None	string value	API version of the Identity API endpoint.
cache = None	string value	Request environment key where the Swift cache object is stored. When <code>auth_token</code> middleware is deployed with a Swift cache, use this option to have the middleware share a caching backend with swift. Otherwise, use the memcached_servers option instead.
cafile = None	string value	A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
certfile = None	string value	Required if identity server requires client certificate
delay_auth_decision = False	boolean value	Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components.
enforce_token_bind = permissive	string value	Used to control the use and type of token binding. Can be set to: "disabled" to not check token binding. "permissive" (default) to validate binding information if the bind type is of a form known to the server and ignore it if not. "strict" like "permissive" but if the bind type is unknown the token will be rejected. "required" any form of token binding is needed to be allowed. Finally the name of a binding method that must be present in tokens.
http_connect_timeout = None	integer value	Request timeout value for communicating with Identity API server.

Configuration option = Default value	Type	Description
http_request_max_retries = 3	integer value	How many times are we trying to reconnect when communicating with Identity API Server.
include_service_catalog = True	boolean value	(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	boolean value	Verify HTTPS connections.
interface = internal	string value	Interface to use for the Identity API endpoint. Valid values are "public", "internal" (default) or "admin".
keyfile = None	string value	Required if identity server requires client certificate
memcache_pool_conn_get_timeout = 10	integer value	(Optional) Number of seconds that an operation will wait to get a memcached client connection from the pool.
memcache_pool_dead_retry = 300	integer value	(Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = 10	integer value	(Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = 3	integer value	(Optional) Socket timeout in seconds for communicating with a memcached server.
memcache_pool_unused_timeout = 60	integer value	(Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
memcache_secret_key = None	string value	(Optional, mandatory if memcache_security_strategy is defined) This string is used for key derivation.
memcache_security_strategy = None	string value	(Optional) If defined, indicate whether token data should be authenticated or authenticated and encrypted. If MAC, token data is authenticated (with HMAC) in the cache. If ENCRYPT, token data is encrypted and authenticated in the cache. If the value is not one of these options or empty, auth_token will raise an exception on initialization.
memcache_use_advanced_pool = False	boolean value	(Optional) Use the advanced (eventlet safe) memcached client pool. The advanced pool will only work under python 2.x.

Configuration option = Default value	Type	Description
memcached_servers = None	list value	Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
region_name = None	string value	The region in which the identity server can be found.
service_token_roles = ['service']	list value	A choice of roles that must be present in a service token. Service tokens are allowed to request that an expired token can be used and so this check should tightly control that only actual services should be sending this token. Roles here are applied as an ANY check so any role in this list must be present. For backwards compatibility reasons this currently only affects the allow_expired check.
service_token_roles_required = False	boolean value	For backwards compatibility reasons we must let valid service tokens pass that don't pass the service_token_roles check as valid. Setting this true will become the default in a future release and should be enabled if possible.
service_type = None	string value	The name or type of the service as it appears in the service catalog. This is used to validate tokens that have restricted access rules.
token_cache_time = 300	integer value	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.
www_authenticate_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint.

6.1.12. oslo_policy

The following table outlines the options available under the **[oslo_policy]** group in the **/etc/ironic-inspector/inspector.conf** file.

Table 6.11. oslo_policy

Configuration option = Default value	Type	Description
enforce_new_defaults = False	boolean value	This option controls whether or not to use old deprecated defaults when evaluating policies. If True , the old deprecated defaults are not going to be evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be disallowed. It is encouraged to enable this flag along with the enforce_scope flag so that you can get the benefits of new defaults and scope_type together
enforce_scope = False	boolean value	This option controls whether or not to enforce scope when evaluating policies. If True , the scope of the token used in the request is compared to the scope_types of the policy being enforced. If the scopes do not match, an InvalidScope exception will be raised. If False , a message will be logged informing operators that policies are being invoked with mismatching scope.
policy_default_rule = default	string value	Default rule. Enforced when a requested rule is not found.
policy_dirs = ['policy.d']	multi valued	Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the <code>config_dir</code> option, or absolute paths. The file defined by <code>policy_file</code> must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = policy.json	string value	The relative or absolute path of a file that maps roles to permissions for a given service. Relative paths must be specified in relation to the configuration file setting this option.
remote_content_type = application/x-www-form-urlencoded	string value	Content Type to send and receive data for REST based policy check
remote_ssl_ca_cert_file = None	string value	Absolute path to ca cert file for REST based policy check
remote_ssl_client_cert_file = None	string value	Absolute path to client cert for REST based policy check
remote_ssl_client_key_file = None	string value	Absolute path client key file REST based policy check

Configuration option = Default value	Type	Description
remote_ssl_verify_server _cert = False	boolean value	server identity verification for REST based policy check

6.1.13. pci_devices

The following table outlines the options available under the **[pci_devices]** group in the **/etc/ironic-inspector/inspector.conf** file.

Table 6.12. pci_devices

Configuration option = Default value	Type	Description
alias = []	multi valued	An alias for PCI device identified by <i>vendor_id</i> and <i>product_id</i> fields. Format: {"vendor_id": "1234", "product_id": "5678", "name": "pci_dev1"}

6.1.14. port_physnet

The following table outlines the options available under the **[port_physnet]** group in the **/etc/ironic-inspector/inspector.conf** file.

Table 6.13. port_physnet

Configuration option = Default value	Type	Description
cidr_map = []	list value	Mapping of IP subnet CIDR to physical network. When the <i>physnet_cidr_map</i> processing hook is enabled the <i>physical_network</i> property of baremetal ports is populated based on this mapping.

6.1.15. processing

The following table outlines the options available under the **[processing]** group in the **/etc/ironic-inspector/inspector.conf** file.

Table 6.14. processing

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
add_ports = pxe	string value	Which MAC addresses to add as ports during introspection. Possible values: all (all MAC addresses), active (MAC addresses of NIC with IP addresses), pxe (only MAC address of NIC node PXE booted from, falls back to "active" if PXE MAC is not supplied by the ramdisk).
always_store_ramdisk_logs = False	boolean value	Whether to store ramdisk logs even if it did not return an error message (dependent upon "ramdisk_logs_dir" option being set).
default_processing_hooks = ramdisk_error,root_disk_selection,scheduler,validate_interfaces,capabilities,pci_devices	string value	Comma-separated list of default hooks for processing pipeline. Hook <i>scheduler</i> updates the node with the minimum properties required by the Nova scheduler. Hook <i>validate_interfaces</i> ensures that valid NIC data was provided by the ramdisk. Do not exclude these two unless you really know what you're doing.
disk_partitioning_spacing = True	boolean value	Whether to leave 1 GiB of disk size untouched for partitioning. Only has effect when used with the IPA as a ramdisk, for older ramdisk local_gb is calculated on the ramdisk side.
keep_ports = all	string value	Which ports (already present on a node) to keep after introspection. Possible values: all (do not delete anything), present (keep ports which MACs were present in introspection data), added (keep only MACs that we added during introspection).
node_not_found_hook = None	string value	The name of the hook to run when inspector receives inspection information from a node it isn't already aware of. This hook is ignored by default.
overwrite_existing = True	boolean value	Whether to overwrite existing values in node database. Disable this option to make introspection a non-destructive operation.
permit_active_introspection = False	boolean value	Whether to process nodes that are in running states.
power_off = True	boolean value	Whether to power off a node after introspection. Nodes in active or rescue states which submit introspection data will be left on if the feature is enabled via the <i>permit_active_introspection</i> configuration option.

Configuration option = Default value	Type	Description
processing_hooks = \$default_processing_hooks	string value	Comma-separated list of enabled hooks for processing pipeline. The default for this is \$default_processing_hooks, hooks can be added before or after the defaults like this: "prehook,\$default_processing_hooks,posthook".
ramdisk_logs_dir = None	string value	If set, logs from ramdisk will be stored in this directory.
ramdisk_logs_filename_format = {uuid}_{dt:%Y%m%d-%H%M%S.%f}.tar.gz	string value	File name template for storing ramdisk logs. The following replacements can be used: {uuid} - node UUID or "unknown", {bmc} - node BMC address or "unknown", {dt} - current UTC date and time, {mac} - PXE booting MAC or "unknown".
store_data = none	string value	The storage backend for storing introspection data. Possible values are: <i>none</i> , <i>database</i> and <i>swift</i> . If set to <i>none</i> , introspection data will not be stored.
update_pxe_enabled = True	boolean value	Whether to update the pxe_enabled value according to the introspection data. This option has no effect if [processing]overwrite_existing is set to False

6.1.16. pxe_filter

The following table outlines the options available under the **[pxe_filter]** group in the `/etc/ironic-inspector/inspector.conf` file.

Table 6.15. pxe_filter

Configuration option = Default value	Type	Description
deny_unknown_macs = False	boolean value	By default inspector will open the DHCP server for any node when introspection is active. Opening DHCP for unknown MAC addresses when introspection is active allow for users to add nodes with no ports to ironic and have ironic-inspector enroll ports based on node introspection results. NOTE: If this option is True, nodes must have at least one enrolled port prior to introspection.
driver = iptables	string value	PXE boot filter driver to use, possible filters are: "iptables", "dnsmasq" and "noop". Set "noop" to disable the firewall filtering.

Configuration option = Default value	Type	Description
sync_period = 15	integer value	Amount of time in seconds, after which repeat periodic update of the filter.

6.1.17. service_catalog

The following table outlines the options available under the **[service_catalog]** group in the **/etc/ironic-inspector/inspector.conf** file.

Table 6.16. service_catalog

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to

Configuration option = Default value	Type	Description
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
max-version = None	string value	The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.
min-version = None	string value	The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is "latest".
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
region-name = None	string value	The default region_name for endpoint URL discovery.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = baremetal-introspection	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrievable HTTP status codes.

Configuration option = Default value	Type	Description
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User id
username = None	string value	Username
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.
version = None	string value	Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

6.1.18. swift

The following table outlines the options available under the **[swift]** group in the `/etc/ironic-inspector/inspector.conf` file.

Table 6.17. swift

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_type = None	string value	Authentication type to load

Configuration option = Default value	Type	Description
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
container = ironic-inspector	string value	Default Swift container to use when creating objects.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
delete_after = 0	integer value	Number of seconds that the Swift object will last before being deleted. (set to 0 to never delete the object).
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file

Configuration option = Default value	Type	Description
max-version = None	string value	The maximum major version of a given API, intended to be used as the upper bound of a range with min_version. Mutually exclusive with version.
min-version = None	string value	The minimum major version of a given API, intended to be used as the lower bound of a range with max_version. Mutually exclusive with version. If min_version is given with no max_version it is as if max version is "latest".
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
region-name = None	string value	The default region_name for endpoint URL discovery.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = object-store	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrieable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name

Configuration option = Default value	Type	Description
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User id
username = None	string value	Username
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.
version = None	string value	Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min_version and max_version

CHAPTER 7. KEYSTONE

The following chapter contains information about the configuration options in the **keystone** service.

7.1. KEYSTONE.CONF

This section contains options for the `/etc/keystone/keystone.conf` file.

7.1.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/keystone/keystone.conf` file.

Configuration option = Default value	Type	Description
admin_token = None	string value	Using this feature is NOT recommended. Instead, use the keystone-manage bootstrap command. The value of this option is treated as a "shared secret" that can be used to bootstrap Keystone through the API. This "token" does not represent a user (it has no identity), and carries no explicit authorization (it effectively bypasses most authorization checks). If set to None , the value is ignored and the admin_token middleware is effectively disabled.
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
control_exchange = keystone	string value	The default exchange under which topics are scoped. May be overridden by an exchange name specified in the <code>transport_url</code> option.
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.

Configuration option = Default value	Type	Description
default_log_levels = <code>['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpids=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' ', 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']</code>	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
default_publisher_id = None	string value	Default publisher_id for outgoing notifications. If left undefined, Keystone will default to using the server's host name.
executor_thread_pool_siz e = 64	integer value	Size of executor thread pool when executor is threading or eventlet.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
insecure_debug = False	boolean value	If set to true, then the server will return information in HTTP responses that may allow an unauthenticated or authenticated user to get more information than normal, such as additional details about why authentication failed. This may be useful for debugging but is insecure.
<code>`instance_format = [instance: %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.

Configuration option = Default value	Type	Description
<code>`instance_uuid_format = [instance: %(uuid)s] `</code>	string value	The format for an instance UUID that is passed with the log message.
list_limit = None	integer value	The maximum number of entities that will be returned in a collection. This global limit may be then overridden for a specific driver, by specifying a <code>list_limit</code> in the appropriate section (for example, [assignment]). No limit is set by default. In larger deployments, it is recommended that you set this to a reasonable number to prevent operations like listing all users and projects from placing an unnecessary load on the system.
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, <code>log-date-format</code>).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code> . This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code> . This option is ignored if <code>log_config_append</code> is set.
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless <code>log_rotation_type</code> is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.

Configuration option = Default value	Type	Description
logging_context_format_string = <code>%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s</code>	string value	Format string to use for log messages with context. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_debug_format_s uffix = <code>%(funcName)s %(pathname)s:%(lineno)d</code>	string value	Additional data to append to log message when logging level for the message is <code>DEBUG</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_default_format_s tring = <code>%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s</code>	string value	Format string to use for log messages when context is undefined. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_exception_prefix = <code>%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s</code>	string value	Prefix each line of exception output with this format. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_user_identity_for mat = <code>%(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s</code>	string value	Defines the format string for <code>%(user_identity)s</code> that is used in <code>logging_context_format_string</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
max_param_size = 64	integer value	Limit the sizes of user & project ID/names.
max_project_tree_depth = 5	integer value	Maximum depth of the project hierarchy, excluding the project acting as a domain at the top of the hierarchy. WARNING: Setting it to a large value may adversely impact performance.
max_token_size = 255	integer value	Similar to [DEFAULT] max_param_size , but provides an exception for token values. With Fernet tokens, this can be set as low as 255.

Configuration option = Default value	Type	Description
notification_format = cadf	string value	Define the notification format for identity service events. A basic notification only 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. The cadf option is entirely backwards compatible with the basic option, but is fully CADF-compliant, and is recommended for auditing use cases.
notification_opt_out = ['identity.authenticate.success', 'identity.authenticate.pending', 'identity.authenticate.failed']	multi valued	You can reduce the number of notifications keystone emits by explicitly opting out. Keystone will not emit notifications that match the patterns expressed in this list. Values are expected to be in the form of identity.<resource_type>.<operation> . By default, all notifications related to authentication are automatically suppressed. This field can be set multiple times in order to opt-out of multiple notification topics. For example, the following suppresses notifications describing user creation or successful authentication events: notification_opt_out=identity.user.create notification_opt_out=identity.authenticate.success
public_endpoint = None	uri value	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. For example, if keystone receives a request to http://server:5000/v3/users , then this will option will be automatically treated as http://server:5000 . You should only need to set option if either the value of the base URL contains a path that keystone does not automatically infer (/prefix/v3), or if the endpoint should be found on a different host.
publish_errors = False	boolean value	Enables or disables publication of error events.
rate_limit_burst = 0	integer value	Maximum number of logged messages per rate_limit_interval.
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.

Configuration option = Default value	Type	Description
rpc_conn_pool_size = 30	integer value	Size of RPC connection pool.
rpc_ping_enabled = False	boolean value	Add an endpoint to answer to ping calls. Endpoint is named oslo_rpc_server_ping
rpc_response_timeout = 60	integer value	Seconds to wait for a response from a call.
strict_password_check = False	boolean value	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.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
transport_url = rabbit://	string value	The network address and optional user credentials for connecting to the messaging backend, in URL format. The expected format is: <pre>driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query</pre> Example: rabbit://rabbitmq:password@127.0.0.1:5672// For full details on the fields in the URL see the documentation of oslo_messaging.TransportURL at https://docs.openstack.org/oslo.messaging/latest/reference/transport.html
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

7.1.2. application_credential

The following table outlines the options available under the **[application_credential]** group in the **/etc/keystone/keystone.conf** file.

Table 7.1. application_credential

Configuration option = Default value	Type	Description
cache_time = None	integer value	Time to cache application credential data in seconds. This has no effect unless global caching is enabled.
caching = True	boolean value	Toggle for application credential caching. This has no effect unless global caching is enabled.
driver = sql	string value	Entry point for the application credential backend driver in the keystone.application_credential namespace. Keystone only provides a sql driver, so there is no reason to change this unless you are providing a custom entry point.
user_limit = -1	integer value	Maximum number of application credentials a user is permitted to create. A value of -1 means unlimited. If a limit is not set, users are permitted to create application credentials at will, which could lead to bloat in the keystone database or open keystone to a DoS attack.

7.1.3. assignment

The following table outlines the options available under the **[assignment]** group in the **/etc/keystone/keystone.conf** file.

Table 7.2. assignment

Configuration option = Default value	Type	Description
driver = sql	string value	Entry point for the assignment backend driver (where role assignments are stored) in the keystone.assignment namespace. Only a SQL driver is supplied by keystone itself. Unless you are writing proprietary drivers for keystone, you do not need to set this option.
prohibited_implied_role = ['admin']	list value	A list of role names which are prohibited from being an implied role.

7.1.4. auth

The following table outlines the options available under the **[auth]** group in the **/etc/keystone/keystone.conf** file.

Table 7.3. auth

Configuration option = Default value	Type	Description
application_credential = None	string value	Entry point for the application_credential auth plugin module in the keystone.auth.application_credential namespace. You do not need to set this unless you are overriding keystone's own application_credential authentication plugin.
external = None	string value	Entry point for the external (REMOTE_USER) auth plugin module in the keystone.auth.external namespace. Supplied drivers are DefaultDomain and Domain . The default driver is DefaultDomain , which assumes that all users identified by the username specified to keystone in the REMOTE_USER variable exist within the context of the default domain. The Domain option expects an additional environment variable be presented to keystone, REMOTE_DOMAIN , containing the domain name of the REMOTE_USER (if REMOTE_DOMAIN is not set, then the default domain will be used instead). You do not need to set this unless you are taking advantage of "external authentication", where the application server (such as Apache) is handling authentication instead of keystone.
mapped = None	string value	Entry point for the mapped auth plugin module in the keystone.auth.mapped namespace. You do not need to set this unless you are overriding keystone's own mapped authentication plugin.

Configuration option = Default value	Type	Description
methods = ['external', 'password', 'token', 'oauth1', 'mapped', 'application_credential']	list value	Allowed authentication methods. Note: You should disable the external auth method if you are currently using federation. External auth and federation both use the REMOTE_USER variable. Since both the mapped and external plugin are being invoked to validate attributes in the request environment, it can cause conflicts.
oauth1 = None	string value	Entry point for the OAuth 1.0a auth plugin module in the keystone.auth.oauth1 namespace. You do not need to set this unless you are overriding keystone's own oauth1 authentication plugin.
password = None	string value	Entry point for the password auth plugin module in the keystone.auth.password namespace. You do not need to set this unless you are overriding keystone's own password authentication plugin.
token = None	string value	Entry point for the token auth plugin module in the keystone.auth.token namespace. You do not need to set this unless you are overriding keystone's own token authentication plugin.

7.1.5. cache

The following table outlines the options available under the **[cache]** group in the **/etc/keystone/keystone.conf** file.

Table 7.4. cache

Configuration option = Default value	Type	Description
backend = dogpile.cache.null	string value	Cache backend module. For eventlet-based or environments with hundreds of threaded servers, Memcache with pooling (oslo_cache.memcache_pool) is recommended. For environments with less than 100 threaded servers, Memcached (dogpile.cache.memcached) or Redis (dogpile.cache.redis) is recommended. Test environments with a single instance of the server can use the dogpile.cache.memory backend.
backend_argument = []	multi valued	Arguments supplied to the backend module. Specify this option once per argument to be passed to the dogpile.cache backend. Example format: "<argname>:<value>".

Configuration option = Default value	Type	Description
config_prefix = cache.oslo	string value	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.
dead_timeout = 60	floating point value	Time in seconds before attempting to add a node back in the pool in the HashClient's internal mechanisms.
debug_cache_backend = False	boolean value	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.
enable_retry_client = False	boolean value	Enable retry client mechanisms to handle failure. Those mechanisms can be used to wrap all kind of pymemcache clients. The wrapper allows you to define how many attempts to make and how long to wait between attempts.
enable_socket_keepalive = False	boolean value	Global toggle for the socket keepalive of dogpile's pymemcache backend
enabled = True	boolean value	Global toggle for caching.
expiration_time = 600	integer value	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.
hashclient_retry_attempts = 2	integer value	Amount of times a client should be tried before it is marked dead and removed from the pool in the HashClient's internal mechanisms.
hashclient_retry_delay = 1	floating point value	Time in seconds that should pass between retry attempts in the HashClient's internal mechanisms.
memcache_dead_retry = 300	integer value	Number of seconds memcached server is considered dead before it is tried again. (dogpile.cache.memcache and oslo_cache.memcache_pool backends only).
memcache_pool_connection_get_timeout = 10	integer value	Number of seconds that an operation will wait to get a memcache client connection.

Configuration option = Default value	Type	Description
memcache_pool_flush_on_reconnect = False	boolean value	Global toggle if memcache will be flushed on reconnect. (oslo_cache.memcache_pool backend only).
memcache_pool_maxsize = 10	integer value	Max total number of open connections to every memcached server. (oslo_cache.memcache_pool backend only).
memcache_pool_unused_timeout = 60	integer value	Number of seconds a connection to memcached is held unused in the pool before it is closed. (oslo_cache.memcache_pool backend only).
memcache_servers = ['localhost:11211']	list value	Memcache servers in the format of "host:port". (dogpile.cache.memcached and oslo_cache.memcache_pool backends only). If a given host refer to an IPv6 or a given domain refer to IPv6 then you should prefix the given address with the address family (inet6) (e.g inet6:::1]:11211 , inet6:[fd12:3456:789a:1::1]:11211 , inet6:[controller-0.internalapi]:11211). If the address family is not given then default address family used will be inet which correspond to IPv4
memcache_socket_timeout = 1.0	floating point value	Timeout in seconds for every call to a server. (dogpile.cache.memcache and oslo_cache.memcache_pool backends only).
proxies = []	list value	Proxy classes to import that will affect the way the dogpile.cache backend functions. See the dogpile.cache documentation on changing-backend-behavior.
retry_attempts = 2	integer value	Number of times to attempt an action before failing.
retry_delay = 0	floating point value	Number of seconds to sleep between each attempt.
socket_keepalive_count = 1	integer value	The maximum number of keepalive probes TCP should send before dropping the connection. Should be a positive integer greater than zero.
socket_keepalive_idle = 1	integer value	The time (in seconds) the connection needs to remain idle before TCP starts sending keepalive probes. Should be a positive integer most greater than zero.

Configuration option = Default value	Type	Description
socket_keepalive_interval = 1	integer value	The time (in seconds) between individual keepalive probes. Should be a positive integer greater than zero.
tls_allowed_ciphers = None	string value	Set the available ciphers for sockets created with the TLS context. It should be a string in the OpenSSL cipher list format. If not specified, all OpenSSL enabled ciphers will be available.
tls_cafile = None	string value	Path to a file of concatenated CA certificates in PEM format necessary to establish the caching servers' authenticity. If <code>tls_enabled</code> is <code>False</code> , this option is ignored.
tls_certfile = None	string value	Path to a single file in PEM format containing the client's certificate as well as any number of CA certificates needed to establish the certificate's authenticity. This file is only required when client side authentication is necessary. If <code>tls_enabled</code> is <code>False</code> , this option is ignored.
tls_enabled = False	boolean value	Global toggle for TLS usage when communicating with the caching servers.
tls_keyfile = None	string value	Path to a single file containing the client's private key in. Otherwise the private key will be taken from the file specified in <code>tls_certfile</code> . If <code>tls_enabled</code> is <code>False</code> , this option is ignored.

7.1.6. catalog

The following table outlines the options available under the **[catalog]** group in the `/etc/keystone/keystone.conf` file.

Table 7.5. catalog

Configuration option = Default value	Type	Description
cache_time = None	integer value	Time to cache catalog data (in seconds). This has no effect unless global and catalog caching are both enabled. Catalog data (services, endpoints, etc.) typically does not change frequently, and so a longer duration than the global default may be desirable.

Configuration option = Default value	Type	Description
caching = True	boolean value	Toggle for catalog caching. This has no effect unless global caching is enabled. In a typical deployment, there is no reason to disable this.
driver = sql	string value	Entry point for the catalog driver in the keystone.catalog namespace. Keystone provides a sql option (which supports basic CRUD operations through SQL), a templated option (which loads the catalog from a templated catalog file on disk), and a endpoint_filter.sql option (which supports arbitrary service catalogs per project).
list_limit = None	integer value	Maximum number of entities that will be returned in a catalog collection. There is typically no reason to set this, as it would be unusual for a deployment to have enough services or endpoints to exceed a reasonable limit.
template_file = default_catalog.templates	string value	Absolute path to the file used for the templated catalog backend. This option is only used if the [catalog] driver is set to templated .

7.1.7. cors

The following table outlines the options available under the **[cors]** group in the **/etc/keystone/keystone.conf** file.

Table 7.6. cors

Configuration option = Default value	Type	Description
allow_credentials = True	boolean value	Indicate that the actual request can include user credentials
allow_headers = ['X-Auth-Token', 'X-Openstack-Request-Id', 'X-Subject-Token', 'X-Project-Id', 'X-Project-Name', 'X-Project-Domain-Id', 'X-Project-Domain-Name', 'X-Domain-Id', 'X-Domain-Name', 'Openstack-Auth-Receipt']	list value	Indicate which header field names may be used during the actual request.

Configuration option = Default value	Type	Description
allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH']	list value	Indicate which methods can be used during the actual request.
allowed_origin = None	list value	Indicate whether this resource may be shared with the domain received in the requests "origin" header. Format: "<protocol>://<host>[:<port>]", no trailing slash. Example: https://horizon.example.com
expose_headers = ['X-Auth-Token', 'X-Openstack-Request-Id', 'X-Subject-Token', 'Openstack-Auth-Receipt']	list value	Indicate which headers are safe to expose to the API. Defaults to HTTP Simple Headers.
max_age = 3600	integer value	Maximum cache age of CORS preflight requests.

7.1.8. credential

The following table outlines the options available under the **[credential]** group in the `/etc/keystone/keystone.conf` file.

Table 7.7. credential

Configuration option = Default value	Type	Description
auth_ttl = 15	integer value	The length of time in minutes for which a signed EC2 or S3 token request is valid from the timestamp contained in the token request.
cache_time = None	integer value	Time to cache credential data in seconds. This has no effect unless global caching is enabled.
caching = True	boolean value	Toggle for caching only on retrieval of user credentials. This has no effect unless global caching is enabled.
driver = sql	string value	Entry point for the credential backend driver in the keystone.credential namespace. Keystone only provides a sql driver, so there's no reason to change this unless you are providing a custom entry point.

Configuration option = Default value	Type	Description
key_repository = /etc/keystone/credential- keys/	string value	Directory containing Fernet keys used to encrypt and decrypt credentials stored in the credential backend. Fernet keys used to encrypt credentials have no relationship to Fernet keys used to encrypt Fernet tokens. Both sets of keys should be managed separately and require different rotation policies. Do not share this repository with the repository used to manage keys for Fernet tokens.
provider = fernet	string value	Entry point for credential encryption and decryption operations in the keystone.credential.provider namespace. Keystone only provides a fernet driver, so there's no reason to change this unless you are providing a custom entry point to encrypt and decrypt credentials.
user_limit = -1	integer value	Maximum number of credentials a user is permitted to create. A value of -1 means unlimited. If a limit is not set, users are permitted to create credentials at will, which could lead to bloat in the keystone database or open keystone to a DoS attack.

7.1.9. database

The following table outlines the options available under the **[database]** group in the **/etc/keystone/keystone.conf** file.

Table 7.8. database

Configuration option = Default value	Type	Description
backend = sqlalchemy	string value	The back end to use for the database.
connection = None	string value	The SQLAlchemy connection string to use to connect to the database.
connection_debug = 0	integer value	Verbosity of SQL debugging information: 0=None, 100=Everything.
connection_parameters = ``	string value	Optional URL parameters to append onto the connection URL at connect time; specify as param1=value1&param2=value2&...

Configuration option = Default value	Type	Description
connection_recycle_time = 3600	integer value	Connections which have been present in the connection pool longer than this number of seconds will be replaced with a new one the next time they are checked out from the pool.
connection_trace = False	boolean value	Add Python stack traces to SQL as comment strings.
db_inc_retry_interval = True	boolean value	If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = 20	integer value	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	integer value	If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = 1	integer value	Seconds between retries of a database transaction.
max_overflow = 50	integer value	If set, use this value for max_overflow with SQLAlchemy.
max_pool_size = 5	integer value	Maximum number of SQL connections to keep open in a pool. Setting a value of 0 indicates no limit.
max_retries = 10	integer value	Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.
mysql_enable_ndb = False	boolean value	If True, transparently enables support for handling MySQL Cluster (NDB).
mysql_sql_mode = TRADITIONAL	string value	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	integer value	If set, use this value for pool_timeout with SQLAlchemy.
retry_interval = 10	integer value	Interval between retries of opening a SQL connection.
slave_connection = None	string value	The SQLAlchemy connection string to use to connect to the slave database.

Configuration option = Default value	Type	Description
sqlite_synchronous = True	boolean value	If True, SQLite uses synchronous mode.
use_db_reconnect = False	boolean value	Enable the experimental use of database reconnect on connection lost.

7.1.10. domain_config

The following table outlines the options available under the **[domain_config]** group in the **/etc/keystone/keystone.conf** file.

Table 7.9. domain_config

Configuration option = Default value	Type	Description
cache_time = 300	integer value	Time-to-live (TTL, in seconds) to cache domain-specific configuration data. This has no effect unless [domain_config] caching is enabled.
caching = True	boolean value	Toggle for caching of the domain-specific configuration backend. This has no effect unless global caching is enabled. There is normally no reason to disable this.
driver = sql	string value	Entry point for the domain-specific configuration driver in the keystone.resource.domain_config namespace. Only a sql option is provided by keystone, so there is no reason to set this unless you are providing a custom entry point.

7.1.11. endpoint_filter

The following table outlines the options available under the **[endpoint_filter]** group in the **/etc/keystone/keystone.conf** file.

Table 7.10. endpoint_filter

Configuration option = Default value	Type	Description
driver = sql	string value	Entry point for the endpoint filter driver in the keystone.endpoint_filter namespace. Only a sql option is provided by keystone, so there is no reason to set this unless you are providing a custom entry point.

Configuration option = Default value	Type	Description
return_all_endpoints_if_no_filter = True	boolean value	This controls keystone's behavior if the configured endpoint filters do not result in any endpoints for a user + project pair (and therefore a potentially empty service catalog). If set to true, keystone will return the entire service catalog. If set to false, keystone will return an empty service catalog.

7.1.12. endpoint_policy

The following table outlines the options available under the **[endpoint_policy]** group in the `/etc/keystone/keystone.conf` file.

Table 7.11. endpoint_policy

Configuration option = Default value	Type	Description
driver = sql	string value	Entry point for the endpoint policy driver in the keystone.endpoint_policy namespace. Only a sql driver is provided by keystone, so there is no reason to set this unless you are providing a custom entry point.

7.1.13. eventlet_server

The following table outlines the options available under the **[eventlet_server]** group in the `/etc/keystone/keystone.conf` file.

Table 7.12. eventlet_server

Configuration option = Default value	Type	Description
admin_bind_host = 0.0.0.0	host address value	The IP address of the network interface for the admin service to listen on. Deprecated since: K *Reason:*Support for running keystone under eventlet has been removed in the Newton release. These options remain for backwards compatibility because they are used for URL substitutions.

Configuration option = Default value	Type	Description
admin_port = 35357	port value	The port number for the admin service to listen on. Deprecated since: K *Reason:*Support for running keystone under eventlet has been removed in the Newton release. These options remain for backwards compatibility because they are used for URL substitutions.
public_bind_host = 0.0.0.0	host address value	The IP address of the network interface for the public service to listen on. Deprecated since: K *Reason:*Support for running keystone under eventlet has been removed in the Newton release. These options remain for backwards compatibility because they are used for URL substitutions.
public_port = 5000	port value	The port number for the public service to listen on. Deprecated since: K *Reason:*Support for running keystone under eventlet has been removed in the Newton release. These options remain for backwards compatibility because they are used for URL substitutions.

7.1.14. federation

The following table outlines the options available under the **[federation]** group in the **/etc/keystone/keystone.conf** file.

Table 7.13. federation

Configuration option = Default value	Type	Description
<code>`assertion_prefix = `</code>	string value	Prefix to use when filtering environment variable names for federated assertions. Matched variables are passed into the federated mapping engine.
caching = True	boolean value	Toggle for federation caching. This has no effect unless global caching is enabled. There is typically no reason to disable this.
default_authorization_ttl = 0	integer value	Default time in minutes for the validity of group memberships carried over from a mapping. Default is 0, which means disabled.

Configuration option = Default value	Type	Description
driver = sql	string value	Entry point for the federation backend driver in the keystone.federation namespace. Keystone only provides a sql driver, so there is no reason to set this option unless you are providing a custom entry point.
federated_domain_name = Federated	string value	An arbitrary 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. Deprecated since: T <i>*Reason:</i>*This option has been superseded by ephemeral users existing in the domain of their identity provider.
remote_id_attribute = None	string value	Default value for all protocols to be used to obtain the entity ID of the Identity Provider from the environment. For mod_shib, this would be Shib-Identity-Provider. For mod_auth_openidc, this could be HTTP_OIDC_ISS. For mod_auth_mellon, this could be MELLON_IDP. This can be overridden on a per-protocol basis by providing a remote_id_attribute to the federation protocol using the API.
sso_callback_template = /etc/keystone/sso_callback_template.html	string value	Absolute path to an HTML file used as a Single Sign-On callback handler. This page is expected to redirect the user from keystone back to a trusted dashboard host, by form encoding a token in a POST request. Keystone's default value should be sufficient for most deployments.
trusted_dashboard = []	multi valued	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 this list. This configuration option may be repeated for multiple values. You must set this in order to use web-based SSO flows. For example: <pre>trusted_dashboard=https://acme.example.com/auth/webssso trusted_dashboard=https://beta.example.com/auth/webssso</pre>

7.1.15. fernet_receipts

The following table outlines the options available under the **[fernet_receipts]** group in the **/etc/keystone/keystone.conf** file.

Table 7.14. fernet_receipts

Configuration option = Default value	Type	Description
key_repository = /etc/keystone/fernet-keys/	string value	Directory containing Fernet receipt keys. This directory must exist before using keystone-manage fernet_setup for the first time, must be writable by the user running keystone-manage fernet_setup or keystone-manage fernet_rotate , and of course must be readable by keystone's server process. The repository may contain keys in one of three states: a single staged key (always index 0) used for receipt validation, a single primary key (always the highest index) used for receipt creation and validation, and any number of secondary keys (all other index values) used for receipt validation. With multiple keystone nodes, each node must share the same key repository contents, with the exception of the staged key (index 0). It is safe to run keystone-manage fernet_rotate once on any one node to promote a staged key (index 0) to be the new primary (incremented from the previous highest index), and produce a new staged key (a new key with index 0); the resulting repository can then be atomically replicated to other nodes without any risk of race conditions (for example, it is safe to run keystone-manage fernet_rotate on host A, wait any amount of time, create a tarball of the directory on host A, unpack it on host B to a temporary location, and atomically move (mv) the directory into place on host B). Running keystone-manage fernet_rotate twice on a key repository without syncing other nodes will result in receipts that can not be validated by all nodes.
max_active_keys = 3	integer value	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 (always index 0), one primary key (the highest numerical index), and one secondary key (every other index). Increasing this value means that additional secondary keys will be kept in the rotation.

7.1.16. fernet_tokens

The following table outlines the options available under the **[fernet_tokens]** group in the **/etc/keystone/keystone.conf** file.

Table 7.15. fernet_tokens

Configuration option = Default value	Type	Description
key_repository = /etc/keystone/fernet-keys/	string value	Directory containing Fernet token keys. This directory must exist before using keystone-manage fernet_setup for the first time, must be writable by the user running keystone-manage fernet_setup or keystone-manage fernet_rotate , and of course must be readable by keystone's server process. The repository may contain keys in one of three states: a single staged key (always index 0) used for token validation, a single primary key (always the highest index) used for token creation and validation, and any number of secondary keys (all other index values) used for token validation. With multiple keystone nodes, each node must share the same key repository contents, with the exception of the staged key (index 0). It is safe to run keystone-manage fernet_rotate once on any one node to promote a staged key (index 0) to be the new primary (incremented from the previous highest index), and produce a new staged key (a new key with index 0); the resulting repository can then be atomically replicated to other nodes without any risk of race conditions (for example, it is safe to run keystone-manage fernet_rotate on host A, wait any amount of time, create a tarball of the directory on host A, unpack it on host B to a temporary location, and atomically move (mv) the directory into place on host B). Running keystone-manage fernet_rotate twice on a key repository without syncing other nodes will result in tokens that can not be validated by all nodes.
max_active_keys = 3	integer value	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 (always index 0), one primary key (the highest numerical index), and one secondary key (every other index). Increasing this value means that additional secondary keys will be kept in the rotation.

7.1.17. healthcheck

The following table outlines the options available under the **[healthcheck]** group in the **/etc/keystone/keystone.conf** file.

Table 7.16. healthcheck

Configuration option = Default value	Type	Description
backends = []	list value	Additional backends that can perform health checks and report that information back as part of a request.
detailed = False	boolean value	Show more detailed information as part of the response. Security note: Enabling this option may expose sensitive details about the service being monitored. Be sure to verify that it will not violate your security policies.
disable_by_file_path = None	string value	Check the presence of a file to determine if an application is running on a port. Used by DisableByFileHealthcheck plugin.
disable_by_file_paths = []	list value	Check the presence of a file based on a port to determine if an application is running on a port. Expects a "port:path" list of strings. Used by DisableByFilesPortsHealthcheck plugin.
path = /healthcheck	string value	The path to respond to healthcheck requests on.

7.1.18. identity

The following table outlines the options available under the **[identity]** group in the **/etc/keystone/keystone.conf** file.

Table 7.17. identity

Configuration option = Default value	Type	Description
cache_time = 600	integer value	Time to cache identity data (in seconds). This has no effect unless global and identity caching are enabled.
caching = True	boolean value	Toggle for identity caching. This has no effect unless global caching is enabled. There is typically no reason to disable this.
default_domain_id = default	string value	This references the domain to use for all Identity API v2 requests (which are not aware of domains). A domain with this ID can optionally be created for you by keystone-manage bootstrap . 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 in order to maintain support for your v2 clients. There is typically no reason to change this value.

Configuration option = Default value	Type	Description
domain_config_dir = /etc/keystone/domains	string value	Absolute path where keystone should locate domain-specific [identity] configuration files. This option has no effect unless [identity] domain_specific_drivers_enabled is set to true. There is typically no reason to change this value.
domain_configurations_from_database = False	boolean value	By default, domain-specific configuration data is read from files in the directory identified by [identity] domain_config_dir . Enabling this configuration option allows you to instead manage domain-specific configurations through the API, which are then persisted in the backend (typically, a SQL database), rather than using configuration files on disk.
domain_specific_drivers_enabled = False	boolean value	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 [identity] domain_configurations_from_database). Only values specific to the domain need to be specified in this manner. This feature is disabled by default, but may be enabled by default in a future release; set to true to enable.
driver = sql	string value	Entry point for the identity backend driver in the keystone.identity namespace. Keystone provides a sql and ldap driver. This option is also used as the default driver selection (along with the other configuration variables in this section) in the event that [identity] domain_specific_drivers_enabled is enabled, but no applicable domain-specific configuration is defined for the domain in question. Unless your deployment primarily relies on ldap AND is not using domain-specific configuration, you should typically leave this set to sql .
list_limit = None	integer value	Maximum number of entities that will be returned in an identity collection.
max_password_length = 4096	integer value	Maximum allowed length for user passwords. Decrease this value to improve performance. Changing this value does not effect existing passwords.
password_hash_algorithm = bcrypt	string value	The password hashing algorithm to use for passwords stored within keystone.

Configuration option = Default value	Type	Description
password_hash_rounds = None	integer value	This option represents a trade off between security and performance. Higher values lead to slower performance, but higher security. Changing this option will only affect newly created passwords as existing password hashes already have a fixed number of rounds applied, so it is safe to tune this option in a running cluster. The default for bcrypt is 12, must be between 4 and 31, inclusive. The default for scrypt is 16, must be within range(1,32) . The default for pbkdf_sha512 is 60000, must be within range(1,1<<32) WARNING: If using scrypt, increasing this value increases BOTH time AND memory requirements to hash a password.
salt_bytesize = None	integer value	Number of bytes to use in scrypt and pbkdf2_sha512 hashing salt. Default for scrypt is 16 bytes. Default for pbkdf2_sha512 is 16 bytes. Limited to a maximum of 96 bytes due to the size of the column used to store password hashes.
scrypt_block_size = None	integer value	Optional block size to pass to scrypt hash function (the r parameter). Useful for tuning scrypt to optimal performance for your CPU architecture. This option is only used when the password_hash_algorithm option is set to scrypt . Defaults to 8.
scrypt_parallelism = None	integer value	Optional parallelism to pass to scrypt hash function (the p parameter). This option is only used when the password_hash_algorithm option is set to scrypt . Defaults to 1.

7.1.19. identity_mapping

The following table outlines the options available under the **[identity_mapping]** group in the **/etc/keystone/keystone.conf** file.

Table 7.18. identity_mapping

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
backward_compatible_ids = True	boolean value	The format of user and group IDs changed in Juno for backends that do not generate UUIDs (for example, 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 ([identity] domain_specific_drivers_enabled), 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 new mapping for all backends, including the default LDAP driver. It is only guaranteed to be safe to enable this option if you do not already have assignments for users and groups from the default LDAP domain, and you consider it to be acceptable for Keystone to provide the different IDs to clients than it did previously (existing IDs in the API will suddenly change). Typically this means that the only time you can set this value to false is when configuring a fresh installation, although that is the recommended value.
driver = sql	string value	Entry point for the identity mapping backend driver in the keystone.identity.id_mapping namespace. Keystone only provides a sql driver, so there is no reason to change this unless you are providing a custom entry point.
generator = sha256	string value	Entry point for the public ID generator for user and group entities in the keystone.identity.id_generator namespace. The Keystone identity mapper only supports generators that produce 64 bytes or less. Keystone only provides a sha256 entry point, so there is no reason to change this value unless you're providing a custom entry point.

7.1.20. jwt_tokens

The following table outlines the options available under the **[jwt_tokens]** group in the **/etc/keystone/keystone.conf** file.

Table 7.19. jwt_tokens

Configuration option = Default value	Type	Description
jws_private_key_repository = /etc/keystone/jws-keys/private	string value	Directory containing private keys for signing JWS tokens. This directory must exist in order for keystone's server process to start. It must also be readable by keystone's server process. It must contain at least one private key that corresponds to a public key in keystone.conf [jwt_tokens] jws_public_key_repository . In the event there are multiple private keys in this directory, keystone will use a key named private.pem to sign tokens. In the future, keystone may support the ability to sign tokens with multiple private keys. For now, only a key named private.pem within this directory is required to issue JWS tokens. This option is only applicable in deployments issuing JWS tokens and setting keystone.conf [token] provider = jws .
jws_public_key_repository = /etc/keystone/jws-keys/public	string value	Directory containing public keys for validating JWS token signatures. This directory must exist in order for keystone's server process to start. It must also be readable by keystone's server process. It must contain at least one public key that corresponds to a private key in keystone.conf [jwt_tokens] jws_private_key_repository . This option is only applicable in deployments issuing JWS tokens and setting keystone.conf [token] provider = jws .

7.1.21. Imap

The following table outlines the options available under the **[imap]** group in the **/etc/keystone/keystone.conf** file.

Table 7.20. Imap

Configuration option = Default value	Type	Description
alias_dereferencing = default	string value	The LDAP dereferencing option to use for queries involving aliases. A value of default falls back to using default dereferencing behavior configured by your ldap.conf . A value of never prevents aliases from being dereferenced at all. A value of searching dereferences aliases only after name resolution. A value of finding dereferences aliases only during name resolution. A value of always dereferences aliases in all cases.

Configuration option = Default value	Type	Description
auth_pool_connection_lifetime = 60	integer value	The maximum end user authentication connection lifetime to the LDAP server in seconds. When this lifetime is exceeded, the connection will be unbound and removed from the connection pool. This option has no effect unless [ldap] use_auth_pool is also enabled.
auth_pool_size = 100	integer value	The size of the connection pool to use for end user authentication. This option has no effect unless [ldap] use_auth_pool is also enabled.
chase_referrals = None	boolean value	Sets keystone's referral chasing behavior across directory partitions. If left unset, the system's default behavior will be used.
connection_timeout = -1	integer value	The connection timeout to use with the LDAP server. A value of -1 means that connections will never timeout.
debug_level = None	integer value	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.
group_ad_nesting = False	boolean value	If enabled, group queries will use Active Directory specific filters for nested groups.
group_additional_attribute_mapping = []	list value	A list of LDAP attribute to keystone group attribute pairs used for mapping additional attributes to groups in keystone. The expected format is <ldap_attr>:<group_attr> , where ldap_attr is the attribute in the LDAP object and group_attr is the attribute which should appear in the identity API.
group_attribute_ignore = []	list value	List of group attributes to ignore on create and update. or whether a specific group attribute should be filtered for list or show group.
group_desc_attribute = description	string value	The LDAP attribute mapped to group descriptions in keystone.
group_filter = None	string value	The LDAP search filter to use for groups.

Configuration option = Default value	Type	Description
group_id_attribute = cn	string value	The LDAP attribute mapped to group IDs in keystone. This must NOT be a multivalued attribute. Group IDs are expected to be globally unique across keystone domains and URL-safe.
group_member_attribute = member	string value	The LDAP attribute used to indicate that a user is a member of the group.
group_members_are_ids = False	boolean value	Enable this option if the members of the group object class are keystone user IDs rather than LDAP DNs. This is the case when using posixGroup as the group object class in Open Directory.
group_name_attribute = ou	string value	The LDAP attribute mapped to group names in keystone. Group names are expected to be unique only within a keystone domain and are not expected to be URL-safe.
group_objectclass = groupOfNames	string value	The LDAP object class to use for groups. If setting this option to posixGroup , you may also be interested in enabling the [ldap] group_members_are_ids option.
group_tree_dn = None	string value	The search base to use for groups. Defaults to ou=UserGroups with the [ldap] suffix appended to it.
page_size = 0	integer value	Defines the maximum number of results per page that keystone should request from the LDAP server when listing objects. A value of zero (0) disables paging.
password = None	string value	The password of the administrator bind DN to use when querying the LDAP server, if your LDAP server requires it.
pool_connection_lifetime = 600	integer value	The maximum connection lifetime to the LDAP server in seconds. When this lifetime is exceeded, the connection will be unbound and removed from the connection pool. This option has no effect unless [ldap] use_pool is also enabled.
pool_connection_timeout = -1	integer value	The connection timeout to use when pooling LDAP connections. A value of -1 means that connections will never timeout. This option has no effect unless [ldap] use_pool is also enabled.

Configuration option = Default value	Type	Description
pool_retry_delay = 0.1	floating point value	The number of seconds to wait before attempting to reconnect to the LDAP server. This option has no effect unless [ldap] use_pool is also enabled.
pool_retry_max = 3	integer value	The maximum number of times to attempt reconnecting to the LDAP server before aborting. A value of zero prevents retries. This option has no effect unless [ldap] use_pool is also enabled.
pool_size = 10	integer value	The size of the LDAP connection pool. This option has no effect unless [ldap] use_pool is also enabled.
query_scope = one	string value	The search scope which defines how deep to search within the search base. A value of one (representing oneLevel or singleLevel) indicates a search of objects immediately below to the base object, but does not include the base object itself. A value of sub (representing subtree or wholeSubtree) indicates a search of both the base object itself and the entire subtree below it.
suffix = cn=example,cn=com	string value	The default LDAP server suffix to use, if a DN is not defined via either [ldap] user_tree_dn or [ldap] group_tree_dn .
tls_cacertdir = None	string value	An absolute path to a CA certificate directory to use when communicating with LDAP servers. There is no reason to set this option if you've also set [ldap] tls_cacertfile .
tls_cacertfile = None	string value	An absolute path to a CA certificate file to use when communicating with LDAP servers. This option will take precedence over [ldap] tls_cacertdir , so there is no reason to set both.
tls_req_cert = demand	string value	Specifies which checks to perform against client certificates on incoming TLS sessions. If set to demand , then a certificate will always be requested and required from the LDAP server. If set to allow , then a certificate will always be requested but not required from the LDAP server. If set to never , then a certificate will never be requested.
url = ldap://localhost	string value	URL(s) for connecting to the LDAP server. Multiple LDAP URLs may be specified as a comma separated string. The first URL to successfully bind is used for the connection.

Configuration option = Default value	Type	Description
use_auth_pool = True	boolean value	Enable LDAP connection pooling for end user authentication. There is typically no reason to disable this.
use_pool = True	boolean value	Enable LDAP connection pooling for queries to the LDAP server. There is typically no reason to disable this.
use_tls = False	boolean value	Enable TLS when communicating with LDAP servers. You should also set the [ldap] tls_cacertfile and [ldap] tls_cacertdir options when using this option. Do not set this option if you are using LDAP over SSL (LDAPS) instead of TLS.
user = None	string value	The user name of the administrator bind DN to use when querying the LDAP server, if your LDAP server requires it.
user_additional_attribute_mapping = []	list value	A list of LDAP attribute to keystone user attribute pairs used for mapping additional attributes to users in keystone. The expected format is <ldap_attr>: <user_attr> , where ldap_attr is the attribute in the LDAP object and user_attr is the attribute which should appear in the identity API.
user_attribute_ignore = ['default_project_id']	list value	List of user attributes to ignore on create and update, or whether a specific user attribute should be filtered for list or show user.
user_default_project_id_attribute = None	string value	The LDAP attribute mapped to a user's default_project_id in keystone. This is most commonly used when keystone has write access to LDAP.
user_description_attribute = description	string value	The LDAP attribute mapped to user descriptions in keystone.
user_enabled_attribute = enabled	string value	The LDAP attribute mapped to the user enabled attribute in keystone. If setting this option to userAccountControl , then you may be interested in setting [ldap] user_enabled_mask and [ldap] user_enabled_default as well.

Configuration option = Default value	Type	Description
user_enabled_default = True	string value	The default value to enable users. This should match an appropriate integer 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 , then the typical value is 512 . This is typically used when [ldap] user_enabled_attribute = userAccountControl .
user_enabled_emulation = False	boolean value	If enabled, keystone uses an alternative method to determine if a user is enabled or not by checking if they are a member of the group defined by the [ldap] user_enabled_emulation_dn option. Enabling this option causes keystone to ignore the value of [ldap] user_enabled_invert .
user_enabled_emulation_dn = None	string value	DN of the group entry to hold enabled users when using enabled emulation. Setting this option has no effect unless [ldap] user_enabled_emulation is also enabled.
user_enabled_emulation_use_group_config = False	boolean value	Use the [ldap] group_member_attribute and [ldap] group_objectclass settings to determine membership in the emulated enabled group. Enabling this option has no effect unless [ldap] user_enabled_emulation is also enabled.
user_enabled_invert = False	boolean value	Logically negate the boolean value of the enabled attribute obtained from the LDAP server. Some LDAP servers use a boolean lock attribute where "true" means an account is disabled. Setting [ldap] user_enabled_invert = true will allow these lock attributes to be used. This option will have no effect if either the [ldap] user_enabled_mask or [ldap] user_enabled_emulation options are in use.
user_enabled_mask = 0	integer value	Bitmask integer to select which bit indicates the enabled value if the LDAP server represents "enabled" as a bit on an integer rather than as a discrete boolean. A value of 0 indicates that the mask is not used. If this is not set to 0 the typical value is 2 . This is typically used when [ldap] user_enabled_attribute = userAccountControl . Setting this option causes keystone to ignore the value of [ldap] user_enabled_invert .
user_filter = None	string value	The LDAP search filter to use for users.

Configuration option = Default value	Type	Description
user_id_attribute = cn	string value	The LDAP attribute mapped to user IDs in keystone. This must NOT be a multivalued attribute. User IDs are expected to be globally unique across keystone domains and URL-safe.
user_mail_attribute = mail	string value	The LDAP attribute mapped to user emails in keystone.
user_name_attribute = sn	string value	The LDAP attribute mapped to user names in keystone. User names are expected to be unique only within a keystone domain and are not expected to be URL-safe.
user_objectclass = inetOrgPerson	string value	The LDAP object class to use for users.
user_pass_attribute = userPassword	string value	The LDAP attribute mapped to user passwords in keystone.
user_tree_dn = None	string value	The search base to use for users. Defaults to ou=Users with the [ldap] suffix appended to it.

7.1.22. memcache

The following table outlines the options available under the **[memcache]** group in the **/etc/keystone/keystone.conf** file.

Table 7.21. memcache

Configuration option = Default value	Type	Description
dead_retry = 300	integer value	Number of seconds memcached server is considered dead before it is tried again. This is used by the key value store system.
pool_connection_get_timeout = 10	integer value	Number of seconds that an operation will wait to get a memcache client connection. This is used by the key value store system.
pool_maxsize = 10	integer value	Max total number of open connections to every memcached server. This is used by the key value store system.
pool_unused_timeout = 60	integer value	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.

Configuration option = Default value	Type	Description
socket_timeout = 3	integer value	Timeout in seconds for every call to a server. This is used by the key value store system. Deprecated since: T *Reason:*This option is duplicated with oslo.cache. Configure ``keystone.conf [cache] memcache_socket_timeout`` option to set the socket_timeout of memcached instead.

7.1.23. oauth1

The following table outlines the options available under the **[oauth1]** group in the **/etc/keystone/keystone.conf** file.

Table 7.22. oauth1

Configuration option = Default value	Type	Description
access_token_duration = 86400	integer value	Number of seconds for the OAuth Access Token to remain valid after being created. This is the amount of time the consumer has to interact with the service provider (which is typically keystone). Setting this option to zero means that access tokens will last forever.
driver = sql	string value	Entry point for the OAuth backend driver in the keystone.oauth1 namespace. Typically, there is no reason to set this option unless you are providing a custom entry point.
request_token_duration = 28800	integer value	Number of seconds for the OAuth Request Token to remain valid after being created. This is the amount of time the user has to authorize the token. Setting this option to zero means that request tokens will last forever.

7.1.24. oslo_messaging_amqp

The following table outlines the options available under the **[oslo_messaging_amqp]** group in the **/etc/keystone/keystone.conf** file.

Table 7.23. oslo_messaging_amqp

Configuration option = Default value	Type	Description
addressing_mode = dynamic	string value	Indicates the addressing mode used by the driver. Permitted values: <i>legacy</i> - use legacy non-routable addressing <i>routable</i> - use routable addresses <i>dynamic</i> - use legacy addresses if the message bus does not support routing otherwise use routable addressing
anycast_address = anycast	string value	Appended to the address prefix when sending to a group of consumers. Used by the message bus to identify messages that should be delivered in a round-robin fashion across consumers.
broadcast_prefix = broadcast	string value	address prefix used when broadcasting to all servers
connection_retry_backoff = 2	integer value	Increase the <code>connection_retry_interval</code> by this many seconds after each unsuccessful failover attempt.
connection_retry_interval = 1	integer value	Seconds to pause before attempting to re-connect.
connection_retry_interval_max = 30	integer value	Maximum limit for <code>connection_retry_interval</code> + <code>connection_retry_backoff</code>
container_name = None	string value	Name for the AMQP container. must be globally unique. Defaults to a generated UUID
default_notification_exchange = None	string value	Exchange name used in notification addresses. Exchange name resolution precedence: Target.exchange if set else <code>default_notification_exchange</code> if set else <code>control_exchange</code> if set else <i>notify</i>
default_notify_timeout = 30	integer value	The deadline for a sent notification message delivery. Only used when caller does not provide a timeout expiry.
default_reply_retry = 0	integer value	The maximum number of attempts to re-send a reply message which failed due to a recoverable error.
default_reply_timeout = 30	integer value	The deadline for an rpc reply message delivery.

Configuration option = Default value	Type	Description
default_rpc_exchange = None	string value	Exchange name used in RPC addresses. Exchange name resolution precedence: Target.exchange if set else default_rpc_exchange if set else control_exchange if set else <i>rpc</i>
default_send_timeout = 30	integer value	The deadline for an rpc cast or call message delivery. Only used when caller does not provide a timeout expiry.
default_sender_link_timeout = 600	integer value	The duration to schedule a purge of idle sender links. Detach link after expiry.
group_request_prefix = unicast	string value	address prefix when sending to any server in group
idle_timeout = 0	integer value	Timeout for inactive connections (in seconds)
link_retry_delay = 10	integer value	Time to pause between re-connecting an AMQP 1.0 link that failed due to a recoverable error.
multicast_address = multicast	string value	Appended to the address prefix when sending a fanout message. Used by the message bus to identify fanout messages.
notify_address_prefix = openstack.org/om/notify	string value	Address prefix for all generated Notification addresses
notify_server_credit = 100	integer value	Window size for incoming Notification messages
pre_settled = ['rpc-cast', 'rpc-reply']	multi valued	Send messages of this type pre-settled. Pre-settled messages will not receive acknowledgement from the peer. Note well: pre-settled messages may be silently discarded if the delivery fails. Permitted values: <i>rpc-call</i> - send RPC Calls pre-settled <i>rpc-reply</i> - send RPC Replies pre-settled <i>rpc-cast</i> - Send RPC Casts pre-settled <i>notify</i> - Send Notifications pre-settled
pseudo_vhost = True	boolean value	Enable virtual host support for those message buses that do not natively support virtual hosting (such as qpidd). When set to true the virtual host name will be added to all message bus addresses, effectively creating a private <i>subnet</i> per virtual host. Set to False if the message bus supports virtual hosting using the <i>hostname</i> field in the AMQP 1.0 Open performative as the name of the virtual host.
reply_link_credit = 200	integer value	Window size for incoming RPC Reply messages.

Configuration option = Default value	Type	Description
rpc_address_prefix = openstack.org/om/rpc	string value	Address prefix for all generated RPC addresses
rpc_server_credit = 100	integer value	Window size for incoming RPC Request messages
<code>`sas_config_dir = `</code>	string value	Path to directory that contains the SASL configuration
<code>`sas_config_name = `</code>	string value	Name of configuration file (without .conf suffix)
<code>`sas_default_realm = `</code>	string value	SASL realm to use if no realm present in username
<code>`sas_mechanisms = `</code>	string value	Space separated list of acceptable SASL mechanisms
server_request_prefix = exclusive	string value	address prefix used when sending to a specific server
ssl = False	boolean value	Attempt to connect via SSL. If no other ssl-related parameters are given, it will use the system's CA-bundle to verify the server's certificate.
<code>`ssl_ca_file = `</code>	string value	CA certificate PEM file used to verify the server's certificate
<code>`ssl_cert_file = `</code>	string value	Self-identifying certificate PEM file for client authentication
<code>`ssl_key_file = `</code>	string value	Private key PEM file used to sign ssl_cert_file certificate (optional)
ssl_key_password = None	string value	Password for decrypting ssl_key_file (if encrypted)
ssl_verify_vhost = False	boolean value	By default SSL checks that the name in the server's certificate matches the hostname in the transport_url. In some configurations it may be preferable to use the virtual hostname instead, for example if the server uses the Server Name Indication TLS extension (rfc6066) to provide a certificate per virtual host. Set ssl_verify_vhost to True if the server's SSL certificate uses the virtual host name instead of the DNS name.
trace = False	boolean value	Debug: dump AMQP frames to stdout

Configuration option = Default value	Type	Description
unicast_address = unicast	string value	Appended to the address prefix when sending to a particular RPC/Notification server. Used by the message bus to identify messages sent to a single destination.

7.1.25. oslo_messaging_kafka

The following table outlines the options available under the **[oslo_messaging_kafka]** group in the **/etc/keystone/keystone.conf** file.

Table 7.24. oslo_messaging_kafka

Configuration option = Default value	Type	Description
compression_codec = none	string value	The compression codec for all data generated by the producer. If not set, compression will not be used. Note that the allowed values of this depend on the kafka version
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
consumer_group = oslo_messaging_consumer	string value	Group id for Kafka consumer. Consumers in one group will coordinate message consumption
enable_auto_commit = False	boolean value	Enable asynchronous consumer commits
kafka_consumer_timeout = 1.0	floating point value	Default timeout(s) for Kafka consumers
kafka_max_fetch_bytes = 1048576	integer value	Max fetch bytes of Kafka consumer
max_poll_records = 500	integer value	The maximum number of records returned in a poll call
pool_size = 10	integer value	Pool Size for Kafka Consumers
producer_batch_size = 16384	integer value	Size of batch for the producer async send

Configuration option = Default value	Type	Description
producer_batch_timeout = 0.0	floating point value	Upper bound on the delay for KafkaProducer batching in seconds
sasl_mechanism = PLAIN	string value	Mechanism when security protocol is SASL
security_protocol = PLAINTEXT	string value	Protocol used to communicate with brokers
<code>`ssl_cafile = `</code>	string value	CA certificate PEM file used to verify the server certificate
<code>`ssl_client_cert_file = `</code>	string value	Client certificate PEM file used for authentication.
<code>`ssl_client_key_file = `</code>	string value	Client key PEM file used for authentication.
<code>`ssl_client_key_password = `</code>	string value	Client key password file used for authentication.

7.1.26. oslo_messaging_notifications

The following table outlines the options available under the **[oslo_messaging_notifications]** group in the **/etc/keystone/keystone.conf** file.

Table 7.25. oslo_messaging_notifications

Configuration option = Default value	Type	Description
driver = []	multi valued	The Drivers(s) to handle sending notifications. Possible values are messaging, messagingv2, routing, log, test, noop
retry = -1	integer value	The maximum number of attempts to re-send a notification message which failed to be delivered due to a recoverable error. 0 - No retry, -1 - indefinite
topics = ['notifications']	list value	AMQP topic used for OpenStack notifications.
transport_url = None	string value	A URL representing the messaging driver to use for notifications. If not set, we fall back to the same configuration used for RPC.

7.1.27. oslo_messaging_rabbit

The following table outlines the options available under the **[oslo_messaging_rabbit]** group in the **/etc/keystone/keystone.conf** file.

Table 7.26. oslo_messaging_rabbit

Configuration option = Default value	Type	Description
amqp_auto_delete = False	boolean value	Auto-delete queues in AMQP.
amqp_durable_queues = False	boolean value	Use durable queues in AMQP.
direct_mandatory_flag = True	boolean value	(DEPRECATED) Enable/Disable the RabbitMQ mandatory flag for direct send. The direct send is used as reply, so the MessageUndeliverable exception is raised in case the client queue does not exist. MessageUndeliverable exception will be used to loop for a timeout to let a chance to sender to recover. This flag is deprecated and it will not be possible to deactivate this functionality anymore
enable_cancel_on_failover = False	boolean value	Enable x-cancel-on-ha-failover flag so that rabbitmq server will cancel and notify consumers when queue is down
heartbeat_in_pthread = True	boolean value	Run the health check heartbeat thread through a native python thread by default. If this option is equal to False then the health check heartbeat will inherit the execution model from the parent process. For example if the parent process has monkey patched the stdlib by using eventlet/greenlet then the heartbeat will be run through a green thread.
heartbeat_rate = 2	integer value	How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = 60	integer value	Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables heartbeat).
kombu_compression = None	string value	EXPERIMENTAL: Possible values are: gzip, bz2. If not set compression will not be used. This option may not be available in future versions.
kombu_failover_strategy = round-robin	string value	Determines how the next RabbitMQ node is chosen in case the one we are currently connected to becomes unavailable. Takes effect only if more than one RabbitMQ node is provided in config.
kombu_missing_consumer_retry_timeout = 60	integer value	How long to wait a missing client before abandoning to send it its replies. This value should not be longer than rpc_response_timeout.

Configuration option = Default value	Type	Description
kombu_reconnect_delay = 1.0	floating point value	How long to wait before reconnecting in response to an AMQP consumer cancel notification.
rabbit_ha_queues = False	boolean value	Try to use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database. In RabbitMQ 3.0, queue mirroring is no longer controlled by the x-ha-policy argument when declaring a queue. If you just want to make sure that all queues (except those with auto-generated names) are mirrored across all nodes, run: <code>"rabbitmqctl set_policy HA ^(?!amq\..)*{\$"ha-mode": "all"}"</code>
rabbit_interval_max = 30	integer value	Maximum interval of RabbitMQ connection retries. Default is 30 seconds.
rabbit_login_method = AMQPLAIN	string value	The RabbitMQ login method.
rabbit_qos_prefetch_count = 0	integer value	Specifies the number of messages to prefetch. Setting to zero allows unlimited messages.
rabbit_retry_backoff = 2	integer value	How long to backoff for between retries when connecting to RabbitMQ.
rabbit_retry_interval = 1	integer value	How frequently to retry connecting with RabbitMQ.
rabbit_transient_queues_ttl = 1800	integer value	Positive integer representing duration in seconds for queue TTL (x-expires). Queues which are unused for the duration of the TTL are automatically deleted. The parameter affects only reply and fanout queues.
ssl = False	boolean value	Connect over SSL.
<code>`ssl_ca_file = `</code>	string value	SSL certification authority file (valid only if SSL enabled).
<code>`ssl_cert_file = `</code>	string value	SSL cert file (valid only if SSL enabled).
<code>`ssl_key_file = `</code>	string value	SSL key file (valid only if SSL enabled).
<code>`ssl_version = `</code>	string value	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.

7.1.28. oslo_middleware

The following table outlines the options available under the **[oslo_middleware]** group in the **/etc/keystone/keystone.conf** file.

Table 7.27. oslo_middleware

Configuration option = Default value	Type	Description
enable_proxy_headers_parsing = False	boolean value	Whether the application is behind a proxy or not. This determines if the middleware should parse the headers or not.
max_request_body_size = 114688	integer value	The maximum body size for each request, in bytes.
secure_proxy_ssl_header = X-Forwarded-Proto	string value	The HTTP Header that will be used to determine what the original request protocol scheme was, even if it was hidden by a SSL termination proxy.

7.1.29. oslo_policy

The following table outlines the options available under the **[oslo_policy]** group in the **/etc/keystone/keystone.conf** file.

Table 7.28. oslo_policy

Configuration option = Default value	Type	Description
enforce_new_defaults = False	boolean value	This option controls whether or not to use old deprecated defaults when evaluating policies. If True , the old deprecated defaults are not going to be evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be disallowed. It is encouraged to enable this flag along with the enforce_scope flag so that you can get the benefits of new defaults and scope_type together
enforce_scope = False	boolean value	This option controls whether or not to enforce scope when evaluating policies. If True , the scope of the token used in the request is compared to the scope_types of the policy being enforced. If the scopes do not match, an InvalidScope exception will be raised. If False , a message will be logged informing operators that policies are being invoked with mismatching scope.
policy_default_rule = default	string value	Default rule. Enforced when a requested rule is not found.

Configuration option = Default value	Type	Description
policy_dirs = ['policy.d']	multi valued	Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the <code>config_dir</code> option, or absolute paths. The file defined by <code>policy_file</code> must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = policy.yaml	string value	The relative or absolute path of a file that maps roles to permissions for a given service. Relative paths must be specified in relation to the configuration file setting this option.
remote_content_type = application/x-www-form-urlencoded	string value	Content Type to send and receive data for REST based policy check
remote_ssl_ca_cert_file = None	string value	Absolute path to ca cert file for REST based policy check
remote_ssl_client_cert_file = None	string value	Absolute path to client cert for REST based policy check
remote_ssl_client_key_file = None	string value	Absolute path client key file REST based policy check
remote_ssl_verify_server_cert = False	boolean value	server identity verification for REST based policy check

7.1.30. policy

The following table outlines the options available under the **[policy]** group in the `/etc/keystone/keystone.conf` file.

Table 7.29. policy

Configuration option = Default value	Type	Description
driver = sql	string value	Entry point for the policy backend driver in the keystone.policy namespace. Supplied drivers are rules (which does not support any CRUD operations for the v3 policy API) and sql . Typically, there is no reason to set this option unless you are providing a custom entry point.
list_limit = None	integer value	Maximum number of entities that will be returned in a policy collection.

7.1.31. profiler

The following table outlines the options available under the **[profiler]** group in the `/etc/keystone/keystone.conf` file.

Table 7.30. profiler

Configuration option = Default value	Type	Description
connection_string = messaging://	string value	Connection string for a notifier backend. Default value is messaging:// which sets the notifier to oslo_messaging. Examples of possible values: ● messaging:// - use oslo_messaging driver for sending spans. ● redis://127.0.0.1:6379 - use redis driver for sending spans. ● mongodb://127.0.0.1:27017 - use mongodb driver for sending spans. ● elasticsearch://127.0.0.1:9200 - use elasticsearch driver for sending spans. ● jaeger://127.0.0.1:6831 - use jaeger tracing as driver for sending spans.
enabled = False	boolean value	Enable the profiling for all services on this node. Default value is False (fully disable the profiling feature). Possible values: ● True: Enables the feature ● False: Disables the feature. The profiling cannot be started via this project operations. If the profiling is triggered by another project, this project part will be empty.
es_doc_type = notification	string value	Document type for notification indexing in elasticsearch.
es_scroll_size = 10000	integer value	Elasticsearch splits large requests in batches. This parameter defines maximum size of each batch (for example: es_scroll_size=10000).

Configuration option = Default value	Type	Description
es_scroll_time = 2m	string value	This parameter is a time value parameter (for example: es_scroll_time=2m), indicating for how long the nodes that participate in the search will maintain relevant resources in order to continue and support it.
filter_error_trace = False	boolean value	Enable filter traces that contain error/exception to a separated place. Default value is set to False. Possible values: • True: Enable filter traces that contain error/exception. • False: Disable the filter.
hmac_keys = SECRET_KEY	string value	Secret key(s) to use for encrypting context data for performance profiling. This string value should have the following format: <key1>[,<key2>,...,<keyn>], where each key is some random string. A user who triggers the profiling via the REST API has to set one of these keys in the headers of the REST API call to include profiling results of this node for this particular project. Both "enabled" flag and "hmac_keys" config options should be set to enable profiling. Also, to generate correct profiling information across all services at least one key needs to be consistent between OpenStack projects. This ensures it can be used from client side to generate the trace, containing information from all possible resources.
sentinel_service_name = mymaster	string value	Redis sentinel uses a service name to identify a master redis service. This parameter defines the name (for example: sentinal_service_name=mymaster).
socket_timeout = 0.1	floating point value	Redis sentinel provides a timeout option on the connections. This parameter defines that timeout (for example: socket_timeout=0.1).

Configuration option = Default value	Type	Description
trace_sqlalchemy = False	boolean value	Enable SQL requests profiling in services. Default value is False (SQL requests won't be traced). Possible values: • True: Enables SQL requests profiling. Each SQL query will be part of the trace and can be analyzed by how much time was spent for that. • False: Disables SQL requests profiling. The spent time is only shown on a higher level of operations. Single SQL queries cannot be analyzed this way.

7.1.32. receipt

The following table outlines the options available under the **[receipt]** group in the `/etc/keystone/keystone.conf` file.

Table 7.31. receipt

Configuration option = Default value	Type	Description
cache_on_issue = True	boolean value	Enable storing issued receipt data to receipt validation cache so that first receipt validation doesn't actually cause full validation cycle. This option has no effect unless global caching and receipt caching are enabled.
cache_time = 300	integer value	The number of seconds to cache receipt creation and validation data. This has no effect unless both global and [receipt] caching are enabled.
caching = True	boolean value	Toggle for caching receipt creation and validation data. This has no effect unless global caching is enabled, or if cache_on_issue is disabled as we only cache receipts on issue.
expiration = 300	integer value	The amount of time that a receipt should remain valid (in seconds). This value should always be very short, as it represents how long a user has to reattempt auth with the missing auth methods.

Configuration option = Default value	Type	Description
provider = fernet	string value	Entry point for the receipt provider in the keystone.receipt.provider namespace. The receipt provider controls the receipt construction and validation operations. Keystone includes just the fernet receipt provider for now. fernet receipts do not need to be persisted at all, but require that you run keystone-manage fernet_setup (also see the keystone-manage fernet_rotate command).

7.1.33. resource

The following table outlines the options available under the **[resource]** group in the **/etc/keystone/keystone.conf** file.

Table 7.32. resource

Configuration option = Default value	Type	Description
admin_project_domain_name = None	string value	Name of the domain that owns the admin_project_name . If left unset, then there is no admin project. [resource] admin_project_name must also be set to use this option.
admin_project_name = None	string value	This is a special project which represents cloud-level administrator privileges across services. Tokens scoped to this project will contain a true is_admin_project attribute to indicate to policy systems that the role assignments on that specific project should apply equally across every project. If left unset, then there is no admin project, and thus no explicit means of cross-project role assignments. [resource] admin_project_domain_name must also be set to use this option.
cache_time = None	integer value	Time to cache resource data in seconds. This has no effect unless global caching is enabled.
caching = True	boolean value	Toggle for resource caching. This has no effect unless global caching is enabled.

Configuration option = Default value	Type	Description
domain_name_url_safe = off	string value	This controls whether the names of domains are restricted from containing URL-reserved characters. If set to new , attempts to create or update a domain with a URL-unsafe name will fail. If set to strict , attempts to scope a token with a URL-unsafe domain name will fail, thereby forcing all domain names to be updated to be URL-safe.
driver = sql	string value	Entry point for the resource driver in the keystone.resource namespace. Only a sql driver is supplied by keystone. Unless you are writing proprietary drivers for keystone, you do not need to set this option.
list_limit = None	integer value	Maximum number of entities that will be returned in a resource collection.
project_name_url_safe = off	string value	This controls whether the names of projects are restricted from containing URL-reserved characters. If set to new , attempts to create or update a project with a URL-unsafe name will fail. If set to strict , attempts to scope a token with a URL-unsafe project name will fail, thereby forcing all project names to be updated to be URL-safe.

7.1.34. revoke

The following table outlines the options available under the **[revoke]** group in the **/etc/keystone/keystone.conf** file.

Table 7.33. revoke

Configuration option = Default value	Type	Description
cache_time = 3600	integer value	Time to cache the revocation list and the revocation events (in seconds). This has no effect unless global and [revoke] caching are both enabled.
caching = True	boolean value	Toggle for revocation event caching. This has no effect unless global caching is enabled.
driver = sql	string value	Entry point for the token revocation backend driver in the keystone.revoke namespace. Keystone only provides a sql driver, so there is no reason to set this option unless you are providing a custom entry point.

Configuration option = Default value	Type	Description
expiration_buffer = 1800	integer value	The number of seconds after a token has expired before a corresponding revocation event may be purged from the backend.

7.1.35. role

The following table outlines the options available under the **[role]** group in the **/etc/keystone/keystone.conf** file.

Table 7.34. role

Configuration option = Default value	Type	Description
cache_time = None	integer value	Time to cache role data, in seconds. This has no effect unless both global caching and [role] caching are enabled.
caching = True	boolean value	Toggle for role caching. This has no effect unless global caching is enabled. In a typical deployment, there is no reason to disable this.
driver = None	string value	Entry point for the role backend driver in the keystone.role namespace. Keystone only provides a sql driver, so there's no reason to change this unless you are providing a custom entry point.
list_limit = None	integer value	Maximum number of entities that will be returned in a role collection. This may be useful to tune if you have a large number of discrete roles in your deployment.

7.1.36. saml

The following table outlines the options available under the **[saml]** group in the **/etc/keystone/keystone.conf** file.

Table 7.35. saml

Configuration option = Default value	Type	Description
assertion_expiration_time = 3600	integer value	Determines the lifetime for any SAML assertions generated by keystone, using NotOnOrAfter attributes.

Configuration option = Default value	Type	Description
certfile = /etc/keystone/ssl/certs/signing_cert.pem	string value	Absolute path to the public certificate file to use for SAML signing. The value cannot contain a comma (,).
idp_contact_company = Example, Inc.	string value	This is the company name of the identity provider's contact person.
idp_contact_email = support@example.com	string value	This is the email address of the identity provider's contact person.
idp_contact_name = SAML Identity Provider Support	string value	This is the given name of the identity provider's contact person.
idp_contact_surname = Support	string value	This is the surname of the identity provider's contact person.
idp_contact_telephone = +1 800 555 0100	string value	This is the telephone number of the identity provider's contact person.
idp_contact_type = other	string value	This is the type of contact that best describes the identity provider's contact person.
idp_entity_id = None	uri value	This is the unique entity identifier of the identity provider (keystone) to use when generating SAML assertions. This value is required to generate identity provider metadata and must be a URI (a URL is recommended). For example: https://keystone.example.com/v3/OS-FEDERATION/saml2/idp.
idp_lang = en	string value	This is the language used by the identity provider's organization.
idp_metadata_path = /etc/keystone/saml2_idp_metadata.xml	string value	Absolute path to the identity provider metadata file. This file should be generated with the keystone-manage saml_idp_metadata command. There is typically no reason to change this value.
idp_organization_display_name = OpenStack SAML Identity Provider	string value	This is the name of the identity provider's organization to be displayed.
idp_organization_name = SAML Identity Provider	string value	This is the name of the identity provider's organization.

Configuration option = Default value	Type	Description
idp_organization_url = https://example.com/	uri value	This is the URL of the identity provider's organization. The URL referenced here should be useful to humans.
idp_sso_endpoint = None	uri value	This is the single sign-on (SSO) service location of the identity provider which accepts HTTP POST requests. A value is required to generate identity provider metadata. For example: https://keystone.example.com/v3/OS-FEDERATION/saml2/sso.
keyfile = /etc/keystone/ssl/private/s igning_key.pem	string value	Absolute path to the private key file to use for SAML signing. The value cannot contain a comma (,).
relay_state_prefix = ss:mem:	string value	The prefix of the RelayState SAML attribute to use when generating enhanced client and proxy (ECP) assertions. In a typical deployment, there is no reason to change this value.
xmlsec1_binary = xmlsec1	string value	Name of, or absolute path to, the binary to be used for XML signing. Although only the XML Security Library (xmlsec1) is supported, it may have a non-standard name or path on your system. If keystone cannot find the binary itself, you may need to install the appropriate package, use this option to specify an absolute path, or adjust keystone's PATH environment variable.

7.1.37. security_compliance

The following table outlines the options available under the **[security_compliance]** group in the **/etc/keystone/keystone.conf** file.

Table 7.36. security_compliance

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
change_password_upon_first_use = False	boolean value	Enabling this option requires users to change their password when the user is created, or upon administrative reset. Before accessing any services, affected users will have to change their password. To ignore this requirement for specific users, such as service users, set the options attribute ignore_change_password_upon_first_use to True for the desired user via the update user API. This feature is disabled by default. This feature is only applicable with the sql backend for the [identity] driver .
disable_user_account_days_inactive = None	integer value	The maximum number of days a user can go without authenticating before being considered "inactive" and automatically disabled (locked). This feature is disabled by default; set any value to enable it. This feature depends on the sql backend for the [identity] driver . When a user exceeds this threshold and is considered "inactive", the user's enabled attribute in the HTTP API may not match the value of the user's enabled column in the user table.
lockout_duration = 1800	integer value	The number of seconds a user account will be locked when the maximum number of failed authentication attempts (as specified by [security_compliance] lockout_failure_attempts) is exceeded. Setting this option will have no effect unless you also set [security_compliance] lockout_failure_attempts to a non-zero value. This feature depends on the sql backend for the [identity] driver .
lockout_failure_attempts = None	integer value	The maximum number of times that a user can fail to authenticate before the user account is locked for the number of seconds specified by [security_compliance] lockout_duration . This feature is disabled by default. If this feature is enabled and [security_compliance] lockout_duration is not set, then users may be locked out indefinitely until the user is explicitly enabled via the API. This feature depends on the sql backend for the [identity] driver .

Configuration option = Default value	Type	Description
minimum_password_age = 0	integer value	The number of days that a password must be used before the user can change it. This prevents users from changing their passwords immediately in order to wipe out their password history and reuse an old password. This feature does not prevent administrators from manually resetting passwords. It is disabled by default and allows for immediate password changes. This feature depends on the sql backend for the [identity] driver . Note: If [security_compliance] password_expires_days is set, then the value for this option should be less than the password_expires_days .
password_expires_days = None	integer value	The number of days for which a password will be considered valid before requiring it to be changed. This feature is disabled by default. If enabled, new password changes will have an expiration date, however existing passwords would not be impacted. This feature depends on the sql backend for the [identity] driver .
password_regex = None	string value	The regular expression used to validate password strength requirements. By default, the regular expression will match any password. The following is an example of a pattern which requires at least 1 letter, 1 digit, and have a minimum length of 7 characters: <code>^(?=.*\d)(?=.*[a-zA-Z]).{7,}\$</code> This feature depends on the sql backend for the [identity] driver .
password_regex_description = None	string value	Describe your password regular expression here in language for humans. If a password fails to match the regular expression, the contents of this configuration variable will be returned to users to explain why their requested password was insufficient.
unique_last_password_count = 0	integer value	This controls the number of previous user password iterations to keep in history, in order to enforce that newly created passwords are unique. The total number which includes the new password should not be greater or equal to this value. Setting the value to zero (the default) disables this feature. Thus, to enable this feature, values must be greater than 0. This feature depends on the sql backend for the [identity] driver .

7.1.38. shadow_users

The following table outlines the options available under the **[shadow_users]** group in the **/etc/keystone/keystone.conf** file.

Table 7.37. shadow_users

Configuration option = Default value	Type	Description
driver = sql	string value	Entry point for the shadow users backend driver in the keystone.identity.shadow_users namespace. This driver is used for persisting local user references to externally-managed identities (via federation, LDAP, etc). Keystone only provides a sql driver, so there is no reason to change this option unless you are providing a custom entry point.

7.1.39. token

The following table outlines the options available under the **[token]** group in the **/etc/keystone/keystone.conf** file.

Table 7.38. token

Configuration option = Default value	Type	Description
allow_expired_window = 172800	integer value	This controls the number of seconds that a token can be retrieved for beyond the built-in expiry time. This allows long running operations to succeed. Defaults to two days.
allow_rescope_scoped_token = True	boolean value	This toggles whether scoped tokens may be re-scoped to a new project or domain, thereby preventing users from exchanging a scoped token (including those with a default project scope) for any other token. This forces users to either authenticate for unscoped tokens (and later exchange that unscoped token for tokens with a more specific scope) or to provide their credentials in every request for a scoped token to avoid re-scoping altogether.

Configuration option = Default value	Type	Description
cache_on_issue = True	boolean value	Enable storing issued token data to token validation cache so that first token validation doesn't actually cause full validation cycle. This option has no effect unless global caching is enabled and will still cache tokens even if [token] caching = False. Deprecated since: S <i>*Reason:</i>*Keystone already exposes a configuration option for caching tokens. Having a separate configuration option to cache tokens when they are issued is redundant, unnecessarily complicated, and is misleading if token caching is disabled because tokens will still be pre-cached by default when they are issued. The ability to pre-cache tokens when they are issued is going to rely exclusively on the ``keystone.conf [token] caching`` option in the future.
cache_time = None	integer value	The number of seconds to cache token creation and validation data. This has no effect unless both global and [token] caching are enabled.
caching = True	boolean value	Toggle for caching token creation and validation data. This has no effect unless global caching is enabled.
expiration = 3600	integer value	The amount of time that a token should remain valid (in seconds). Drastically reducing this value may break "long-running" operations that involve multiple services to coordinate together, and will force users to authenticate with keystone more frequently. Drastically increasing this value will increase the number of tokens that will be simultaneously valid. Keystone tokens are also bearer tokens, so a shorter duration will also reduce the potential security impact of a compromised token.

Configuration option = Default value	Type	Description
provider = fernet	string value	Entry point for the token provider in the keystone.token.provider namespace. The token provider controls the token construction, validation, and revocation operations. Supported upstream providers are fernet and jws . Neither fernet or jws tokens require persistence and both require additional setup. If using fernet , you're required to run keystone-manage fernet_setup , which creates symmetric keys used to encrypt tokens. If using jws , you're required to generate an ECDSA keypair using a SHA-256 hash algorithm for signing and validating token, which can be done with keystone-manage create_jws_keypair . Note that fernet tokens are encrypted and jws tokens are only signed. Please be sure to consider this if your deployment has security requirements regarding payload contents used to generate token IDs.
revoke_by_id = True	boolean value	This toggles support for revoking individual tokens by the token identifier and thus various token enumeration operations (such as listing all tokens issued to a specific user). These operations are used to determine the list of tokens to consider revoked. Do not disable this option if you're using the kvs [revoke] driver .

7.1.40. tokenless_auth

The following table outlines the options available under the **[tokenless_auth]** group in the **/etc/keystone/keystone.conf** file.

Table 7.39. tokenless_auth

Configuration option = Default value	Type	Description
issuer_attribute = SSL_CLIENT_I_DN	string value	The name of the WSGI environment variable used to pass the issuer of the client certificate to keystone. This attribute is used as an identity provider ID for the X.509 tokenless authorization along with the protocol to look up its corresponding mapping. In a typical deployment, there is no reason to change this value.

Configuration option = Default value	Type	Description
protocol = x509	string value	The federated protocol ID used to represent X.509 tokenless authorization. This is used in combination with the value of [tokenless_auth] issuer_attribute to find a corresponding federated mapping. In a typical deployment, there is no reason to change this value.
trusted_issuer = []	multi valued	The list of distinguished names which identify trusted issuers of client certificates allowed to use X.509 tokenless authorization. If the option is absent then no certificates will be allowed. The format for the values of a distinguished name (DN) must be separated by a comma and contain no spaces. Furthermore, because an individual DN may contain commas, this configuration option may be repeated multiple times to represent multiple values. For example, keystone.conf would include two consecutive lines in order to trust two different DNs, such as trusted_issuer = CN=john,OU=keystone,O=openstack and trusted_issuer = CN=mary,OU=eng,O=abc .

7.1.41. totp

The following table outlines the options available under the **[totp]** group in the **/etc/keystone/keystone.conf** file.

Table 7.40. totp

Configuration option = Default value	Type	Description
included_previous_windows = 1	integer value	The number of previous windows to check when processing TOTP passcodes.

7.1.42. trust

The following table outlines the options available under the **[trust]** group in the **/etc/keystone/keystone.conf** file.

Table 7.41. trust

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
allow_redelegation = False	boolean value	Allows authorization to be redelegated from one user to another, effectively chaining trusts together. When disabled, the remaining_uses attribute of a trust is constrained to be zero.
driver = sql	string value	Entry point for the trust backend driver in the keystone.trust namespace. Keystone only provides a sql driver, so there is no reason to change this unless you are providing a custom entry point.
max_redelegation_count = 3	integer value	Maximum number of times that authorization can be redelegated from one user to another in a chain of trusts. This number may be reduced further for a specific trust.

7.1.43. unified_limit

The following table outlines the options available under the **[unified_limit]** group in the **/etc/keystone/keystone.conf** file.

Table 7.42. unified_limit

Configuration option = Default value	Type	Description
cache_time = None	integer value	Time to cache unified limit data, in seconds. This has no effect unless both global caching and [unified_limit] caching are enabled.
caching = True	boolean value	Toggle for unified limit caching. This has no effect unless global caching is enabled. In a typical deployment, there is no reason to disable this.
driver = sql	string value	Entry point for the unified limit backend driver in the keystone.unified_limit namespace. Keystone only provides a sql driver, so there's no reason to change this unless you are providing a custom entry point.
enforcement_model = flat	string value	The enforcement model to use when validating limits associated to projects. Enforcement models will behave differently depending on the existing limits, which may result in backwards incompatible changes if a model is switched in a running deployment.
list_limit = None	integer value	Maximum number of entities that will be returned in a unified limit collection. This may be useful to tune if you have a large number of unified limits in your deployment.

7.1.44. wsgi

The following table outlines the options available under the **[wsgi]** group in the **/etc/keystone/keystone.conf** file.

Table 7.43. wsgi

Configuration option = Default value	Type	Description
debug_middleware = False	boolean value	If set to true, this enables the oslo debug middleware in Keystone. This Middleware prints a lot of information about the request and the response. It is useful for getting information about the data on the wire (decoded) and passed to the WSGI application pipeline. This middleware has no effect on the "debug" setting in the [DEFAULT] section of the config file or setting Keystone's log-level to "DEBUG"; it is specific to debugging the WSGI data as it enters and leaves Keystone (specific request-related data). This option is used for introspection on the request and response data between the web server (apache, nginx, etc) and Keystone. This middleware is inserted as the first element in the middleware chain and will show the data closest to the wire. WARNING: NOT INTENDED FOR USE IN PRODUCTION. THIS MIDDLEWARE CAN AND WILL EMIT SENSITIVE/PRIVILEGED DATA.

CHAPTER 8. NEUTRON

The following chapter contains information about the configuration options in the **neutron** service.

8.1. DHCP_AGENT.INI

This section contains options for the `/etc/neutron/dhcp_agent.ini` file.

8.1.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/neutron/dhcp_agent.ini` file.

Configuration option = Default value	Type	Description
bulk_reload_interval = 0	integer value	Time to sleep between reloading the DHCP allocations. This will only be invoked if the value is not 0. If a network has N updates in X seconds then we will reload once with the port changes in the X seconds and not N times.
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.

Configuration option = Default value	Type	Description
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
dhcp_broadcast_reply = False	boolean value	Use broadcast in DHCP replies.
dhcp_confs = \$state_path/dhcp	string value	Location to store DHCP server config files.
dhcp_driver = neutron.agent.linux.dhcp. Dnsmasq	string value	The driver used to manage the DHCP server.
dhcp_rebinding_time = 0	integer value	DHCP rebinding time T2 (in seconds). If set to 0, it will default to 7/8 of the lease time.
dhcp_renewal_time = 0	integer value	DHCP renewal time T1 (in seconds). If set to 0, it will default to half of the lease time.

Configuration option = Default value	Type	Description
dnsmasq_base_log_dir = None	string value	Base log dir for dnsmasq logging. The log contains DHCP and DNS log information and is useful for debugging issues with either DHCP or DNS. If this section is null, disable dnsmasq log.
<code>`dnsmasq_config_file = `</code>	string value	Override the default dnsmasq settings with this file.
dnsmasq_dns_servers = []	list value	Comma-separated list of the DNS servers which will be used as forwarders.
dnsmasq_enable_addr6_list = False	boolean value	Enable dhcp-host entry with list of addresses when port has multiple IPv6 addresses in the same subnet.
dnsmasq_lease_max = 16777216	integer value	Limit number of leases to prevent a denial-of-service.
dnsmasq_local_resolv = False	boolean value	Enables the dnsmasq service to provide name resolution for instances via DNS resolvers on the host running the DHCP agent. Effectively removes the <code>--no-resolv</code> option from the dnsmasq process arguments. Adding custom DNS resolvers to the <code>dnsmasq_dns_servers</code> option disables this feature.
enable_isolated_metadata = False	boolean value	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). This option doesn't have any effect when <code>force_metadata</code> is set to True.
enable_metadata_network = False	boolean value	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:1 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 <code>enable_isolated_metadata = True</code> .
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.

Configuration option = Default value	Type	Description
force_metadata = False	boolean value	In some cases the Neutron router is not present to provide the metadata IP but the DHCP server can be used to provide this info. Setting this value will force the DHCP server to append specific host routes to the DHCP request. If this option is set, then the metadata service will be activated for all the networks.
<code>`instance_format = [instance: %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.
<code>`instance_uuid_format = [instance: %(uuid)s] `</code>	string value	The format for an instance UUID that is passed with the log message.
interface_driver = None	string value	The driver used to manage the virtual interface.
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code> . This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code> . This option is ignored if <code>log_config_append</code> is set.
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless <code>log_rotation_type</code> is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.

Configuration option = Default value	Type	Description
logging_context_format_string = <code>%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s</code>	string value	Format string to use for log messages with context. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_debug_format_s uffix = <code>%(funcName)s %(pathname)s:%(lineno)d</code>	string value	Additional data to append to log message when logging level for the message is <code>DEBUG</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_default_format_s tring = <code>%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s</code>	string value	Format string to use for log messages when context is undefined. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_exception_prefix = <code>%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s</code>	string value	Prefix each line of exception output with this format. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_user_identity_for mat = <code>%(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s</code>	string value	Defines the format string for <code>%(user_identity)s</code> that is used in <code>logging_context_format_string</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
num_sync_threads = 4	integer value	Number of threads to use during sync process. Should not exceed connection pool size configured on server.
ovs_integration_bridge = br-int	string value	Name of Open vSwitch bridge to use
ovs_use_veth = False	boolean value	Uses veth for an OVS interface or not. Support kernels with limited namespace support (e.g. RHEL 6.5) and rate limiting on router's gateway port so long as <code>ovs_use_veth</code> is set to <code>True</code> .

Configuration option = Default value	Type	Description
publish_errors = False	boolean value	Enables or disables publication of error events.
rate_limit_burst = 0	integer value	Maximum number of logged messages per <code>rate_limit_interval</code> .
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to <code>rate_limit_except_level</code> are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
resync_interval = 5	integer value	The DHCP agent will resync its state with Neutron to recover from any transient notification or RPC errors. The interval is maximum number of seconds between attempts. The resync can be done more often based on the events triggered.
resync_throttle = 1	integer value	Throttle the number of resync state events between the local DHCP state and Neutron to only once per <code>resync_throttle</code> seconds. The value of throttle introduces a minimum interval between resync state events. Otherwise the resync may end up in a busy-loop. The value must be less than <code>resync_interval</code> .
rpc_response_max_timeout = 600	integer value	Maximum seconds to wait for a response from an RPC call.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if <code>log_config_append</code> is set.
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if <code>log_config_append</code> is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if <code>log_config_append</code> is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if <code>log_config_append</code> is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.

Configuration option = Default value	Type	Description
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

8.1.2. agent

The following table outlines the options available under the **[agent]** group in the **/etc/neutron/dhcp_agent.ini** file.

Table 8.1. agent

Configuration option = Default value	Type	Description
availability_zone = nova	string value	Availability zone of this node
log_agent_heartbeats = False	boolean value	Log agent heartbeats
report_interval = 30	floating point value	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.

8.1.3. ovs

The following table outlines the options available under the **[ovs]** group in the **/etc/neutron/dhcp_agent.ini** file.

Table 8.2. ovs

Configuration option = Default value	Type	Description
bridge_mac_table_size = 50000	integer value	The maximum number of MAC addresses to learn on a bridge managed by the Neutron OVS agent. Values outside a reasonable range (10 to 1,000,000) might be overridden by Open vSwitch according to the documentation.

Configuration option = Default value	Type	Description
igmp_snooping_enable = False	boolean value	Enable IGMP snooping for integration bridge. If this option is set to True, support for Internet Group Management Protocol (IGMP) is enabled in integration bridge. Setting this option to True will also enable Open vSwitch mcast-snooping-disable-flood-unregistered flag. This option will disable flooding of unregistered multicast packets to all ports. The switch will send unregistered multicast packets only to ports connected to multicast routers.
ovsdb_connection = tcp:127.0.0.1:6640	string value	The connection string for the OVSDb backend. Will be used for all ovsdb commands and by ovsdb-client when monitoring
ovsdb_debug = False	boolean value	Enable OVSDb debug logs
ovsdb_timeout = 10	integer value	Timeout in seconds for ovsdb commands. If the timeout expires, ovsdb commands will fail with ALARMCLOCK error.
ssl_ca_cert_file = None	string value	The Certificate Authority (CA) certificate to use when interacting with OVSDb. Required when using an "ssl:" prefixed ovsdb_connection
ssl_cert_file = None	string value	The SSL certificate file to use when interacting with OVSDb. Required when using an "ssl:" prefixed ovsdb_connection
ssl_key_file = None	string value	The SSL private key file to use when interacting with OVSDb. Required when using an "ssl:" prefixed ovsdb_connection

8.2. L3_AGENT.INI

This section contains options for the `/etc/neutron/l3_agent.ini` file.

8.2.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/neutron/l3_agent.ini` file.

.

Configuration option = Default value	Type	Description
agent_mode = legacy	string value	The working mode for the agent. Allowed modes are: <i>legacy</i> - 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. <i>dvr</i> - this mode enables DVR functionality and must be used for an L3 agent that runs on a compute host. <i>dvr_snat</i> - 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). <i>dvr_no_external</i> - this mode enables only East/West DVR routing functionality for a L3 agent that runs on a compute host, the North/South functionality such as DNAT and SNAT will be provided by the centralized network node that is running in <i>dvr_snat</i> mode. This mode should be used when there is no external network connectivity on the compute host.
api_workers = None	integer value	Number of separate API worker processes for service. If not specified, the default is equal to the number of CPUs available for best performance, capped by potential RAM usage.
cleanup_on_shutdown = False	boolean value	Delete all routers on L3 agent shutdown. For L3 HA routers it includes a shutdown of keepalived and the state change monitor. NOTE: Setting to True could affect the data plane when stopping or restarting the L3 agent.
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.

Configuration option = Default value	Type	Description
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
enable_metadata_proxy = True	boolean value	Allow running metadata proxy.
external_ingress_mark = 0x2	string value	Iptables mangle mark used to mark ingress from external network. This mark will be masked with 0xffff so that only the lower 16 bits will be used.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
ha_confs_path = \$state_path/ha_confs	string value	Location to store keepalived config files
ha_keepalived_state_cha nge_server_threads = <based on operating system>	integer value	Number of concurrent threads for keepalived server connection requests. More threads create a higher CPU load on the agent node.
ha_vrrp_advert_int = 2	integer value	The advertisement interval in seconds

Configuration option = Default value	Type	Description
ha_vrrp_auth_password = None	string value	VRRP authentication password
ha_vrrp_auth_type = PASS	string value	VRRP authentication type
ha_vrrp_garp_master_delay = 5	integer value	The delay for second set of gratuitous ARPs after lower priority advert received when MASTER. NOTE: this config option will be available only in OSP13 and OSP16. Future releases will implement a template form to provide the "keepalived" configuration.
ha_vrrp_garp_master_rep eat = 5	integer value	The number of gratuitous ARP messages to send at a time after transition to MASTER. NOTE: this config option will be available only in OSP13 and OSP16. Future releases will implement a template form to provide the "keepalived" configuration.
ha_vrrp_health_check_int erval = 0	integer value	The VRRP health check interval in seconds. Values > 0 enable VRRP health checks. Setting it to 0 disables VRRP health checks. Recommended value is 5. This will cause pings to be sent to the gateway IP address(es) - requires ICMP_ECHO_REQUEST to be enabled on the gateway(s). If a gateway fails, all routers will be reported as primary, and a primary election will be repeated in a round-robin fashion, until one of the routers restores the gateway connection.
handle_internal_only_rou ters = True	boolean value	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.
<code>`instance_format = [instance: %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.
<code>`instance_uuid_format = [instance: %(uuid)s] `</code>	string value	The format for an instance UUID that is passed with the log message.
interface_driver = None	string value	The driver used to manage the virtual interface.

Configuration option = Default value	Type	Description
<code>`ipv6_gateway = `</code>	string value	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 <code>ipv6_gateway</code> , 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.
keepalived_use_no_track = True	boolean value	If keepalived without support for "no_track" option is used, this should be set to False. Support for this option was introduced in keepalived 2.x
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, <code>log-date-format</code>).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code> . This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code> . This option is ignored if <code>log_config_append</code> is set.
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless <code>log_rotation_type</code> is set to "interval".

Configuration option = Default value	Type	Description
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.% (msecs)03d %(process)d %(levelname)s %(name)s [% (request_id)s % (user_identity)s] % (instance)s%(message)s	string value	Format string to use for log messages with context. Used by oslo_log.formatters.ContextFormatter
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is DEBUG. Used by oslo_log.formatters.ContextFormatter
logging_default_format_s tring = %(asctime)s.% (msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s% (message)s	string value	Format string to use for log messages when context is undefined. Used by oslo_log.formatters.ContextFormatter
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	string value	Prefix each line of exception output with this format. Used by oslo_log.formatters.ContextFormatter
logging_user_identity_for mat = %(user)s % (tenant)s %(domain)s % (user_domain)s % (project_domain)s	string value	Defines the format string for %(user_identity)s that is used in logging_context_format_string. Used by oslo_log.formatters.ContextFormatter
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
max_rtr_adv_interval = 100	integer value	MaxRtrAdvInterval setting for radvd.conf
metadata_access_mark = 0x1	string value	Iptables mangle mark used to mark metadata valid requests. This mark will be masked with 0xffff so that only the lower 16 bits will be used.

Configuration option = Default value	Type	Description
metadata_port = 9697	port value	TCP Port used by Neutron metadata namespace proxy.
min_rtr_adv_interval = 30	integer value	MinRtrAdvInterval setting for radvd.conf
ovs_integration_bridge = br-int	string value	Name of Open vSwitch bridge to use
ovs_use_veth = False	boolean value	Uses veth for an OVS interface or not. Support kernels with limited namespace support (e.g. RHEL 6.5) and rate limiting on router's gateway port so long as ovs_use_veth is set to True.
pd_confs = \$state_path/pd	string value	Location to store IPv6 PD files.
periodic_fuzzy_delay = 5	integer value	Range of seconds to randomly delay when starting the periodic task scheduler to reduce stampeding. (Disable by setting to 0)
periodic_interval = 40	integer value	Seconds between running periodic tasks.
prefix_delegation_driver = dibbler	string value	Driver used for ipv6 prefix delegation. This needs to be an entry point defined in the neutron.agent.linux.pd_drivers namespace. See setup.cfg for entry points included with the neutron source.
publish_errors = False	boolean value	Enables or disables publication of error events.
ra_confs = \$state_path/ra	string value	Location to store IPv6 RA config files
`radvd_user = `	string value	The username passed to radvd, used to drop root privileges and change user ID to username and group ID to the primary group of username. If no user specified (by default), the user executing the L3 agent will be passed. If "root" specified, because radvd is spawned as root, no "username" parameter will be passed.
rate_limit_burst = 0	integer value	Maximum number of logged messages per rate_limit_interval.

Configuration option = Default value	Type	Description
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
rpc_response_max_timeout = 600	integer value	Maximum seconds to wait for a response from an RPC call.
rpc_state_report_workers = 1	integer value	Number of RPC worker processes dedicated to state reports queue.
rpc_workers = None	integer value	Number of RPC worker processes for service. If not specified, the default is equal to half the number of API workers.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
vendor_pen = 8888	string value	A decimal value as Vendor's Registered Private Enterprise Number as required by RFC3315 DUID-EN.

Configuration option = Default value	Type	Description
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

8.2.2. agent

The following table outlines the options available under the **[agent]** group in the `/etc/neutron/l3_agent.ini` file.

Table 8.3. agent

Configuration option = Default value	Type	Description
availability_zone = nova	string value	Availability zone of this node
extensions = []	list value	Extensions list to use
log_agent_heartbeats = False	boolean value	Log agent heartbeats
report_interval = 30	floating point value	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.

8.2.3. network_log

The following table outlines the options available under the **[network_log]** group in the `/etc/neutron/l3_agent.ini` file.

Table 8.4. network_log

Configuration option = Default value	Type	Description
burst_limit = 25	integer value	Maximum number of packets per rate_limit.
local_output_log_base = None	string value	Output logfile path on agent side, default syslog file.
rate_limit = 100	integer value	Maximum packets logging per second.

8.2.4. ovs

The following table outlines the options available under the **[ovs]** group in the `/etc/neutron/l3_agent.ini` file.

Table 8.5. ovs

Configuration option = Default value	Type	Description
bridge_mac_table_size = 50000	integer value	The maximum number of MAC addresses to learn on a bridge managed by the Neutron OVS agent. Values outside a reasonable range (10 to 1,000,000) might be overridden by Open vSwitch according to the documentation.
igmp_snooping_enable = False	boolean value	Enable IGMP snooping for integration bridge. If this option is set to True, support for Internet Group Management Protocol (IGMP) is enabled in integration bridge. Setting this option to True will also enable Open vSwitch mcast-snooping-disable-flood-unregistered flag. This option will disable flooding of unregistered multicast packets to all ports. The switch will send unregistered multicast packets only to ports connected to multicast routers.
ovsdb_connection = tcp:127.0.0.1:6640	string value	The connection string for the OVSDDB backend. Will be used for all ovsdb commands and by ovsdb-client when monitoring
ovsdb_debug = False	boolean value	Enable OVSDDB debug logs
ovsdb_timeout = 10	integer value	Timeout in seconds for ovsdb commands. If the timeout expires, ovsdb commands will fail with ALARMCLOCK error.
ssl_ca_cert_file = None	string value	The Certificate Authority (CA) certificate to use when interacting with OVSDDB. Required when using an "ssl:" prefixed ovsdb_connection
ssl_cert_file = None	string value	The SSL certificate file to use when interacting with OVSDDB. Required when using an "ssl:" prefixed ovsdb_connection
ssl_key_file = None	string value	The SSL private key file to use when interacting with OVSDDB. Required when using an "ssl:" prefixed ovsdb_connection

8.3. LINUXBRIDGE_AGENT.INI

This section contains options for the `/etc/neutron/plugins/ml2/linuxbridge_agent.ini` file.

8.3.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/neutron/plugins/ml2/linuxbridge_agent.ini` file.

Configuration option = Default value	Type	Description
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpidd=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystone.middleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
<code>`instance_format = [instance: %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.
<code>`instance_uuid_format = [instance: %(uuid)s] `</code>	string value	The format for an instance UUID that is passed with the log message.

Configuration option = Default value	Type	Description
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code> . This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code> . This option is ignored if <code>log_config_append</code> is set.
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless <code>log_rotation_type</code> is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	string value	Format string to use for log messages with context. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is <code>DEBUG</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>

Configuration option = Default value	Type	Description
logging_default_format_string = <code>%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s</code>	string value	Format string to use for log messages when context is undefined. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_exception_prefix = <code>%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s</code>	string value	Prefix each line of exception output with this format. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_user_identity_format = <code>%(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s</code>	string value	Defines the format string for <code>%(user_identity)s</code> that is used in <code>logging_context_format_string</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
publish_errors = False	boolean value	Enables or disables publication of error events.
rate_limit_burst = 0	integer value	Maximum number of logged messages per <code>rate_limit_interval</code> .
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to <code>rate_limit_except_level</code> are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
rpc_response_max_timeout = 600	integer value	Maximum seconds to wait for a response from an RPC call.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if <code>log_config_append</code> is set.

Configuration option = Default value	Type	Description
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

8.3.2. agent

The following table outlines the options available under the **[agent]** group in the `/etc/neutron/plugins/ml2/linuxbridge_agent.ini` file.

Table 8.6. agent

Configuration option = Default value	Type	Description
dscp = None	integer value	The DSCP value to use for outer headers during tunnel encapsulation.
dscp_inherit = False	boolean value	If set to True, the DSCP value of tunnel interfaces is overwritten and set to inherit. The DSCP value of the inner header is then copied to the outer header.
extensions = []	list value	Extensions list to use

Configuration option = Default value	Type	Description
polling_interval = 2	integer value	The number of seconds the agent will wait between polling for local device changes.
quitting_rpc_timeout = 10	integer value	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

8.3.3. linux_bridge

The following table outlines the options available under the **[linux_bridge]** group in the `/etc/neutron/plugins/ml2/linuxbridge_agent.ini` file.

Table 8.7. linux_bridge

Configuration option = Default value	Type	Description
bridge_mappings = []	list value	List of <physical_network>:<physical_bridge>
physical_interface_mappings = []	list value	Comma-separated list of <physical_network>:<physical_interface> tuples mapping physical network names to the agent's node-specific physical network interfaces to be used for flat and VLAN networks. All physical networks listed in <code>network_vlan_ranges</code> on the server should have mappings to appropriate interfaces on each agent.

8.3.4. network_log

The following table outlines the options available under the **[network_log]** group in the `/etc/neutron/plugins/ml2/linuxbridge_agent.ini` file.

Table 8.8. network_log

Configuration option = Default value	Type	Description
burst_limit = 25	integer value	Maximum number of packets per rate_limit.
local_output_log_base = None	string value	Output logfile path on agent side, default syslog file.
rate_limit = 100	integer value	Maximum packets logging per second.

8.3.5. securitygroup

The following table outlines the options available under the **[securitygroup]** group in the **/etc/neutron/plugins/ml2/linuxbridge_agent.ini** file.

Table 8.9. securitygroup

Configuration option = Default value	Type	Description
enable_ipset = True	boolean value	Use ipset to speed-up the iptables based security groups. Enabling ipset support requires that ipset is installed on L2 agent node.
enable_security_group = True	boolean value	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	string value	Driver for security groups firewall in the L2 agent
permitted_ethertypes = []	list value	Comma-separated list of ethertypes to be permitted, in hexadecimal (starting with "0x"). For example, "0x4008" to permit InfiniBand.

8.3.6. vxlan

The following table outlines the options available under the **[vxlan]** group in the **/etc/neutron/plugins/ml2/linuxbridge_agent.ini** file.

Table 8.10. vxlan

Configuration option = Default value	Type	Description
arp_responder = False	boolean value	Enable local ARP responder which provides local responses instead of performing ARP broadcast into the overlay. Enabling local ARP responder is not fully compatible with the allowed-address-pairs extension.
enable_vxlan = True	boolean value	Enable VXLAN on the agent. Can be enabled when agent is managed by ml2 plugin using linuxbridge mechanism driver
l2_population = False	boolean value	Extension to use alongside ml2 plugin's l2population mechanism driver. It enables the plugin to populate VXLAN forwarding table.

Configuration option = Default value	Type	Description
local_ip = None	IP address value	IP address of local overlay (tunnel) network endpoint. Use either an IPv4 or IPv6 address that resides on one of the host network interfaces. The IP version of this value must match the value of the <i>overlay_ip_version</i> option in the ML2 plug-in configuration file on the neutron server node(s).
multicast_ranges = []	list value	Optional comma-separated list of <multicast address>:<vni_min>:<vni_max> triples describing how to assign a multicast address to VXLAN according to its VNI ID.
tos = None	integer value	TOS for vxlan interface protocol packets. This option is deprecated in favor of the dscp option in the AGENT section and will be removed in a future release. To convert the TOS value to DSCP, divide by 4.
ttl = None	integer value	TTL for vxlan interface protocol packets.
udp_dstport = None	port value	The UDP port used for VXLAN communication. By default, the Linux kernel doesn't use the IANA assigned standard value, so if you want to use it, this option must be set to 4789. It is not set by default because of backward compatibility.
udp_srcport_max = 0	port value	The maximum of the UDP source port range used for VXLAN communication.
udp_srcport_min = 0	port value	The minimum of the UDP source port range used for VXLAN communication.
vxlan_group = 224.0.0.1	string value	Multicast group(s) for vxlan interface. A range of group addresses may be specified by using CIDR notation. Specifying a range allows different VNIs to use different group addresses, reducing or eliminating spurious broadcast traffic to the tunnel endpoints. To reserve a unique group for each possible (24-bit) VNI, use a /8 such as 239.0.0.0/8. This setting must be the same on all the agents.

8.4. METADATA_AGENT.INI

This section contains options for the `/etc/neutron/metadata_agent.ini` file.

8.4.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/neutron/metadata_agent.ini` file.

Configuration option = Default value	Type	Description
auth_ca_cert = None	string value	Certificate Authority public key (CA cert) file for ssl
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.
default_log_levels = ['amqp=WARN', 'amqp-lib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
`instance_format = [instance: %(uuid)s] `	string value	The format for an instance that is passed with the log message.
`instance_uuid_format = [instance: %(uuid)s] `	string value	The format for an instance UUID that is passed with the log message.

Configuration option = Default value	Type	Description
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code> . This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code> . This option is ignored if <code>log_config_append</code> is set.
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless <code>log_rotation_type</code> is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	string value	Format string to use for log messages with context. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_debug_format_suffix = %(funcName)s %(pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is <code>DEBUG</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>

Configuration option = Default value	Type	Description
logging_default_format_string = <code>%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s</code>	string value	Format string to use for log messages when context is undefined. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_exception_prefix = <code>%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s</code>	string value	Prefix each line of exception output with this format. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_user_identity_format = <code>%(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s</code>	string value	Defines the format string for <code>%(user_identity)s</code> that is used in <code>logging_context_format_string</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
metadata_backlog = 4096	integer value	Number of backlog requests to configure the metadata server socket with
<code>`metadata_proxy_group` = `</code>	string value	Group (gid or name) running metadata proxy after its initialization (if empty: agent effective group).
<code>`metadata_proxy_shared_secret` = `</code>	string value	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 <code>[neutron]</code> section.
metadata_proxy_socket = <code>\$state_path/metadata_proxy</code>	string value	Location for Metadata Proxy UNIX domain socket.

Configuration option = Default value	Type	Description
metadata_proxy_socket_mode = deduce	string value	Metadata Proxy UNIX domain socket mode, 4 values allowed: <i>deduce</i> : deduce mode from metadata_proxy_user/group values, <i>user</i> : set metadata proxy socket mode to 0o644, to use when metadata_proxy_user is agent effective user or root, <i>group</i> : set metadata proxy socket mode to 0o664, to use when metadata_proxy_group is agent effective group or root, <i>all</i> : set metadata proxy socket mode to 0o666, to use otherwise.
<code>`metadata_proxy_user = `</code>	string value	User (uid or name) running metadata proxy after its initialization (if empty: agent effective user).
metadata_workers = <based on operating system>	integer value	Number of separate worker processes for metadata server (defaults to 2 when used with ML2/OVN and half of the number of CPUs with other backend drivers)
<code>`nova_client_cert = `</code>	string value	Client certificate for nova metadata api server.
<code>`nova_client_priv_key = `</code>	string value	Private key of client certificate.
nova_metadata_host = 127.0.0.1	host address value	IP address or DNS name of Nova metadata server.
nova_metadata_insecure = False	boolean value	Allow to perform insecure SSL (https) requests to nova metadata
nova_metadata_port = 8775	port value	TCP Port used by Nova metadata server.
nova_metadata_protocol = http	string value	Protocol to access nova metadata, http or https
publish_errors = False	boolean value	Enables or disables publication of error events.
rate_limit_burst = 0	integer value	Maximum number of logged messages per rate_limit_interval.
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.

Configuration option = Default value	Type	Description
rpc_response_max_timeout = 600	integer value	Maximum seconds to wait for a response from an RPC call.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

8.4.2. agent

The following table outlines the options available under the **[agent]** group in the `/etc/neutron/metadata_agent.ini` file.

Table 8.11. agent

Configuration option = Default value	Type	Description
log_agent_heartbeats = False	boolean value	Log agent heartbeats

Configuration option = Default value	Type	Description
report_interval = 30	floating point value	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.

8.4.3. cache

The following table outlines the options available under the **[cache]** group in the **/etc/neutron/metadata_agent.ini** file.

Table 8.12. cache

Configuration option = Default value	Type	Description
backend = dogpile.cache.null	string value	Cache backend module. For eventlet-based or environments with hundreds of threaded servers, Memcache with pooling (oslo_cache.memcache_pool) is recommended. For environments with less than 100 threaded servers, Memcached (dogpile.cache.memcached) or Redis (dogpile.cache.redis) is recommended. Test environments with a single instance of the server can use the dogpile.cache.memory backend.
backend_argument = []	multi valued	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.oslo	string value	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.
dead_timeout = 60	floating point value	Time in seconds before attempting to add a node back in the pool in the HashClient's internal mechanisms.
debug_cache_backend = False	boolean value	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.

Configuration option = Default value	Type	Description
enable_retry_client = False	boolean value	Enable retry client mechanisms to handle failure. Those mechanisms can be used to wrap all kind of pymemcache clients. The wrapper allows you to define how many attempts to make and how long to wait between attempts.
enable_socket_keepalive = False	boolean value	Global toggle for the socket keepalive of dogpile's pymemcache backend
enabled = False	boolean value	Global toggle for caching.
expiration_time = 600	integer value	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.
hashclient_retry_attempts = 2	integer value	Amount of times a client should be tried before it is marked dead and removed from the pool in the HashClient's internal mechanisms.
hashclient_retry_delay = 1	floating point value	Time in seconds that should pass between retry attempts in the HashClient's internal mechanisms.
memcache_dead_retry = 300	integer value	Number of seconds memcached server is considered dead before it is tried again. (dogpile.cache.memcache and oslo_cache.memcache_pool backends only).
memcache_pool_connection_get_timeout = 10	integer value	Number of seconds that an operation will wait to get a memcache client connection.
memcache_pool_flush_on_reconnect = False	boolean value	Global toggle if memcache will be flushed on reconnect. (oslo_cache.memcache_pool backend only).
memcache_pool_maxsize = 10	integer value	Max total number of open connections to every memcached server. (oslo_cache.memcache_pool backend only).
memcache_pool_unused_timeout = 60	integer value	Number of seconds a connection to memcached is held unused in the pool before it is closed. (oslo_cache.memcache_pool backend only).

Configuration option = Default value	Type	Description
memcache_servers = ['localhost:11211']	list value	Memcache servers in the format of "host:port". (dogpile.cache.memcached and oslo_cache.memcache_pool backends only). If a given host refer to an IPv6 or a given domain refer to IPv6 then you should prefix the given address with the address family (inet6) (e.g inet6[::1]:11211 , inet6:[fd12:3456:789a:1::1]:11211 , inet6:[controller-0.internalapi]:11211). If the address family is not given then default address family used will be inet which correspond to IPv4
memcache_socket_timeout = 1.0	floating point value	Timeout in seconds for every call to a server. (dogpile.cache.memcache and oslo_cache.memcache_pool backends only).
proxies = []	list value	Proxy classes to import that will affect the way the dogpile.cache backend functions. See the dogpile.cache documentation on changing-backend-behavior.
retry_attempts = 2	integer value	Number of times to attempt an action before failing.
retry_delay = 0	floating point value	Number of seconds to sleep between each attempt.
socket_keepalive_count = 1	integer value	The maximum number of keepalive probes TCP should send before dropping the connection. Should be a positive integer greater than zero.
socket_keepalive_idle = 1	integer value	The time (in seconds) the connection needs to remain idle before TCP starts sending keepalive probes. Should be a positive integer most greater than zero.
socket_keepalive_interval = 1	integer value	The time (in seconds) between individual keepalive probes. Should be a positive integer greater than zero.
tls_allowed_ciphers = None	string value	Set the available ciphers for sockets created with the TLS context. It should be a string in the OpenSSL cipher list format. If not specified, all OpenSSL enabled ciphers will be available.
tls_cafile = None	string value	Path to a file of concatenated CA certificates in PEM format necessary to establish the caching servers' authenticity. If <code>tls_enabled</code> is <code>False</code> , this option is ignored.

Configuration option = Default value	Type	Description
tls_certfile = None	string value	Path to a single file in PEM format containing the client's certificate as well as any number of CA certificates needed to establish the certificate's authenticity. This file is only required when client side authentication is necessary. If <code>tls_enabled</code> is <code>False</code> , this option is ignored.
tls_enabled = False	boolean value	Global toggle for TLS usage when communicating with the caching servers.
tls_keyfile = None	string value	Path to a single file containing the client's private key in. Otherwise the private key will be taken from the file specified in <code>tls_certfile</code> . If <code>tls_enabled</code> is <code>False</code> , this option is ignored.

8.5. METERING_AGENT.INI

This section contains options for the `/etc/neutron/metering_agent.ini` file.

8.5.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/neutron/metering_agent.ini` file.

Configuration option = Default value	Type	Description
debug = False	boolean value	If set to true, the logging level will be set to <code>DEBUG</code> instead of the default <code>INFO</code> level.

Configuration option = Default value	Type	Description
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
driver = neutron.services.meterin g.drivers.noop.noop_driv er.NoopMeteringDriver	string value	Metering driver
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
granular_traffic_data = False	boolean value	Defines if the metering agent driver should present traffic data in a granular fashion, instead of grouping all of the traffic data for all projects and routers where the labels were assigned to. The default value is False for backward compatibility.
`instance_format = [instance: %(uuid)s] `	string value	The format for an instance that is passed with the log message.
`instance_uuid_format = [instance: %(uuid)s] `	string value	The format for an instance UUID that is passed with the log message.

Configuration option = Default value	Type	Description
interface_driver = None	string value	The driver used to manage the virtual interface.
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code> . This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code> . This option is ignored if <code>log_config_append</code> is set.
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless <code>log_rotation_type</code> is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	string value	Format string to use for log messages with context. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is <code>DEBUG</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>

Configuration option = Default value	Type	Description
logging_default_format_string = <code>%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s</code>	string value	Format string to use for log messages when context is undefined. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_exception_prefix = <code>%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s</code>	string value	Prefix each line of exception output with this format. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_user_identity_format = <code>%(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s</code>	string value	Defines the format string for <code>%(user_identity)s</code> that is used in <code>logging_context_format_string</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
measure_interval = 30	integer value	Interval between two metering measures
ovs_integration_bridge = br-int	string value	Name of Open vSwitch bridge to use
ovs_use_veth = False	boolean value	Uses veth for an OVS interface or not. Support kernels with limited namespace support (e.g. RHEL 6.5) and rate limiting on router's gateway port so long as <code>ovs_use_veth</code> is set to True.
publish_errors = False	boolean value	Enables or disables publication of error events.
rate_limit_burst = 0	integer value	Maximum number of logged messages per <code>rate_limit_interval</code> .
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to <code>rate_limit_except_level</code> are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.

Configuration option = Default value	Type	Description
report_interval = 300	integer value	Interval between two metering reports
rpc_response_max_timeout = 600	integer value	Maximum seconds to wait for a response from an RPC call.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

8.5.2. agent

The following table outlines the options available under the **[agent]** group in the **/etc/neutron/metering_agent.ini** file.

Table 8.13. agent

Configuration option = Default value	Type	Description
log_agent_heartbeats = False	boolean value	Log agent heartbeats
report_interval = 30	floating point value	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.

8.5.3. ovs

The following table outlines the options available under the **[ovs]** group in the `/etc/neutron/metering_agent.ini` file.

Table 8.14. ovs

Configuration option = Default value	Type	Description
bridge_mac_table_size = 50000	integer value	The maximum number of MAC addresses to learn on a bridge managed by the Neutron OVS agent. Values outside a reasonable range (10 to 1,000,000) might be overridden by Open vSwitch according to the documentation.
igmp_snooping_enable = False	boolean value	Enable IGMP snooping for integration bridge. If this option is set to True, support for Internet Group Management Protocol (IGMP) is enabled in integration bridge. Setting this option to True will also enable Open vSwitch mcast-snooping-disable-flood-unregistered flag. This option will disable flooding of unregistered multicast packets to all ports. The switch will send unregistered multicast packets only to ports connected to multicast routers.
ovsdb_connection = tcp:127.0.0.1:6640	string value	The connection string for the OVSDb backend. Will be used for all ovsdb commands and by ovsdb-client when monitoring
ovsdb_debug = False	boolean value	Enable OVSDb debug logs
ovsdb_timeout = 10	integer value	Timeout in seconds for ovsdb commands. If the timeout expires, ovsdb commands will fail with ALARMCLOCK error.
ssl_ca_cert_file = None	string value	The Certificate Authority (CA) certificate to use when interacting with OVSDb. Required when using an "ssl:" prefixed ovsdb_connection

Configuration option = Default value	Type	Description
ssl_cert_file = None	string value	The SSL certificate file to use when interacting with OVSDb. Required when using an "ssl:" prefixed ovbdb_connection
ssl_key_file = None	string value	The SSL private key file to use when interacting with OVSDb. Required when using an "ssl:" prefixed ovbdb_connection

8.6. ML2_CONF.INI

This section contains options for the `/etc/neutron/plugins/ml2/ml2_conf.ini` file.

8.6.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/neutron/plugins/ml2/ml2_conf.ini` file.

Configuration option = Default value	Type	Description
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.

Configuration option = Default value	Type	Description
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpids=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
`instance_format = [instance: %(uuid)s] `	string value	The format for an instance that is passed with the log message.
`instance_uuid_format = [instance:%(uuid)s] `	string value	The format for an instance UUID that is passed with the log message.
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).

Configuration option = Default value	Type	Description
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code> . This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code> . This option is ignored if <code>log_config_append</code> is set.
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless <code>log_rotation_type</code> is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	string value	Format string to use for log messages with context. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_debug_format_suffix = %(funcName)s %(pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is <code>DEBUG</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_default_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s	string value	Format string to use for log messages when context is undefined. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s	string value	Prefix each line of exception output with this format. Used by <code>oslo_log.formatters.ContextFormatter</code>

Configuration option = Default value	Type	Description
logging_user_identity_format = <code>%(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s</code>	string value	Defines the format string for <code>%(user_identity)s</code> that is used in <code>logging_context_format_string</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if <code>"log_rotation_type"</code> is not set to <code>"size"</code> .
publish_errors = False	boolean value	Enables or disables publication of error events.
rate_limit_burst = 0	integer value	Maximum number of logged messages per <code>rate_limit_interval</code> .
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to <code>rate_limit_except_level</code> are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if <code>log_config_append</code> is set.
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if <code>log_config_append</code> is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if <code>log_config_append</code> is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if <code>log_config_append</code> is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.

Configuration option = Default value	Type	Description
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

8.6.2. ml2

The following table outlines the options available under the **[ml2]** group in the **/etc/neutron/plugins/ml2/ml2_conf.ini** file.

Table 8.15. ml2

Configuration option = Default value	Type	Description
extension_drivers = []	list value	An ordered list of extension driver entrypoints to be loaded from the neutron.ml2.extension_drivers namespace. For example: extension_drivers = port_security,qos
external_network_type = None	string value	Default network type for external networks when no provider attributes are specified. By default it is None, which means that if provider attributes are not specified while creating external networks then they will have the same type as tenant networks. Allowed values for external_network_type config option depend on the network type values configured in type_drivers config option.
mechanism_drivers = []	list value	An ordered list of networking mechanism driver entrypoints to be loaded from the neutron.ml2.mechanism_drivers namespace.
overlay_ip_version = 4	integer value	IP version of all overlay (tunnel) network endpoints. Use a value of 4 for IPv4 or 6 for IPv6.
path_mtu = 0	integer value	Maximum size of an IP packet (MTU) that can traverse the underlying physical network infrastructure without fragmentation when using an overlay/tunnel protocol. This option allows specifying a physical network MTU value that differs from the default global_physnet_mtu value.

Configuration option = Default value	Type	Description
physical_network_mtu = []	list value	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 <code>global_physnet_mtu</code> value.
tenant_network_types = ['local']	list value	Ordered list of network_types to allocate as tenant networks. The default value <code>local</code> is useful for single-box testing but provides no connectivity between hosts.
type_drivers = ['local', 'flat', 'vlan', 'gre', 'vxlan', 'geneve']	list value	List of network type driver entrypoints to be loaded from the <code>neutron.ml2.type_drivers</code> namespace.

8.6.3. ml2_type_flat

The following table outlines the options available under the **[ml2_type_flat]** group in the `/etc/neutron/plugins/ml2/ml2_conf.ini` file.

Table 8.16. ml2_type_flat

Configuration option = Default value	Type	Description
flat_networks = *	list value	List of physical_network names with which flat networks can be created. Use default <code>*</code> to allow flat networks with arbitrary physical_network names. Use an empty list to disable flat networks.

8.6.4. ml2_type_geneve

The following table outlines the options available under the **[ml2_type_geneve]** group in the `/etc/neutron/plugins/ml2/ml2_conf.ini` file.

Table 8.17. ml2_type_geneve

Configuration option = Default value	Type	Description
max_header_size = 30	integer value	Geneve encapsulation header size is dynamic, this value is used to calculate the maximum MTU for the driver. The default size for this field is 30, which is the size of the Geneve header without any additional option headers.

Configuration option = Default value	Type	Description
vni_ranges = []	list value	Comma-separated list of <vni_min>:<vni_max> tuples enumerating ranges of Geneve VNI IDs that are available for tenant network allocation

8.6.5. ml2_type_gre

The following table outlines the options available under the **[ml2_type_gre]** group in the **/etc/neutron/plugins/ml2/ml2_conf.ini** file.

Table 8.18. ml2_type_gre

Configuration option = Default value	Type	Description
tunnel_id_ranges = []	list value	Comma-separated list of <tun_min>:<tun_max> tuples enumerating ranges of GRE tunnel IDs that are available for tenant network allocation

8.6.6. ml2_type_vlan

The following table outlines the options available under the **[ml2_type_vlan]** group in the **/etc/neutron/plugins/ml2/ml2_conf.ini** file.

Table 8.19. ml2_type_vlan

Configuration option = Default value	Type	Description
network_vlan_ranges = []	list value	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.

8.6.7. ml2_type_vxlan

The following table outlines the options available under the **[ml2_type_vxlan]** group in the **/etc/neutron/plugins/ml2/ml2_conf.ini** file.

Table 8.20. ml2_type_vxlan

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
vni_ranges = []	list value	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	string value	Multicast group for VXLAN. When configured, will enable sending all broadcast traffic to this multicast group. When left unconfigured, will disable multicast VXLAN mode.

8.6.8. ovs_driver

The following table outlines the options available under the **[ovs_driver]** group in the **/etc/neutron/plugins/ml2/ml2_conf.ini** file.

Table 8.21. ovs_driver

Configuration option = Default value	Type	Description
vnic_type_prohibit_list = []	list value	Comma-separated list of VNIC types for which support is administratively prohibited by the mechanism driver. Please note that the supported vnic_types depend on your network interface card, on the kernel version of your operating system, and on other factors, like OVS version. In case of ovs mechanism driver the valid vnic types are normal and direct. Note that direct is supported only from kernel 4.8, and from ovs 2.8.0. Bind DIRECT (SR-IOV) port allows to offload the OVS flows using tc to the SR-IOV NIC. This allows to support hardware offload via tc and that allows us to manage the VF by OpenFlow control plane using representor net-device.

8.6.9. securitygroup

The following table outlines the options available under the **[securitygroup]** group in the **/etc/neutron/plugins/ml2/ml2_conf.ini** file.

Table 8.22. securitygroup

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
enable_ipset = True	boolean value	Use ipset to speed-up the iptables based security groups. Enabling ipset support requires that ipset is installed on L2 agent node.
enable_security_group = True	boolean value	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	string value	Driver for security groups firewall in the L2 agent
permitted_ethertypes = []	list value	Comma-separated list of ethertypes to be permitted, in hexadecimal (starting with "0x"). For example, "0x4008" to permit InfiniBand.

8.6.10. sriov_driver

The following table outlines the options available under the **[sriov_driver]** group in the `/etc/neutron/plugins/ml2/ml2_conf.ini` file.

Table 8.23. sriov_driver

Configuration option = Default value	Type	Description
vnic_type_prohibit_list = []	list value	Comma-separated list of VNIC types for which support is administratively prohibited by the mechanism driver. Please note that the supported vnic_types depend on your network interface card, on the kernel version of your operating system, and on other factors. In case of sriov mechanism driver the valid VNIC types are direct and direct-physical.

8.7. NEUTRON.CONF

This section contains options for the `/etc/neutron/neutron.conf` file.

8.7.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/neutron/neutron.conf` file.

.

Configuration option = Default value	Type	Description
agent_down_time = 75	integer value	Seconds to regard the agent is down; should be at least twice report_interval, to be sure the agent is down for good.
allow_automatic_dhcp_failover = True	boolean value	Automatically remove networks from offline DHCP agents.
allow_automatic_l3agent_failover = False	boolean value	Automatically reschedule routers from offline L3 agents to online L3 agents.
allow_bulk = True	boolean value	Allow the usage of the bulk API
allow_overlapping_ips = False	boolean value	Allow overlapping IP support in Neutron. Attention: the following parameter MUST be set to False if Neutron is being used in conjunction with Nova security groups.
allowed_contrack_helpers = [{<i>'amanda'</i>: <i>'tcp'</i>}, {<i>'ftp'</i>: <i>'tcp'</i>}, {<i>'h323'</i>: <i>'udp'</i>}, {<i>'h323'</i>: <i>'tcp'</i>}, {<i>'irc'</i>: <i>'tcp'</i>}, {<i>'netbios-ns'</i>: <i>'udp'</i>}, {<i>'pptp'</i>: <i>'tcp'</i>}, {<i>'sane'</i>: <i>'tcp'</i>}, {<i>'sip'</i>: <i>'udp'</i>}, {<i>'sip'</i>: <i>'tcp'</i>}, {<i>'snmp'</i>: <i>'udp'</i>}, {<i>'tftp'</i>: <i>'udp'</i>}]	list value	Defines the allowed contrack helpers, and contrack helper module protocol constraints.
api_extensions_path = `	string value	The path for API extensions. Note that this can be a colon-separated list of paths. For example: api_extensions_path = extensions:/path/to/more/exts:/even/more/exts. The <i>path</i> of neutron.extensions is appended to this, so if your extensions are in there you don't need to specify them here.
api_paste_config = api-paste.ini	string value	File name for the paste.deploy config for api service
api_workers = None	integer value	Number of separate API worker processes for service. If not specified, the default is equal to the number of CPUs available for best performance, capped by potential RAM usage.
auth_strategy = keystone	string value	The type of authentication to use

Configuration option = Default value	Type	Description
backdoor_port = None	string value	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.
backdoor_socket = None	string value	Enable eventlet backdoor, using the provided path as a unix socket that can receive connections. This option is mutually exclusive with <i>backdoor_port</i> in that only one should be provided. If both are provided then the existence of this option overrides the usage of that option. Inside the path {pid} will be replaced with the PID of the current process.
backlog = 4096	integer value	Number of backlog requests to configure the socket with
base_mac = fa:16:3e:00:00:00	string value	The base MAC address Neutron will use for VIFs. 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.
bind_host = 0.0.0.0	host address value	The host IP to bind to.
bind_port = 9696	port value	The port to bind to
client_socket_timeout = 900	integer value	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.
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
control_exchange = neutron	string value	The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.
core_plugin = None	string value	The core plugin Neutron will use
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.

Configuration option = Default value	Type	Description
default_availability_zones = []	list value	Default value of availability zone hints. The availability zone aware schedulers use this when the resources availability_zone_hints is empty. Multiple availability zones can be specified by a comma separated string. This value can be empty. In this case, even if availability_zone_hints for a resource is empty, availability zone is considered for high availability while scheduling the resource.
default_log_levels = ['amqp=WARN', 'amqp-lib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
dhcp_agent_notification = True	boolean value	Allow sending resource operation notification to DHCP agent
dhcp_agents_per_network = 1	integer value	Number of DHCP agents scheduled to host a tenant network. If this number is greater than 1, the scheduler automatically assigns multiple DHCP agents for a given tenant network, providing high availability for DHCP service.
dhcp_lease_duration = 86400	integer value	DHCP lease duration (in seconds). Use -1 to tell dnsmasq to use infinite lease times.

Configuration option = Default value	Type	Description
dhcp_load_type = networks	string value	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
dns_domain = openstacklocal	string value	Domain to use for building the hostnames
dvr_base_mac = fa:16:3f:00:00:00	string value	The base mac address used for unique DVR instances by Neutron. 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 <i>dvr_base_mac</i> must be different from <i>base_mac</i> 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
enable_dvr = True	boolean value	Determine if setup is configured for DVR. If False, DVR API extension will be disabled.
enable_new_agents = True	boolean value	Agent starts with admin_state_up=False when enable_new_agents=False. In the case, user's resources will not be scheduled automatically to the agent until admin changes admin_state_up to True.
enable_services_on_agents_with_admin_state_down = False	boolean value	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.
enable_snat_by_default = True	boolean value	Define the default value of enable_snat if not provided in external_gateway_info.

Configuration option = Default value	Type	Description
enable_traditional_dhcp = True	boolean value	If False, neutron-server will disable the following DHCP-agent related functions:1. DHCP provisioning block 2. DHCP scheduler API extension 3. Network scheduling mechanism 4. DHCP RPC/notification
executor_thread_pool_size = 64	integer value	Size of executor thread pool when executor is threading or eventlet.
external_dns_driver = None	string value	Driver for external DNS integration.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
filter_validation = True	boolean value	If True, then allow plugins to decide whether to perform validations on filter parameters. Filter validation is enabled if this config is turned on and it is supported by all plugins
global_physnet_mtu = 1500	integer value	MTU of the underlying physical network. Neutron uses this value to calculate MTU for all virtual network components. For flat and VLAN networks, neutron uses this value without modification. For overlay networks such as VXLAN, neutron automatically subtracts the overlay protocol overhead from this value. Defaults to 1500, the standard value for Ethernet.
graceful_shutdown_timeout = 60	integer value	Specify a timeout after which a gracefully shutdown server will exit. Zero value means endless wait.
host = <based on operating system>	host address value	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.
host_dvr_for_dhcp = True	boolean value	Flag to determine if hosting a DVR local router to the DHCP agent is desired. If False, any L3 function supported by the DHCP agent instance will not be possible, for instance: DNS.

Configuration option = Default value	Type	Description
http_retries = 3	integer value	Number of times client connections (nova, ironic) should be retried on a failed HTTP call. 0 (zero) means connection is attempted only once (not retried). Setting to any positive integer means that on failure the connection is retried that many times. For example, setting to 3 means total attempts to connect will be 4.
<code>`instance_format = [instance: %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.
<code>`instance_uuid_format = [instance: %(uuid)s] `</code>	string value	The format for an instance UUID that is passed with the log message.
interface_driver = None	string value	The driver used to manage the virtual interface.
ipam_driver = internal	string value	Neutron IPAM (IP address management) driver to use. By default, the reference implementation of the Neutron IPAM driver is used.
ipv6_pd_enabled = False	boolean value	Enables IPv6 Prefix Delegation for automatic subnet CIDR allocation. Set to True to enable IPv6 Prefix Delegation for subnet allocation in a PD-capable environment. Users making subnet creation requests for IPv6 subnets without providing a CIDR or subnetpool ID will be given a CIDR via the Prefix Delegation mechanism. Note that enabling PD will override the behavior of the default IPv6 subnetpool.
l3_ha = False	boolean value	Enable HA mode for virtual routers.
l3_ha_net_cidr = 169.254.192.0/18	string value	Subnet used for the l3 HA admin network.
<code>`l3_ha_network_physical_name = `</code>	string value	The physical network name with which the HA network can be created.
<code>`l3_ha_network_type = `</code>	string value	The network type to use when creating the HA network for an HA router. By default or if empty, the first <i>tenant_network_types</i> is used. This is helpful when the VRRP traffic should use a specific network which is not the default one.

Configuration option = Default value	Type	Description
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for %(asctime)s in log records. Default: %(default)s . This option is ignored if log_config_append is set.
log-dir = None	string value	(Optional) The base directory used for relative log_file paths. This option is ignored if log_config_append is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to stderr as defined by use_stderr. This option is ignored if log_config_append is set.
log_options = True	boolean value	Enables or disables logging values of all registered options when starting a service (at DEBUG level).
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless log_rotation_type is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	string value	Format string to use for log messages with context. Used by oslo_log.formatters.ContextFormatter
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is DEBUG. Used by oslo_log.formatters.ContextFormatter

Configuration option = Default value	Type	Description
logging_default_format_string = <code>%(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s</code>	string value	Format string to use for log messages when context is undefined. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_exception_prefix = <code>%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s</code>	string value	Prefix each line of exception output with this format. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_user_identity_format = <code>%(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s</code>	string value	Defines the format string for <code>%(user_identity)s</code> that is used in <code>logging_context_format_string</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
max_allowed_address_pairs = <code>10</code>	integer value	Maximum number of allowed address pairs
max_dns_nameservers = <code>5</code>	integer value	Maximum number of DNS nameservers per subnet
max_header_line = <code>16384</code>	integer value	Maximum line size of message headers to be accepted. <code>max_header_line</code> may need to be increased when using large tokens (typically those generated when keystone is configured to use PKI tokens with big service catalogs).
max_l3_agents_per_router = <code>3</code>	integer value	Maximum number of L3 agents which a HA router will be scheduled on. If it is set to 0 then the router will be scheduled on every agent.
max_logfile_count = <code>30</code>	integer value	Maximum number of rotated log files.
max_logfile_size_mb = <code>200</code>	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
max_routes = <code>30</code>	integer value	Maximum number of routes per router
max_subnet_host_routes = <code>20</code>	integer value	Maximum number of host routes per subnet
<code>`metadata_proxy_group =`</code>	string value	Group (gid or name) running metadata proxy after its initialization (if empty: agent effective group).

Configuration option = Default value	Type	Description
metadata_proxy_socket = \$state_path/metadata_pro xy	string value	Location for Metadata Proxy UNIX domain socket.
metadata_proxy_user = ``	string value	User (uid or name) running metadata proxy after its initialization (if empty: agent effective user).
network_auto_schedule = True	boolean value	Allow auto scheduling networks to DHCP agent.
network_link_prefix = None	string value	This string is prepended to the normal URL that is returned in links to the OpenStack Network API. If it is empty (the default), the URLs are returned unchanged.
network_scheduler_drive r = neutron.scheduler.dhcp_ agent_scheduler.WeightS cheduler	string value	Driver to use for scheduling network to DHCP agent
notify_nova_on_port_dat a_changes = True	boolean value	Send notification to nova when port data (fixed_ips/floatingip) changes so nova can update its cache.
notify_nova_on_port_stat us_changes = True	boolean value	Send notification to nova when port status changes
pagination_max_limit = -1	string value	The maximum number of items returned in a single response, value was <i>infinite</i> or negative integer means no limit
periodic_fuzzy_delay = 5	integer value	Range of seconds to randomly delay when starting the periodic task scheduler to reduce stampeding. (Disable by setting to 0)
periodic_interval = 40	integer value	Seconds between running periodic tasks.
publish_errors = False	boolean value	Enables or disables publication of error events.
rate_limit_burst = 0	integer value	Maximum number of logged messages per rate_limit_interval.

Configuration option = Default value	Type	Description
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
retry_until_window = 30	integer value	Number of seconds to keep retrying to listen
router_auto_schedule = True	boolean value	Allow auto scheduling of routers to L3 agent.
router_distributed = False	boolean value	System-wide flag to determine the type of router that tenants can create. Only admin can override.
router_scheduler_driver = neutron.scheduler.L3_agent_scheduler.LeastRoutersScheduler	string value	Driver to use for scheduling router to a default L3 agent
rpc_conn_pool_size = 30	integer value	Size of RPC connection pool.
rpc_ping_enabled = False	boolean value	Add an endpoint to answer to ping calls. Endpoint is named oslo_rpc_server_ping
rpc_response_max_timeout = 600	integer value	Maximum seconds to wait for a response from an RPC call.
rpc_response_timeout = 60	integer value	Seconds to wait for a response from a call.
rpc_state_report_workers = 1	integer value	Number of RPC worker processes dedicated to state reports queue.
rpc_workers = None	integer value	Number of RPC worker processes for service. If not specified, the default is equal to half the number of API workers.
run_external_periodic_tasks = True	boolean value	Some periodic tasks can be run in a separate process. Should we run them here?
send_events_interval = 2	integer value	Number of seconds between sending events to nova if there are any events to send.

Configuration option = Default value	Type	Description
service_plugins = []	list value	The service plugins Neutron will use
setproctitle = on	string value	Set process name to match child worker role. Available options are: <i>off</i> - retains the previous behavior; <i>on</i> - renames processes to <i>neutron-server: role (original string)</i> ; <i>brief</i> - renames the same as <i>on</i> , but without the original string, such as <i>neutron-server: role</i> .
state_path = /var/lib/neutron	string value	Where to store Neutron state files. This directory must be writable by the agent.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
tcp_keepidle = 600	integer value	Sets the value of TCP_KEEPIDLE in seconds for each server socket. Not supported on OS X.
transport_url = rabbit://	string value	The network address and optional user credentials for connecting to the messaging backend, in URL format. The expected format is: <pre>driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query</pre> Example: <pre>rabbit://rabbitmq:password@127.0.0.1:5672//</pre> For full details on the fields in the URL see the documentation of oslo_messaging.TransportURL at https://docs.openstack.org/oslo.messaging/latest/reference/transport.html
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.

Configuration option = Default value	Type	Description
use_ssl = False	boolean value	Enable SSL on the API server
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
vlan_transparent = False	boolean value	If True, then allow plugins that support it to create VLAN transparent networks.
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.
wsgi_default_pool_size = 100	integer value	Size of the pool of greenthreads used by wsgi
wsgi_keep_alive = True	boolean value	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	string value	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.
wsgi_server_debug = False	boolean value	True if the server should send exception tracebacks to the clients on 500 errors. If False, the server will respond with empty bodies.

8.7.2. agent

The following table outlines the options available under the **[agent]** group in the **/etc/neutron/neutron.conf** file.

Table 8.24. agent

Configuration option = Default value	Type	Description
availability_zone = nova	string value	Availability zone of this node
check_child_processes_action = respawn	string value	Action to be executed when a child process dies

Configuration option = Default value	Type	Description
check_child_processes_interval = 60	integer value	Interval between checks of child process liveness (seconds), use 0 to disable
comment_iptables_rules = True	boolean value	Add comments to iptables rules. Set to false to disallow the addition of comments to generated iptables rules that describe each rule's purpose. System must support the iptables comments module for addition of comments.
debug_iptables_rules = False	boolean value	Duplicate every iptables difference calculation to ensure the format being generated matches the format of iptables-save. This option should not be turned on for production systems because it imposes a performance penalty.
kill_scripts_path = /etc/neutron/kill_scripts/	string value	Location of scripts used to kill external processes. Names of scripts here must follow the pattern: "<process-name>-kill" where <process-name> is name of the process which should be killed using this script. For example, kill script for dnsmasq process should be named "dnsmasq-kill". If path is set to None, then default "kill" command will be used to stop processes.
log_agent_heartbeats = False	boolean value	Log agent heartbeats
report_interval = 30	floating point value	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.
root_helper = sudo	string value	Root helper application. Use <i>sudo neutron-rootwrap /etc/neutron/rootwrap.conf</i> to use the real root filter facility. Change to <i>sudo</i> to skip the filtering and just run the command directly.
root_helper_daemon = None	string value	Root helper daemon application to use when possible. Use <i>sudo neutron-rootwrap-daemon /etc/neutron/rootwrap.conf</i> to run rootwrap in "daemon mode" which has been reported to improve performance at scale. For more information on running rootwrap in "daemon mode", see: https://docs.openstack.org/oslo.rootwrap/latest/user/usage.html#daemon-mode

Configuration option = Default value	Type	Description
use_helper_for_ns_read = True	boolean value	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.

8.7.3. cors

The following table outlines the options available under the **[cors]** group in the `/etc/neutron/neutron.conf` file.

Table 8.25. cors

Configuration option = Default value	Type	Description
allow_credentials = True	boolean value	Indicate that the actual request can include user credentials
allow_headers = ['X-Auth-Token', 'X-Identity-Status', 'X-Roles', 'X-Service-Catalog', 'X-User-Id', 'X-Tenant-Id', 'X-OpenStack-Request-ID']	list value	Indicate which header field names may be used during the actual request.
allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH']	list value	Indicate which methods can be used during the actual request.
allowed_origin = None	list value	Indicate whether this resource may be shared with the domain received in the requests "origin" header. Format: "<protocol>://<host>[:<port>]", no trailing slash. Example: https://horizon.example.com
expose_headers = ['X-Auth-Token', 'X-Subject-Token', 'X-Service-Token', 'X-OpenStack-Request-ID', 'OpenStack-Volume-microversion']	list value	Indicate which headers are safe to expose to the API. Defaults to HTTP Simple Headers.
max_age = 3600	integer value	Maximum cache age of CORS preflight requests.

8.7.4. database

The following table outlines the options available under the **[database]** group in the `/etc/neutron/neutron.conf` file.

Table 8.26. database

Configuration option = Default value	Type	Description
backend = sqlalchemy	string value	The back end to use for the database.
connection = None	string value	The SQLAlchemy connection string to use to connect to the database.
connection_debug = 0	integer value	Verbosity of SQL debugging information: 0=None, 100=Everything.
<code>`connection_parameters = `</code>	string value	Optional URL parameters to append onto the connection URL at connect time; specify as param1=value1&param2=value2&...
connection_recycle_time = 3600	integer value	Connections which have been present in the connection pool longer than this number of seconds will be replaced with a new one the next time they are checked out from the pool.
connection_trace = False	boolean value	Add Python stack traces to SQL as comment strings.
db_inc_retry_interval = True	boolean value	If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = 20	integer value	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	integer value	If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = 1	integer value	Seconds between retries of a database transaction.
<code>`engine = `</code>	string value	Database engine for which script will be generated when using offline migration.
max_overflow = 50	integer value	If set, use this value for max_overflow with SQLAlchemy.
max_pool_size = 5	integer value	Maximum number of SQL connections to keep open in a pool. Setting a value of 0 indicates no limit.
max_retries = 10	integer value	Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.

Configuration option = Default value	Type	Description
mysql_enable_ndb = False	boolean value	If True, transparently enables support for handling MySQL Cluster (NDB).
mysql_sql_mode = TRADITIONAL	string value	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: <code>mysql_sql_mode=</code>
pool_timeout = None	integer value	If set, use this value for pool_timeout with SQLAlchemy.
retry_interval = 10	integer value	Interval between retries of opening a SQL connection.
slave_connection = None	string value	The SQLAlchemy connection string to use to connect to the slave database.
sqlite_synchronous = True	boolean value	If True, SQLite uses synchronous mode.
use_db_reconnect = False	boolean value	Enable the experimental use of database reconnect on connection lost.

8.7.5. healthcheck

The following table outlines the options available under the **[healthcheck]** group in the `/etc/neutron/neutron.conf` file.

Table 8.27. healthcheck

Configuration option = Default value	Type	Description
backends = []	list value	Additional backends that can perform health checks and report that information back as part of a request.
detailed = False	boolean value	Show more detailed information as part of the response. Security note: Enabling this option may expose sensitive details about the service being monitored. Be sure to verify that it will not violate your security policies.
disable_by_file_path = None	string value	Check the presence of a file to determine if an application is running on a port. Used by DisableByFileHealthcheck plugin.

Configuration option = Default value	Type	Description
disable_by_file_paths = []	list value	Check the presence of a file based on a port to determine if an application is running on a port. Expects a "port:path" list of strings. Used by DisableByFilesPortsHealthcheck plugin.
path = /healthcheck	string value	The path to respond to healthcheck requests on.

8.7.6. ironic

The following table outlines the options available under the **[ironic]** group in the **/etc/neutron/neutron.conf** file.

Table 8.28. ironic

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
enable_notifications = False	boolean value	Send notification events to ironic. (For example on relevant port status changes.)

Configuration option = Default value	Type	Description
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
split-loggers = False	boolean value	Log requests to multiple loggers.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User id
username = None	string value	Username

8.7.7. keystone_authtoken

The following table outlines the options available under the **[keystone_authtoken]** group in the **/etc/neutron/neutron.conf** file.

Table 8.29. keystone_authtoken

Configuration option = Default value	Type	Description
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load
auth_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint. This option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release. Deprecated since: Queens <i>*Reason:</i>*The <code>auth_uri</code> option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release.
auth_version = None	string value	API version of the Identity API endpoint.
cache = None	string value	Request environment key where the Swift cache object is stored. When <code>auth_token</code> middleware is deployed with a Swift cache, use this option to have the middleware share a caching backend with swift. Otherwise, use the <code>memcached_servers</code> option instead.
cafile = None	string value	A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
certfile = None	string value	Required if identity server requires client certificate
delay_auth_decision = False	boolean value	Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components.

Configuration option = Default value	Type	Description
enforce_token_bind = permissive	string value	Used to control the use and type of token binding. Can be set to: "disabled" to not check token binding. "permissive" (default) to validate binding information if the bind type is of a form known to the server and ignore it if not. "strict" like "permissive" but if the bind type is unknown the token will be rejected. "required" any form of token binding is needed to be allowed. Finally the name of a binding method that must be present in tokens.
http_connect_timeout = None	integer value	Request timeout value for communicating with Identity API server.
http_request_max_retries = 3	integer value	How many times are we trying to reconnect when communicating with Identity API Server.
include_service_catalog = True	boolean value	(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	boolean value	Verify HTTPS connections.
interface = internal	string value	Interface to use for the Identity API endpoint. Valid values are "public", "internal" (default) or "admin".
keyfile = None	string value	Required if identity server requires client certificate
memcache_pool_conn_get_timeout = 10	integer value	(Optional) Number of seconds that an operation will wait to get a memcached client connection from the pool.
memcache_pool_dead_retry = 300	integer value	(Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = 10	integer value	(Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = 3	integer value	(Optional) Socket timeout in seconds for communicating with a memcached server.
memcache_pool_unused_timeout = 60	integer value	(Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.

Configuration option = Default value	Type	Description
memcache_secret_key = None	string value	(Optional, mandatory if memcache_security_strategy is defined) This string is used for key derivation.
memcache_security_strategy = None	string value	(Optional) If defined, indicate whether token data should be authenticated or authenticated and encrypted. If MAC, token data is authenticated (with HMAC) in the cache. If ENCRYPT, token data is encrypted and authenticated in the cache. If the value is not one of these options or empty, auth_token will raise an exception on initialization.
memcache_use_advanced_pool = False	boolean value	(Optional) Use the advanced (eventlet safe) memcached client pool. The advanced pool will only work under python 2.x.
memcached_servers = None	list value	Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
region_name = None	string value	The region in which the identity server can be found.
service_token_roles = ['service']	list value	A choice of roles that must be present in a service token. Service tokens are allowed to request that an expired token can be used and so this check should tightly control that only actual services should be sending this token. Roles here are applied as an ANY check so any role in this list must be present. For backwards compatibility reasons this currently only affects the allow_expired check.
service_token_roles_required = False	boolean value	For backwards compatibility reasons we must let valid service tokens pass that don't pass the service_token_roles check as valid. Setting this true will become the default in a future release and should be enabled if possible.
service_type = None	string value	The name or type of the service as it appears in the service catalog. This is used to validate tokens that have restricted access rules.
token_cache_time = 300	integer value	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.

Configuration option = Default value	Type	Description
www_authenticate_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint.

8.7.8. nova

The following table outlines the options available under the **[nova]** group in the `/etc/neutron/neutron.conf` file.

Table 8.30. nova

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to

Configuration option = Default value	Type	Description
endpoint_type = public	string value	Type of the nova endpoint to use. This endpoint will be looked up in the keystone catalog and should be one of public, internal or admin.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
live_migration_events = False	boolean value	When this option is enabled, during the live migration, the OVS agent will only send the "vif-plugged-event" when the destination host interface is bound. This option also disables any other agent (like DHCP) to send to Nova this event when the port is provisioned. This option can be enabled if Nova patch https://review.opendev.org/c/openstack/nova/+767368 is in place. This option is temporary and will be removed in Y and the behavior will be "True".
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
region_name = None	string value	Name of nova region to use. Useful if keystone manages more than one region.
split-loggers = False	boolean value	Log requests to multiple loggers.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id

Configuration option = Default value	Type	Description
user-domain-name = None	string value	User's domain name
user-id = None	string value	User id
username = None	string value	Username

8.7.9. oslo_concurrency

The following table outlines the options available under the **[oslo_concurrency]** group in the **/etc/neutron/neutron.conf** file.

Table 8.31. oslo_concurrency

Configuration option = Default value	Type	Description
disable_process_locking = False	boolean value	Enables or disables inter-process locks.
lock_path = None	string value	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.

8.7.10. oslo_messaging_amqp

The following table outlines the options available under the **[oslo_messaging_amqp]** group in the **/etc/neutron/neutron.conf** file.

Table 8.32. oslo_messaging_amqp

Configuration option = Default value	Type	Description
addressing_mode = dynamic	string value	Indicates the addressing mode used by the driver. Permitted values: <i>legacy</i> - use legacy non-routable addressing <i>routable</i> - use routable addresses <i>dynamic</i> - use legacy addresses if the message bus does not support routing otherwise use routable addressing

Configuration option = Default value	Type	Description
anycast_address = anycast	string value	Appended to the address prefix when sending to a group of consumers. Used by the message bus to identify messages that should be delivered in a round-robin fashion across consumers.
broadcast_prefix = broadcast	string value	address prefix used when broadcasting to all servers
connection_retry_backoff = 2	integer value	Increase the connection_retry_interval by this many seconds after each unsuccessful failover attempt.
connection_retry_interval = 1	integer value	Seconds to pause before attempting to re-connect.
connection_retry_interval _max = 30	integer value	Maximum limit for connection_retry_interval + connection_retry_backoff
container_name = None	string value	Name for the AMQP container. must be globally unique. Defaults to a generated UUID
default_notification_exch ange = None	string value	Exchange name used in notification addresses. Exchange name resolution precedence: Target.exchange if set else default_notification_exchange if set else control_exchange if set else <i>notify</i>
default_notify_timeout = 30	integer value	The deadline for a sent notification message delivery. Only used when caller does not provide a timeout expiry.
default_reply_retry = 0	integer value	The maximum number of attempts to re-send a reply message which failed due to a recoverable error.
default_reply_timeout = 30	integer value	The deadline for an rpc reply message delivery.
default_rpc_exchange = None	string value	Exchange name used in RPC addresses. Exchange name resolution precedence: Target.exchange if set else default_rpc_exchange if set else control_exchange if set else <i>rpc</i>
default_send_timeout = 30	integer value	The deadline for an rpc cast or call message delivery. Only used when caller does not provide a timeout expiry.
default_sender_link_time out = 600	integer value	The duration to schedule a purge of idle sender links. Detach link after expiry.

Configuration option = Default value	Type	Description
group_request_prefix = unicast	string value	address prefix when sending to any server in group
idle_timeout = 0	integer value	Timeout for inactive connections (in seconds)
link_retry_delay = 10	integer value	Time to pause between re-connecting an AMQP 1.0 link that failed due to a recoverable error.
multicast_address = multicast	string value	Appended to the address prefix when sending a fanout message. Used by the message bus to identify fanout messages.
notify_address_prefix = openstack.org/om/notify	string value	Address prefix for all generated Notification addresses
notify_server_credit = 100	integer value	Window size for incoming Notification messages
pre_settled = ['rpc-cast', 'rpc-reply']	multi valued	Send messages of this type pre-settled. Pre-settled messages will not receive acknowledgement from the peer. Note well: pre-settled messages may be silently discarded if the delivery fails. Permitted values: <i>rpc-call</i> - send RPC Calls pre-settled <i>rpc-reply</i> - send RPC Replies pre-settled <i>rpc-cast</i> - Send RPC Casts pre-settled <i>notify</i> - Send Notifications pre-settled
pseudo_vhost = True	boolean value	Enable virtual host support for those message buses that do not natively support virtual hosting (such as qpid). When set to true the virtual host name will be added to all message bus addresses, effectively creating a private <i>subnet</i> per virtual host. Set to False if the message bus supports virtual hosting using the <i>hostname</i> field in the AMQP 1.0 Open performative as the name of the virtual host.
reply_link_credit = 200	integer value	Window size for incoming RPC Reply messages.
rpc_address_prefix = openstack.org/om/rpc	string value	Address prefix for all generated RPC addresses
rpc_server_credit = 100	integer value	Window size for incoming RPC Request messages
<code>`sasl_config_dir = `</code>	string value	Path to directory that contains the SASL configuration
<code>`sasl_config_name = `</code>	string value	Name of configuration file (without .conf suffix)
<code>`sasl_default_realm = `</code>	string value	SASL realm to use if no realm present in username

Configuration option = Default value	Type	Description
<code>`sasl_mechanisms = `</code>	string value	Space separated list of acceptable SASL mechanisms
server_request_prefix = exclusive	string value	address prefix used when sending to a specific server
ssl = False	boolean value	Attempt to connect via SSL. If no other ssl-related parameters are given, it will use the system's CA-bundle to verify the server's certificate.
<code>`ssl_ca_file = `</code>	string value	CA certificate PEM file used to verify the server's certificate
<code>`ssl_cert_file = `</code>	string value	Self-identifying certificate PEM file for client authentication
<code>`ssl_key_file = `</code>	string value	Private key PEM file used to sign <code>ssl_cert_file</code> certificate (optional)
ssl_key_password = None	string value	Password for decrypting <code>ssl_key_file</code> (if encrypted)
ssl_verify_vhost = False	boolean value	By default SSL checks that the name in the server's certificate matches the hostname in the <code>transport_url</code> . In some configurations it may be preferable to use the virtual hostname instead, for example if the server uses the Server Name Indication TLS extension (rfc6066) to provide a certificate per virtual host. Set <code>ssl_verify_vhost</code> to True if the server's SSL certificate uses the virtual host name instead of the DNS name.
trace = False	boolean value	Debug: dump AMQP frames to stdout
unicast_address = unicast	string value	Appended to the address prefix when sending to a particular RPC/Notification server. Used by the message bus to identify messages sent to a single destination.

8.7.11. oslo_messaging_kafka

The following table outlines the options available under the **[oslo_messaging_kafka]** group in the `/etc/neutron/neutron.conf` file.

Table 8.33. oslo_messaging_kafka

Configuration option = Default value	Type	Description
compression_codec = none	string value	The compression codec for all data generated by the producer. If not set, compression will not be used. Note that the allowed values of this depend on the kafka version
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
consumer_group = oslo_messaging_consumer	string value	Group id for Kafka consumer. Consumers in one group will coordinate message consumption
enable_auto_commit = False	boolean value	Enable asynchronous consumer commits
kafka_consumer_timeout = 1.0	floating point value	Default timeout(s) for Kafka consumers
kafka_max_fetch_bytes = 1048576	integer value	Max fetch bytes of Kafka consumer
max_poll_records = 500	integer value	The maximum number of records returned in a poll call
pool_size = 10	integer value	Pool Size for Kafka Consumers
producer_batch_size = 16384	integer value	Size of batch for the producer async send
producer_batch_timeout = 0.0	floating point value	Upper bound on the delay for KafkaProducer batching in seconds
sasl_mechanism = PLAIN	string value	Mechanism when security protocol is SASL
security_protocol = PLAINTEXT	string value	Protocol used to communicate with brokers
<code>`ssl_cafile = `</code>	string value	CA certificate PEM file used to verify the server certificate
<code>`ssl_client_cert_file = `</code>	string value	Client certificate PEM file used for authentication.
<code>`ssl_client_key_file = `</code>	string value	Client key PEM file used for authentication.
<code>`ssl_client_key_password = `</code>	string value	Client key password file used for authentication.

8.7.12. oslo_messaging_notifications

The following table outlines the options available under the **[oslo_messaging_notifications]** group in the **/etc/neutron/neutron.conf** file.

Table 8.34. oslo_messaging_notifications

Configuration option = Default value	Type	Description
driver = []	multi valued	The Drivers(s) to handle sending notifications. Possible values are messaging, messagingv2, routing, log, test, noop
retry = -1	integer value	The maximum number of attempts to re-send a notification message which failed to be delivered due to a recoverable error. 0 - No retry, -1 - indefinite
topics = ['notifications']	list value	AMQP topic used for OpenStack notifications.
transport_url = None	string value	A URL representing the messaging driver to use for notifications. If not set, we fall back to the same configuration used for RPC.

8.7.13. oslo_messaging_rabbit

The following table outlines the options available under the **[oslo_messaging_rabbit]** group in the **/etc/neutron/neutron.conf** file.

Table 8.35. oslo_messaging_rabbit

Configuration option = Default value	Type	Description
amqp_auto_delete = False	boolean value	Auto-delete queues in AMQP.
amqp_durable_queues = False	boolean value	Use durable queues in AMQP.
direct_mandatory_flag = True	boolean value	(DEPRECATED) Enable/Disable the RabbitMQ mandatory flag for direct send. The direct send is used as reply, so the MessageUndeliverable exception is raised in case the client queue does not exist. MessageUndeliverable exception will be used to loop for a timeout to let a chance to sender to recover. This flag is deprecated and it will not be possible to deactivate this functionality anymore
enable_cancel_on_failover = False	boolean value	Enable x-cancel-on-ha-failover flag so that rabbitmq server will cancel and notify consumers when queue is down

Configuration option = Default value	Type	Description
heartbeat_in_pthread = True	boolean value	Run the health check heartbeat thread through a native python thread by default. If this option is equal to False then the health check heartbeat will inherit the execution model from the parent process. For example if the parent process has monkey patched the stdlib by using eventlet/greenlet then the heartbeat will be run through a green thread.
heartbeat_rate = 2	integer value	How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = 60	integer value	Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables heartbeat).
kombu_compression = None	string value	EXPERIMENTAL: Possible values are: gzip, bz2. If not set compression will not be used. This option may not be available in future versions.
kombu_failover_strategy = round-robin	string value	Determines how the next RabbitMQ node is chosen in case the one we are currently connected to becomes unavailable. Takes effect only if more than one RabbitMQ node is provided in config.
kombu_missing_consumer_retry_timeout = 60	integer value	How long to wait a missing client before abandoning to send it its replies. This value should not be longer than rpc_response_timeout.
kombu_reconnect_delay = 1.0	floating point value	How long to wait before reconnecting in response to an AMQP consumer cancel notification.
rabbit_ha_queues = False	boolean value	Try to use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database. In RabbitMQ 3.0, queue mirroring is no longer controlled by the x-ha-policy argument when declaring a queue. If you just want to make sure that all queues (except those with auto-generated names) are mirrored across all nodes, run: <code>"rabbitmqctl set_policy HA ^{?!amq\}. * {\"ha-mode\": \"all\"}"</code>
rabbit_interval_max = 30	integer value	Maximum interval of RabbitMQ connection retries. Default is 30 seconds.
rabbit_login_method = AMQPLAIN	string value	The RabbitMQ login method.

Configuration option = Default value	Type	Description
rabbit_qos_prefetch_count = 0	integer value	Specifies the number of messages to prefetch. Setting to zero allows unlimited messages.
rabbit_retry_backoff = 2	integer value	How long to backoff for between retries when connecting to RabbitMQ.
rabbit_retry_interval = 1	integer value	How frequently to retry connecting with RabbitMQ.
rabbit_transient_queues_ttl = 1800	integer value	Positive integer representing duration in seconds for queue TTL (x-expires). Queues which are unused for the duration of the TTL are automatically deleted. The parameter affects only reply and fanout queues.
ssl = False	boolean value	Connect over SSL.
<code>`ssl_ca_file = `</code>	string value	SSL certification authority file (valid only if SSL enabled).
<code>`ssl_cert_file = `</code>	string value	SSL cert file (valid only if SSL enabled).
<code>`ssl_key_file = `</code>	string value	SSL key file (valid only if SSL enabled).
<code>`ssl_version = `</code>	string value	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.

8.7.14. oslo_middleware

The following table outlines the options available under the **[oslo_middleware]** group in the `/etc/neutron/neutron.conf` file.

Table 8.36. oslo_middleware

Configuration option = Default value	Type	Description
enable_proxy_headers_parsing = False	boolean value	Whether the application is behind a proxy or not. This determines if the middleware should parse the headers or not.

8.7.15. oslo_policy

The following table outlines the options available under the **[oslo_policy]** group in the `/etc/neutron/neutron.conf` file.

Table 8.37. oslo_policy

Configuration option = Default value	Type	Description
enforce_new_defaults = False	boolean value	This option controls whether or not to use old deprecated defaults when evaluating policies. If True , the old deprecated defaults are not going to be evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be disallowed. It is encouraged to enable this flag along with the enforce_scope flag so that you can get the benefits of new defaults and scope_type together
enforce_scope = False	boolean value	This option controls whether or not to enforce scope when evaluating policies. If True , the scope of the token used in the request is compared to the scope_types of the policy being enforced. If the scopes do not match, an InvalidScope exception will be raised. If False , a message will be logged informing operators that policies are being invoked with mismatching scope.
policy_default_rule = default	string value	Default rule. Enforced when a requested rule is not found.
policy_dirs = ['policy.d']	multi valued	Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the <code>config_dir</code> option, or absolute paths. The file defined by <code>policy_file</code> must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = policy.yaml	string value	The relative or absolute path of a file that maps roles to permissions for a given service. Relative paths must be specified in relation to the configuration file setting this option.
remote_content_type = application/x-www-form-urlencoded	string value	Content Type to send and receive data for REST based policy check
remote_ssl_ca_cert_file = None	string value	Absolute path to ca cert file for REST based policy check
remote_ssl_client_cert_file = None	string value	Absolute path to client cert for REST based policy check
remote_ssl_client_key_file = None	string value	Absolute path client key file REST based policy check

Configuration option = Default value	Type	Description
remote_ssl_verify_server _cert = False	boolean value	server identity verification for REST based policy check

8.7.16. oslo_reports

The following table outlines the options available under the **[oslo_reports]** group in the **/etc/neutron/neutron.conf** file.

Table 8.38. oslo_reports

Configuration option = Default value	Type	Description
file_event_handler = None	string value	The path to a file to watch for changes to trigger the reports, instead of signals. Setting this option disables the signal trigger for the reports. If application is running as a WSGI application it is recommended to use this instead of signals.
file_event_handler_interval = 1	integer value	How many seconds to wait between polls when file_event_handler is set
log_dir = None	string value	Path to a log directory where to create a file

8.7.17. privsep

The following table outlines the options available under the **[privsep]** group in the **/etc/neutron/neutron.conf** file.

Table 8.39. privsep

Configuration option = Default value	Type	Description
capabilities = []	list value	List of Linux capabilities retained by the privsep daemon.
group = None	string value	Group that the privsep daemon should run as.
helper_command = None	string value	Command to invoke to start the privsep daemon if not using the "fork" method. If not specified, a default is generated using "sudo privsep-helper" and arguments designed to recreate the current configuration. This command must accept suitable --privsep_context and --privsep_sock_path arguments.

Configuration option = Default value	Type	Description
thread_pool_size = <based on operating system>	integer value	The number of threads available for privsep to concurrently run processes. Defaults to the number of CPU cores in the system.
user = None	string value	User that the privsep daemon should run as.

8.7.18. quotas

The following table outlines the options available under the **[quotas]** group in the **/etc/neutron/neutron.conf** file.

Table 8.40. quotas

Configuration option = Default value	Type	Description
default_quota = -1	integer value	Default number of resource allowed per tenant. A negative value means unlimited.
quota_driver = neutron.db.quota.driver_ noLOCK.DbQuotaNoLockD river	string value	Default driver to use for quota checks.
quota_floatingip = 50	integer value	Number of floating IPs allowed per tenant. A negative value means unlimited.
quota_network = 100	integer value	Number of networks allowed per tenant. A negative value means unlimited.
quota_port = 500	integer value	Number of ports allowed per tenant. A negative value means unlimited.
quota_router = 10	integer value	Number of routers allowed per tenant. A negative value means unlimited.
quota_security_group = 10	integer value	Number of security groups allowed per tenant. A negative value means unlimited.
quota_security_group_rul e = 100	integer value	Number of security rules allowed per tenant. A negative value means unlimited.
quota_subnet = 100	integer value	Number of subnets allowed per tenant, A negative value means unlimited.

Configuration option = Default value	Type	Description
track_quota_usage = True	boolean value	Keep in track in the database of current resource quota usage. Plugins which do not leverage the neutron database should set this flag to False.

8.7.19. ssl

The following table outlines the options available under the **[ssl]** group in the **/etc/neutron/neutron.conf** file.

Table 8.41. ssl

Configuration option = Default value	Type	Description
ca_file = None	string value	CA certificate file to use to verify connecting clients.
cert_file = None	string value	Certificate file to use when starting the server securely.
ciphers = None	string value	Sets the list of available ciphers. value should be a string in the OpenSSL cipher list format.
key_file = None	string value	Private key file to use when starting the server securely.
version = None	string value	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.

8.8. OPENVSWITCH_AGENT.INI

This section contains options for the **/etc/neutron/plugins/ml2/openvswitch_agent.ini** file.

8.8.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the **/etc/neutron/plugins/ml2/openvswitch_agent.ini** file.

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
<code>`instance_format = [instance: %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.
<code>`instance_uuid_format = [instance: %(uuid)s] `</code>	string value	The format for an instance UUID that is passed with the log message.
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).

Configuration option = Default value	Type	Description
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code> . This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code> . This option is ignored if <code>log_config_append</code> is set.
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless <code>log_rotation_type</code> is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	string value	Format string to use for log messages with context. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is <code>DEBUG</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_default_format_s tring = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s	string value	Format string to use for log messages when context is undefined. Used by <code>oslo_log.formatters.ContextFormatter</code>

Configuration option = Default value	Type	Description
logging_exception_prefix = <code>%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s</code>	string value	Prefix each line of exception output with this format. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_user_identity_for mat = <code>%(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s</code>	string value	Defines the format string for <code>%(user_identity)s</code> that is used in <code>logging_context_format_string</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
publish_errors = False	boolean value	Enables or disables publication of error events.
rate_limit_burst = 0	integer value	Maximum number of logged messages per <code>rate_limit_interval</code> .
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to <code>rate_limit_except_level</code> are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
rpc_response_max_timeout = 600	integer value	Maximum seconds to wait for a response from an RPC call.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if <code>log_config_append</code> is set.
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if <code>log_config_append</code> is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if <code>log_config_append</code> is set.

Configuration option = Default value	Type	Description
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

8.8.2. agent

The following table outlines the options available under the **[agent]** group in the **/etc/neutron/plugins/ml2/openvswitch_agent.ini** file.

Table 8.42. agent

Configuration option = Default value	Type	Description
arp_responder = False	boolean value	Enable local ARP responder if it is supported. Requires OVS 2.1 and ML2 I2population driver. Allows the switch (when supporting an overlay) to respond to an ARP request locally without performing a costly ARP broadcast into the overlay. NOTE: If enable_distributed_routing is set to True then arp_responder will automatically be set to True in the agent, regardless of the setting in the config file.
baremetal_smartnic = False	boolean value	Enable the agent to process Smart NIC ports.
dont_fragment = True	boolean value	Set or un-set the don't fragment (DF) bit on outgoing IP packet carrying GRE/VXLAN tunnel.
drop_flows_on_start = False	boolean value	Reset flow table on start. Setting this to True will cause brief traffic interruption.
enable_distributed_routing = False	boolean value	Make the I2 agent run in DVR mode.

Configuration option = Default value	Type	Description
explicitly_egress_direct = False	boolean value	When set to True, the accepted egress unicast traffic will not use action NORMAL. The accepted egress packets will be taken care of in the final egress tables direct output flows for unicast traffic.
extensions = []	list value	Extensions list to use
l2_population = False	boolean value	Use ML2 l2population mechanism driver to learn remote MAC and IPs and improve tunnel scalability.
minimize_polling = True	boolean value	Minimize polling by monitoring ovsdb for interface changes.
ovsdb_monitor_respawn_interval = 30	integer value	The number of seconds to wait before respawning the ovsdb monitor after losing communication with it.
tunnel_csum = False	boolean value	Set or un-set the tunnel header checksum on outgoing IP packet carrying GRE/VXLAN tunnel.
tunnel_types = []	list value	Network types supported by the agent (gre, vxlan and/or geneve).
veth_mtu = 9000	integer value	MTU size of veth interfaces
vxlan_udp_port = 4789	port value	The UDP port to use for VXLAN tunnels.

8.8.3. network_log

The following table outlines the options available under the **[network_log]** group in the `/etc/neutron/plugins/ml2/openvswitch_agent.ini` file.

Table 8.43. network_log

Configuration option = Default value	Type	Description
burst_limit = 25	integer value	Maximum number of packets per rate_limit.
local_output_log_base = None	string value	Output logfile path on agent side, default syslog file.
rate_limit = 100	integer value	Maximum packets logging per second.

8.8.4. ovs

The following table outlines the options available under the **[ovs]** group in the **/etc/neutron/plugins/ml2/openvswitch_agent.ini** file.

Table 8.44. ovs

Configuration option = Default value	Type	Description
bridge_mappings = []	list value	Comma-separated list of <physical_network>: <bridge> tuples mapping physical network names to the agent's node-specific Open vSwitch bridge names to be used for flat and VLAN networks. The length of bridge names should be no more than 11. Each bridge must exist, and should have a physical network interface configured as a port. All physical networks configured on the server should have mappings to appropriate bridges on each agent. Note: If you remove a bridge from this mapping, make sure to disconnect it from the integration bridge as it won't be managed by the agent anymore.
datapath_type = system	string value	OVS datapath to use. <i>system</i> is the default value and corresponds to the kernel datapath. To enable the userspace datapath set this value to <i>netdev</i> .
disable_packet_marking = False	boolean value	Disables the packet marking when the QoS extension is enabled. This option needs to be enabled when using OVS with hardware offload until the <i>skb_priority</i> , <i>skb_mark</i> and output queue fields are supported and can be offloaded. If this options is enabled, no rate QoS rule (bandwidth limit nor minimum bandwidth) will work for VirtIO ports.
int_peer_patch_port = patch-tun	string value	Peer patch port in integration bridge for tunnel bridge.
integration_bridge = br-int	string value	Integration bridge to use. Do not change this parameter unless you have a good reason to. This is the name of the OVS integration bridge. There is one per hypervisor. The integration bridge acts as a virtual <i>patch bay</i> . All VM VIFs are attached to this bridge and then <i>patched</i> according to their network connectivity.
local_ip = None	IP address value	IP address of local overlay (tunnel) network endpoint. Use either an IPv4 or IPv6 address that resides on one of the host network interfaces. The IP version of this value must match the value of the <i>overlay_ip_version</i> option in the ML2 plug-in configuration file on the neutron server node(s).
of_connect_timeout = 300	integer value	Timeout in seconds to wait for the local switch connecting the controller.

Configuration option = Default value	Type	Description
of_inactivity_probe = 10	integer value	The inactivity_probe interval in seconds for the local switch connection to the controller. A value of 0 disables inactivity probes.
of_listen_address = 127.0.0.1	IP address value	Address to listen on for OpenFlow connections.
of_listen_port = 6633	port value	Port to listen on for OpenFlow connections.
of_request_timeout = 300	integer value	Timeout in seconds to wait for a single OpenFlow request.
ovsdb_connection = tcp:127.0.0.1:6640	string value	The connection string for the OVSDB backend. Will be used for all ovsdb commands and by ovsdb-client when monitoring
ovsdb_debug = False	boolean value	Enable OVSDB debug logs
resource_provider_bandwidths = []	list value	Comma-separated list of <bridge>:<egress_bw>:<ingress_bw> tuples, showing the available bandwidth for the given bridge in the given direction. The direction is meant from VM perspective. Bandwidth is measured in kilobits per second (kbps). The bridge must appear in bridge_mappings as the value. But not all bridges in bridge_mappings must be listed here. For a bridge not listed here we neither create a resource provider in placement nor report inventories against. An omitted direction means we do not report an inventory for the corresponding class.
resource_provider_default_hypervisor = None	string value	The default hypervisor name used to locate the parent of the resource provider. If this option is not set, canonical name is used
resource_provider_hypervisors = {}	dict value	Mapping of bridges to hypervisors: <bridge>:<hypervisor>,... hypervisor name is used to locate the parent of the resource provider tree. Only needs to be set in the rare case when the hypervisor name is different from the resource_provider_default_hypervisor config option value as known by the nova-compute managing that hypervisor.

Configuration option = Default value	Type	Description
resource_provider_inventory_defaults = {'allocation_ratio': 1.0, 'min_unit': 1, 'reserved': 0, 'step_size': 1}	dict value	Key:value pairs to specify defaults used while reporting resource provider inventories. Possible keys with their types: allocation_ratio:float, max_unit:int, min_unit:int, reserved:int, step_size:int, See also: https://docs.openstack.org/api-ref/placement/#update-resource-provider-inventories
ssl_ca_cert_file = None	string value	The Certificate Authority (CA) certificate to use when interacting with OVSDB. Required when using an "ssl:" prefixed ovsdb_connection
ssl_cert_file = None	string value	The SSL certificate file to use when interacting with OVSDB. Required when using an "ssl:" prefixed ovsdb_connection
ssl_key_file = None	string value	The SSL private key file to use when interacting with OVSDB. Required when using an "ssl:" prefixed ovsdb_connection
tun_peer_patch_port = patch-int	string value	Peer patch port in tunnel bridge for integration bridge.
tunnel_bridge = br-tun	string value	Tunnel bridge to use.
vhostuser_socket_dir = /var/run/openvswitch	string value	OVS vhost-user socket directory.

8.8.5. securitygroup

The following table outlines the options available under the **[securitygroup]** group in the `/etc/neutron/plugins/ml2/openvswitch_agent.ini` file.

Table 8.45. securitygroup

Configuration option = Default value	Type	Description
enable_ipset = True	boolean value	Use ipset to speed-up the iptables based security groups. Enabling ipset support requires that ipset is installed on L2 agent node.
enable_security_group = True	boolean value	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.

Configuration option = Default value	Type	Description
firewall_driver = None	string value	Driver for security groups firewall in the L2 agent
permitted_ethertypes = []	list value	Comma-separated list of ethertypes to be permitted, in hexadecimal (starting with "0x"). For example, "0x4008" to permit InfiniBand.

8.9. SRIOV_AGENT.INI

This section contains options for the `/etc/neutron/plugins/ml2/sriov_agent.ini` file.

8.9.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/neutron/plugins/ml2/sriov_agent.ini` file.

Configuration option = Default value	Type	Description
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.

Configuration option = Default value	Type	Description
default_log_levels = <code>['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' ', 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']</code>	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
<code>`instance_format = [instance:  %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.
<code>`instance_uuid_format =  [instance: %(uuid)s] `</code>	string value	The format for an instance UUID that is passed with the log message.
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).

Configuration option = Default value	Type	Description
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for <code>%(asctime)s</code> in log records. Default: <code>%(default)s</code> . This option is ignored if <code>log_config_append</code> is set.
log-dir = None	string value	(Optional) The base directory used for relative <code>log_file</code> paths. This option is ignored if <code>log_config_append</code> is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to <code>stderr</code> as defined by <code>use_stderr</code> . This option is ignored if <code>log_config_append</code> is set.
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless <code>log_rotation_type</code> is set to "interval".
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	string value	Format string to use for log messages with context. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_debug_format_suffix = %(funcName)s %(pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is <code>DEBUG</code> . Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_default_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s	string value	Format string to use for log messages when context is undefined. Used by <code>oslo_log.formatters.ContextFormatter</code>
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s	string value	Prefix each line of exception output with this format. Used by <code>oslo_log.formatters.ContextFormatter</code>

Configuration option = Default value	Type	Description
logging_user_identity_format = %(user)s %(tenant)s %(domain)s %(user_domain)s %(project_domain)s	string value	Defines the format string for %(user_identity)s that is used in logging_context_format_string. Used by oslo_log.formatters.ContextFormatter
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
publish_errors = False	boolean value	Enables or disables publication of error events.
rate_limit_burst = 0	integer value	Maximum number of logged messages per rate_limit_interval.
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
rpc_response_max_timeout = 600	integer value	Maximum seconds to wait for a response from an RPC call.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.

8.9.2. agent

The following table outlines the options available under the **[agent]** group in the **/etc/neutron/plugins/ml2/sriov_agent.ini** file.

Table 8.46. agent

Configuration option = Default value	Type	Description
extensions = []	list value	Extensions list to use

8.9.3. sriov_nic

The following table outlines the options available under the **[sriov_nic]** group in the **/etc/neutron/plugins/ml2/sriov_agent.ini** file.

Table 8.47. sriov_nic

Configuration option = Default value	Type	Description
exclude_devices = []	list value	Comma-separated list of <network_device>: <vfs_to_exclude> tuples, mapping network_device to the agent's node-specific list of virtual functions that should not be used for virtual networking. vfs_to_exclude is a semicolon-separated list of virtual functions to exclude from network_device. The network_device in the mapping should appear in the physical_device_mappings list.

Configuration option = Default value	Type	Description
physical_device_mappings = []	list value	Comma-separated list of <physical_network>: <network_device> tuples mapping physical network names to the agent's node-specific physical network device interfaces 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.
resource_provider_bandwidths = []	list value	Comma-separated list of <network_device>: <egress_bw>:<ingress_bw> tuples, showing the available bandwidth for the given device in the given direction. The direction is meant from VM perspective. Bandwidth is measured in kilobits per second (kbps). The device must appear in physical_device_mappings as the value. But not all devices in physical_device_mappings must be listed here. For a device not listed here we neither create a resource provider in placement nor report inventories against. An omitted direction means we do not report an inventory for the corresponding class.
resource_provider_default_hypervisor = None	string value	The default hypervisor name used to locate the parent of the resource provider. If this option is not set, canonical name is used
resource_provider_hypervisors = {}	dict value	Mapping of network devices to hypervisors: <network_device>:<hypervisor>,... hypervisor name is used to locate the parent of the resource provider tree. Only needs to be set in the rare case when the hypervisor name is different from the resource_provider_default_hypervisor config option value as known by the nova-compute managing that hypervisor.
resource_provider_inventory_defaults = {'allocation_ratio': 1.0, 'min_unit': 1, 'reserved': 0, 'step_size': 1}	dict value	Key:value pairs to specify defaults used while reporting resource provider inventories. Possible keys with their types: allocation_ratio:float, max_unit:int, min_unit:int, reserved:int, step_size:int, See also: https://docs.openstack.org/api-ref/placement/#update-resource-provider-inventories

CHAPTER 9. NOVA

The following chapter contains information about the configuration options in the **nova** service.

9.1. NOVA.CONF

This section contains options for the `/etc/nova/nova.conf` file.

9.1.1. DEFAULT

The following table outlines the options available under the **[DEFAULT]** group in the `/etc/nova/nova.conf` file.

Configuration option = Default value	Type	Description
allow_resize_to_same_host = False	boolean value	Allow destination machine to match source for resize. Useful when testing in single-host environments. By default it is not allowed to resize to the same host. Setting this option to true will add the same host to the destination options. Also set to true if you allow the ServerGroupAffinityFilter and need to resize.
arq_binding_timeout = 300	integer value	Timeout for Accelerator Request (ARQ) bind event message arrival. Number of seconds to wait for ARQ bind resolution event to arrive. The event indicates that every ARQ for an instance has either bound successfully or failed to bind. If it does not arrive, instance bringup is aborted with an exception.
backdoor_port = None	string value	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.
backdoor_socket = None	string value	Enable eventlet backdoor, using the provided path as a unix socket that can receive connections. This option is mutually exclusive with <code>backdoor_port</code> in that only one should be provided. If both are provided then the existence of this option overrides the usage of that option. Inside the path {pid} will be replaced with the PID of the current process.

Configuration option = Default value	Type	Description
block_device_allocate_retries = 60	integer value	The number of times to check for a volume to be "available" before attaching it during server create. When creating a server with block device mappings where source_type is one of blank, image or snapshot and the destination_type is volume, the nova-compute service will create a volume and then attach it to the server. Before the volume can be attached, it must be in status "available". This option controls how many times to check for the created volume to be "available" before it is attached. If the operation times out, the volume will be deleted if the block device mapping delete_on_termination value is True. It is recommended to configure the image cache in the block storage service to speed up this operation. See https://docs.openstack.org/cinder/latest/admin/blockstorage-image-volume-cache.html for details. Possible values: ● 60 (default) ● If value is 0, then one attempt is made. ● For any value > 0, total attempts are (value + 1) Related options: ● block_device_allocate_retries_interval - controls the interval between checks

Configuration option = Default value	Type	Description
block_device_allocate_retries_interval = 3	integer value	Interval (in seconds) between block device allocation retries on failures. This option allows the user to specify the time interval between consecutive retries. The block_device_allocate_retries option specifies the maximum number of retries. Possible values: ● 0: Disables the option. ● Any positive integer in seconds enables the option. Related options: ● block_device_allocate_retries - controls the number of retries
cert = self.pem	string value	Path to SSL certificate file. Related options: ● key ● ssl_only ● [console] ssl_ciphers ● [console] ssl_minimum_version
compute_driver = None	string value	Defines which driver to use for controlling virtualization. Possible values: ● libvirt.LibvirtDriver ● fake.FakeDriver ● ironic.IronicDriver ● vmwareapi.VMwareVCDriver ● hyperv.HyperVDriver ● powervm.PowerVMDriver ● zvm.ZVMDriver

Configuration option = Default value	Type	Description
compute_monitors = []	list value	A comma-separated list of monitors that can be used for getting compute metrics. You can use the alias/name from the setuptools entry points for nova.compute.monitors.* namespaces. If no namespace is supplied, the "cpu." namespace is assumed for backwards-compatibility. NOTE Only one monitor per namespace (For example: cpu) can be loaded at a time. Possible values: • An empty list will disable the feature (Default). • An example value that would enable the CPU bandwidth monitor that uses the virt driver variant <pre>compute_monitors = cpu.virt_driver</pre>

Configuration option = Default value	Type	Description
config_drive_format = iso9660	string value	Config drive format. Config drive format that will contain metadata attached to the instance when it boots. Related options: • This option is meaningful when one of the following alternatives occur: 1. force_config_drive option set to true 2. the REST API call to create the instance contains an enable flag for config drive option 3. the image used to create the instance requires a config drive, this is defined by img_config_drive property for that image. • A compute node running Hyper-V hypervisor can be configured to attach config drive as a CD drive. To attach the config drive as a CD drive, set the [hyperv] config_drive_cdrom option to true. Deprecated since: 19.0.0 Reason: This option was originally added as a workaround for bug in libvirt, #1246201, that was resolved in libvirt v1.2.17. As a result, this option is no longer necessary or useful.
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
console_host = <based on operating system>	string value	Console proxy host to be used to connect to instances on this host. It is the publicly visible name for the console host. Possible values: • Current hostname (default) or any string representing hostname.
control_exchange = nova	string value	The default exchange under which topics are scoped. May be overridden by an exchange name specified in the transport_url option.

Configuration option = Default value	Type	Description
cpu_allocation_ratio = None	floating point value	Virtual CPU to physical CPU allocation ratio. This option is used to influence the hosts selected by the Placement API by configuring the allocation ratio for VCPU inventory. In addition, the AggregateCoreFilter (deprecated) will fall back to this configuration value if no per-aggregate setting is found. a. note:: This option does not affect `PCPU` inventory, which cannot be overcommitted. b. note:: If this option is set to something *other than* `None` or `0.0`, the allocation ratio will be overwritten by the value of this option, otherwise, the allocation ratio will not change. Once set to a non-default value, it is not possible to "unset" the config to get back to the default behavior. If you want to reset back to the initial value, explicitly specify it to the value of `initial_cpu_allocation_ratio`. Possible values: - Any valid positive integer or float value Related options: initial_cpu_allocation_ratio
daemon = False	boolean value	Run as a background process.
debug = False	boolean value	If set to true, the logging level will be set to DEBUG instead of the default INFO level.

Configuration option = Default value	Type	Description
default_access_ip_network_name = None	string value	Name of the network to be used to set access IPs for instances. If there are multiple IPs to choose from, an arbitrary one will be chosen. Possible values: • None (default) • Any string representing network name.
default_availability_zone = nova	string value	Default availability zone for compute services. This option determines the default availability zone for <i>nova-compute</i> services, which will be used if the service(s) do not belong to aggregates with availability zone metadata. Possible values: • Any string representing an existing availability zone name.
default_ephemeral_format = None	string value	The default format an ephemeral_volume will be formatted with on creation. Possible values: • ext2 • ext3 • ext4 • xfs • ntfs (only for Windows guests)

Configuration option = Default value	Type	Description
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O', 'glanceclient=WARN', 'oslo.privsep.daemon=IN FO']	list value	List of package logging levels in logger=LEVEL pairs. This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
default_schedule_zone = None	string value	Default availability zone for instances. This option determines the default availability zone for instances, which will be used when a user does not specify one when creating an instance. The instance(s) will be bound to this availability zone for their lifetime. Possible values: • Any string representing an existing availability zone name. • None, which means that the instance can move from one availability zone to another during its lifetime if it is moved from one compute node to another. Related options: • [cinder]/cross_az_attach

Configuration option = Default value	Type	Description
disk_allocation_ratio = None	floating point value	Virtual disk to physical disk allocation ratio. This option is used to influence the hosts selected by the Placement API by configuring the allocation ratio for DISK_GB inventory. In addition, the AggregateDiskFilter (deprecated) will fall back to this configuration value if no per-aggregate setting is found. When configured, a ratio greater than 1.0 will result in over-subscription of the available physical disk, which can be useful for more efficiently packing instances created with images that do not use the entire virtual disk, such as sparse or compressed images. It can be set to a value between 0.0 and 1.0 in order to preserve a percentage of the disk for uses other than instances. a. note:: If the value is set to <code>>1`</code>, we recommend keeping track of the free disk space, as the value approaching <code>`0`</code> may result in the incorrect functioning of instances using it at the moment.</p> </div> <p>b. note::</p> <div> <p>If this option is set to something <i>*other than*</i> <code>`None`</code> or <code>`0.0`</code>, the allocation ratio will be overwritten by the value of this option, otherwise, the allocation ratio will not change. Once set to a non-default value, it is not possible to "unset" the config to get back to the default behavior. If you want to reset back to the initial value, explicitly specify it to the value of <code>`initial_disk_allocation_ratio`</code>. Possible values: - Any valid positive integer or float value Related options: initial_disk_allocation_ratio

Configuration option = Default value	Type	Description
enable_new_services = True	boolean value	Enable new nova-compute services on this host automatically. When a new nova-compute service starts up, it gets registered in the database as an enabled service. Sometimes it can be useful to register new compute services in disabled state and then enabled them at a later point in time. This option only sets this behavior for nova-compute services, it does not auto-disable other services like nova-conductor, nova-scheduler, or nova-osapi_compute. Possible values: ● True: Each new compute service is enabled as soon as it registers itself. ● False: Compute services must be enabled via an os-services REST API call or with the CLI with nova service-enable <hostname> <binary>, otherwise they are not ready to use.
enabled_apis = ['osapi_compute', 'metadata']	list value	List of APIs to be enabled by default.
enabled_ssl_apis = []	list value	List of APIs with enabled SSL. Nova provides SSL support for the API servers. enabled_ssl_apis option allows configuring the SSL support.
executor_thread_pool_size = 64	integer value	Size of executor thread pool when executor is threading or eventlet.
fatal_deprecations = False	boolean value	Enables or disables fatal status of deprecations.
flat_injected = False	boolean value	This option determines whether the network setup information is injected into the VM before it is booted. While it was originally designed to be used only by nova-network, it is also used by the vmware virt driver to control whether network information is injected into a VM. The libvirt virt driver also uses it when we use config_drive to configure network to control whether network information is injected into a VM.

Configuration option = Default value	Type	Description
force_config_drive = False	boolean value	Force injection to take place on a config drive When this option is set to true config drive functionality will be forced enabled by default, otherwise users can still enable config drives via the REST API or image metadata properties. Launched instances are not affected by this option. Possible values: • True: Force to use of config drive regardless the user's input in the REST API call. • False: Do not force use of config drive. Config drives can still be enabled via the REST API or image metadata properties. Related options: • Use the <i>mkisofs_cmd</i> flag to set the path where you install the genisoimage program. If genisoimage is in same path as the nova-compute service, you do not need to set this flag. • To use a config drive with Hyper-V, you must set the <i>mkisofs_cmd</i> value to the full path to an mkisofs.exe installation. Additionally, you must set the <i>qemu_img_cmd</i> value in the hyperv configuration section to the full path to an qemu-img command installation.
force_raw_images = True	boolean value	Force conversion of backing images to raw format. Possible values: • True: Backing image files will be converted to raw image format • False: Backing image files will not be converted Related options: • compute_driver: Only the libvirt driver uses this option. • [libvirt]/images_type: If images_type is rbd, setting this option to False is not allowed. See the bug https://bugs.launchpad.net/nova/+bug/1816686 for more details.

Configuration option = Default value	Type	Description
graceful_shutdown_timeout = 60	integer value	Specify a timeout after which a gracefully shutdown server will exit. Zero value means endless wait.
heal_instance_info_cache_interval = 60	integer value	Interval between instance network information cache updates. Number of seconds after which each compute node runs the task of querying Neutron for all of its instances networking information, then updates the Nova db with that information. Nova will never update it's cache if this option is set to 0. If we don't update the cache, the metadata service and nova-api endpoints will be proxying incorrect network data about the instance. So, it is not recommended to set this option to 0. Possible values: - Any positive integer in seconds. - Any value ≤ 0 will disable the sync. This is not recommended.
host = <based on operating system>	string value	Hostname, FQDN or IP address of this host. Used as: - the oslo.messaging queue name for nova-compute worker - we use this value for the binding_host sent to neutron. This means if you use a neutron agent, it should have the same value for host. - cinder host attachment information Must be valid within AMQP key. Possible values: - String with hostname, FQDN or IP address. Default is hostname of this host.

Configuration option = Default value	Type	Description
initial_cpu_allocation_ratio o = 16.0	floating point value	Initial virtual CPU to physical CPU allocation ratio. This is only used when initially creating the computes_nodes table record for a given nova-compute service. See https://docs.openstack.org/nova/latest/admin/configuration/schedulers.html for more details and usage scenarios. Related options: • cpu_allocation_ratio
initial_disk_allocation_ratio io = 1.0	floating point value	Initial virtual disk to physical disk allocation ratio. This is only used when initially creating the computes_nodes table record for a given nova-compute service. See https://docs.openstack.org/nova/latest/admin/configuration/schedulers.html for more details and usage scenarios. Related options: • disk_allocation_ratio
initial_ram_allocation_ratio o = 1.5	floating point value	Initial virtual RAM to physical RAM allocation ratio. This is only used when initially creating the computes_nodes table record for a given nova-compute service. See https://docs.openstack.org/nova/latest/admin/configuration/schedulers.html for more details and usage scenarios. Related options: • ram_allocation_ratio

Configuration option = Default value	Type	Description
injected_network_template = \$pybasedir/nova/virt/interfaces.template	string value	Path to <code>/etc/network/interfaces</code> template. The path to a template file for the <code>/etc/network/interfaces</code>-style file, which will be populated by nova and subsequently used by cloudinit. This provides a method to configure network connectivity in environments without a DHCP server. The template will be rendered using Jinja2 template engine, and receive a top-level key called interfaces. This key will contain a list of dictionaries, one for each interface. Refer to the cloudinit documentaion for more information: https://cloudinit.readthedocs.io/en/latest/topics/datasources.html Possible values: • A path to a Jinja2-formatted template for a Debian <code>/etc/network/interfaces</code> file. This applies even if using a non Debian-derived guest. Related options: • flat_inject: This must be set to True to ensure nova embeds network configuration information in the metadata provided through the config drive.
instance_build_timeout = 0	integer value	Maximum time in seconds that an instance can take to build. If this timer expires, instance status will be changed to ERROR. Enabling this option will make sure an instance will not be stuck in BUILD state for a longer period. Possible values: • 0: Disables the option (default) • Any positive integer in seconds: Enables the option.

Configuration option = Default value	Type	Description
instance_delete_interval = 300	integer value	Interval for retrying failed instance file deletes. This option depends on <i>maximum_instance_delete_attempts</i>. This option specifies how often to retry deletes whereas <i>maximum_instance_delete_attempts</i> specifies the maximum number of retry attempts that can be made. Possible values: ● 0: Will run at the default periodic interval. ● Any value < 0: Disables the option. ● Any positive integer in seconds. Related options: ● maximum_instance_delete_attempts from instance_cleaning_opts group.
<code>`instance_format = [instance: %(uuid)s] `</code>	string value	The format for an instance that is passed with the log message.
instance_name_template = instance-%08x	string value	Template string to be used to generate instance names. This template controls the creation of the database name of an instance. This is not the display name you enter when creating an instance (via Horizon or CLI). For a new deployment it is advisable to change the default value (which uses the database autoincrement) to another value which makes use of the attributes of an instance, like instance-%(uuid)s. If you already have instances in your deployment when you change this, your deployment will break. Possible values: ● A string which either uses the instance database ID (like the default) ● A string with a list of named database columns, for example %(id)d or %(uuid)s or %(hostname)s.

Configuration option = Default value	Type	Description
instance_usage_audit = False	boolean value	This option enables periodic compute.instance.exists notifications. Each compute node must be configured to generate system usage data. These notifications are consumed by OpenStack Telemetry service.
instance_usage_audit_period = month	string value	Time period to generate instance usages for. It is possible to define optional offset to given period by appending @ character followed by a number defining offset. Possible values: - period, example: hour, day, month or year - period with offset, example: month@15 will result in monthly audits starting on 15th day of month.
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	string value	The format for an instance UUID that is passed with the log message.
instances_path = \$state_path/instances	string value	Specifies where instances are stored on the hypervisor's disk. It can point to locally attached storage or a directory on NFS. Possible values: <code>\$state_path/instances</code> where <code>state_path</code> is a config option that specifies the top-level directory for maintaining nova's state. (default) or Any string representing directory path. Related options: [workarounds]/ensure_libvirt_rbd_instance_dir_cleanup
internal_service_availability_zone = internal	string value	Availability zone for internal services. This option determines the availability zone for the various internal nova services, such as <i>nova-scheduler</i>, <i>nova-conductor</i>, etc. Possible values: Any string representing an existing availability zone name.

Configuration option = Default value	Type	Description
key = None	string value	SSL key file (if separate from cert). Related options: cert
live_migration_retry_count = 30	integer value	Maximum number of 1 second retries in live_migration. It specifies number of retries to iptables when it complains. It happens when an user continuously sends live-migration request to same host leading to concurrent request to iptables. Possible values: Any positive integer representing retry count.
log-config-append = None	string value	The name of a logging configuration file. This file is appended to any existing logging configuration files. For details about logging configuration files, see the Python logging module documentation. Note that when logging configuration files are used then all logging configuration is set in the configuration file and other logging configuration options are ignored (for example, log-date-format).
log-date-format = %Y-%m-%d %H:%M:%S	string value	Defines the format string for %(asctime)s in log records. Default: %(default)s . This option is ignored if log_config_append is set.
log-dir = None	string value	(Optional) The base directory used for relative log_file paths. This option is ignored if log_config_append is set.
log-file = None	string value	(Optional) Name of log file to send logging output to. If no default is set, logging will go to stderr as defined by use_stderr. This option is ignored if log_config_append is set.
log_options = True	boolean value	Enables or disables logging values of all registered options when starting a service (at DEBUG level).
log_rotate_interval = 1	integer value	The amount of time before the log files are rotated. This option is ignored unless log_rotation_type is set to "interval".

Configuration option = Default value	Type	Description
log_rotate_interval_type = days	string value	Rotation interval type. The time of the last file change (or the time when the service was started) is used when scheduling the next rotation.
log_rotation_type = none	string value	Log rotation type.
logging_context_format_string = %(asctime)s.% (msecs)03d %(process)d %(levelname)s %(name)s [% (request_id)s % (user_identity)s] % (instance)s%(message)s	string value	Format string to use for log messages with context. Used by oslo_log.formatters.ContextFormatter
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	string value	Additional data to append to log message when logging level for the message is DEBUG. Used by oslo_log.formatters.ContextFormatter
logging_default_format_s tring = %(asctime)s.% (msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s% (message)s	string value	Format string to use for log messages when context is undefined. Used by oslo_log.formatters.ContextFormatter
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	string value	Prefix each line of exception output with this format. Used by oslo_log.formatters.ContextFormatter
logging_user_identity_for mat = %(user)s % (tenant)s %(domain)s % (user_domain)s % (project_domain)s	string value	Defines the format string for %(user_identity)s that is used in logging_context_format_string. Used by oslo_log.formatters.ContextFormatter

Configuration option = Default value	Type	Description
long_rpc_timeout = 1800	integer value	This option allows setting an alternate timeout value for RPC calls that have the potential to take a long time. If set, RPC calls to other services will use this value for the timeout (in seconds) instead of the global <code>rpc_response_timeout</code> value. Operations with RPC calls that utilize this value: ● live migration ● scheduling ● enabling/disabling a compute service ● image pre-caching ● snapshot-based / cross-cell resize ● resize / cold migration ● volume attach Related options: ● <code>rpc_response_timeout</code>
max_concurrent_builds = 10	integer value	Limits the maximum number of instance builds to run concurrently by nova-compute. Compute service can attempt to build an infinite number of instances, if asked to do so. This limit is enforced to avoid building unlimited instance concurrently on a compute node. This value can be set per compute node. Possible Values: ● 0 : treated as unlimited. ● Any positive integer representing maximum concurrent builds.

Configuration option = Default value	Type	Description
max_concurrent_live_migrations = 1	integer value	Maximum number of live migrations to run concurrently. This limit is enforced to avoid outbound live migrations overwhelming the host/network and causing failures. It is not recommended that you change this unless you are very sure that doing so is safe and stable in your environment. Possible values: ● 0 : treated as unlimited. ● Any positive integer representing maximum number of live migrations to run concurrently.
max_concurrent_snapshots = 5	integer value	Maximum number of instance snapshot operations to run concurrently. This limit is enforced to prevent snapshots overwhelming the host/network/storage and causing failure. This value can be set per compute node. Possible Values: ● 0 : treated as unlimited. ● Any positive integer representing maximum concurrent snapshots.

Configuration option = Default value	Type	Description
max_local_block_devices = 3	integer value	Maximum number of devices that will result in a local image being created on the hypervisor node. A negative number means unlimited. Setting max_local_block_devices to 0 means that any request that attempts to create a local disk will fail. This option is meant to limit the number of local discs (so root local disc that is the result of imageRef being used when creating a server, and any other ephemeral and swap disks). 0 does not mean that images will be automatically converted to volumes and boot instances from volumes - it just means that all requests that attempt to create a local disk will fail. Possible values: ● 0: Creating a local disk is not allowed. ● Negative number: Allows unlimited number of local discs. ● Positive number: Allows only these many number of local discs.
max_logfile_count = 30	integer value	Maximum number of rotated log files.
max_logfile_size_mb = 200	integer value	Log file maximum size in MB. This option is ignored if "log_rotation_type" is not set to "size".
maximum_instance_delete_attempts = 5	integer value	The number of times to attempt to reap an instance's files. This option specifies the maximum number of retry attempts that can be made. Possible values: ● Any positive integer defines how many attempts are made. Related options: ● [DEFAULT] instance_delete_interval can be used to disable this option.
metadata_listen = 0.0.0.0	string value	IP address on which the metadata API will listen. The metadata API service listens on this IP address for incoming requests.

Configuration option = Default value	Type	Description
metadata_listen_port = 8775	port value	Port on which the metadata API will listen. The metadata API service listens on this port number for incoming requests.
metadata_workers = <based on operating system>	integer value	Number of workers for metadata service. If not specified the number of available CPUs will be used. The metadata service can be configured to run as multi-process (workers). This overcomes the problem of reduction in throughput when API request concurrency increases. The metadata service will run in the specified number of processes. Possible Values: - Any positive integer - None (default value)
migrate_max_retries = -1	integer value	Number of times to retry live-migration before failing. Possible values: - If == -1, try until out of hosts (default) - If == 0, only try once, no retries - Integer greater than 0

Configuration option = Default value	Type	Description
mkisofs_cmd = genisoimage	string value	Name or path of the tool used for ISO image creation. Use the mkisofs_cmd flag to set the path where you install the genisoimage program. If genisoimage is on the system path, you do not need to change the default value. To use a config drive with Hyper-V, you must set the mkisofs_cmd value to the full path to an mkisofs.exe installation. Additionally, you must set the qemu_img_cmd value in the hyperv configuration section to the full path to an qemu-img command installation. Possible values: ● Name of the ISO image creator program, in case it is in the same directory as the nova-compute service ● Path to ISO image creator program Related options: ● This option is meaningful when config drives are enabled. ● To use config drive with Hyper-V, you must set the qemu_img_cmd value in the hyperv configuration section to the full path to an qemu-img command installation.
my_block_storage_ip = \$my_ip	string value	The IP address which is used to connect to the block storage network. Possible values: ● String with valid IP address. Default is IP address of this host. Related options: ● my_ip - if my_block_storage_ip is not set, then my_ip value is used.

Configuration option = Default value	Type	Description
my_ip = <based on operating system>	string value	The IP address which the host is using to connect to the management network. Possible values: ● String with valid IP address. Default is IPv4 address of this host. Related options: ● my_block_storage_ip
network_allocate_retries = 0	integer value	Number of times to retry network allocation. It is required to attempt network allocation retries if the virtual interface plug fails. Possible values: ● Any positive integer representing retry count.

Configuration option = Default value	Type	Description
non_inheritable_image_properties = ['cache_in_nova', 'bittorrent']	list value	Image properties that should not be inherited from the instance when taking a snapshot. This option gives an opportunity to select which image-properties should not be inherited by newly created snapshots. a. note:: The following image properties are <i>*never*</i> inherited regardless of whether they are listed in this configuration option or not: • cinder_encryption_key_id • cinder_encryption_key_deletion_policy • img_signature • img_signature_hash_method • img_signature_key_type • img_signature_certificate_uuid Possible values: • A comma-separated list whose item is an image property. Usually only the image properties that are only needed by base images can be included here, since the snapshots that are created from the base images don't need them. • Default list: cache_in_nova, bittorrent
osapi_compute_listen = 0.0.0.0	string value	IP address on which the OpenStack API will listen. The OpenStack API service listens on this IP address for incoming requests.
osapi_compute_listen_port = 8774	port value	Port on which the OpenStack API will listen. The OpenStack API service listens on this port number for incoming requests.

Configuration option = Default value	Type	Description
<code>`osapi_compute_unique_serv er_name_scope = `</code>	string value	Sets the scope of the check for unique instance names. The default doesn't check for unique names. If a scope for the name check is set, a launch of a new instance or an update of an existing instance with a duplicate name will result in an <i>'InstanceExists'</i> error. The uniqueness is case-insensitive. Setting this option can increase the usability for end users as they don't have to distinguish among instances with the same name by their IDs.
osapi_compute_workers = None	integer value	Number of workers for OpenStack API service. The default will be the number of CPUs available. OpenStack API services can be configured to run as multi-process (workers). This overcomes the problem of reduction in throughput when API request concurrency increases. OpenStack API service will run in the specified number of processes. Possible Values: - Any positive integer - None (default value)
password_length = 12	integer value	Length of generated instance admin passwords.
periodic_enable = True	boolean value	Enable periodic tasks. If set to true, this option allows services to periodically run tasks on the manager. In case of running multiple schedulers or conductors you may want to run periodic tasks on only one host - in this case disable this option for all hosts but one.

Configuration option = Default value	Type	Description
periodic_fuzzy_delay = 60	integer value	Number of seconds to randomly delay when starting the periodic task scheduler to reduce stampeding. When compute workers are restarted in unison across a cluster, they all end up running the periodic tasks at the same time causing problems for the external services. To mitigate this behavior, <code>periodic_fuzzy_delay</code> option allows you to introduce a random initial delay when starting the periodic task scheduler. Possible Values: - Any positive integer (in seconds) 0 : disable the random delay
pointer_model = usbtouchpad	string value	Generic property to specify the pointer type. Input devices allow interaction with a graphical framebuffer. For example to provide a graphic tablet for absolute cursor movement. If set, either the hw_input_bus or hw_pointer_model image metadata properties will take precedence over this configuration option. Related options: <code>usbtouchpad</code> must be configured with VNC enabled or SPICE enabled and SPICE agent disabled. When used with libvirt the instance mode should be configured as HVM.
preallocate_images = none	string value	The image preallocation mode to use. Image preallocation allows storage for instance images to be allocated up front when the instance is initially provisioned. This ensures immediate feedback is given if enough space isn't available. In addition, it should significantly improve performance on writes to new blocks and may even improve I/O performance to prewritten blocks due to reduced fragmentation.
publish_errors = False	boolean value	Enables or disables publication of error events.

Configuration option = Default value	Type	Description
pybasedir = /usr/lib/python3.9/site- packages	string value	The directory where the Nova python modules are installed. This directory is used to store template files for networking and remote console access. It is also the default path for other config options which need to persist Nova internal data. It is very unlikely that you need to change this option from its default value. Possible values: • The full path to a directory. Related options: • state_path
ram_allocation_ratio = None	floating point value	Virtual RAM to physical RAM allocation ratio. This option is used to influence the hosts selected by the Placement API by configuring the allocation ratio for MEMORY_MB inventory. In addition, the AggregateRamFilter (deprecated) will fall back to this configuration value if no per-aggregate setting is found. a. note:: If this option is set to something *other than* `None` or `0.0`, the allocation ratio will be overwritten by the value of this option, otherwise, the allocation ratio will not change. Once set to a non-default value, it is not possible to "unset" the config to get back to the default behavior. If you want to reset back to the initial value, explicitly specify it to the value of `initial_ram_allocation_ratio`. Possible values: • Any valid positive integer or float value Related options: • initial_ram_allocation_ratio
rate_limit_burst = 0	integer value	Maximum number of logged messages per rate_limit_interval.

Configuration option = Default value	Type	Description
rate_limit_except_level = CRITICAL	string value	Log level name used by rate limiting: CRITICAL, ERROR, INFO, WARNING, DEBUG or empty string. Logs with level greater or equal to rate_limit_except_level are not filtered. An empty string means that all levels are filtered.
rate_limit_interval = 0	integer value	Interval, number of seconds, of log rate limiting.
reboot_timeout = 0	integer value	Time interval after which an instance is hard rebooted automatically. When doing a soft reboot, it is possible that a guest kernel is completely hung in a way that causes the soft reboot task to not ever finish. Setting this option to a time period in seconds will automatically hard reboot an instance if it has been stuck in a rebooting state longer than N seconds. Possible values: ● 0: Disables the option (default). ● Any positive integer in seconds: Enables the option.

Configuration option = Default value	Type	Description
reclaim_instance_interval = 0	integer value	Interval for reclaiming deleted instances. A value greater than 0 will enable SOFT_DELETE of instances. This option decides whether the server to be deleted will be put into the SOFT_DELETED state. If this value is greater than 0, the deleted server will not be deleted immediately, instead it will be put into a queue until it's too old (deleted time greater than the value of reclaim_instance_interval). The server can be recovered from the delete queue by using the restore action. If the deleted server remains longer than the value of reclaim_instance_interval, it will be deleted by a periodic task in the compute service automatically. Note that this option is read from both the API and compute nodes, and must be set globally otherwise servers could be put into a soft deleted state in the API and never actually reclaimed (deleted) on the compute node. a. note:: When using this option, you should also configure the [cinder] auth options, e.g. auth_type, auth_url, username, etc. Since the reclaim happens in a periodic task, there is no user token to cleanup volumes attached to any SOFT_DELETED servers so nova must be configured with administrator role access to cleanup those resources in cinder. Possible values: - Any positive integer(in seconds) greater than 0 will enable this option. - Any value ≤ 0 will disable the option. Related options: [cinder] auth options for cleaning up volumes attached to servers during the reclaim process
record = None	string value	Filename that will be used for storing websocket frames received and sent by a proxy service (like VNC, spice, serial) running on this host. If this is not set, no recording will be done.

Configuration option = Default value	Type	Description
report_interval = 10	integer value	Number of seconds indicating how frequently the state of services on a given hypervisor is reported. Nova needs to know this to determine the overall health of the deployment. Related Options: ● <code>service_down_time</code> <code>report_interval</code> should be less than <code>service_down_time</code>. If <code>service_down_time</code> is less than <code>report_interval</code>, services will routinely be considered down, because they report in too rarely.
rescue_timeout = 0	integer value	Interval to wait before un-rescuing an instance stuck in RESCUE. Possible values: ● 0: Disables the option (default) ● Any positive integer in seconds: Enables the option.
reserved_host_cpus = 0	integer value	Number of host CPUs to reserve for host processes. The host resources usage is reported back to the scheduler continuously from nova-compute running on the compute node. This value is used to determine the reserved value reported to placement. This option cannot be set if the <code>[compute] cpu_shared_set</code> or <code>[compute] cpu_dedicated_set</code> config options have been defined. When these options are defined, any host CPUs not included in these values are considered reserved for the host. Possible values: ● Any positive integer representing number of physical CPUs to reserve for the host. Related options: ● <code>[compute] cpu_shared_set</code> ● <code>[compute] cpu_dedicated_set</code>

Configuration option = Default value	Type	Description
reserved_host_disk_mb = 0	integer value	Amount of disk resources in MB to make them always available to host. The disk usage gets reported back to the scheduler from nova-compute running on the compute nodes. To prevent the disk resources from being considered as available, this option can be used to reserve disk space for that host. Possible values: Any positive integer representing amount of disk in MB to reserve for the host.
reserved_host_memory_mb = 512	integer value	Amount of memory in MB to reserve for the host so that it is always available to host processes. The host resources usage is reported back to the scheduler continuously from nova-compute running on the compute node. To prevent the host memory from being considered as available, this option is used to reserve memory for the host. Possible values: Any positive integer representing amount of memory in MB to reserve for the host.
reserved_huge_pages = None	dict value	Number of huge/large memory pages to reserved per NUMA host cell. Possible values: A list of valid key=value which reflect NUMA node ID, page size (Default unit is KiB) and number of pages to be reserved. For example <pre>reserved_huge_pages = node:0,size:2048,count:64 reserved_huge_pages = node:1,size:1GB,count:1</pre> In this example we are reserving on NUMA node 0 64 pages of 2MiB and on NUMA node 1 1 page of 1GiB.

Configuration option = Default value	Type	Description
resize_confirm_window = 0	integer value	Automatically confirm resizes after N seconds. Resize functionality will save the existing server before resizing. After the resize completes, user is requested to confirm the resize. The user has the opportunity to either confirm or revert all changes. Confirm resize removes the original server and changes server status from resized to active. Setting this option to a time period (in seconds) will automatically confirm the resize if the server is in resized state longer than that time. Possible values: ● 0: Disables the option (default) ● Any positive integer in seconds: Enables the option.
resize_fs_using_block_device = False	boolean value	Enable resizing of filesystems via a block device. If enabled, 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).
resume_guests_state_on_host_boot = False	boolean value	This option specifies whether to start guests that were running before the host rebooted. It ensures that all of the instances on a Nova compute node resume their state each time the compute node boots or restarts.
rootwrap_config = /etc/nova/rootwrap.conf	string value	Path to the rootwrap configuration file. Goal of the root wrapper is to allow a service-specific unprivileged user to run a number of actions as the root user in the safest manner possible. The configuration file used here must match the one defined in the sudoers entry.
rpc_conn_pool_size = 30	integer value	Size of RPC connection pool.
rpc_ping_enabled = False	boolean value	Add an endpoint to answer to ping calls. Endpoint is named oslo_rpc_server_ping

Configuration option = Default value	Type	Description
rpc_response_timeout = 60	integer value	Seconds to wait for a response from a call.
run_external_periodic_tasks = True	boolean value	Some periodic tasks can be run in a separate process. Should we run them here?
running_deleted_instance_action = reap	string value	The compute service periodically checks for instances that have been deleted in the database but remain running on the compute node. The above option enables action to be taken when such instances are identified. Related options: ● running_deleted_instance_poll_interval ● running_deleted_instance_timeout
running_deleted_instance_poll_interval = 1800	integer value	Time interval in seconds to wait between runs for the clean up action. If set to 0, above check will be disabled. If "running_deleted_instance_action" is set to "log" or "reap", a value greater than 0 must be set. Possible values: ● Any positive integer in seconds enables the option. ● 0: Disables the option. ● 1800: Default value. Related options: ● running_deleted_instance_action
running_deleted_instance_timeout = 0	integer value	Time interval in seconds to wait for the instances that have been marked as deleted in database to be eligible for cleanup. Possible values: ● Any positive integer in seconds(default is 0). Related options: ● "running_deleted_instance_action"

Configuration option = Default value	Type	Description
scheduler_instance_sync_interval = 120	integer value	Interval 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 <i>scheduler_tracks_instance_changes</i> is False, the sync calls will not be made. So, changing this option will have no effect. If the out of sync situations are not very common, this interval can be increased to lower the number of RPC messages being sent. Likewise, if sync issues turn out to be a problem, the interval can be lowered to check more frequently. Possible values: ● 0: Will run at the default periodic interval. ● Any value < 0: Disables the option. ● Any positive integer in seconds. Related options: ● This option has no impact if scheduler_tracks_instance_changes is set to False.
service_down_time = 60	integer value	Maximum time in seconds since last check-in for up service Each compute node periodically updates their database status based on the specified report interval. If the compute node hasn't updated the status for more than <i>service_down_time</i>, then the compute node is considered down. Related Options: ● <i>report_interval</i> (<i>service_down_time</i> should not be less than <i>report_interval</i>)

Configuration option = Default value	Type	Description
servicegroup_driver = db	string value	This option specifies the driver to be used for the servicegroup service. ServiceGroup API in nova enables checking status of a compute node. When a compute worker running the nova-compute daemon starts, it calls the join API to join the compute group. Services like nova scheduler can query the ServiceGroup API to check if a node is alive. Internally, the ServiceGroup client driver automatically updates the compute worker status. There are multiple backend implementations for this service: Database ServiceGroup driver and Memcache ServiceGroup driver. Related Options: ● service_down_time (maximum time since last check-in for up service)
shelved_offload_time = 0	integer value	Time before a shelved instance is eligible for removal from a host. By default this option is set to 0 and the shelved instance will be removed from the hypervisor immediately after shelve operation. Otherwise, the instance will be kept for the value of shelved_offload_time(in seconds) so that during the time period the unshelve action will be faster, then the periodic task will remove the instance from hypervisor after shelved_offload_time passes. Possible values: ● 0: Instance will be immediately offloaded after being shelved. ● Any value < 0: An instance will never offload. ● Any positive integer in seconds: The instance will exist for the specified number of seconds before being offloaded.

Configuration option = Default value	Type	Description
shelved_poll_interval = 3600	integer value	Interval for polling shelved instances to offload. The periodic task runs for every shelved_poll_interval number of seconds and checks if there are any shelved instances. If it finds a shelved instance, based on the <i>shelved_offload_time</i> config value it offloads the shelved instances. Check <i>shelved_offload_time</i> config option description for details. Possible values: - Any value ≤ 0: Disables the option. - Any positive integer in seconds. Related options: shelved_offload_time
shutdown_timeout = 60	integer value	Total time to wait in seconds for an instance to perform a clean shutdown. It determines the overall period (in seconds) a VM is allowed to perform a clean shutdown. While performing stop, rescue and shelve, rebuild operations, configuring this option gives the VM a chance to perform a controlled shutdown before the instance is powered off. The default timeout is 60 seconds. A value of 0 (zero) means the guest will be powered off immediately with no opportunity for guest OS clean-up. The timeout value can be overridden on a per image basis by means of <i>os_shutdown_timeout</i> that is an image metadata setting allowing different types of operating systems to specify how much time they need to shut down cleanly. Possible values: A positive integer or 0 (default value is 60).
source_is_ipv6 = False	boolean value	Set to True if source host is addressed with IPv6.

Configuration option = Default value	Type	Description
ssl_only = False	boolean value	Disallow non-encrypted connections. Related options: ● cert ● key
state_path = \$pybasedir	string value	The top-level directory for maintaining Nova's state. This directory is used to store Nova's internal state. It is used by a variety of other config options which derive from this. In some scenarios (for example migrations) it makes sense to use a storage location which is shared between multiple compute hosts (for example via NFS). Unless the option instances_path gets overwritten, this directory can grow very large. Possible values: ● The full path to a directory. Defaults to value provided in pybasedir.
sync_power_state_interval = 600	integer value	Interval to sync power states between the database and the hypervisor. The interval that Nova checks the actual virtual machine power state and the power state that Nova has in its database. If a user powers down their VM, Nova updates the API to report the VM has been powered down. Should something turn on the VM unexpectedly, Nova will turn the VM back off to keep the system in the expected state. Possible values: ● 0: Will run at the default periodic interval. ● Any value < 0: Disables the option. ● Any positive integer in seconds. Related options: ● If handle_virt_lifecycle_events in the workarounds group is false and this option is negative, then instances that get out of sync between the hypervisor and the Nova database will have to be synchronized manually.

Configuration option = Default value	Type	Description
sync_power_state_pool_size = 1000	integer value	Number of greenthreads available for use to sync power states. This option can be used to reduce the number of concurrent requests made to the hypervisor or system with real instance power states for performance reasons, for example, with Ironic. Possible values: Any positive integer representing greenthreads count.
syslog-log-facility = LOG_USER	string value	Syslog facility to receive log lines. This option is ignored if log_config_append is set.
tempdir = None	string value	Explicitly specify the temporary working directory.
timeout_nbd = 10	integer value	Amount of time, in seconds, to wait for NBD device start up.
transport_url = rabbit://	string value	The network address and optional user credentials for connecting to the messaging backend, in URL format. The expected format is: driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query Example: rabbit://rabbitmq:password@127.0.0.1:5672// For full details on the fields in the URL see the documentation of oslo_messaging.TransportURL at https://docs.openstack.org/oslo.messaging/latest/reference/transport.html

Configuration option = Default value	Type	Description
update_resources_interval = 0	integer value	Interval for updating compute resources. This option specifies how often the update_available_resource periodic task should run. A number less than 0 means to disable the task completely. Leaving this at the default of 0 will cause this to run at the default periodic interval. Setting it to any positive value will cause it to run at approximately that number of seconds. Possible values: ● 0: Will run at the default periodic interval. ● Any value < 0: Disables the option. ● Any positive integer in seconds.
use-journal = False	boolean value	Enable journald for logging. If running in a systemd environment you may wish to enable journal support. Doing so will use the journal native protocol which includes structured metadata in addition to log messages. This option is ignored if log_config_append is set.
use-json = False	boolean value	Use JSON formatting for logging. This option is ignored if log_config_append is set.
use-syslog = False	boolean value	Use syslog for logging. Existing syslog format is DEPRECATED and will be changed later to honor RFC5424. This option is ignored if log_config_append is set.
use_cow_images = True	boolean value	Enable use of copy-on-write (cow) images. QEMU/KVM allow the use of qcow2 as backing files. By disabling this, backing files will not be used.
use_eventlog = False	boolean value	Log output to Windows Event Log.
use_rootwrap_daemon = False	boolean value	Start and use a daemon that can run the commands that need to be run with root privileges. This option is usually enabled on nodes that run nova compute processes.
use_stderr = False	boolean value	Log output to standard error. This option is ignored if log_config_append is set.

Configuration option = Default value	Type	Description
vcpu_pin_set = None	string value	Mask of host CPUs that can be used for VCPU resources. The behavior of this option depends on the definition of the [compute] cpu_dedicated_set option and affects the behavior of the [compute] cpu_shared_set option. • If [compute] cpu_dedicated_set is defined, defining this option will result in an error. • If [compute] cpu_dedicated_set is not defined, this option will be used to determine inventory for VCPU resources and to limit the host CPUs that both pinned and unpinned instances can be scheduled to, overriding the [compute] cpu_shared_set option. Possible values: • A comma-separated list of physical CPU numbers that virtual CPUs can be allocated from. Each element should be either a single CPU number, a range of CPU numbers, or a caret followed by a CPU number to be excluded from a previous range. For example <pre>vcpu_pin_set = "4-12,^8,15"</pre> Related options: • [compute] cpu_dedicated_set • [compute] cpu_shared_set Deprecated since: 20.0.0 Reason: This option has been superseded by the <code>``[compute] cpu_dedicated_set``</code> and <code>``[compute] cpu_shared_set``</code> options, which allow things like the co-existence of pinned and unpinned instances on the same host (for the libvirt driver).

Configuration option = Default value	Type	Description
vif_plugging_is_fatal = True	boolean value	Determine if instance should boot or fail on VIF plugging timeout. Nova sends a port update to Neutron after an instance has been scheduled, providing Neutron with the necessary information to finish setup of the port. Once completed, Neutron notifies Nova that it has finished setting up the port, at which point Nova resumes the boot of the instance since network connectivity is now supposed to be present. A timeout will occur if the reply is not received after a given interval. This option determines what Nova does when the VIF plugging timeout event happens. When enabled, the instance will error out. When disabled, the instance will continue to boot on the assumption that the port is ready. Possible values: • True: Instances should fail after VIF plugging timeout • False: Instances should continue booting after VIF plugging timeout
vif_plugging_timeout = 300	integer value	Timeout for Neutron VIF plugging event message arrival. Number of seconds to wait for Neutron vif plugging events to arrive before continuing or failing (see <i>vif_plugging_is_fatal</i>). If you are hitting timeout failures at scale, consider running rootwrap in "daemon mode" in the neutron agent via the [agent]/root_helper_daemon neutron configuration option. Related options: • <i>vif_plugging_is_fatal</i> - If vif_plugging_timeout is set to zero and vif_plugging_is_fatal is False, events should not be expected to arrive at all.
virt_mkfs = []	multi valued	Name of the mkfs commands for ephemeral device. The format is <os_type>=<mkfs command>

Configuration option = Default value	Type	Description
volume_usage_poll_interval = 0	integer value	Interval for gathering volume usages. This option updates the volume usage cache for every volume_usage_poll_interval number of seconds. Possible values: - Any positive integer(in seconds) greater than 0 will enable this option. - Any value ≤ 0 will disable the option.
watch-log-file = False	boolean value	Uses logging handler designed to watch file system. When log file is moved or removed this handler will open a new log file with specified path instantaneously. It makes sense only if log_file option is specified and Linux platform is used. This option is ignored if log_config_append is set.
web = /usr/share/spice-html5	string value	Path to directory with content which will be served by a web server.

9.1.2. api

The following table outlines the options available under the **[api]** group in the **/etc/nova/nova.conf** file.

Table 9.1. api

Configuration option = Default value	Type	Description
auth_strategy = keystone	string value	Determine the strategy to use for authentication. Deprecated since: 21.0.0 Reason: The only non-default choice, ``noauth2``, is for internal development and testing purposes only and should not be used in deployments. This option and its middleware, NoAuthMiddleware[V2_18], will be removed in a future release.

Configuration option = Default value	Type	Description
compute_link_prefix = None	string value	This string is prepended to the normal URL that is returned in links to the OpenStack Compute API. If it is empty (the default), the URLs are returned unchanged. Possible values: Any string, including an empty string (the default).
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	string value	When gathering the existing metadata for a config drive, the EC2-style metadata is returned for all versions that don't appear in this option. As of the Liberty release, the available versions are: 1.0 2007-01-19 2007-03-01 2007-08-29 2007-10-10 2007-12-15 2008-02-01 2008-09-01 2009-04-04 The option is in the format of a single string, with each version separated by a space. Possible values: - Any string that represents zero or more versions, separated by spaces.
dhcp_domain = novalocal	string value	Domain name used to configure FQDN for instances. Configure a fully-qualified domain name for instance hostnames. If unset, only the hostname without a domain will be configured. Possible values: Any string that is a valid domain name.

Configuration option = Default value	Type	Description
enable_instance_password = True	boolean value	Enables returning of the instance password by the relevant server API calls such as create, rebuild, evacuate, or rescue. If the hypervisor does not support password injection, then the password returned will not be correct, so if your hypervisor does not support password injection, set this to False.
glance_link_prefix = None	string value	This string is prepended to the normal URL that is returned in links to Glance resources. If it is empty (the default), the URLs are returned unchanged. Possible values: Any string, including an empty string (the default).
instance_list_cells_batch_fixed_size = 100	integer value	This controls the batch size of instances requested from each cell database if instance_list_cells_batch_strategy is set to fixed. This integral value will define the limit issued to each cell every time a batch of instances is requested, regardless of the number of cells in the system or any other factors. Per the general logic called out in the documentation for instance_list_cells_batch_strategy, the minimum value for this is 100 records per batch. Related options: - instance_list_cells_batch_strategy - max_limit

Configuration option = Default value	Type	Description
instance_list_cells_batch_strategy = distributed	string value	This controls the method by which the API queries cell databases in smaller batches during large instance list operations. If batching is performed, a large instance list operation will request some fraction of the overall API limit from each cell database initially, and will re-request that same batch size as records are consumed (returned) from each cell as necessary. Larger batches mean less chattiness between the API and the database, but potentially more wasted effort processing the results from the database which will not be returned to the user. Any strategy will yield a batch size of at least 100 records, to avoid a user causing many tiny database queries in their request. Related options: ● instance_list_cells_batch_fixed_size ● max_limit
instance_list_per_project_cells = False	boolean value	When enabled, this will cause the API to only query cell databases in which the tenant has mapped instances. This requires an additional (fast) query in the API database before each list, but also (potentially) limits the number of cell databases that must be queried to provide the result. If you have a small number of cells, or tenants are likely to have instances in all cells, then this should be False. If you have many cells, especially if you confine tenants to a small subset of those cells, this should be True.
list_records_by_skipping_down_cells = True	boolean value	When set to False, this will cause the API to return a 500 error if there is an infrastructure failure like non-responsive cells. If you want the API to skip the down cells and return the results from the up cells set this option to True. Note that from API microversion 2.69 there could be transient conditions in the deployment where certain records are not available and the results could be partial for certain requests containing those records. In those cases this option will be ignored. See "Handling Down Cells" section of the Compute API guide (https://docs.openstack.org/api-guide/compute/down_cells.html) for more information.

Configuration option = Default value	Type	Description
local_metadata_per_cell = False	boolean value	Indicates that the nova-metadata API service has been deployed per-cell, so that we can have better performance and data isolation in a multi-cell deployment. Users should consider the use of this configuration depending on how neutron is setup. If you have networks that span cells, you might need to run nova-metadata API service globally. If your networks are segmented along cell boundaries, then you can run nova-metadata API service per cell. When running nova-metadata API service per cell, you should also configure each Neutron metadata-agent to point to the corresponding nova-metadata API service.
max_limit = 1000	integer value	As a query can potentially return many thousands of items, you can limit the maximum number of items in a single response by setting this option.
metadata_cache_expiration = 15	integer value	This option is the time (in seconds) to cache metadata. When set to 0, metadata caching is disabled entirely; this is generally not recommended for performance reasons. Increasing this setting should improve response times of the metadata API when under heavy load. Higher values may increase memory usage, and result in longer times for host metadata changes to take effect.
neutron_default_tenant_id = default	string value	Tenant ID for getting the default network from Neutron API (also referred in some places as the <i>project ID</i>) to use. Related options: ● <code>use_neutron_default_nets</code>
use_forwarded_for = False	boolean value	When True, the <i>X-Forwarded-For</i> header is treated as the canonical remote address. When False (the default), the <i>remote_address</i> header is used. You should only enable this if you have an HTML sanitizing proxy.
use_neutron_default_nets = False	boolean value	When True, the TenantNetworkController will query the Neutron API to get the default networks to use. Related options: ● <code>neutron_default_tenant_id</code>

Configuration option = Default value	Type	Description
vendordata_dynamic_connect_timeout = 5	integer value	Maximum wait time for an external REST service to connect. Possible values: - Any integer with a value greater than three (the TCP packet retransmission timeout). Note that instance start may be blocked during this wait time, so this value should be kept small. Related options: <code>vendordata_providers</code> <code>vendordata_dynamic_targets</code> <code>vendordata_dynamic_ssl_certfile</code> <code>vendordata_dynamic_read_timeout</code> <code>vendordata_dynamic_failure_fatal</code>
vendordata_dynamic_failure_fatal = False	boolean value	Should failures to fetch dynamic vendordata be fatal to instance boot? Related options: <code>vendordata_providers</code> <code>vendordata_dynamic_targets</code> <code>vendordata_dynamic_ssl_certfile</code> <code>vendordata_dynamic_connect_timeout</code> <code>vendordata_dynamic_read_timeout</code>

Configuration option = Default value	Type	Description
vendordata_dynamic_read_timeout = 5	integer value	Maximum wait time for an external REST service to return data once connected. Possible values: - Any integer. Note that instance start is blocked during this wait time, so this value should be kept small. Related options: - vendordata_providers - vendordata_dynamic_targets - vendordata_dynamic_ssl_certfile - vendordata_dynamic_connect_timeout - vendordata_dynamic_failure_fatal
vendordata_dynamic_ssl_certfile = ``	string value	Path to an optional certificate file or CA bundle to verify dynamic vendordata REST services ssl certificates against. Possible values: - An empty string, or a path to a valid certificate file Related options: - vendordata_providers - vendordata_dynamic_targets - vendordata_dynamic_connect_timeout - vendordata_dynamic_read_timeout - vendordata_dynamic_failure_fatal
vendordata_dynamic_targets = []	list value	A list of targets for the dynamic vendordata provider. These targets are of the form <name>@<url>. The dynamic vendordata provider collects metadata by contacting external REST services and querying them for information about the instance. This behaviour is documented in the vendordata.rst file in the nova developer reference.

Configuration option = Default value	Type	Description
vendordata_jsonfile_path = None	string value	Cloud providers may store custom data in vendor data file that will then be available to the instances via the metadata service, and to the rendering of config-drive. The default class for this, <code>JsonFileVendorData</code>, loads this information from a JSON file, whose path is configured by this option. If there is no path set by this option, the class returns an empty dictionary. Note that when using this to provide static vendor data to a configuration drive, the nova-compute service must be configured with this option and the file must be accessible from the nova-compute host. Possible values: Any string representing the path to the data file, or an empty string (default).
vendordata_providers = ['StaticJSON']	list value	A list of vendordata providers. vendordata providers are how deployers can provide metadata via configdrive and metadata that is specific to their deployment. For more information on the requirements for implementing a vendordata dynamic endpoint, please see the <code>vendordata.rst</code> file in the nova developer reference. Related options: vendordata_dynamic_targets vendordata_dynamic_ssl_certfile vendordata_dynamic_connect_timeout vendordata_dynamic_read_timeout vendordata_dynamic_failure_fatal

9.1.3. api_database

The following table outlines the options available under the **[api_database]** group in the `/etc/nova/nova.conf` file.

Table 9.2. api_database

Configuration option = Default value	Type	Description
connection = None	string value	The SQLAlchemy connection string to use to connect to the database. Do not set this for the nova-compute service.
connection_debug = 0	integer value	Verbosity of SQL debugging information: 0=None, 100=Everything.
<code>connection_parameters = `</code>	string value	Optional URL parameters to append onto the connection URL at connect time; specify as param1=value1&param2=value2&...
connection_recycle_time = 3600	integer value	Connections which have been present in the connection pool longer than this number of seconds will be replaced with a new one the next time they are checked out from the pool.
connection_trace = False	boolean value	Add Python stack traces to SQL as comment strings.
max_overflow = None	integer value	If set, use this value for max_overflow with SQLAlchemy.
max_pool_size = None	integer value	Maximum number of SQL connections to keep open in a pool. Setting a value of 0 indicates no limit.
max_retries = 10	integer value	Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.
mysql_sql_mode = TRADITIONAL	string value	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	integer value	If set, use this value for pool_timeout with SQLAlchemy.
retry_interval = 10	integer value	Interval between retries of opening a SQL connection.
slave_connection = None	string value	The SQLAlchemy connection string to use to connect to the slave database.
sqlite_synchronous = True	boolean value	If True, SQLite uses synchronous mode.

9.1.4. barbican

The following table outlines the options available under the **[barbican]** group in the **/etc/nova/nova.conf** file.

Table 9.3. barbican

Configuration option = Default value	Type	Description
auth_endpoint = http://localhost/identity/v3	string value	Use this endpoint to connect to Keystone
barbican_api_version = None	string value	Version of the Barbican API, for example: "v1"
barbican_endpoint = None	string value	Use this endpoint to connect to Barbican, for example: "http://localhost:9311/"
barbican_endpoint_type = public	string value	Specifies the type of endpoint. Allowed values are: public, private, and admin
number_of_retries = 60	integer value	Number of times to retry poll for key creation completion
retry_delay = 1	integer value	Number of seconds to wait before retrying poll for key creation completion
verify_ssl = True	boolean value	Specifies if insecure TLS (https) requests. If False, the server's certificate will not be validated, if True, we can set the verify_ssl_path config meanwhile.
verify_ssl_path = None	string value	A path to a bundle or CA certs to check against, or None for requests to attempt to locate and use certificates which verify_ssh is True. If verify_ssl is False, this is ignored.

9.1.5. cache

The following table outlines the options available under the **[cache]** group in the **/etc/nova/nova.conf** file.

Table 9.4. cache

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
backend = dogpile.cache.null	string value	Cache backend module. For eventlet-based or environments with hundreds of threaded servers, Memcache with pooling (oslo_cache.memcache_pool) is recommended. For environments with less than 100 threaded servers, Memcached (dogpile.cache.memcached) or Redis (dogpile.cache.redis) is recommended. Test environments with a single instance of the server can use the dogpile.cache.memory backend.
backend_argument = []	multi valued	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.oslo	string value	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.
dead_timeout = 60	floating point value	Time in seconds before attempting to add a node back in the pool in the HashClient's internal mechanisms.
debug_cache_backend = False	boolean value	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.
enable_retry_client = False	boolean value	Enable retry client mechanisms to handle failure. Those mechanisms can be used to wrap all kind of pymemcache clients. The wrapper allows you to define how many attempts to make and how long to wait between attempts.
enable_socket_keepalive = False	boolean value	Global toggle for the socket keepalive of dogpile's pymemcache backend
enabled = False	boolean value	Global toggle for caching.
expiration_time = 600	integer value	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.

Configuration option = Default value	Type	Description
hashclient_retry_attempts = 2	integer value	Amount of times a client should be tried before it is marked dead and removed from the pool in the HashClient's internal mechanisms.
hashclient_retry_delay = 1	floating point value	Time in seconds that should pass between retry attempts in the HashClient's internal mechanisms.
memcache_dead_retry = 300	integer value	Number of seconds memcached server is considered dead before it is tried again. (dogpile.cache.memcache and oslo_cache.memcache_pool backends only).
memcache_pool_connection_get_timeout = 10	integer value	Number of seconds that an operation will wait to get a memcache client connection.
memcache_pool_flush_on_reconnect = False	boolean value	Global toggle if memcache will be flushed on reconnect. (oslo_cache.memcache_pool backend only).
memcache_pool_maxsize = 10	integer value	Max total number of open connections to every memcached server. (oslo_cache.memcache_pool backend only).
memcache_pool_unused_timeout = 60	integer value	Number of seconds a connection to memcached is held unused in the pool before it is closed. (oslo_cache.memcache_pool backend only).
memcache_servers = ['localhost:11211']	list value	Memcache servers in the format of "host:port". (dogpile.cache.memcached and oslo_cache.memcache_pool backends only). If a given host refer to an IPv6 or a given domain refer to IPv6 then you should prefix the given address with the address family (inet6) (e.g inet6[::1]:11211 , inet6:[fd12:3456:789a:1::1]:11211 , inet6:[controller-0.internalapi]:11211). If the address family is not given then default address family used will be inet which correspond to IPv4
memcache_socket_timeout = 1.0	floating point value	Timeout in seconds for every call to a server. (dogpile.cache.memcache and oslo_cache.memcache_pool backends only).
proxies = []	list value	Proxy classes to import that will affect the way the dogpile.cache backend functions. See the dogpile.cache documentation on changing-backend-behavior.

Configuration option = Default value	Type	Description
retry_attempts = 2	integer value	Number of times to attempt an action before failing.
retry_delay = 0	floating point value	Number of seconds to sleep between each attempt.
socket_keepalive_count = 1	integer value	The maximum number of keepalive probes TCP should send before dropping the connection. Should be a positive integer greater than zero.
socket_keepalive_idle = 1	integer value	The time (in seconds) the connection needs to remain idle before TCP starts sending keepalive probes. Should be a positive integer most greater than zero.
socket_keepalive_interval = 1	integer value	The time (in seconds) between individual keepalive probes. Should be a positive integer greater than zero.
tls_allowed_ciphers = None	string value	Set the available ciphers for sockets created with the TLS context. It should be a string in the OpenSSL cipher list format. If not specified, all OpenSSL enabled ciphers will be available.
tls_cafile = None	string value	Path to a file of concatenated CA certificates in PEM format necessary to establish the caching servers' authenticity. If <code>tls_enabled</code> is <code>False</code> , this option is ignored.
tls_certfile = None	string value	Path to a single file in PEM format containing the client's certificate as well as any number of CA certificates needed to establish the certificate's authenticity. This file is only required when client side authentication is necessary. If <code>tls_enabled</code> is <code>False</code> , this option is ignored.
tls_enabled = False	boolean value	Global toggle for TLS usage when communicating with the caching servers.
tls_keyfile = None	string value	Path to a single file containing the client's private key in. Otherwise the private key will be taken from the file specified in <code>tls_certfile</code> . If <code>tls_enabled</code> is <code>False</code> , this option is ignored.

9.1.6. cinder

The following table outlines the options available under the **[cinder]** group in the `/etc/nova/nova.conf` file.

Table 9.5. cinder

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
catalog_info = volumev3::publicURL	string value	Info to match when looking for cinder in the service catalog. The <service_name> is optional and omitted by default since it should not be necessary in most deployments. Possible values: - Format is separated values of the form: <service_type>:<service_name>: <endpoint_type> Note: Nova does not support the Cinder v2 API since the Nova 17.0.0 Queens release. Related options: - endpoint_template - Setting this option will override catalog_info
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.

Configuration option = Default value	Type	Description
cross_az_attach = True	boolean value	Allow attach between instance and volume in different availability zones. If False, volumes attached to an instance must be in the same availability zone in Cinder as the instance availability zone in Nova. This also means care should be taken when booting an instance from a volume where source is not "volume" because Nova will attempt to create a volume using the same availability zone as what is assigned to the instance. If that AZ is not in Cinder (or allow_availability_zone_fallback=False in cinder.conf), the volume create request will fail and the instance will fail the build request. By default there is no availability zone restriction on volume attach. Related options: • [DEFAULT]/default_schedule_zone
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to

Configuration option = Default value	Type	Description
endpoint_template = None	string value	If this option is set then it will override service catalog lookup with this template for cinder endpoint Possible values: ● URL for cinder endpoint API e.g. http://localhost:8776/v3/%(project_id)s Note: Nova does not support the Cinder v2 API since the Nova 17.0.0 Queens release. Related options: ● catalog_info - If endpoint_template is not set, catalog_info will be used.
http_retries = 3	integer value	Number of times cinderclient should retry on any failed http call. 0 means connection is attempted only once. Setting it to any positive integer means that on failure connection is retried that many times e.g. setting it to 3 means total attempts to connect will be 4. Possible values: ● Any integer value. 0 means connection is attempted only once
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
os_region_name = None	string value	Region name of this node. This is used when picking the URL in the service catalog. Possible values: ● Any string representing region name
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to

Configuration option = Default value	Type	Description
project-name = None	string value	Project name to scope to
split-loggers = False	boolean value	Log requests to multiple loggers.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User ID
username = None	string value	Username

9.1.7. compute

The following table outlines the options available under the **[compute]** group in the **/etc/nova/nova.conf** file.

Table 9.6. compute

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
consecutive_build_service_disable_threshold = 10	integer value	Enables reporting of build failures to the scheduler. Any nonzero value will enable sending build failure statistics to the scheduler for use by the BuildFailureWeigher. Possible values: • Any positive integer enables reporting build failures. • Zero to disable reporting build failures. Related options: • [filter_scheduler]/build_failure_weight_multiplier

Configuration option = Default value	Type	Description
cpu_dedicated_set = None	string value	Mask of host CPUs that can be used for PCPU resources. The behavior of this option affects the behavior of the deprecated vcpu_pin_set option. ● If this option is defined, defining vcpu_pin_set will result in an error. ● If this option is not defined, vcpu_pin_set will be used to determine inventory for VCPUs resources and to limit the host CPUs that both pinned and unpinned instances can be scheduled to. This behavior will be simplified in a future release when vcpu_pin_set is removed. Possible values: ● A comma-separated list of physical CPU numbers that instance VCPUs can be allocated from. Each element should be either a single CPU number, a range of CPU numbers, or a caret followed by a CPU number to be excluded from a previous range. For example <code>cpu_dedicated_set = "4-12,^8,15"</code> Related options: ● [compute] cpu_shared_set: This is the counterpart option for defining where VCPUs resources should be allocated from. ● vcpu_pin_set: A legacy option that this option partially replaces.

Configuration option = Default value	Type	Description
cpu_shared_set = None	string value	Mask of host CPUs that can be used for VCPU resources and offloaded emulator threads. The behavior of this option depends on the definition of the deprecated vcpu_pin_set option. • If vcpu_pin_set is not defined, [compute] cpu_shared_set will be used to provide VCPU inventory and to determine the host CPUs that unpinned instances can be scheduled to. It will also be used to determine the host CPUs that instance emulator threads should be offloaded to for instances configured with the share emulator thread policy (hw:emulator_threads_policy=share). • If vcpu_pin_set is defined, [compute] cpu_shared_set will only be used to determine the host CPUs that instance emulator threads should be offloaded to for instances configured with the share emulator thread policy (hw:emulator_threads_policy=share). vcpu_pin_set will be used to provide VCPU inventory and to determine the host CPUs that both pinned and unpinned instances can be scheduled to. This behavior will be simplified in a future release when vcpu_pin_set is removed. Possible values: • A comma-separated list of physical CPU numbers that instance VCPUs can be allocated from. Each element should be either a single CPU number, a range of CPU numbers, or a caret followed by a CPU number to be excluded from a previous range. For example <pre>cpu_shared_set = "4-12,^8,15"</pre> Related options: • [compute] cpu_dedicated_set: This is the counterpart option for defining where PCPU resources should be allocated from. • vcpu_pin_set: A legacy option whose definition may change the behavior of this option.

Configuration option = Default value	Type	Description
image_type_exclude_list = []	list value	A list of image formats that should not be advertised as supported by this compute node. In some situations, it may be desirable to have a compute node refuse to support an expensive or complex image format. This factors into the decisions made by the scheduler about which compute node to select when booted with a given image. Possible values: - Any glance image disk_format name (i.e. raw, qcow2, etc) Related options: [scheduler]query_placement_for_image_type_support - enables filtering computes based on supported image types, which is required to be enabled for this to take effect.
live_migration_wait_for_vif_plug = True	boolean value	Determine if the source compute host should wait for a network-vif-plugged event from the (neutron) networking service before starting the actual transfer of the guest to the destination compute host. Note that this option is read on the destination host of a live migration. If you set this option the same on all of your compute hosts, which you should do if you use the same networking backend universally, you do not have to worry about this. Before starting the transfer of the guest, some setup occurs on the destination compute host, including plugging virtual interfaces. Depending on the networking backend on the destination host, a network-vif-plugged event may be triggered and then received on the source compute host and the source compute can wait for that event to ensure networking is set up on the destination host before starting the guest transfer in the hypervisor. a. note:: The compute service cannot reliably determine which types of virtual interfaces (<code>`port.binding:vif_type`</code>) will send <code>`network-vif-plugged`</code> events without an accompanying port <code>`binding:host_id`</code> change. Open vSwitch and linuxbridge should

Configuration option = Default value	Type	Description
		be OK, but OpenDaylight is at least one known backend that will not currently work in this case, see bug https://launchpad.net/bugs/1755890 for more details. Possible values: • True: wait for network-vif-plugged events before starting guest transfer • False: do not wait for network-vif-plugged events before starting guest transfer (this is the legacy behavior) Related options: • [DEFAULT]/vif_plugging_is_fatal: if live_migration_wait_for_vif_plug is True and vif_plugging_timeout is greater than 0, and a timeout is reached, the live migration process will fail with an error but the guest transfer will not have started to the destination host • [DEFAULT]/vif_plugging_timeout: if live_migration_wait_for_vif_plug is True, this controls the amount of time to wait before timing out and either failing if vif_plugging_is_fatal is True, or simply continuing with the live migration
max_concurrent_disk_ops = 0	integer value	Number of concurrent disk-IO-intensive operations (glance image downloads, image format conversions, etc.) that we will do in parallel. If this is set too high then response time suffers. The default value of 0 means no limit.
max_disk_devices_to_attach = -1	integer value	Maximum number of disk devices allowed to attach to a single server. Note that the number of disks supported by an server depends on the bus used. For example, the ide disk bus is limited to 4 attached devices. The configured maximum is enforced during server create, rebuild, evacuate, unshelve, live migrate, and attach volume. Usually, disk bus is determined automatically from the device type or disk device, and the virtualization type. However, disk bus can also be specified via a block device mapping or an image property. See the disk_bus field in :doc:/user/block-device-mapping for more information about specifying disk bus in a block device mapping, and see https://docs.openstack.org/glance/latest/admin/useful-image-properties.html for more information about the hw_disk_bus image property. Operators changing the

Configuration option = Default value	Type	Description
		[compute]/max_disk_devices_to_attach on a compute service that is hosting servers should be aware that it could cause rebuilds to fail, if the maximum is decreased lower than the number of devices already attached to servers. For example, if server A has 26 devices attached and an operators changes [compute]/max_disk_devices_to_attach to 20, a request to rebuild server A will fail and go into ERROR state because 26 devices are already attached and exceed the new configured maximum of 20. Operators setting [compute]/max_disk_devices_to_attach should also be aware that during a cold migration, the configured maximum is only enforced in-place and the destination is not checked before the move. This means if an operator has set a maximum of 26 on compute host A and a maximum of 20 on compute host B, a cold migration of a server with 26 attached devices from compute host A to compute host B will succeed. Then, once the server is on compute host B, a subsequent request to rebuild the server will fail and go into ERROR state because 26 devices are already attached and exceed the configured maximum of 20 on compute host B. The configured maximum is not enforced on shelved offloaded servers, as they have no compute host. a. warning:: If this option is set to 0, the nova-compute service will fail to start, as 0 disk devices is an invalid configuration that would prevent instances from being able to boot. Possible values: ● -1 means unlimited ● Any integer ≥ 1 represents the maximum allowed. A value of 0 will cause the nova-compute service to fail to start, as 0 disk devices is an invalid configuration that would prevent instances from being able to boot.

Configuration option = Default value	Type	Description
provider_config_location = /etc/nova/provider_config /	string value	Location of YAML files containing resource provider configuration data. These files allow the operator to specify additional custom inventory and traits to assign to one or more resource providers. Additional documentation is available here: https://docs.openstack.org/nova/latest/admin/managing-resource-providers.html
resource_provider_association_refresh = 300	integer value	Interval for updating nova-compute-side cache of the compute node resource provider's inventories, aggregates, and traits. This option specifies the number of seconds between attempts to update a provider's inventories, aggregates and traits in the local cache of the compute node. A value of zero disables cache refresh completely. The cache can be cleared manually at any time by sending SIGHUP to the compute process, causing it to be repopulated the next time the data is accessed. Possible values: Any positive integer in seconds, or zero to disable refresh.
shutdown_retry_interval = 10	integer value	Time to wait in seconds before resending an ACPI shutdown signal to instances. The overall time to wait is set by shutdown_timeout. Possible values: - Any integer greater than 0 in seconds Related options: shutdown_timeout

Configuration option = Default value	Type	Description
vmdk_allowed_types = ['streamOptimized', 'monolithicSparse']	list value	A list of strings describing allowed VMDK "create-type" subformats that will be allowed. This is recommended to only include single-file-with-sparse-header variants to avoid potential host file exposure due to processing named extents. If this list is empty, then no form of VMDK image will be allowed.

9.1.8. conductor

The following table outlines the options available under the **[conductor]** group in the `/etc/nova/nova.conf` file.

Table 9.7. conductor

Configuration option = Default value	Type	Description
workers = None	integer value	Number of workers for OpenStack Conductor service. The default will be the number of CPUs available.

9.1.9. console

The following table outlines the options available under the **[console]** group in the `/etc/nova/nova.conf` file.

Table 9.8. console

Configuration option = Default value	Type	Description
allowed_origins = []	list value	Adds list of allowed origins to the console websocket proxy to allow connections from other origin hostnames. Websocket proxy matches the host header with the origin header to prevent cross-site requests. This list specifies if any there are values other than host are allowed in the origin header. Possible values: • A list where each element is an allowed origin hostnames, else an empty list

Configuration option = Default value	Type	Description
ssl_ciphers = None	string value	OpenSSL cipher preference string that specifies what ciphers to allow for TLS connections from clients. For example:: <pre>ssl_ciphers = "kEECDH+aECDH+AES:kEECDH+AES+aR SA:kEDH+aRSA+AES"</pre> See the man page for the OpenSSL ciphers command for details of the cipher preference string format and allowed values:: <pre>https://www.openssl.org/docs/man1.1.0/man1/ ciphers.html</pre> Related options: • [DEFAULT] cert • [DEFAULT] key
ssl_minimum_version = default	string value	Minimum allowed SSL/TLS protocol version. Related options: • [DEFAULT] cert • [DEFAULT] key

9.1.10. consoleauth

The following table outlines the options available under the **[consoleauth]** group in the **/etc/nova/nova.conf** file.

Table 9.9. consoleauth

Configuration option = Default value	Type	Description
token_ttl = 600	integer value	The lifetime of a console auth token (in seconds). A console auth token is used in authorizing console access for a user. Once the auth token time to live count has elapsed, the token is considered expired. Expired tokens are then deleted.

9.1.11. cors

The following table outlines the options available under the **[cors]** group in the `/etc/nova/nova.conf` file.

Table 9.10. cors

Configuration option = Default value	Type	Description
allow_credentials = True	boolean value	Indicate that the actual request can include user credentials
allow_headers = ['X-Auth-Token', 'X-Openstack-Request-Id', 'X-Identity-Status', 'X-Roles', 'X-Service-Catalog', 'X-User-Id', 'X-Tenant-Id', 'X-OpenStack-Nova-API-Version', 'OpenStack-API-Version']	list value	Indicate which header field names may be used during the actual request.
allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH']	list value	Indicate which methods can be used during the actual request.
allowed_origin = None	list value	Indicate whether this resource may be shared with the domain received in the requests "origin" header. Format: "<protocol>://<host>[:<port>]", no trailing slash. Example: https://horizon.example.com
expose_headers = ['X-Auth-Token', 'X-Openstack-Request-Id', 'X-Subject-Token', 'X-Service-Token', 'X-OpenStack-Nova-API-Version', 'OpenStack-API-Version']	list value	Indicate which headers are safe to expose to the API. Defaults to HTTP Simple Headers.
max_age = 3600	integer value	Maximum cache age of CORS preflight requests.

9.1.12. cyborg

The following table outlines the options available under the **[cyborg]** group in the `/etc/nova/nova.conf` file.

Table 9.11. cyborg

Configuration option = Default value	Type	Description
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.

Configuration option = Default value	Type	Description
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
region-name = None	string value	The default region_name for endpoint URL discovery.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = accelerator	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrieable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
timeout = None	integer value	Timeout value for http requests
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.

9.1.13. database

The following table outlines the options available under the **[database]** group in the **/etc/nova/nova.conf** file.

Table 9.12. database

Configuration option = Default value	Type	Description
backend = sqlalchemy	string value	The back end to use for the database.
connection = None	string value	The SQLAlchemy connection string to use to connect to the database.
connection_debug = 0	integer value	Verbosity of SQL debugging information: 0=None, 100=Everything.
connection_parameters = ``	string value	Optional URL parameters to append onto the connection URL at connect time; specify as param1=value1&param2=value2&...
connection_recycle_time = 3600	integer value	Connections which have been present in the connection pool longer than this number of seconds will be replaced with a new one the next time they are checked out from the pool.
connection_trace = False	boolean value	Add Python stack traces to SQL as comment strings.
db_inc_retry_interval = True	boolean value	If True, increases the interval between retries of a database operation up to db_max_retry_interval.
db_max_retries = 20	integer value	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	integer value	If db_inc_retry_interval is set, the maximum seconds between retries of a database operation.
db_retry_interval = 1	integer value	Seconds between retries of a database transaction.
max_overflow = 50	integer value	If set, use this value for max_overflow with SQLAlchemy.
max_pool_size = 5	integer value	Maximum number of SQL connections to keep open in a pool. Setting a value of 0 indicates no limit.
max_retries = 10	integer value	Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.

Configuration option = Default value	Type	Description
mysql_enable_ndb = False	boolean value	If True, transparently enables support for handling MySQL Cluster (NDB).
mysql_sql_mode = TRADITIONAL	string value	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: <code>mysql_sql_mode=</code>
pool_timeout = None	integer value	If set, use this value for pool_timeout with SQLAlchemy.
retry_interval = 10	integer value	Interval between retries of opening a SQL connection.
slave_connection = None	string value	The SQLAlchemy connection string to use to connect to the slave database.
sqlite_synchronous = True	boolean value	If True, SQLite uses synchronous mode.
use_db_reconnect = False	boolean value	Enable the experimental use of database reconnect on connection lost.
use_tpool = False	boolean value	Enable the experimental use of thread pooling for all DB API calls

9.1.14. devices

The following table outlines the options available under the **[devices]** group in the `/etc/nova/nova.conf` file.

Table 9.13. devices

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
enabled_vgpu_types = []	list value	The vGPU types enabled in the compute node. Some pGPUs (e.g. NVIDIA GRID K1) support different vGPU types. User can use this option to specify a list of enabled vGPU types that may be assigned to a guest instance. If more than one single vGPU type is provided, then for each vGPU type an additional section, [vgpu_\$(VGPU_TYPE)], must be added to the configuration file. Each section then must be configured with a single configuration option, device_addresses, which should be a list of PCI addresses corresponding to the physical GPU(s) to assign to this type. If one or more sections are missing (meaning that a specific type is not wanted to use for at least one physical GPU) or if no device addresses are provided, then Nova will only use the first type that was provided by [devices]/enabled_vgpu_types. If the same PCI address is provided for two different types, nova-compute will return an InvalidLibvirtGPUConfig exception at restart. An example is as the following <pre>[devices] enabled_vgpu_types = nvidia-35, nvidia-36</pre> <pre>[vgpu_nvidia-35] device_addresses = 0000:84:00.0,0000:85:00.0</pre> <pre>[vgpu_nvidia-36] device_addresses = 0000:86:00.0</pre>

9.1.15. ephemeral_storage_encryption

The following table outlines the options available under the **[ephemeral_storage_encryption]** group in the **/etc/nova/nova.conf** file.

Table 9.14. ephemeral_storage_encryption

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
cipher = aes-xts-plain64	string value	Cipher-mode string to be used. The cipher and mode to be used to encrypt ephemeral storage. The set of cipher-mode combinations available depends on kernel support. According to the dm-crypt documentation, the cipher is expected to be in the format: "<cipher>-<chainmode>-<ivmode>". Possible values: Any crypto option listed in /proc/crypto.
enabled = False	boolean value	Enables/disables LVM ephemeral storage encryption.
key_size = 512	integer value	Encryption key length in bits. 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.

9.1.16. filter_scheduler

The following table outlines the options available under the **[filter_scheduler]** group in the **/etc/nova/nova.conf** file.

Table 9.15. filter_scheduler

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
aggregate_image_properties_isolation_namespace = None	string value	Image property namespace for use in the host aggregate. Images and hosts can be configured so that certain images can only be scheduled to hosts in a particular aggregate. This is done with metadata values set on the host aggregate that are identified by beginning with the value of this option. If the host is part of an aggregate with such a metadata key, the image in the request spec must have the value of that metadata in its properties in order for the scheduler to consider the host as acceptable. Note that this setting only affects scheduling if the AggregateImagePropertiesIsolation filter is enabled. Possible values: • A string, where the string corresponds to an image property namespace Related options: • [filter_scheduler] aggregate_image_properties_isolation_separator

Configuration option = Default value	Type	Description
aggregate_image_properties_isolation_separator = .	string value	Separator character(s) for image property namespace and name. When using the <code>aggregate_image_properties_isolation</code> filter, the relevant metadata keys are prefixed with the namespace defined in the <code>aggregate_image_properties_isolation_namespace</code> configuration option plus a separator. This option defines the separator to be used. Note that this setting only affects scheduling if the <code>AggregateImagePropertiesIsolation</code> filter is enabled. Possible values: • A string, where the string corresponds to an image property namespace separator character Related options: • <code>[filter_scheduler] aggregate_image_properties_isolation_namespace</code>
available_filters = <code>['nova.scheduler.filters.all_filters']</code>	multi valued	Filters that the scheduler can use. An unordered list of the filter classes the nova scheduler may apply. Only the filters specified in the <code>[filter_scheduler] enabled_filters</code> option will be used, but any filter appearing in that option must also be included in this list. By default, this is set to all filters that are included with nova. Possible values: • A list of zero or more strings, where each string corresponds to the name of a filter that may be used for selecting a host Related options: • <code>[filter_scheduler] enabled_filters</code>

Configuration option = Default value	Type	Description
build_failure_weight_multiplier = 1000000.0	floating point value	Multiplier used for weighing hosts that have had recent build failures. This option determines how much weight is placed on a compute node with recent build failures. Build failures may indicate a failing, misconfigured, or otherwise ailing compute node, and avoiding it during scheduling may be beneficial. The weight is inversely proportional to the number of recent build failures the compute node has experienced. This value should be set to some high value to offset weight given by other enabled weighers due to available resources. To disable weighing compute hosts by the number of recent failures, set this to zero. Note that this setting only affects scheduling if the BuildFailureWeigher weigher is enabled. Possible values: • An integer or float value, where the value corresponds to the multiplier ratio for this weigher. Related options: • [compute] consecutive_build_service_disable_threshold - Must be nonzero for a compute to report data considered by this weigher. • [filter_scheduler] weight_classes
cpu_weight_multiplier = 1.0	floating point value	CPU weight multiplier ratio. Multiplier used for weighting free vCPUs. Negative numbers indicate stacking rather than spreading. Note that this setting only affects scheduling if the CPUWeigher weigher is enabled. Possible values: • An integer or float value, where the value corresponds to the multiplier ratio for this weigher. Related options: • [filter_scheduler] weight_classes

Configuration option = Default value	Type	Description
cross_cell_move_weight_multiplier = 1000000.0	floating point value	Multiplier used for weighing hosts during a cross-cell move. This option determines how much weight is placed on a host which is within the same source cell when moving a server, for example during cross-cell resize. By default, when moving an instance, the scheduler will prefer hosts within the same cell since cross-cell move operations can be slower and riskier due to the complicated nature of cross-cell migrations. Note that this setting only affects scheduling if the CrossCellWeigher weigher is enabled. If your cloud is not configured to support cross-cell migrations, then this option has no effect. The value of this configuration option can be overridden per host aggregate by setting the aggregate metadata key with the same name (cross_cell_move_weight_multiplier). Possible values: - An integer or float value, where the value corresponds to the multiplier ratio for this weigher. Positive values mean the weigher will prefer hosts within the same cell in which the instance is currently running. Negative values mean the weigher will prefer hosts in other cells from which the instance is currently running. Related options: [filter_scheduler] weight_classes
disk_weight_multiplier = 1.0	floating point value	Disk weight multiplier ratio. Multiplier used for weighing free disk space. Negative numbers mean to stack vs spread. Note that this setting only affects scheduling if the DiskWeigher weigher is enabled. Possible values: An integer or float value, where the value corresponds to the multiplier ratio for this weigher.

Configuration option = Default value	Type	Description
enabled_filters = ['AvailabilityZoneFilter', 'ComputeFilter', 'ComputeCapabilitiesFilter', 'ImagePropertiesFilter', 'ServerGroupAntiAffinityFilter', 'ServerGroupAffinityFilter']	list value	Filters that the scheduler will use. An ordered list of filter class names that will be used for filtering hosts. These filters will be applied in the order they are listed so place your most restrictive filters first to make the filtering process more efficient. All of the filters in this option must be present in the [scheduler_filter] available_filter option, or a SchedulerHostFilterNotFound exception will be raised. Possible values: - A list of zero or more strings, where each string corresponds to the name of a filter to be used for selecting a host Related options: [filter_scheduler] available_filters
host_subset_size = 1	integer value	Size of subset of best hosts selected by scheduler. New instances will be scheduled on a host chosen randomly from a subset of the N best hosts, where N is the value set by this option. Setting this to a value greater than 1 will reduce the chance that multiple scheduler processes handling similar requests will select the same host, creating a potential race condition. By selecting a host randomly from the N hosts that best fit the request, the chance of a conflict is reduced. However, the higher you set this value, the less optimal the chosen host may be for a given request. Possible values: An integer, where the integer corresponds to the size of a host subset.

Configuration option = Default value	Type	Description
image_properties_default_architecture = None	string value	The default architecture to be used when using the image properties filter. When using the ImagePropertiesFilter, it is possible that you want to define a default architecture to make the user experience easier and avoid having something like x86_64 images landing on AARCH64 compute nodes because the user did not specify the hw_architecture property in Glance. Possible values: • CPU Architectures such as x86_64, aarch64, s390x.
io_ops_weight_multiplier = -1.0	floating point value	IO operations weight multiplier ratio. This option determines how hosts with differing workloads are weighed. Negative values, such as the default, will result in the scheduler preferring hosts with lighter workloads whereas positive values will prefer hosts with heavier workloads. Another way to look at it is that positive values for this option will tend to schedule instances onto hosts that are already busy, while negative values will tend to distribute the workload across more hosts. The absolute value, whether positive or negative, controls how strong the io_ops weigher is relative to other weighers. Note that this setting only affects scheduling if the ioOpsWeigher weigher is enabled. Possible values: • An integer or float value, where the value corresponds to the multiplier ratio for this weigher. Related options: • [filter_scheduler] weight_classes

Configuration option = Default value	Type	Description
isolated_hosts = []	list value	List of hosts that can only run certain images. If there is a need to restrict some images to only run on certain designated hosts, list those host names here. Note that this setting only affects scheduling if the IsolatedHostsFilter filter is enabled. Possible values: • A list of strings, where each string corresponds to the name of a host Related options: • [filter_scheduler] isolated_images • [filter_scheduler] restrict_isolated_hosts_to_isolated_images
isolated_images = []	list value	List of UUIDs for images that can only be run on certain hosts. If there is a need to restrict some images to only run on certain designated hosts, list those image UUIDs here. Note that this setting only affects scheduling if the IsolatedHostsFilter filter is enabled. Possible values: • A list of UUID strings, where each string corresponds to the UUID of an image Related options: • [filter_scheduler] isolated_hosts • [filter_scheduler] restrict_isolated_hosts_to_isolated_images

Configuration option = Default value	Type	Description
max_instances_per_host = 50	integer value	Maximum number of instances that can exist on a host. If you need to limit the number of instances on any given host, set this option to the maximum number of instances you want to allow. The NumInstancesFilter and AggregateNumInstancesFilter will reject any host that has at least as many instances as this option's value. Note that this setting only affects scheduling if the NumInstancesFilter or AggregateNumInstancesFilter filter is enabled. Possible values: • An integer, where the integer corresponds to the max instances that can be scheduled on a host. Related options: • [filter_scheduler] enabled_filters
max_io_ops_per_host = 8	integer value	The number of instances that can be actively performing IO on a host. Instances performing IO includes those in the following states: build, resize, snapshot, migrate, rescue, unshelve. Note that this setting only affects scheduling if the IoOpsFilter filter is enabled. Possible values: • An integer, where the integer corresponds to the max number of instances that can be actively performing IO on any given host. Related options: • [filter_scheduler] enabled_filters

Configuration option = Default value	Type	Description
pci_weight_multiplier = 1.0	floating point value	PCI device affinity weight multiplier. The PCI device affinity weighter computes a weighting based on the number of PCI devices on the host and the number of PCI devices requested by the instance. Note that this setting only affects scheduling if the PCIWeigher weigher and NUMATopologyFilter filter are enabled. Possible values: • A positive integer or float value, where the value corresponds to the multiplier ratio for this weigher. Related options: • [filter_scheduler] weight_classes
ram_weight_multiplier = 1.0	floating point value	RAM weight multiplier ratio. This option determines how hosts with more or less available RAM are weighed. A positive value will result in the scheduler preferring hosts with more available RAM, and a negative number will result in the scheduler preferring hosts with less available RAM. Another way to look at it is that positive values for this option will tend to spread instances across many hosts, while negative values will tend to fill up (stack) hosts as much as possible before scheduling to a less-used host. The absolute value, whether positive or negative, controls how strong the RAM weigher is relative to other weighers. Note that this setting only affects scheduling if the RAMWeigher weigher is enabled. Possible values: • An integer or float value, where the value corresponds to the multiplier ratio for this weigher. Related options: • [filter_scheduler] weight_classes

Configuration option = Default value	Type	Description
restrict_isolated_hosts_to_isolated_images = True	boolean value	Prevent non-isolated images from being built on isolated hosts. Note that this setting only affects scheduling if the IsolatedHostsFilter filter is enabled. Even then, this option doesn't affect the behavior of requests for isolated images, which will always be restricted to isolated hosts. Related options: • [filter_scheduler] isolated_images • [filter_scheduler] isolated_hosts
shuffle_best_same_weighted_hosts = False	boolean value	Enable spreading the instances between hosts with the same best weight. Enabling it is beneficial for cases when [filter_scheduler] host_subset_size is 1 (default), but there is a large number of hosts with same maximal weight. This scenario is common in ironic deployments where there are typically many baremetal nodes with identical weights returned to the scheduler. In such case enabling this option will reduce contention and chances for rescheduling events. At the same time it will make the instance packing (even in unweighed case) less dense.
soft_affinity_weight_multiplier = 1.0	floating point value	Multiplier used for weighing hosts for group soft-affinity. Note that this setting only affects scheduling if the ServerGroupSoftAffinityWeigher weigher is enabled. Possible values: • A non-negative integer or float value, where the value corresponds to weight multiplier for hosts with group soft affinity. Related options: • [filter_scheduler] weight_classes

Configuration option = Default value	Type	Description
soft_anti_affinity_weight_multiplier = 1.0	floating point value	Multiplier used for weighing hosts for group soft-anti-affinity. Note that this setting only affects scheduling if the ServerGroupSoftAntiAffinityWeigher weigher is enabled. Possible values: • A non-negative integer or float value, where the value corresponds to weight multiplier for hosts with group soft anti-affinity. Related options: • [filter_scheduler] weight_classes

Configuration option = Default value	Type	Description
track_instance_changes = True	boolean value	Enable querying of individual hosts for instance information. The scheduler may need information about the instances on a host in order to evaluate its filters and weighers. The most common need for this information is for the (anti-)affinity filters, which need to choose a host based on the instances already running on a host. If the configured filters and weighers do not need this information, disabling this option will improve performance. It may also be disabled when the tracking overhead proves too heavy, although this will cause classes requiring host usage data to query the database on each request instead. a. note:: In a multi-cell (v2) setup where the cell MQ is separated from the top-level, computes cannot directly communicate with the scheduler. Thus, this option cannot be enabled in that scenario. See also the <code>[workarounds] disable_group_policy_check_upcall`</code> option. Related options: • [filter_scheduler] enabled_filters • [workarounds] disable_group_policy_check_upcall

Configuration option = Default value	Type	Description
weight_classes = ['nova.scheduler.weights. all_weighers']	list value	Weighers that the scheduler will use. Only hosts which pass the filters are weighed. The weight for any host starts at 0, and the weighers order these hosts by adding to or subtracting from the weight assigned by the previous weigher. Weights may become negative. An instance will be scheduled to one of the N most-weighted hosts, where N is [filter_scheduler] host_subset_size. By default, this is set to all weighers that are included with Nova. Possible values: • A list of zero or more strings, where each string corresponds to the name of a weigher that will be used for selecting a host

9.1.17. glance

The following table outlines the options available under the **[glance]** group in the **/etc/nova/nova.conf** file.

Table 9.16. glance

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
api_servers = None	list value	List of glance api servers endpoints available to nova. https is used for ssl-based glance api servers.  NOTE The preferred mechanism for endpoint discovery is via keystoneauth1 loading options. Only use api_servers if you need multiple endpoints and are unable to use a load balancer for some reason. Possible values: • A list of any fully qualified url of the form "scheme://hostname:port[/path]" (i.e. "http://10.0.1.0:9292" or "https://my.glance.server/image"). Deprecated since: 21.0.0 Reason: Support for image service configuration via standard keystoneauth1 Adapter options was added in the 17.0.0 Queens release. The api_servers option was retained temporarily to allow consumers time to cut over to a real load balancing solution.
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
debug = False	boolean value	Enable or disable debug logging with glanceclient.

Configuration option = Default value	Type	Description
default_trusted_certificate_ids = []	list value	List of certificate IDs for certificates that should be trusted. May be used as a default list of trusted certificate IDs for certificate validation. The value of this option will be ignored if the user provides a list of trusted certificate IDs with an instance API request. The value of this option will be persisted with the instance data if signature verification and certificate validation are enabled and if the user did not provide an alternative list. If left empty when certificate validation is enabled the user must provide a list of trusted certificate IDs otherwise certificate validation will fail. Related options: The value of this option may be used if both <code>verify_glance_signatures</code> and <code>enable_certificate_validation</code> are enabled.
enable_certificate_validation = False	boolean value	Enable certificate validation for image signature verification. During image signature verification nova will first verify the validity of the image's signing certificate using the set of trusted certificates associated with the instance. If certificate validation fails, signature verification will not be performed and the instance will be placed into an error state. This provides end users with stronger assurances that the image data is unmodified and trustworthy. If left disabled, image signature verification can still occur but the end user will not have any assurance that the signing certificate used to generate the image signature is still trustworthy. Related options: - This option only takes effect if <code>verify_glance_signatures</code> is enabled. - The value of <code>default_trusted_certificate_ids</code> may be used when this option is enabled. Deprecated since: 16.0.0 Reason: This option is intended to ease the transition for deployments leveraging image signature verification. The intended state long-term is for signature verification and certificate validation to always happen together.

Configuration option = Default value	Type	Description
enable_rbd_download = False	boolean value	Enable download of Glance images directly via RBD. Allow compute hosts to quickly download and cache images locally directly from Ceph rather than slow downloads from the Glance API. This can reduce download time for images in the ten to hundreds of GBs from tens of minutes to tens of seconds, but requires a Ceph-based deployment and access from the compute nodes to Ceph. Related options: • [glance] rbd_user • [glance] rbd_connect_timeout • [glance] rbd_pool • [glance] rbd_ceph_conf
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version, min-version, and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
num_retries = 3	integer value	Enable glance operation retries. Specifies the number of retries when uploading / downloading an image to / from glance. 0 means no retries.
`rbd_ceph_conf = `	string value	Path to the ceph configuration file to use. Related options: • This option is only used if [glance] enable_rbd_download is set to True.
rbd_connect_timeout = 5	integer value	The RADOS client timeout in seconds when initially connecting to the cluster. Related options: • This option is only used if [glance] enable_rbd_download is set to True.

Configuration option = Default value	Type	Description
<code>`rbd_pool = `</code>	string value	The RADOS pool in which the Glance images are stored as rbd volumes. Related options: This option is only used if [glance] enable_rbd_download is set to True.
<code>`rbd_user = `</code>	string value	The RADOS client name for accessing Glance images stored as rbd volumes. Related options: This option is only used if [glance] enable_rbd_download is set to True.
region-name = None	string value	The default region_name for endpoint URL discovery.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = image	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrieable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
timeout = None	integer value	Timeout value for http requests
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.

Configuration option = Default value	Type	Description
verify_glance_signatures = False	boolean value	Enable image signature verification. nova uses the image signature metadata from glance and verifies the signature of a signed image while downloading that image. If the image signature cannot be verified or if the image signature metadata is either incomplete or unavailable, then nova will not boot the image and instead will place the instance into an error state. This provides end users with stronger assurances of the integrity of the image data they are using to create servers. Related options: • The options in the key_manager group, as the key_manager is used for the signature validation. • Both enable_certificate_validation and default_trusted_certificate_ids below depend on this option being enabled.

9.1.18. guestfs

The following table outlines the options available under the **[guestfs]** group in the `/etc/nova/nova.conf` file.

Table 9.17. guestfs

Configuration option = Default value	Type	Description
debug = False	boolean value	Enable/disables guestfs logging. This configures guestfs to debug messages and push them to OpenStack logging system. When set to True, it traces libguestfs API calls and enable verbose debug messages. In order to use the above feature, "libguestfs" package must be installed. Related options: Since libguestfs access and modifies VM's managed by libvirt, below options should be set to give access to those VM's. • libvirt.inject_key • libvirt.inject_partition • libvirt.inject_password

9.1.19. healthcheck

The following table outlines the options available under the **[healthcheck]** group in the **/etc/nova/nova.conf** file.

Table 9.18. healthcheck

Configuration option = Default value	Type	Description
backends = []	list value	Additional backends that can perform health checks and report that information back as part of a request.
detailed = False	boolean value	Show more detailed information as part of the response. Security note: Enabling this option may expose sensitive details about the service being monitored. Be sure to verify that it will not violate your security policies.
disable_by_file_path = None	string value	Check the presence of a file to determine if an application is running on a port. Used by DisableByFileHealthcheck plugin.
disable_by_file_paths = []	list value	Check the presence of a file based on a port to determine if an application is running on a port. Expects a "port:path" list of strings. Used by DisableByFilesPortsHealthcheck plugin.
path = /healthcheck	string value	The path to respond to healthcheck requests on.

9.1.20. hyperv

The following table outlines the options available under the **[hyperv]** group in the **/etc/nova/nova.conf** file.

Table 9.19. hyperv

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
config_drive_cdrom = False	boolean value	Mount config drive as a CD drive. OpenStack can be configured to write instance metadata to a config drive, which is then attached to the instance before it boots. The config drive can be attached as a disk drive (default) or as a CD drive. Related options: • This option is meaningful with force_config_drive option set to True or when the REST API call to create an instance will have --config-drive=True flag. • config_drive_format option must be set to iso9660 in order to use CD drive as the config drive image. • To use config drive with Hyper-V, you must set the mkisofs_cmd value to the full path to an mkisofs.exe installation. Additionally, you must set the qemu_img_cmd value to the full path to an qemu-img command installation. • You can configure the Compute service to always create a configuration drive by setting the force_config_drive option to True.
config_drive_inject_password = False	boolean value	Inject password to config drive. When enabled, the admin password will be available from the config drive image. Related options: • This option is meaningful when used with other options that enable config drive usage with Hyper-V, such as force_config_drive.

Configuration option = Default value	Type	Description
dynamic_memory_ratio = 1.0	floating point value	Dynamic memory ratio 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. Possible values: 1.0: Disables dynamic memory allocation (Default). - Float values greater than 1.0: Enables allocation of total implied RAM divided by this value for startup.
enable_instance_metrics_collection = False	boolean value	Enable instance metrics collection Enables metrics collections for an instance by using Hyper-V's metric APIs. Collected data can be retrieved by other apps and services, e.g.: Ceilometer.
enable_remotefx = False	boolean value	Enable RemoteFX feature This requires at least one DirectX 11 capable graphics adapter for Windows / Hyper-V Server 2012 R2 or newer and RDS-Virtualization feature has to be enabled. Instances with RemoteFX can be requested with the following flavor extra specs: os:resolution. Guest VM screen resolution size. Acceptable values 1024x768, 1280x1024, 1600x1200, 1920x1200, 2560x1600, 3840x2160 3840x2160 is only available on Windows / Hyper-V Server 2016. os:monitors. Guest VM number of monitors. Acceptable values [1, 4] - Windows / Hyper-V Server 2012 R2 [1, 8] - Windows / Hyper-V Server 2016 os:vram. Guest VM VRAM amount. Only available on Windows / Hyper-V Server 2016. Acceptable values:: 64, 128, 256, 512, 1024

Configuration option = Default value	Type	Description
<code>`instances_path_share = `</code>	string value	Instances path share The name of a Windows share 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 (hidden network share) will be used, looking for the same "instances_path" used locally. Possible values: • "": An administrative share will be used (Default). • Name of a Windows share. Related options: • "instances_path": The directory which will be used if this option here is left blank.
<code>iscsi_initiator_list = []</code>	list value	List of iSCSI initiators that will be used for establishing iSCSI sessions. If none are specified, the Microsoft iSCSI initiator service will choose the initiator.
<code>limit_cpu_features = False</code>	boolean value	Limit CPU features This flag is needed to support live migration to hosts with different CPU features and checked during instance creation in order to limit the CPU features used by the instance.
<code>mounted_disk_query_retry_count = 10</code>	integer value	Mounted disk query retry count The number of times to retry checking for a mounted disk. The query runs until the device can be found or the retry count is reached. Possible values: • Positive integer values. Values greater than 1 is recommended (Default: 10). Related options: • Time interval between disk mount retries is declared with "mounted_disk_query_retry_interval" option.

Configuration option = Default value	Type	Description
mounted_disk_query_retry_interval = 5	integer value	Mounted disk query retry interval Interval between checks for a mounted disk, in seconds. Possible values: - Time in seconds (Default: 5). Related options: - This option is meaningful when the <code>mounted_disk_query_retry_count</code> is greater than 1. - The retry loop runs with <code>mounted_disk_query_retry_count</code> and <code>mounted_disk_query_retry_interval</code> configuration options.
power_state_check_timeframe = 60	integer value	Power state check timeframe The timeframe to be checked for instance power state changes. This option is used to fetch the state of the instance from Hyper-V through the WMI interface, within the specified timeframe. Possible values: Timeframe in seconds (Default: 60).
power_state_event_polling_interval = 2	integer value	Power state event polling interval Instance power state change event polling frequency. Sets the listener interval for power state events to the given value. This option enhances the internal lifecycle notifications of instances that reboot themselves. It is unlikely that an operator has to change this value. Possible values: Time in seconds (Default: 2).

Configuration option = Default value	Type	Description
qemu_img_cmd = qemu-img.exe	string value	qemu-img command qemu-img is required for some of the image related operations like converting between different image types. You can get it from here: (http://qemu.weilnetz.de/) or you can install the Cloudbase OpenStack Hyper-V Compute Driver (https://cloudbase.it/openstack-hyperv-driver/) which automatically sets the proper path for this config option. You can either give the full path of qemu-img.exe or set its path in the PATH environment variable and leave this option to the default value. Possible values: ● Name of the qemu-img executable, in case it is in the same directory as the nova-compute service or its path is in the PATH environment variable (Default). ● Path of qemu-img command (DRIVELETTER:\PATH\TO\QEMU-IMG\COMMAND). Related options: ● If the config_drive_cdrom option is False, qemu-img will be used to convert the ISO to a VHD, otherwise the config drive will remain an ISO. To use config drive with Hyper-V, you must set the mkisofs_cmd value to the full path to an mkisofs.exe installation.
use_multipath_io = False	boolean value	Use multipath connections when attaching iSCSI or FC disks. This requires the Multipath IO Windows feature to be enabled. MPIO must be configured to claim such devices.

Configuration option = Default value	Type	Description
volume_attach_retry_count = 10	integer value	Volume attach retry count The number of times to retry attaching a volume. Volume attachment is retried until success or the given retry count is reached. Possible values: • Positive integer values (Default: 10). Related options: • Time interval between attachment attempts is declared with <code>volume_attach_retry_interval</code> option.
volume_attach_retry_interval = 5	integer value	Volume attach retry interval Interval between volume attachment attempts, in seconds. Possible values: • Time in seconds (Default: 5). Related options: • This options is meaningful when <code>volume_attach_retry_count</code> is greater than 1. • The retry loop runs with <code>volume_attach_retry_count</code> and <code>volume_attach_retry_interval</code> configuration options.

Configuration option = Default value	Type	Description
vswitch_name = None	string value	External virtual switch name The Hyper-V Virtual Switch is a software-based layer-2 Ethernet network switch that is available with the installation of the Hyper-V server role. The switch includes programmatically managed and extensible capabilities to connect virtual machines to both virtual networks and the physical network. In addition, Hyper-V Virtual Switch provides policy enforcement for security, isolation, and service levels. The vSwitch represented by this config option must be an external one (not internal or private). Possible values: • If not provided, the first of a list of available vswitches is used. This list is queried using WQL. • Virtual switch name.
wait_soft_reboot_seconds = 60	integer value	Wait soft reboot seconds 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. Possible values: • Time in seconds (Default: 60).

9.1.21. image_cache

The following table outlines the options available under the **[image_cache]** group in the **/etc/nova/nova.conf** file.

Table 9.20. image_cache

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
manager_interval = 2400	integer value	Number of seconds to wait between runs of the image cache manager. Note that when using shared storage for the [DEFAULT]/instances_path configuration option across multiple nova-compute services, this periodic could process a large number of instances. Similarly, using a compute driver that manages a cluster (like <code>vmwareapi.VMwareVCDriver</code>) could result in processing a large number of instances. Therefore you may need to adjust the time interval for the anticipated load, or only run on one nova-compute service within a shared storage aggregate. Possible values: ● 0: run at the default interval of 60 seconds (not recommended) ● -1: disable ● Any other value Related options: ● [DEFAULT]/compute_driver ● [DEFAULT]/instances_path
precache_concurrency = 1	integer value	Maximum number of compute hosts to trigger image precaching in parallel. When an image precache request is made, compute nodes will be contacted to initiate the download. This number constrains the number of those that will happen in parallel. Higher numbers will cause more computes to work in parallel and may result in reduced time to complete the operation, but may also DDoS the image service. Lower numbers will result in more sequential operation, lower image service load, but likely longer runtime to completion.
remove_unused_base_images = True	boolean value	Should unused base images be removed?
remove_unused_original_minimum_age_seconds = 86400	integer value	Unused unresized base images younger than this will not be removed.

Configuration option = Default value	Type	Description
remove_unused_resized_minimum_age_seconds = 3600	integer value	Unused resized base images younger than this will not be removed.
subdirectory_name = _base	string value	Location of cached images. This is NOT the full path - just a folder name relative to <i>\$instances_path</i> . For per-compute-host cached images, set to <i>base\$my_ip</i>

9.1.22. ironic

The following table outlines the options available under the **[ironic]** group in the */etc/nova/nova.conf* file.

Table 9.21. ironic

Configuration option = Default value	Type	Description
api_max_retries = 60	integer value	The number of times to retry when a request conflicts. If set to 0, only try once, no retries. Related options: • <code>api_retry_interval</code>
api_retry_interval = 2	integer value	The number of seconds to wait before retrying the request. Related options: • <code>api_max_retries</code>
auth-url = None	string value	Authentication URL
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file

Configuration option = Default value	Type	Description
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
partition_key = None	string value	Case-insensitive key to limit the set of nodes that may be managed by this service to the set of nodes in Ironic which have a matching conductor_group property. If unset, all available nodes will be eligible to be managed by this service. Note that setting this to the empty string ("") will match the default conductor group, and is different than leaving the option unset.
password = None	string value	User's password
peer_list = []	list value	List of hostnames for all nova-compute services (including this host) with this partition_key config value. Nodes matching the partition_key value will be distributed between all services specified here. If partition_key is unset, this option is ignored.
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to

Configuration option = Default value	Type	Description
project-name = None	string value	Project name to scope to
region-name = None	string value	The default region_name for endpoint URL discovery.
serial_console_state_timeout = 10	integer value	Timeout (seconds) to wait for node serial console state changed. Set to 0 to disable timeout.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = baremetal	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrieable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
system-scope = None	string value	Scope for system operations
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User ID
username = None	string value	Username
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.

9.1.23. key_manager

The following table outlines the options available under the **[key_manager]** group in the **/etc/nova/nova.conf** file.

Table 9.22. key_manager

Configuration option = Default value	Type	Description
auth_type = None	string value	The type of authentication credential to create. Possible values are <i>token</i> , <i>password</i> , <i>keystone_token</i> , and <i>keystone_password</i> . Required if no context is passed to the credential factory.
auth_url = None	string value	Use this endpoint to connect to Keystone.
backend = barbican	string value	Specify the key manager implementation. Options are "barbican" and "vault". Default is "barbican". Will support the values earlier set using [key_manager]/api_class for some time.
domain_id = None	string value	Domain ID for domain scoping. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
domain_name = None	string value	Domain name for domain scoping. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
fixed_key = None	string value	Fixed key returned by key manager, specified in hex. Possible values: ● Empty string or a key in hex value
password = None	string value	Password for authentication. Required for <i>password</i> and <i>keystone_password</i> auth_type.
project_domain_id = None	string value	Project's domain ID for project. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
project_domain_name = None	string value	Project's domain name for project. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
project_id = None	string value	Project ID for project scoping. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
project_name = None	string value	Project name for project scoping. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
reauthenticate = True	boolean value	Allow fetching a new token if the current one is going to expire. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.

Configuration option = Default value	Type	Description
token = None	string value	Token for authentication. Required for <i>token</i> and <i>keystone_token</i> auth_type if no context is passed to the credential factory.
trust_id = None	string value	Trust ID for trust scoping. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
user_domain_id = None	string value	User's domain ID for authentication. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
user_domain_name = None	string value	User's domain name for authentication. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
user_id = None	string value	User ID for authentication. Optional for <i>keystone_token</i> and <i>keystone_password</i> auth_type.
username = None	string value	Username for authentication. Required for <i>password</i> auth_type. Optional for the <i>keystone_password</i> auth_type.

9.1.24. keystone

The following table outlines the options available under the **[keystone]** group in the `/etc/nova/nova.conf` file.

Table 9.23. keystone

Configuration option = Default value	Type	Description
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

Configuration option = Default value	Type	Description
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
region-name = None	string value	The default region_name for endpoint URL discovery.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = identity	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrieable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
timeout = None	integer value	Timeout value for http requests
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.

9.1.25. keystone_authtoken

The following table outlines the options available under the **[keystone_authtoken]** group in the **/etc/nova/nova.conf** file.

Table 9.24. keystone_authtoken

Configuration option = Default value	Type	Description
auth_section = None	string value	Config Section from which to load plugin specific options

Configuration option = Default value	Type	Description
auth_type = None	string value	Authentication type to load
auth_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint. This option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release. Deprecated since: Queens *Reason:*The <code>auth_uri</code> option is deprecated in favor of <code>www_authenticate_uri</code> and will be removed in the S release.
auth_version = None	string value	API version of the Identity API endpoint.
cache = None	string value	Request environment key where the Swift cache object is stored. When <code>auth_token</code> middleware is deployed with a Swift cache, use this option to have the middleware share a caching backend with swift. Otherwise, use the <code>memcached_servers</code> option instead.
cafile = None	string value	A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.
certfile = None	string value	Required if identity server requires client certificate
delay_auth_decision = False	boolean value	Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components.
enforce_token_bind = permissive	string value	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	Type	Description
http_connect_timeout = None	integer value	Request timeout value for communicating with Identity API server.
http_request_max_retries = 3	integer value	How many times are we trying to reconnect when communicating with Identity API Server.
include_service_catalog = True	boolean value	(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	boolean value	Verify HTTPS connections.
interface = internal	string value	Interface to use for the Identity API endpoint. Valid values are "public", "internal" (default) or "admin".
keyfile = None	string value	Required if identity server requires client certificate
memcache_pool_conn_get_timeout = 10	integer value	(Optional) Number of seconds that an operation will wait to get a memcached client connection from the pool.
memcache_pool_dead_retry = 300	integer value	(Optional) Number of seconds memcached server is considered dead before it is tried again.
memcache_pool_maxsize = 10	integer value	(Optional) Maximum total number of open connections to every memcached server.
memcache_pool_socket_timeout = 3	integer value	(Optional) Socket timeout in seconds for communicating with a memcached server.
memcache_pool_unused_timeout = 60	integer value	(Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.
memcache_secret_key = None	string value	(Optional, mandatory if memcache_security_strategy is defined) This string is used for key derivation.
memcache_security_strategy = None	string value	(Optional) If defined, indicate whether token data should be authenticated or authenticated and encrypted. If MAC, token data is authenticated (with HMAC) in the cache. If ENCRYPT, token data is encrypted and authenticated in the cache. If the value is not one of these options or empty, auth_token will raise an exception on initialization.

Configuration option = Default value	Type	Description
memcache_use_advanced_pool = False	boolean value	(Optional) Use the advanced (eventlet safe) memcached client pool. The advanced pool will only work under python 2.x.
memcached_servers = None	list value	Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.
region_name = None	string value	The region in which the identity server can be found.
service_token_roles = ['service']	list value	A choice of roles that must be present in a service token. Service tokens are allowed to request that an expired token can be used and so this check should tightly control that only actual services should be sending this token. Roles here are applied as an ANY check so any role in this list must be present. For backwards compatibility reasons this currently only affects the allow_expired check.
service_token_roles_required = False	boolean value	For backwards compatibility reasons we must let valid service tokens pass that don't pass the service_token_roles check as valid. Setting this true will become the default in a future release and should be enabled if possible.
service_type = None	string value	The name or type of the service as it appears in the service catalog. This is used to validate tokens that have restricted access rules.
token_cache_time = 300	integer value	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.
www_authenticate_uri = None	string value	Complete "public" Identity API endpoint. This endpoint should not be an "admin" endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should not be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint.

9.1.26. libvirt

The following table outlines the options available under the **[libvirt]** group in the **/etc/nova/nova.conf** file.

Table 9.25. libvirt

Configuration option = Default value	Type	Description
<code>connection_uri =</code>	string value	Overrides the default libvirt URI of the chosen virtualization type. If set, Nova will use this URI to connect to libvirt. Possible values: - An URI like qemu:///system. This is only necessary if the URI differs to the commonly known URIs for the chosen virtualization type. Related options: virt_type: Influences what is used as default value here.
cpu_mode = None	string value	Is used to set the CPU mode an instance should have. If virt_type="kvm&verbar;qemu", it will default to host-model, otherwise it will default to none. Related options: cpu_models: This should be set ONLY when cpu_mode is set to custom. Otherwise, it would result in an error and the instance launch will fail.
cpu_model_extra_flags = []	list value	Enable or disable guest CPU flags. To explicitly enable or disable CPU flags, use the +flag or -flag notation – the + sign will enable the CPU flag for the guest, while a - sign will disable it. If neither + nor - is specified, the flag will be enabled, which is the default behaviour. For example, if you specify the following (assuming the said CPU model and features are supported by the host hardware and software):: <pre>[libvirt] cpu_mode = custom cpu_models = Cascadelake-Server cpu_model_extra_flags = -hle, -rtm, +ssbd, mtrr</pre>

Configuration option = Default value	Type	Description
		Nova will disable the hle and rtm flags for the guest; and it will enable ssbd and mttr (because it was specified with neither + nor - prefix). The CPU flags are case-insensitive. In the following example, the pdpe1gb flag will be disabled for the guest; vmx and pcid flags will be enabled:: <pre>[libvirt] cpu_mode = custom cpu_models = Haswell-noTSX-IBRS cpu_model_extra_flags = -PDPE1GB, +VMX, pcid</pre> Specifying extra CPU flags is valid in combination with all the three possible values of cpu_mode config attribute: custom (this also requires an explicit CPU model to be specified via the cpu_models config attribute), host-model, or host-passthrough. There can be scenarios where you may need to configure extra CPU flags even for host-passthrough CPU mode, because sometimes QEMU may disable certain CPU features. An example of this is Intel's "invts" (Invariable Time Stamp Counter) CPU flag – if you need to expose this flag to a Nova instance, you need to explicitly enable it. The possible values for cpu_model_extra_flags depends on the CPU model in use. Refer to /usr/share/libvirt/cpu_map/*.xml for possible CPU feature flags for a given CPU model. A special note on a particular CPU flag: pcid (an Intel processor feature that alleviates guest performance degradation as a result of applying the <i>Meltdown</i> CVE fixes). When configuring this flag with the custom CPU mode, not all CPU models (as defined by QEMU and libvirt) need it: • The only virtual CPU models that include the pcid capability are Intel "Haswell", "Broadwell", and "Skylake" variants. • The libvirt / QEMU CPU models "Nehalem", "Westmere", "SandyBridge", and "IvyBridge" will <i>not</i> expose the pcid capability by default, even if the host CPUs by the same name include it. I.e. <i>PCID</i> needs to be explicitly specified when using the said virtual CPU models. The libvirt driver's default CPU mode, host-model, will do the right thing with respect to handling <i>PCID</i> CPU flag for the guest – assuming you are running updated processor microcode, host and guest kernel, libvirt, and QEMU. The other mode, host-

Configuration option = Default value	Type	Description
		passthrough, checks if <i>PCID</i> is available in the hardware, and if so directly passes it through to the Nova guests. Thus, in context of <i>PCID</i>, with either of these CPU modes (host-model or host-passthrough), there is no need to use the cpu_model_extra_flags. Related options: • cpu_mode • cpu_models
cpu_models = []	list value	An ordered list of CPU models the host supports. It is expected that the list is ordered so that the more common and less advanced CPU models are listed earlier. Here is an example: SandyBridge,IvyBridge,Haswell,Broadwell, the latter CPU model's features is richer than the previous CPU model. Possible values: • The named CPU models can be found via virsh cpu-models ARCH, where ARCH is your host architecture. Related options: • cpu_mode: This should be set to custom ONLY when you want to configure (via cpu_models) a specific named CPU model. Otherwise, it would result in an error and the instance launch will fail. • virt_type: Only the virtualization types kvm and qemu use this. a. note:: Be careful to only specify models which can be fully supported in hardware.
device_detach_attempts = 8	integer value	Maximum number of attempts the driver tries to detach a device in libvirt. Related options: • :oslo.config.option:libvirt.device_detach_timeout

Configuration option = Default value	Type	Description
device_detach_timeout = 20	integer value	Maximum number of seconds the driver waits for the success or the failure event from libvirt for a given device detach attempt before it re-trigger the detach. Related options: • :oslo.config.option:libvirt.device_detach_attempts
disk_cachemodes = []	list value	Specific cache modes to use for different disk types. For example: file=directsync,block=none,network=writeback For local or direct-attached storage, it is recommended that you use writethrough (default) mode, as it ensures data integrity and has acceptable I/O performance for applications running in the guest, especially for read operations. However, caching mode none is recommended for remote NFS storage, because direct I/O operations (O_DIRECT) perform better than synchronous I/O operations (with O_SYNC). Caching mode none effectively turns all guest I/O operations into direct I/O operations on the host, which is the NFS client in this environment. Possible cache modes: • default: "It Depends" – For Nova-managed disks, none, if the host file system is capable of Linux's <i>O_DIRECT</i> semantics; otherwise writeback. For volume drivers, the default is driver-dependent: none for everything except for SMBFS and Virtuzzo (which use writeback). • none: With caching mode set to none, the host page cache is disabled, but the disk write cache is enabled for the guest. In this mode, the write performance in the guest is optimal because write operations bypass the host page cache and go directly to the disk write cache. If the disk write cache is battery-backed, or if the applications or storage stack in the guest transfer data properly (either through fsync operations or file system barriers), then data integrity can be ensured. However, because the host page cache is disabled, the read performance in the guest would not be as good as in the modes where the host page cache is enabled, such as writethrough mode. Shareable disk devices, like for a

Configuration option = Default value	Type	Description
		multi-attachable block storage volume, will have their cache mode set to <i>none</i> regardless of configuration. ● writethrough: With caching set to writethrough mode, the host page cache is enabled, but the disk write cache is disabled for the guest. Consequently, this caching mode ensures data integrity even if the applications and storage stack in the guest do not transfer data to permanent storage properly (either through fsync operations or file system barriers). Because the host page cache is enabled in this mode, the read performance for applications running in the guest is generally better. However, the write performance might be reduced because the disk write cache is disabled. ● writeback: With caching set to writeback mode, both the host page cache and the disk write cache are enabled for the guest. Because of this, the I/O performance for applications running in the guest is good, but the data is not protected in a power failure. As a result, this caching mode is recommended only for temporary data where potential data loss is not a concern. NOTE: Certain backend disk mechanisms may provide safe writeback cache semantics. Specifically those that bypass the host page cache, such as QEMU's integrated RBD driver. Ceph documentation recommends setting this to writeback for maximum performance while maintaining data safety. ● directsync: Like "writethrough", but it bypasses the host page cache. ● unsafe: Caching mode of unsafe ignores cache transfer operations completely. As its name implies, this caching mode should be used only for temporary data where data loss is not a concern. This mode can be useful for speeding up guest installations, but you should switch to another caching mode in production environments.

Configuration option = Default value	Type	Description
disk_prefix = None	string value	Override the default disk prefix for the devices attached to an instance. If set, this is used to identify a free disk device name for a bus. Possible values: - Any prefix which will result in a valid disk device name like <i>sda</i> or <i>hda</i> for example. This is only necessary if the device names differ to the commonly known device name prefixes for a virtualization type such as: <i>sd</i>, <i>xvd</i>, <i>uud</i>, <i>vd</i>. Related options: virt_type: Influences which device type is used, which determines the default disk prefix.
enabled_perf_events = []	list value	Performance events to monitor and collect statistics for. This will allow you to specify a list of events to monitor low-level performance of guests, and collect related statistics via the libvirt driver, which in turn uses the Linux kernel's perf infrastructure. With this config attribute set, Nova will generate libvirt guest XML to monitor the specified events. For example, to monitor the count of CPU cycles (total/elapsed) and the count of cache misses, enable them as follows:: <pre>[libvirt] enabled_perf_events = cpu_clock, cache_misses</pre> Possible values: A string list. The list of supported events can be found here here. Note that Intel CMT events - <code>cmt</code>, <code>mbmbt</code> and <code>mbml</code> - are unsupported by recent Linux kernel versions (4.14+) and will be ignored by nova. — https://libvirt.org/formatdomain.html#elementsPerf.

Configuration option = Default value	Type	Description
file_backed_memory = 0	integer value	Available capacity in MiB for file-backed memory. Set to 0 to disable file-backed memory. When enabled, instances will create memory files in the directory specified in <code>/etc/libvirt/qemu.conf</code>'s memory_backing_dir option. The default location is <code>/var/lib/libvirt/qemu/ram</code>. When enabled, the value defined for this option is reported as the node memory capacity. Compute node system memory will be used as a cache for file-backed memory, via the kernel's pagecache mechanism. a. note:: This feature is not compatible with hugepages. b. note:: This feature is not compatible with memory overcommit. Related options: ● virt_type must be set to kvm or qemu. ● ram_allocation_ratio must be set to 1.0.
gid_maps = []	list value	List of guid targets and ranges. Syntax is guest-gid:host-gid:count. Maximum of 5 allowed.
hw_disk_discard = None	string value	Discard option for nova managed disks. Requires: ● Libvirt >= 1.0.6 ● Qemu >= 1.5 (raw format) ● Qemu >= 1.6 (qcow2 format)
hw_machine_type = None	list value	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 <code>:command:virsh capabilities</code> command. The format of the value for this config option is host-arch=machine-type. For example: x86_64=machinetype1,armv7l=machinetype2.
<code>`images_rbd_ceph_conf = `</code>	string value	Path to the ceph configuration file to use

Configuration option = Default value	Type	Description
images_rbd_glance_copy_poll_interval = 15	integer value	The interval in seconds with which to poll Glance after asking for it to copy an image to the local rbd store. This affects how often we ask Glance to report on copy completion, and thus should be short enough that we notice quickly, but not too aggressive that we generate undue load on the Glance server. Related options: ● <code>images_type</code> - must be set to rbd ● <code>images_rbd_glance_store_name</code> - must be set to a store name
images_rbd_glance_copy_timeout = 600	integer value	The overall maximum time we will wait for Glance to complete an image copy to our local rbd store. This should be long enough to allow large images to be copied over the network link between our local store and the one where images typically reside. The downside of setting this too long is just to catch the case where the image copy is stalled or proceeding too slowly to be useful. Actual errors will be reported by Glance and noticed according to the poll interval. Related options: * <code>images_type</code> - must be set to rbd * <code>images_rbd_glance_store_name</code> - must be set to a store name * <code>images_rbd_glance_copy_poll_interval</code> - controls the failure time-to-notice
images_rbd_glance_store_name = `	string value	The name of the Glance store that represents the rbd cluster in use by this node. If set, this will allow Nova to request that Glance copy an image from an existing non-local store into the one named by this option before booting so that proper Copy-on-Write behavior is maintained. Related options: ● <code>images_type</code> - must be set to rbd ● <code>images_rbd_glance_copy_poll_interval</code> - controls the status poll frequency ● <code>images_rbd_glance_copy_timeout</code> - controls the overall copy timeout
images_rbd_pool = rbd	string value	The RADOS pool in which rbd volumes are stored

Configuration option = Default value	Type	Description
images_type = default	string value	VM Images format. If default is specified, then use_cow_images flag is used instead of this one. Related options: ● compute.use_cow_images ● images_volume_group ● [workarounds]/ensure_libvirt_rbd_instance_dir_cleanup ● compute.force_raw_images
images_volume_group = None	string value	LVM Volume Group that is used for VM images, when you specify images_type=lv Related options: ● images_type
inject_key = False	boolean value	Allow the injection of an SSH key at boot time. There is no agent needed within the image to do this. If libguestfs is available on the host, it will be used. Otherwise nbd is used. The file system of the image will be mounted and the SSH key, which is provided in the REST API call will be injected as SSH key for the root user and appended to the authorized_keys of that user. The SELinux context will be set if necessary. Be aware that the injection is not possible when the instance gets launched from a volume. This config option will enable directly modifying the instance disk and does not affect what cloud-init may do using data from config_drive option or the metadata service. Linux distribution guest only. Related options: ● inject_partition: That option will decide about the discovery and usage of the file system. It also can disable the injection at all.

Configuration option = Default value	Type	Description
inject_partition = -2	integer value	Determines how the file system is chosen to inject data into it. libguestfs is used to inject data. If libguestfs is not able to determine the root partition (because there are more or less than one root partition) or cannot mount the file system it will result in an error and the instance won't boot. Possible values: • -2 ⇒ disable the injection of data. • -1 ⇒ find the root partition with the file system to mount with libguestfs • 0 ⇒ The image is not partitioned • >0 ⇒ The number of the partition to use for the injection Linux distribution guest only. Related options: • inject_key: If this option allows the injection of a SSH key it depends on value greater or equal to -1 for inject_partition. • inject_password: If this option allows the injection of an admin password it depends on value greater or equal to -1 for inject_partition. • [guestfs]/debug You can enable the debug log level of libguestfs with this config option. A more verbose output will help in debugging issues. • virt_type: If you use lxc as virt_type it will be treated as a single partition image

Configuration option = Default value	Type	Description
inject_password = False	boolean value	Allow the injection of an admin password for instance only at create and rebuild process. There is no agent needed within the image to do this. If libguestfs is available on the host, it will be used. Otherwise nbd is used. The file system of the image will be mounted and the admin password, which is provided in the REST API call will be injected as password for the root user. If no root user is available, the instance won't be launched and an error is thrown. Be aware that the injection is not possible when the instance gets launched from a volume. Linux distribution guest only. Possible values: • True: Allows the injection. • False: Disallows the injection. Any via the REST API provided admin password will be silently ignored. Related options: • inject_partition: That option will decide about the discovery and usage of the file system. It also can disable the injection at all.
iscsi_iface = None	string value	The iSCSI transport iface to use to connect to target in case offload support is desired. Default format is of the form <transport_name>.<hwaddress>, where <transport_name> is one of (be2iscsi, bnx2i, cxgb3i, cxgb4i, qla4xxx, ocs, tcp) and <hwaddress> is the MAC address of the interface and can be generated via the iscsiadm -m iface command. Do not confuse the iscsi_iface parameter to be provided here with the actual transport name.
iser_use_multipath = False	boolean value	Use multipath connection of the iSER volume. iSER volumes can be connected as multipath devices. This will provide high availability and fault tolerance.

Configuration option = Default value	Type	Description
live_migration_bandwidth = 0	integer value	Maximum bandwidth(in MiB/s) to be used during migration. If set to 0, the hypervisor will choose a suitable default. Some hypervisors do not support this feature and will return an error if bandwidth is not 0. Please refer to the libvirt documentation for further details.
live_migration_completion_timeout = 800	integer value	Time to wait, in seconds, for migration to successfully complete transferring data before aborting the operation. Value is per GiB of guest RAM + disk to be transferred, with lower bound of a minimum of 2 GiB. Should usually be larger than downtime delay * downtime steps. Set to 0 to disable timeouts. Related options: ● live_migration_downtime ● live_migration_downtime_steps ● live_migration_downtime_delay
live_migration_downtime = 500	integer value	Maximum permitted downtime, in milliseconds, for live migration switchover. Will be rounded up to a minimum of 100ms. You can increase this value if you want to allow live-migrations to complete faster, or avoid live-migration timeout errors by allowing the guest to be paused for longer during the live-migration switch over. Related options: ● live_migration_completion_timeout
live_migration_downtime_delay = 75	integer value	Time to wait, in seconds, between each step increase of the migration downtime. Minimum delay is 3 seconds. Value is per GiB of guest RAM + disk to be transferred, with lower bound of a minimum of 2 GiB per device.
live_migration_downtime_steps = 10	integer value	Number of incremental steps to reach max downtime value. Will be rounded up to a minimum of 3 steps.

Configuration option = Default value	Type	Description
live_migration_inbound_address = None	host address value	IP address used as the live migration address for this host. This option indicates the IP address which should be used as the target for live migration traffic when migrating to this hypervisor. This metadata is then used by the source of the live migration traffic to construct a migration URI. If this option is set to None, the hostname of the migration target compute node will be used. This option is useful in environments where the live-migration traffic can impact the network plane significantly. A separate network for live-migration traffic can then use this config option and avoids the impact on the management network.
live_migration_permit_auto_converge = False	boolean value	This option allows nova to start live migration with auto converge on. Auto converge throttles down CPU if a progress of on-going live migration is slow. Auto converge will only be used if this flag is set to True and post copy is not permitted or post copy is unavailable due to the version of libvirt and QEMU in use. Related options: ● live_migration_permit_post_copy

Configuration option = Default value	Type	Description
live_migration_permit_post_copy = False	boolean value	This option allows nova to switch an on-going live migration to post-copy mode, i.e., switch the active VM to the one on the destination node before the migration is complete, therefore ensuring an upper bound on the memory that needs to be transferred. Post-copy requires libvirt>=1.3.3 and QEMU>=2.5.0. When permitted, post-copy mode will be automatically activated if we reach the timeout defined by live_migration_completion_timeout and live_migration_timeout_action is set to <i>force_complete</i>. Note if you change to no timeout or choose to use <i>abort</i>, i.e. live_migration_completion_timeout = 0, then there will be no automatic switch to post-copy. The live-migration force complete API also uses post-copy when permitted. If post-copy mode is not available, force complete falls back to pausing the VM to ensure the live-migration operation will complete. When using post-copy mode, if the source and destination hosts lose network connectivity, the VM being live-migrated will need to be rebooted. For more details, please see the Administration guide. Related options: ● live_migration_permit_auto_converge ● live_migration_timeout_action

Configuration option = Default value	Type	Description
live_migration_scheme = None	string value	URI scheme for live migration used by the source of live migration traffic. Override the default libvirt live migration scheme (which is dependent on virt_type). If this option is set to None, nova will automatically choose a sensible default based on the hypervisor. It is not recommended that you change this unless you are very sure that hypervisor supports a particular scheme. Related options: ● virt_type: This option is meaningful only when virt_type is set to kvm or qemu. ● live_migration_uri: If live_migration_uri value is not None, the scheme used for live migration is taken from live_migration_uri instead.
live_migration_timeout_action = abort	string value	This option will be used to determine what action will be taken against a VM after live_migration_completion_timeout expires. By default, the live migrate operation will be aborted after completion timeout. If it is set to force_complete, the compute service will either pause the VM or trigger post-copy depending on if post copy is enabled and available (live_migration_permit_post_copy is set to True). Related options: ● live_migration_completion_timeout ● live_migration_permit_post_copy

Configuration option = Default value	Type	Description
live_migration_tunelled = False	boolean value	Enable tunnelled migration. This option enables the tunnelled migration feature, where migration data is transported over the libvirt connection. If enabled, we use the VIR_MIGRATE_TUNNELLED migration flag, avoiding the need to configure the network to allow direct hypervisor to hypervisor communication. If False, use the native transport. If not set, Nova will choose a sensible default based on, for example the availability of native encryption support in the hypervisor. Enabling this option will definitely impact performance massively. Note that this option is NOT compatible with use of block migration. Deprecated since: 23.0.0 Reason: The "tunnelled live migration" has two inherent limitations: it cannot handle live migration of disks in a non-shared storage setup; and it has a huge performance cost. Both these problems are solved by ``live_migration_with_native_tls`` (requires a pre-configured TLS environment), which is the recommended approach for securing all live migration streams.

Configuration option = Default value	Type	Description
live_migration_uri = None	string value	Live migration target URI used by the source of live migration traffic. Override the default libvirt live migration target URI (which is dependent on <code>virt_type</code>). Any included "%s" is replaced with the migration target hostname, or live_migration_inbound_addr if set. If this option is set to None (which is the default), Nova will automatically generate the live_migration_uri value based on only 4 supported virt_type in following list: • <i>kvm: qemu+tcp://%s/system</i> • <i>qemu: qemu+tcp://%s/system</i> • <i>parallels: parallels+tcp://%s/system</i> Related options: • live_migration_inbound_addr: If live_migration_inbound_addr value is not None and live_migration_tunneled is False, the ip/hostname address of target compute node is used instead of live_migration_uri as the uri for live migration. • live_migration_scheme: If live_migration_uri is not set, the scheme used for live migration is taken from live_migration_scheme instead. Deprecated since: 15.0.0 Reason: live_migration_uri is deprecated for removal in favor of two other options that allow to change live migration scheme and target URI: <code>live_migration_scheme` and  <code>live_migration_inbound_addr` respectively.</code></code>

Configuration option = Default value	Type	Description
live_migration_with_native_tls = False	boolean value	Use QEMU-native TLS encryption when live migrating. This option will allow both migration stream (guest RAM plus device state) and disk stream to be transported over native TLS, i.e. TLS support built into QEMU. Prerequisite: TLS environment is configured correctly on all relevant Compute nodes. This means, Certificate Authority (CA), server, client certificates, their corresponding keys, and their file permissions are in place, and are validated. Notes: • To have encryption for migration stream and disk stream (also called: "block migration"), live_migration_with_native_tls is the preferred config attribute instead of live_migration_tunneled. • The live_migration_tunneled will be deprecated in the long-term for two main reasons: (a) it incurs a huge performance penalty; and (b) it is not compatible with block migration. Therefore, if your compute nodes have at least libvirt 4.4.0 and QEMU 2.11.0, it is strongly recommended to use live_migration_with_native_tls. • The live_migration_tunneled and live_migration_with_native_tls should not be used at the same time. • Unlike live_migration_tunneled, the live_migration_with_native_tls is compatible with block migration. That is, with this option, NBD stream, over which disks are migrated to a target host, will be encrypted. Related options: live_migration_tunneled: This transports migration stream (but not disk stream) over libvirtd.
max_queues = None	integer value	The maximum number of virtio queue pairs that can be enabled when creating a multiqueue guest. The number of virtio queues allocated will be the lesser of the CPUs requested by the guest and the max value defined. By default, this value is set to none meaning the legacy limits based on the reported kernel major version will be used.

Configuration option = Default value	Type	Description
mem_stats_period_seconds = 10	integer value	A number of seconds to memory usage statistics period. Zero or negative value mean to disable memory usage statistics.
nfs_mount_options = None	string value	Mount options passed to the NFS client. See section of the <code>nfs</code> man page for details. Mount options controls the way the filesystem is mounted and how the NFS client behaves when accessing files on this mount point. Possible values: - Any string representing mount options separated by commas. - Example string: <code>vers=3,lookupcache=pos</code>
nfs_mount_point_base = \$state_path/mnt	string value	Directory where the NFS volume is mounted on the compute node. The default is <code>mnt</code> directory of the location where nova's Python module is installed. NFS provides shared storage for the OpenStack Block Storage service. Possible values: A string representing absolute path of mount point.
num_aoe_discover_tries = 3	integer value	Number of times to rediscover AoE target to find volume. Nova provides support for block storage attaching to hosts via AOE (ATA over Ethernet). This option allows the user to specify the maximum number of retry attempts that can be made to discover the AoE device.
num_iscsi_scan_tries = 5	integer value	Number of times to scan iSER target to find volume. iSER is a server network protocol that extends iSCSI protocol to use Remote Direct Memory Access (RDMA). This option allows the user to specify the maximum number of scan attempts that can be made to find iSER volume.

Configuration option = Default value	Type	Description
num_memory_encrypted_guests = None	integer value	Maximum number of guests with encrypted memory which can run concurrently on this compute host. For now this is only relevant for AMD machines which support SEV (Secure Encrypted Virtualization). Such machines have a limited number of slots in their memory controller for storing encryption keys. Each running guest with encrypted memory will consume one of these slots. The option may be reused for other equivalent technologies in the future. If the machine does not support memory encryption, the option will be ignored and inventory will be set to 0. If the machine does support memory encryption, for now a value of None means an effectively unlimited inventory, i.e. no limit will be imposed by Nova on the number of SEV guests which can be launched, even though the underlying hardware will enforce its own limit. However it is expected that in the future, auto-detection of the inventory from the hardware will become possible, at which point None will cause auto-detection to automatically impose the correct limit. a. note:: It is recommended to read :ref:`the deployment documentation's section on this option <num_memory_encrypted_guests>` before deciding whether to configure this setting or leave it at the default. Related options: • :oslo.config:option:libvirt.virt_type must be set to kvm. • It's recommended to consider including x86_64=q35 in :oslo.config:option:libvirt.hw_machine_type; see :ref:`deploying-sev-capable-infrastructure` for more on this.

Configuration option = Default value	Type	Description
num_nvme_discover_tries = 5	integer value	Number of times to rediscover NVMe target to find volume Nova provides support for block storage attaching to hosts via NVMe (Non-Volatile Memory Express). This option allows the user to specify the maximum number of retry attempts that can be made to discover the NVMe device.
num_pcie_ports = 0	integer value	The number of PCIe ports an instance will get. Libvirt allows a custom number of PCIe ports (pcie-root-port controllers) a target instance will get. Some will be used by default, rest will be available for hotplug use. By default we have just 1-2 free ports which limits hotplug. More info: https://github.com/qemu/qemu/blob/master/docs/pcie.txt Due to QEMU limitations for aarch64/virt maximum value is set to 28. Default value 0 moves calculating amount of ports to libvirt.
num_volume_scan_tries = 5	integer value	Number of times to scan given storage protocol to find volume.

Configuration option = Default value	Type	Description
pmem_namespaces = []	list value	Configure persistent memory(pmem) namespaces. These namespaces must have been already created on the host. This config option is in the following format:: <pre>"\$LABEL:\$NSNAME[&verbar;\$NSNAME] [, \$LABEL:\$NSNAME[&verbar;\$NSNAME]]"</pre> ● \$NSNAME is the name of the pmem namespace. ● \$LABEL represents one resource class, this is used to generate the resource class name as CUSTOM_PMEM_NAMESPACE_\$LABEL. For example <pre>[libvirt] pmem_namespaces=128G:ns0 ns1 ns2 ns3,262144MB:ns4 ns5,MEDIUM:ns6 ns7</pre>
quobyte_client_cfg = None	string value	Path to a Quobyte Client configuration file.
quobyte_mount_point_base = \$state_path/mnt	string value	Directory where the Quobyte volume is mounted on the compute node. Nova supports Quobyte volume driver that enables storing Block Storage service volumes on a Quobyte storage back end. This Option specifies the path of the directory where Quobyte volume is mounted. Possible values: ● A string representing absolute path of mount point.
rbd_connect_timeout = 5	integer value	The RADOS client timeout in seconds when initially connecting to the cluster.
rbd_destroy_volume_retries = 12	integer value	Number of retries to destroy a RBD volume. Related options: ● [libvirt]/images_type = <i>rbd</i>

Configuration option = Default value	Type	Description
rbd_destroy_volume_retry_interval = 5	integer value	Number of seconds to wait between each consecutive retry to destroy a RBD volume. Related options: • [libvirt]/images_type = <i>rbd</i>
rbd_secret_uuid = None	string value	The libvirt UUID of the secret for the rbd_user volumes.
rbd_user = None	string value	The RADOS client name for accessing rbd(RADOS Block Devices) volumes. Libvirt will refer to this user when connecting and authenticating with the Ceph RBD server.
realtime_scheduler_priority = 1	integer value	In a realtime host context vCPUs for guest will run in that scheduling priority. Priority depends on the host kernel (usually 1-99)
remote_filesystem_transport = ssh	string value	libvirt's transport method for remote file operations. Because libvirt cannot use RPC to copy files over network to/from other compute nodes, other method must be used for: • creating directory on remote host • creating file on remote host • removing file from remote host • copying file to remote host

Configuration option = Default value	Type	Description
rescue_image_id = None	string value	The ID of the image to boot from to rescue data from a corrupted instance. If the rescue REST API operation doesn't provide an ID of an image to use, the image which is referenced by this ID is used. If this option is not set, the image from the instance is used. Possible values: • An ID of an image or nothing. If it points to an Amazon Machine Image (AMI), consider to set the config options rescue_kernel_id and rescue_ramdisk_id too. If nothing is set, the image of the instance is used. Related options: • rescue_kernel_id: If the chosen rescue image allows the separate definition of its kernel disk, the value of this option is used, if specified. This is the case when Amazon's AMI/AKI/ARI image format is used for the rescue image. • rescue_ramdisk_id: If the chosen rescue image allows the separate definition of its RAM disk, the value of this option is used if, specified. This is the case when Amazon's AMI/AKI/ARI image format is used for the rescue image.
rescue_kernel_id = None	string value	The ID of the kernel (AKI) image to use with the rescue image. If the chosen rescue image allows the separate definition of its kernel disk, the value of this option is used, if specified. This is the case when Amazon's AMI/AKI/ARI image format is used for the rescue image. Possible values: • An ID of an kernel image or nothing. If nothing is specified, the kernel disk from the instance is used if it was launched with one. Related options: • rescue_image_id: If that option points to an image in Amazon's AMI/AKI/ARI image format, it's useful to use rescue_kernel_id too.

Configuration option = Default value	Type	Description
rescue_ramdisk_id = None	string value	The ID of the RAM disk (ARI) image to use with the rescue image. If the chosen rescue image allows the separate definition of its RAM disk, the value of this option is used, if specified. This is the case when Amazon's AMI/AKI/ARI image format is used for the rescue image. Possible values: ● An ID of a RAM disk image or nothing. If nothing is specified, the RAM disk from the instance is used if it was launched with one. Related options: ● rescue_image_id: If that option points to an image in Amazon's AMI/AKI/ARI image format, it's useful to use rescue_ramdisk_id too.
rng_dev_path = /dev/urandom	string value	The path to an RNG (Random Number Generator) device that will be used as the source of entropy on the host. Since libvirt 1.3.4, any path (that returns random numbers when read) is accepted. The recommended source of entropy is /dev/urandom – it is non-blocking, therefore relatively fast; and avoids the limitations of /dev/random, which is a legacy interface. For more details (and comparison between different RNG sources), refer to the "Usage" section in the Linux kernel API documentation for [u]random: http://man7.org/linux/man-pages/man4/urandom.4.html and http://man7.org/linux/man-pages/man7/random.7.html.
rx_queue_size = None	integer value	Configure virtio rx queue size. This option is only usable for virtio-net device with vhost and vhost-user backend. Available only with QEMU/KVM. Requires libvirt v2.3 QEMU v2.7.
`smbfs_mount_options = `	string value	Mount options passed to the SMBFS client. Provide SMBFS options as a single string containing all parameters. See mount.cifs man page for details. Note that the libvirt-qemu uid and gid must be specified.

Configuration option = Default value	Type	Description
smbfs_mount_point_base = <code>\$state_path/mnt</code>	string value	Directory where the SMBFS shares are mounted on the compute node.
snapshot_compression = False	boolean value	Enable snapshot compression for qcow2 images. Note: you can set snapshot_image_format to qcow2 to force all snapshots to be in qcow2 format, independently from their original image type. Related options: • <code>snapshot_image_format</code>
snapshot_image_format = None	string value	Determine the snapshot image format when sending to the image service. If set, this decides what format is used when sending the snapshot to the image service. If not set, defaults to same type as source image.
snapshots_directory = \$instances_path/snapshots	string value	Location where libvirt driver will store snapshots before uploading them to image service
sparse_logical_volumes = False	boolean value	Create sparse logical volumes (with <code>virtalsize</code>) if this flag is set to <code>True</code>. Deprecated since: 18.0.0 Reason: Sparse logical volumes is a feature that is not tested hence not supported. LVM logical volumes are preallocated by default. If you want thin provisioning, use Cinder thin-provisioned volumes.
swtpm_enabled = False	boolean value	Enable emulated TPM (Trusted Platform Module) in guests.

Configuration option = Default value	Type	Description
swtpm_group = tss	string value	Group that swtpm binary runs as. When using emulated TPM, the swtpm binary will run to emulate a TPM device. The user this binary runs as depends on libvirt configuration, with tss being the default. In order to support cold migration and resize, nova needs to know what group the swtpm binary is running as in order to ensure that files get the proper ownership after being moved between nodes. Related options: ● swtpm_user must also be set.
swtpm_user = tss	string value	User that swtpm binary runs as. When using emulated TPM, the swtpm binary will run to emulate a TPM device. The user this binary runs as depends on libvirt configuration, with tss being the default. In order to support cold migration and resize, nova needs to know what user the swtpm binary is running as in order to ensure that files get the proper ownership after being moved between nodes. Related options: ● swtpm_group must also be set.
sysinfo_serial = unique	string value	The data source used to the populate the host "serial" UUID exposed to guest in the virtual BIOS. All choices except unique will change the serial when migrating the instance to another host. Changing the choice of this option will also affect existing instances on this host once they are stopped and started again. It is recommended to use the default choice (unique) since that will not change when an instance is migrated. However, if you have a need for per-host serials in addition to per-instance serial numbers, then consider restricting flavors via host aggregates.

Configuration option = Default value	Type	Description
tx_queue_size = None	integer value	Configure virtio tx queue size. This option is only usable for virtio-net device with vhost-user backend. Available only with QEMU/KVM. Requires libvirt v3.7 QEMU v2.10.
uid_maps = []	list value	List of uid targets and ranges. Syntax is guest-uid:host-uid:count. Maximum of 5 allowed.
use_virtio_for_bridges = True	boolean value	Use virtio for bridge interfaces with KVM/QEMU
virt_type = kvm	string value	Describes the virtualization type (or so called domain type) libvirt should use. The choice of this type must match the underlying virtualization strategy you have chosen for this host. Related options: ● connection_uri: depends on this ● disk_prefix: depends on this ● cpu_mode: depends on this ● cpu_models: depends on this
volume_clear = zero	string value	Method used to wipe ephemeral disks when they are deleted. Only takes effect if LVM is set as backing storage. Related options: ● images_type - must be set to lvm ● volume_clear_size

Configuration option = Default value	Type	Description
volume_clear_size = 0	integer value	Size of area in MiB, counting from the beginning of the allocated volume, that will be cleared using method set in volume_clear option. Possible values: ● 0 - clear whole volume ● >0 - clear specified amount of MiB Related options: ● images_type - must be set to lvm ● volume_clear - must be set and the value must be different than none for this option to have any impact
volume_use_multipath = False	boolean value	Use multipath connection of the iSCSI or FC volume Volumes can be connected in the LibVirt as multipath devices. This will provide high availability and fault tolerance.
vzstorage_cache_path = None	string value	Path to the SSD cache file. You can attach an SSD drive to a client and configure the drive to store a local cache of frequently accessed data. By having a local cache on a client's SSD drive, you can increase the overall cluster performance by up to 10 and more times. WARNING! There is a lot of SSD models which are not server grade and may loose arbitrary set of data changes on power loss. Such SSDs should not be used in Vstorage and are dangerous as may lead to data corruptions and inconsistencies. Please consult with the manual on which SSD models are known to be safe or verify it using vzstorage-hwflush-check(1) utility. This option defines the path which should include "%(cluster_name)s" template to separate caches from multiple shares. Related options: ● vzstorage_mount_opts may include more detailed cache options.

Configuration option = Default value	Type	Description
vzstorage_log_path = /var/log/vstorage/% (cluster_name)s/nova.log. gz	string value	Path to vzstorage client log. This option defines the log of cluster operations, it should include "%(cluster_name)s" template to separate logs from multiple shares. Related options: ● vzstorage_mount_opts may include more detailed logging options.
vzstorage_mount_group = qemu	string value	Mount owner group name. This option defines the owner group of Vzstorage cluster mountpoint. Related options: ● vzstorage_mount_* group of parameters
vzstorage_mount_opts = []	list value	Extra mount options for pstorage-mount For full description of them, see https://static.openvz.org/vz-man/man1/pstorage-mount.1.gz.html Format is a python string representation of arguments list, like: "[-v, -R, 500]" Shouldn't include -c, -l, -C, -u, -g and -m as those have explicit vzstorage_* options. Related options: ● All other vzstorage_* options
vzstorage_mount_perms = 0770	string value	Mount access mode. This option defines the access bits of Vzstorage cluster mountpoint, in the format similar to one of chmod(1) utility, like this: 0770. It consists of one to four digits ranging from 0 to 7, with missing lead digits assumed to be 0's. Related options: ● vzstorage_mount_* group of parameters

Configuration option = Default value	Type	Description
vzstorage_mount_point_base = \$state_path/mnt	string value	Directory where the Virtuoizzo Storage clusters are mounted on the compute node. This option defines non-standard mountpoint for Vzstorage cluster. Related options: ● vzstorage_mount_* group of parameters
vzstorage_mount_user = stack	string value	Mount owner user name. This option defines the owner user of Vzstorage cluster mountpoint. Related options: ● vzstorage_mount_* group of parameters
wait_soft_reboot_seconds = 120	integer value	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.

9.1.27. metrics

The following table outlines the options available under the **[metrics]** group in the **/etc/nova/nova.conf** file.

Table 9.26. metrics

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
required = True	boolean value	Whether metrics are required. This setting determines how any unavailable metrics are treated. If this option is set to True, any hosts for which a metric is unavailable will raise an exception, so it is recommended to also use the MetricFilter to filter out those hosts before weighing. Possible values: • A boolean value, where False ensures any metric being unavailable for a host will set the host weight to [metrics] weight_of_unavailable. Related options: • [metrics] weight_of_unavailable

Configuration option = Default value	Type	Description
weight_multiplier = 1.0	floating point value	Multiplier used for weighing hosts based on reported metrics. When using metrics to weight the suitability of a host, you can use this option to change how the calculated weight influences the weight assigned to a host as follows: ● >1.0: increases the effect of the metric on overall weight ● 1.0: no change to the calculated weight ● >0.0,<1.0: reduces the effect of the metric on overall weight ● 0.0: the metric value is ignored, and the value of the [metrics] weight_of_unavailable option is returned instead ● >-1.0,<0.0: the effect is reduced and reversed ● -1.0: the effect is reversed ● <-1.0: the effect is increased proportionally and reversed Possible values: ● An integer or float value, where the value corresponds to the multiplier ratio for this weigher. Related options: ● [filter_scheduler] weight_classes ● [metrics] weight_of_unavailable

Configuration option = Default value	Type	Description
weight_of_unavailable = -10000.0	floating point value	Default weight for unavailable metrics. When any of the following conditions are met, this value will be used in place of any actual metric value: - One of the metrics named in [metrics] weight_setting is not available for a host, and the value of required is False. - The ratio specified for a metric in [metrics] weight_setting is 0. - The [metrics] weight_multiplier option is set to 0. Possible values: - An integer or float value, where the value corresponds to the multiplier ratio for this weigher. Related options: [metrics] weight_setting [metrics] required [metrics] weight_multiplier

Configuration option = Default value	Type	Description
weight_setting = []	list value	Mapping of metric to weight modifier. This setting specifies the metrics to be weighed and the relative ratios for each metric. This should be a single string value, consisting of a series of one or more <i>name=ratio</i> pairs, separated by commas, where name is the name of the metric to be weighed, and ratio is the relative weight for that metric. Note that if the ratio is set to 0, the metric value is ignored, and instead the weight will be set to the value of the [metrics] weight_of_unavailable option. As an example, let's consider the case where this option is set to: <pre>`name1=1.0, name2=-1.3`</pre> The final weight will be: <pre>`(name1.value * 1.0) + (name2.value * -1.3)`</pre> Possible values: - A list of zero or more key/value pairs separated by commas, where the key is a string representing the name of a metric and the value is a numeric weight for that metric. If any value is set to 0, the value is ignored and the weight will be set to the value of the [metrics] weight_of_unavailable option. Related options: [metrics] weight_of_unavailable

9.1.28. mks

The following table outlines the options available under the **[mks]** group in the **/etc/nova/nova.conf** file.

Table 9.27. mks

Configuration option = Default value	Type	Description
enabled = False	boolean value	Enables graphical console access for virtual machines.

Configuration option = Default value	Type	Description
mksproxy_base_url = http://127.0.0.1:6090/	uri value	Location of MKS web console proxy The URL in the response points to a WebMKS proxy which starts proxying between client and corresponding vCenter server where instance runs. In order to use the web based console access, WebMKS proxy should be installed and configured Possible values: Must be a valid URL of the form:http://host:port/ or https://host:port/

9.1.29. neutron

The following table outlines the options available under the **[neutron]** group in the `/etc/nova/nova.conf` file.

Table 9.28. neutron

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

Configuration option = Default value	Type	Description
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default_floating_pool = nova	string value	Default name for the floating IP pool. Specifies the name of floating IP pool used for allocating floating IPs. This option is only used if Neutron does not specify the floating IP pool name in port binding responses.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
extension_sync_interval = 600	integer value	Integer value representing the number of seconds to wait before querying Neutron for extensions. After this number of seconds the next time Nova needs to create a resource in Neutron it will requery Neutron for the extensions that it has loaded. Setting value to 0 will refresh the extensions with no wait.
http_retries = 3	integer value	Number of times neutronclient should retry on any failed http call. 0 means connection is attempted only once. Setting it to any positive integer means that on failure connection is retried that many times e.g. setting it to 3 means total attempts to connect will be 4. Possible values: Any integer value. 0 means connection is attempted only once

Configuration option = Default value	Type	Description
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
<code>`metadata_proxy_shared_secret = `</code>	string value	This option holds the shared secret string used to validate proxy requests to Neutron metadata requests. In order to be used, the <i>X-Metadata-Provider-Signature</i> header must be supplied in the request. Related options: • <code>service_metadata_proxy</code>
ovs_bridge = br-int	string value	Default name for the Open vSwitch integration bridge. Specifies the name of an integration bridge interface used by OpenvSwitch. This option is only used if Neutron does not specify the OVS bridge name in port binding responses.
password = None	string value	User's password

Configuration option = Default value	Type	Description
physnets = []	list value	List of physnets present on this host. For each physnet listed, an additional section, [neutron_physnet_\$PHYSNET], will be added to the configuration file. Each section must be configured with a single configuration option, numa_nodes, which should be a list of node IDs for all NUMA nodes this physnet is associated with. For example:: <pre>[neutron] physnets = foo, bar [neutron_physnet_foo] numa_nodes = 0 [neutron_physnet_bar] numa_nodes = 0,1</pre> Any physnet that is not listed using this option will be treated as having no particular NUMA node affinity. Tunnelled networks (VXLAN, GRE, ...) cannot be accounted for in this way and are instead configured using the [neutron_tunnel] group. For example:: <pre>[neutron_tunnel] numa_nodes = 1</pre> Related options: • [neutron_tunnel] numa_nodes can be used to configure NUMA affinity for all tunneled networks • [neutron_physnet_\$PHYSNET] numa_nodes must be configured for each value of \$PHYSNET specified by this option
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to

Configuration option = Default value	Type	Description
region-name = None	string value	The default region_name for endpoint URL discovery.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = network	string value	The default service_type for endpoint URL discovery.
service_metadata_proxy = False	boolean value	When set to True, this option indicates that Neutron will be used to proxy metadata requests and resolve instance ids. Otherwise, the instance ID must be passed to the metadata request in the <i>X-Instance-ID</i> header. Related options: ● metadata_proxy_shared_secret
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrievable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User ID
username = None	string value	Username

Configuration option = Default value	Type	Description
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.

9.1.30. notifications

The following table outlines the options available under the **[notifications]** group in the **/etc/nova/nova.conf** file.

Table 9.29. notifications

Configuration option = Default value	Type	Description
bdms_in_notifications = False	boolean value	If enabled, include block device information in the versioned notification payload. Sending block device information is disabled by default as providing that information can incur some overhead on the system since the information may need to be loaded from the database.
default_level = INFO	string value	Default notification level for outgoing notifications.
notification_format = unversioned	string value	Specifies which notification format shall be emitted by nova. The versioned notification interface are in feature parity with the legacy interface and the versioned interface is actively developed so new consumers should used the versioned interface. However, the legacy interface is heavily used by ceilometer and other mature OpenStack components so it remains the default. Note that notifications can be completely disabled by setting driver=noop in the [oslo_messaging_notifications] group. The list of versioned notifications is visible in https://docs.openstack.org/nova/latest/reference/notifications.html
notify_on_state_change = None	string value	If set, send compute.instance.update notifications on instance state changes. Please refer to https://docs.openstack.org/nova/latest/reference/notifications.html for additional information on notifications.

Configuration option = Default value	Type	Description
versioned_notifications_topics = ['versioned_notifications']	list value	Specifies the topics for the versioned notifications issued by nova. The default value is fine for most deployments and rarely needs to be changed. However, if you have a third-party service that consumes versioned notifications, it might be worth getting a topic for that service. Nova will send a message containing a versioned notification payload to each topic queue in this list. The list of versioned notifications is visible in https://docs.openstack.org/nova/latest/reference/notifications.html

9.1.31. oslo_concurrency

The following table outlines the options available under the **[oslo_concurrency]** group in the **/etc/nova/nova.conf** file.

Table 9.30. oslo_concurrency

Configuration option = Default value	Type	Description
disable_process_locking = False	boolean value	Enables or disables inter-process locks.
lock_path = None	string value	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.

9.1.32. oslo_messaging_amqp

The following table outlines the options available under the **[oslo_messaging_amqp]** group in the **/etc/nova/nova.conf** file.

Table 9.31. oslo_messaging_amqp

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
addressing_mode = dynamic	string value	Indicates the addressing mode used by the driver. Permitted values: <i>legacy</i> - use legacy non-routable addressing <i>routable</i> - use routable addresses <i>dynamic</i> - use legacy addresses if the message bus does not support routing otherwise use routable addressing
anycast_address = anycast	string value	Appended to the address prefix when sending to a group of consumers. Used by the message bus to identify messages that should be delivered in a round-robin fashion across consumers.
broadcast_prefix = broadcast	string value	address prefix used when broadcasting to all servers
connection_retry_backoff = 2	integer value	Increase the connection_retry_interval by this many seconds after each unsuccessful failover attempt.
connection_retry_interval = 1	integer value	Seconds to pause before attempting to re-connect.
connection_retry_interval_max = 30	integer value	Maximum limit for connection_retry_interval + connection_retry_backoff
container_name = None	string value	Name for the AMQP container. must be globally unique. Defaults to a generated UUID
default_notification_exchange = None	string value	Exchange name used in notification addresses. Exchange name resolution precedence: Target.exchange if set else default_notification_exchange if set else control_exchange if set else <i>notify</i>
default_notify_timeout = 30	integer value	The deadline for a sent notification message delivery. Only used when caller does not provide a timeout expiry.
default_reply_retry = 0	integer value	The maximum number of attempts to re-send a reply message which failed due to a recoverable error.
default_reply_timeout = 30	integer value	The deadline for an rpc reply message delivery.
default_rpc_exchange = None	string value	Exchange name used in RPC addresses. Exchange name resolution precedence: Target.exchange if set else default_rpc_exchange if set else control_exchange if set else <i>rpc</i>

Configuration option = Default value	Type	Description
default_send_timeout = 30	integer value	The deadline for an rpc cast or call message delivery. Only used when caller does not provide a timeout expiry.
default_sender_link_timeout = 600	integer value	The duration to schedule a purge of idle sender links. Detach link after expiry.
group_request_prefix = unicast	string value	address prefix when sending to any server in group
idle_timeout = 0	integer value	Timeout for inactive connections (in seconds)
link_retry_delay = 10	integer value	Time to pause between re-connecting an AMQP 1.0 link that failed due to a recoverable error.
multicast_address = multicast	string value	Appended to the address prefix when sending a fanout message. Used by the message bus to identify fanout messages.
notify_address_prefix = openstack.org/om/notify	string value	Address prefix for all generated Notification addresses
notify_server_credit = 100	integer value	Window size for incoming Notification messages
pre_settled = ['rpc-cast', 'rpc-reply']	multi valued	Send messages of this type pre-settled. Pre-settled messages will not receive acknowledgement from the peer. Note well: pre-settled messages may be silently discarded if the delivery fails. Permitted values: <i>rpc-call</i> - send RPC Calls pre-settled <i>rpc-reply</i> - send RPC Replies pre-settled <i>rpc-cast</i> - Send RPC Casts pre-settled <i>notify</i> - Send Notifications pre-settled
pseudo_vhost = True	boolean value	Enable virtual host support for those message buses that do not natively support virtual hosting (such as qpidd). When set to true the virtual host name will be added to all message bus addresses, effectively creating a private <i>subnet</i> per virtual host. Set to False if the message bus supports virtual hosting using the <i>hostname</i> field in the AMQP 1.0 Open performative as the name of the virtual host.
reply_link_credit = 200	integer value	Window size for incoming RPC Reply messages.

Configuration option = Default value	Type	Description
rpc_address_prefix = openstack.org/om/rpc	string value	Address prefix for all generated RPC addresses
rpc_server_credit = 100	integer value	Window size for incoming RPC Request messages
<code>`sas_config_dir = `</code>	string value	Path to directory that contains the SASL configuration
<code>`sas_config_name = `</code>	string value	Name of configuration file (without .conf suffix)
<code>`sas_default_realm = `</code>	string value	SASL realm to use if no realm present in username
<code>`sas_mechanisms = `</code>	string value	Space separated list of acceptable SASL mechanisms
server_request_prefix = exclusive	string value	address prefix used when sending to a specific server
ssl = False	boolean value	Attempt to connect via SSL. If no other ssl-related parameters are given, it will use the system's CA-bundle to verify the server's certificate.
<code>`ssl_ca_file = `</code>	string value	CA certificate PEM file used to verify the server's certificate
<code>`ssl_cert_file = `</code>	string value	Self-identifying certificate PEM file for client authentication
<code>`ssl_key_file = `</code>	string value	Private key PEM file used to sign ssl_cert_file certificate (optional)
ssl_key_password = None	string value	Password for decrypting ssl_key_file (if encrypted)
ssl_verify_vhost = False	boolean value	By default SSL checks that the name in the server's certificate matches the hostname in the transport_url. In some configurations it may be preferable to use the virtual hostname instead, for example if the server uses the Server Name Indication TLS extension (rfc6066) to provide a certificate per virtual host. Set ssl_verify_vhost to True if the server's SSL certificate uses the virtual host name instead of the DNS name.
trace = False	boolean value	Debug: dump AMQP frames to stdout

Configuration option = Default value	Type	Description
unicast_address = unicast	string value	Appended to the address prefix when sending to a particular RPC/Notification server. Used by the message bus to identify messages sent to a single destination.

9.1.33. oslo_messaging_kafka

The following table outlines the options available under the **[oslo_messaging_kafka]** group in the **/etc/nova/nova.conf** file.

Table 9.32. oslo_messaging_kafka

Configuration option = Default value	Type	Description
compression_codec = none	string value	The compression codec for all data generated by the producer. If not set, compression will not be used. Note that the allowed values of this depend on the kafka version
conn_pool_min_size = 2	integer value	The pool size limit for connections expiration policy
conn_pool_ttl = 1200	integer value	The time-to-live in sec of idle connections in the pool
consumer_group = oslo_messaging_consumer	string value	Group id for Kafka consumer. Consumers in one group will coordinate message consumption
enable_auto_commit = False	boolean value	Enable asynchronous consumer commits
kafka_consumer_timeout = 1.0	floating point value	Default timeout(s) for Kafka consumers
kafka_max_fetch_bytes = 1048576	integer value	Max fetch bytes of Kafka consumer
max_poll_records = 500	integer value	The maximum number of records returned in a poll call
pool_size = 10	integer value	Pool Size for Kafka Consumers
producer_batch_size = 16384	integer value	Size of batch for the producer async send

Configuration option = Default value	Type	Description
producer_batch_timeout = 0.0	floating point value	Upper bound on the delay for KafkaProducer batching in seconds
sasl_mechanism = PLAIN	string value	Mechanism when security protocol is SASL
security_protocol = PLAINTEXT	string value	Protocol used to communicate with brokers
<code>`ssl_cafile = `</code>	string value	CA certificate PEM file used to verify the server certificate
<code>`ssl_client_cert_file = `</code>	string value	Client certificate PEM file used for authentication.
<code>`ssl_client_key_file = `</code>	string value	Client key PEM file used for authentication.
<code>`ssl_client_key_password = `</code>	string value	Client key password file used for authentication.

9.1.34. oslo_messaging_notifications

The following table outlines the options available under the **[oslo_messaging_notifications]** group in the **/etc/nova/nova.conf** file.

Table 9.33. oslo_messaging_notifications

Configuration option = Default value	Type	Description
driver = []	multi valued	The Drivers(s) to handle sending notifications. Possible values are messaging, messagingv2, routing, log, test, noop
retry = -1	integer value	The maximum number of attempts to re-send a notification message which failed to be delivered due to a recoverable error. 0 - No retry, -1 - indefinite
topics = ['notifications']	list value	AMQP topic used for OpenStack notifications.
transport_url = None	string value	A URL representing the messaging driver to use for notifications. If not set, we fall back to the same configuration used for RPC.

9.1.35. oslo_messaging_rabbit

The following table outlines the options available under the **[oslo_messaging_rabbit]** group in the **/etc/nova/nova.conf** file.

Table 9.34. oslo_messaging_rabbit

Configuration option = Default value	Type	Description
amqp_auto_delete = False	boolean value	Auto-delete queues in AMQP.
amqp_durable_queues = False	boolean value	Use durable queues in AMQP.
direct_mandatory_flag = True	boolean value	(DEPRECATED) Enable/Disable the RabbitMQ mandatory flag for direct send. The direct send is used as reply, so the MessageUndeliverable exception is raised in case the client queue does not exist. MessageUndeliverable exception will be used to loop for a timeout to let a chance to sender to recover. This flag is deprecated and it will not be possible to deactivate this functionality anymore
enable_cancel_on_failover = False	boolean value	Enable x-cancel-on-ha-failover flag so that rabbitmq server will cancel and notify consumers when queue is down
heartbeat_in_pthread = True	boolean value	Run the health check heartbeat thread through a native python thread by default. If this option is equal to False then the health check heartbeat will inherit the execution model from the parent process. For example if the parent process has monkey patched the stdlib by using eventlet/greenlet then the heartbeat will be run through a green thread.
heartbeat_rate = 2	integer value	How often times during the heartbeat_timeout_threshold we check the heartbeat.
heartbeat_timeout_threshold = 60	integer value	Number of seconds after which the Rabbit broker is considered down if heartbeat's keep-alive fails (0 disables heartbeat).
kombu_compression = None	string value	EXPERIMENTAL: Possible values are: gzip, bz2. If not set compression will not be used. This option may not be available in future versions.
kombu_failover_strategy = round-robin	string value	Determines how the next RabbitMQ node is chosen in case the one we are currently connected to becomes unavailable. Takes effect only if more than one RabbitMQ node is provided in config.
kombu_missing_consumer_retry_timeout = 60	integer value	How long to wait a missing client before abandoning to send it its replies. This value should not be longer than rpc_response_timeout.

Configuration option = Default value	Type	Description
kombu_reconnect_delay = 1.0	floating point value	How long to wait before reconnecting in response to an AMQP consumer cancel notification.
rabbit_ha_queues = False	boolean value	Try to use HA queues in RabbitMQ (x-ha-policy: all). If you change this option, you must wipe the RabbitMQ database. In RabbitMQ 3.0, queue mirroring is no longer controlled by the x-ha-policy argument when declaring a queue. If you just want to make sure that all queues (except those with auto-generated names) are mirrored across all nodes, run: <code>"rabbitmqctl set_policy HA ^(?!amq\.).*{"ha-mode": "all"}"</code>
rabbit_interval_max = 30	integer value	Maximum interval of RabbitMQ connection retries. Default is 30 seconds.
rabbit_login_method = AMQPLAIN	string value	The RabbitMQ login method.
rabbit_qos_prefetch_count = 0	integer value	Specifies the number of messages to prefetch. Setting to zero allows unlimited messages.
rabbit_retry_backoff = 2	integer value	How long to backoff for between retries when connecting to RabbitMQ.
rabbit_retry_interval = 1	integer value	How frequently to retry connecting with RabbitMQ.
rabbit_transient_queues_ttl = 1800	integer value	Positive integer representing duration in seconds for queue TTL (x-expires). Queues which are unused for the duration of the TTL are automatically deleted. The parameter affects only reply and fanout queues.
ssl = False	boolean value	Connect over SSL.
<code>`ssl_ca_file = `</code>	string value	SSL certification authority file (valid only if SSL enabled).
<code>`ssl_cert_file = `</code>	string value	SSL cert file (valid only if SSL enabled).
<code>`ssl_key_file = `</code>	string value	SSL key file (valid only if SSL enabled).
<code>`ssl_version = `</code>	string value	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.

9.1.36. oslo_middleware

The following table outlines the options available under the **[oslo_middleware]** group in the **/etc/nova/nova.conf** file.

Table 9.35. oslo_middleware

Configuration option = Default value	Type	Description
enable_proxy_headers_parsing = False	boolean value	Whether the application is behind a proxy or not. This determines if the middleware should parse the headers or not.
max_request_body_size = 114688	integer value	The maximum body size for each request, in bytes.
secure_proxy_ssl_header = X-Forwarded-Proto	string value	The HTTP Header that will be used to determine what the original request protocol scheme was, even if it was hidden by a SSL termination proxy.

9.1.37. oslo_policy

The following table outlines the options available under the **[oslo_policy]** group in the **/etc/nova/nova.conf** file.

Table 9.36. oslo_policy

Configuration option = Default value	Type	Description
enforce_new_defaults = False	boolean value	This option controls whether or not to use old deprecated defaults when evaluating policies. If True , the old deprecated defaults are not going to be evaluated. This means if any existing token is allowed for old defaults but is disallowed for new defaults, it will be disallowed. It is encouraged to enable this flag along with the enforce_scope flag so that you can get the benefits of new defaults and scope_type together
enforce_scope = False	boolean value	This option controls whether or not to enforce scope when evaluating policies. If True , the scope of the token used in the request is compared to the scope_types of the policy being enforced. If the scopes do not match, an InvalidScope exception will be raised. If False , a message will be logged informing operators that policies are being invoked with mismatching scope.
policy_default_rule = default	string value	Default rule. Enforced when a requested rule is not found.

Configuration option = Default value	Type	Description
policy_dirs = ['policy.d']	multi valued	Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the <code>config_dir</code> option, or absolute paths. The file defined by <code>policy_file</code> must exist for these directories to be searched. Missing or empty directories are ignored.
policy_file = policy.yaml	string value	The relative or absolute path of a file that maps roles to permissions for a given service. Relative paths must be specified in relation to the configuration file setting this option.
remote_content_type = application/x-www-form-urlencoded	string value	Content Type to send and receive data for REST based policy check
remote_ssl_ca_cert_file = None	string value	Absolute path to ca cert file for REST based policy check
remote_ssl_client_cert_file = None	string value	Absolute path to client cert for REST based policy check
remote_ssl_client_key_file = None	string value	Absolute path client key file REST based policy check
remote_ssl_verify_server_cert = False	boolean value	server identity verification for REST based policy check

9.1.38. pci

The following table outlines the options available under the **[pci]** group in the `/etc/nova/nova.conf` file.

Table 9.37. pci

Configuration option = Default value	Type	Description
alias = []	multi valued	An alias for a PCI passthrough device requirement. This allows users to specify the alias in the extra specs for a flavor, without needing to repeat all the PCI property requirements. This should be configured for the nova-api service and, assuming you wish to use move operations, for each nova-compute service. Possible Values:

Configuration option = Default value	Type	Description
		- A dictionary of JSON values which describe the aliases. For example:: <pre>alias = { "name": "QuickAssist", "product_id": "0443", "vendor_id": "8086", "device_type": "type-PCI", "numa_policy": "required" }</pre> This defines an alias for the Intel QuickAssist card. (multi valued). Valid key values are : <code>`name`</code> Name of the PCI alias. <code>`product_id`</code> Product ID of the device in hexadecimal. <code>`vendor_id`</code> Vendor ID of the device in hexadecimal. <code>`device_type`</code> Type of PCI device. Valid values are: <code>`type-PCI`</code>, <code>`type-PF`</code> and <code>`type-VF`</code>. Note that <code>`"device_type": "type-PF"`</code> must be specified if you wish to passthrough a device that supports SR-IOV in its entirety. <code>`numa_policy`</code> Required NUMA affinity of device. Valid values are: <code>`legacy`</code>, <code>`preferred`</code> and <code>`required`</code>. - Supports multiple aliases by repeating the option (not by specifying a list value) <pre>alias = { "name": "QuickAssist-1", "product_id": "0443", "vendor_id": "8086", "device_type": "type-PCI", "numa_policy": "required" } alias = { "name": "QuickAssist-2", "product_id": "0444", "vendor_id": "8086", "device_type": "type-PCI", "numa_policy": "required" }</pre>

passthrough_whitelist = [] Configuration option = Default value	multi valued Type	White list of PCI devices available to VMs. Description Possible values:
		● A JSON dictionary which describe a whitelisted PCI device. It should take the following format <pre>[{"vendor_id": "<id>","product_id": "<id>","address": "[[[[<domain>]:<bus>]:<slot>][.<function>]]" "devname": "<name>","tag": "<tag_value>"}]</pre> Where <code>[]</code> indicates zero or one occurrences, <code>{}</code> indicates zero or multiple occurrences, and <code>`&verbar`</code> mutually exclusive options. Note that any missing fields are automatically wildcarded. Valid key values are : <code>`vendor_id`</code> Vendor ID of the device in hexadecimal. <code>`product_id`</code> Product ID of the device in hexadecimal. <code>`address`</code> PCI address of the device. Both traditional glob style and regular expression syntax is supported. Please note that the address fields are restricted to the following maximum values: ● domain - 0xFFFF ● bus - 0xFF ● slot - 0x1F ● function - 0x7 <code>`devname`</code> Device name of the device (for e.g. interface name). Not all PCI devices have a name. <code>`<tag>`</code>

Configuration option = Default value	Type	Description Additional ` <code><tag></code> ` and ` <code><tag_value></code> ` used for matching PCI devices.
		Supported `<code><tag></code>` values are : - o physical_network - o trusted Valid examples are <pre> passthrough_whitelist = {"devname": "eth0", "physical_network": "physnet"} passthrough_whitelist = {"address": "0a:00:"} passthrough_whitelist = {"address": "0a:00:", "physical_network": "physnet1"} passthrough_whitelist = {"vendor_id": "1137", "product_id": "0071"} passthrough_whitelist = {"vendor_id": "1137", "product_id": "0071", "address": "0000:0a:00.1", "physical_network": "physnet1"} passthrough_whitelist = {"address": {"domain": ".", "bus": "02", "slot": "01", "function": "[2-7]"}, "physical_network": "physnet1"} passthrough_whitelist = {"address": {"domain": ".", "bus": "02", "slot": "0[1-2]", "function": ".*"}, "physical_network": "physnet1"} passthrough_whitelist = {"devname": "eth0", "physical_network": "physnet1", "trusted": "true"} </pre> The following are invalid, as they specify mutually exclusive options <pre> passthrough_whitelist = {"devname": "eth0", "physical_network": "physnet", "address": "0a:00:"} </pre> ● A JSON list of JSON dictionaries corresponding to the above format. For example <pre> passthrough_whitelist = [{"product_id": "0001", "vendor_id": "8086"}, {"product_id": "0002", "vendor_id": "8086"}] </pre>

9.1.39. placement

The following table outlines the options available under the **[placement]** group in the `/etc/nova/nova.conf` file.

Table 9.38. placement

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
connect-retries = None	integer value	The maximum number of retries that should be attempted for connection errors.
connect-retry-delay = None	floating point value	Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
endpoint-override = None	string value	Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the version , min-version , and/or max-version options.
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file

Configuration option = Default value	Type	Description
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
region-name = None	string value	The default region_name for endpoint URL discovery.
service-name = None	string value	The default service_name for endpoint URL discovery.
service-type = placement	string value	The default service_type for endpoint URL discovery.
split-loggers = False	boolean value	Log requests to multiple loggers.
status-code-retries = None	integer value	The maximum number of retries that should be attempted for retrievable HTTP status codes.
status-code-retry-delay = None	floating point value	Delay (in seconds) between two retries for retrievable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User ID

Configuration option = Default value	Type	Description
username = None	string value	Username
valid-interfaces = ['internal', 'public']	list value	List of interfaces, in order of preference, for endpoint URL.

9.1.40. powervm

The following table outlines the options available under the **[powervm]** group in the **/etc/nova/nova.conf** file.

Table 9.39. powervm

Configuration option = Default value	Type	Description
disk_driver = localdisk	string value	The disk driver to use for PowerVM disks. PowerVM provides support for localdisk and PowerVM Shared Storage Pool disk drivers. Related options: ● <code>volume_group_name</code> - required when using localdisk
proc_units_factor = 0.1	floating point value	Factor used to calculate the amount of physical processor compute power given to each vCPU. E.g. A value of 1.0 means a whole physical processor, whereas 0.05 means 1/20th of a physical processor.
<code>`volume_group_name = `</code>	string value	Volume Group to use for block device operations. If <code>disk_driver</code> is <code>localdisk</code> , then this attribute must be specified. It is strongly recommended NOT to use <code>rootvg</code> since that is used by the management partition and filling it will cause failures.

9.1.41. privsep

The following table outlines the options available under the **[privsep]** group in the **/etc/nova/nova.conf** file.

Table 9.40. privsep

Configuration option = Default value	Type	Description
capabilities = []	list value	List of Linux capabilities retained by the privsep daemon.

Configuration option = Default value	Type	Description
group = None	string value	Group that the privsep daemon should run as.
helper_command = None	string value	Command to invoke to start the privsep daemon if not using the "fork" method. If not specified, a default is generated using "sudo privsep-helper" and arguments designed to recreate the current configuration. This command must accept suitable --privsep_context and --privsep_sock_path arguments.
thread_pool_size = <based on operating system>	integer value	The number of threads available for privsep to concurrently run processes. Defaults to the number of CPU cores in the system.
user = None	string value	User that the privsep daemon should run as.

9.1.42. profiler

The following table outlines the options available under the **[profiler]** group in the `/etc/nova/nova.conf` file.

Table 9.41. profiler

Configuration option = Default value	Type	Description
connection_string = messaging://	string value	Connection string for a notifier backend. Default value is messaging:// which sets the notifier to oslo_messaging. Examples of possible values: ● messaging:// - use oslo_messaging driver for sending spans. ● redis://127.0.0.1:6379 - use redis driver for sending spans. ● mongodb://127.0.0.1:27017 - use mongodb driver for sending spans. ● elasticsearch://127.0.0.1:9200 - use elasticsearch driver for sending spans. ● jaeger://127.0.0.1:6831 - use jaeger tracing as driver for sending spans.

Configuration option = Default value	Type	Description
enabled = False	boolean value	Enable the profiling for all services on this node. Default value is False (fully disable the profiling feature). Possible values: ● True: Enables the feature ● False: Disables the feature. The profiling cannot be started via this project operations. If the profiling is triggered by another project, this project part will be empty.
es_doc_type = notification	string value	Document type for notification indexing in elasticsearch.
es_scroll_size = 10000	integer value	Elasticsearch splits large requests in batches. This parameter defines maximum size of each batch (for example: es_scroll_size=10000).
es_scroll_time = 2m	string value	This parameter is a time value parameter (for example: es_scroll_time=2m), indicating for how long the nodes that participate in the search will maintain relevant resources in order to continue and support it.
filter_error_trace = False	boolean value	Enable filter traces that contain error/exception to a separated place. Default value is set to False. Possible values: ● True: Enable filter traces that contain error/exception. ● False: Disable the filter.

Configuration option = Default value	Type	Description
hmac_keys = SECRET_KEY	string value	Secret key(s) to use for encrypting context data for performance profiling. This string value should have the following format: <key1>[,<key2>,...<keyn>], where each key is some random string. A user who triggers the profiling via the REST API has to set one of these keys in the headers of the REST API call to include profiling results of this node for this particular project. Both "enabled" flag and "hmac_keys" config options should be set to enable profiling. Also, to generate correct profiling information across all services at least one key needs to be consistent between OpenStack projects. This ensures it can be used from client side to generate the trace, containing information from all possible resources.
sentinel_service_name = mymaster	string value	Redis sentinel uses a service name to identify a master redis service. This parameter defines the name (for example: sentinel_service_name=mymaster).
socket_timeout = 0.1	floating point value	Redis sentinel provides a timeout option on the connections. This parameter defines that timeout (for example: socket_timeout=0.1).
trace_sqlalchemy = False	boolean value	Enable SQL requests profiling in services. Default value is False (SQL requests won't be traced). Possible values: • True: Enables SQL requests profiling. Each SQL query will be part of the trace and can be analyzed by how much time was spent for that. • False: Disables SQL requests profiling. The spent time is only shown on a higher level of operations. Single SQL queries cannot be analyzed this way.

9.1.43. quota

The following table outlines the options available under the **[quota]** group in the `/etc/nova/nova.conf` file.

Table 9.42. quota

Configuration option = Default value	Type	Description
cores = 20	integer value	The number of instance cores or vCPUs allowed per project. Possible values: • A positive integer or 0. • -1 to disable the quota.
count_usage_from_placement = False	boolean value	Enable the counting of quota usage from the placement service. Starting in Train, it is possible to count quota usage for cores and ram from the placement service and instances from the API database instead of counting from cell databases. This works well if there is only one Nova deployment running per placement deployment. However, if an operator is running more than one Nova deployment sharing a placement deployment, they should not set this option to True because currently the placement service has no way to partition resource providers per Nova deployment. When this option is left as the default or set to False, Nova will use the legacy counting method to count quota usage for instances, cores, and ram from its cell databases. Note that quota usage behavior related to resizes will be affected if this option is set to True. Placement resource allocations are claimed on the destination while holding allocations on the source during a resize, until the resize is confirmed or reverted. During this time, when the server is in VERIFY_RESIZE state, quota usage will reflect resource consumption on both the source and the destination. This can be beneficial as it reserves space for a revert of a downsize, but it also means quota usage will be inflated until a resize is confirmed or reverted. Behavior will also be different for unscheduled servers in ERROR state. A server in ERROR state that has never been scheduled to a compute host will not have placement allocations, so it will not consume quota usage for cores and ram. Behavior will be different for servers in SHELVED_OFFLOADED state. A server in SHELVED_OFFLOADED state will not have placement allocations, so it will not consume quota usage for cores and ram. Note that because of this, it

Configuration option = Default value	Type	Description
		will be possible for a request to unshelve a server to be rejected if the user does not have enough quota available to support the cores and ram needed by the server to be unshelved. The populate_queued_for_delete and populate_user_id online data migrations must be completed before usage can be counted from placement. Until the data migration is complete, the system will fall back to legacy quota usage counting from cell databases depending on the result of an EXISTS database query during each quota check, if this configuration option is set to True. Operators who want to avoid the performance hit from the EXISTS queries should wait to set this configuration option to True until after they have completed their online data migrations via nova-manage db online_data_migrations.
driver = nova.quota.DbQuotaDriver	string value	Provides abstraction for quota checks. Users can configure a specific driver to use for quota checks.
injected_file_content_bytes = 10240	integer value	The number of bytes allowed per injected file. Possible values: • A positive integer or 0. • -1 to disable the quota.
injected_file_path_length = 255	integer value	The maximum allowed injected file path length. Possible values: • A positive integer or 0. • -1 to disable the quota.
injected_files = 5	integer value	The number of injected files allowed. File injection allows users to customize the personality of an instance by injecting data into it upon boot. Only text file injection is permitted: binary or ZIP files are not accepted. During file injection, any existing files that match specified files are renamed to include .bak extension appended with a timestamp. Possible values: • A positive integer or 0. • -1 to disable the quota.

Configuration option = Default value	Type	Description
instances = 10	integer value	The number of instances allowed per project. Possible Values • A positive integer or 0. • -1 to disable the quota.
key_pairs = 100	integer value	The maximum number of key pairs allowed per user. Users can create at least one key pair for each project and use the key pair for multiple instances that belong to that project. Possible values: • A positive integer or 0. • -1 to disable the quota.
metadata_items = 128	integer value	The number of metadata items allowed per instance. Users can associate metadata with an instance during instance creation. This metadata takes the form of key-value pairs. Possible values: • A positive integer or 0. • -1 to disable the quota.
ram = 51200	integer value	The number of megabytes of instance RAM allowed per project. Possible values: • A positive integer or 0. • -1 to disable the quota.

Configuration option = Default value	Type	Description
recheck_quota = True	boolean value	Recheck quota after resource creation to prevent allowing quota to be exceeded. This defaults to True (recheck quota after resource creation) but can be set to False to avoid additional load if allowing quota to be exceeded because of racing requests is considered acceptable. For example, when set to False, if a user makes highly parallel REST API requests to create servers, it will be possible for them to create more servers than their allowed quota during the race. If their quota is 10 servers, they might be able to create 50 during the burst. After the burst, they will not be able to create any more servers but they will be able to keep their 50 servers until they delete them. The initial quota check is done before resources are created, so if multiple parallel requests arrive at the same time, all could pass the quota check and create resources, potentially exceeding quota. When recheck_quota is True, quota will be checked a second time after resources have been created and if the resource is over quota, it will be deleted and OverQuota will be raised, usually resulting in a 403 response to the REST API user. This makes it impossible for a user to exceed their quota with the caveat that it will, however, be possible for a REST API user to be rejected with a 403 response in the event of a collision close to reaching their quota limit, even if the user has enough quota available when they made the request.
server_group_members = 10	integer value	The maximum number of servers per server group. Possible values: • A positive integer or 0. • -1 to disable the quota.

Configuration option = Default value	Type	Description
server_groups = 10	integer value	The maximum number of server groups per project. Server groups are used to control the affinity and anti-affinity scheduling policy for a group of servers or instances. Reducing the quota will not affect any existing group, but new servers will not be allowed into groups that have become over quota. Possible values: • A positive integer or 0. • -1 to disable the quota.

9.1.44. rdp

The following table outlines the options available under the **[rdp]** group in the `/etc/nova/nova.conf` file.

Table 9.43. rdp

Configuration option = Default value	Type	Description
enabled = False	boolean value	Enable Remote Desktop Protocol (RDP) related features. Hyper-V, unlike the majority of the hypervisors employed on Nova compute nodes, uses RDP instead of VNC and SPICE as a desktop sharing protocol to provide instance console access. This option enables RDP for graphical console access for virtual machines created by Hyper-V. Note: RDP should only be enabled on compute nodes that support the Hyper-V virtualization platform. Related options: • compute_driver: Must be hyperv.

Configuration option = Default value	Type	Description
html5_proxy_base_url = http://127.0.0.1:6083/	uri value	The URL an end user would use to connect to the RDP HTML5 console proxy. The console proxy service is called with this token-embedded URL and establishes the connection to the proper instance. An RDP HTML5 console proxy service will need to be configured to listen on the address configured here. Typically the console proxy service would be run on a controller node. The localhost address used as default would only work in a single node environment i.e. devstack. An RDP HTML5 proxy allows a user to access via the web the text or graphical console of any Windows server or workstation using RDP. RDP HTML5 console proxy services include FreeRDP, wsgate. See https://github.com/FreeRDP/FreeRDP-WebConnect Possible values: • <code><scheme>://<ip-address>:<port-number>/</code> - The scheme must be identical to the scheme configured for the RDP HTML5 console proxy service. It is <code>`http`</code> or <code>`https`</code>. - The IP address must be identical to the address on which the RDP HTML5 console proxy service is listening. - The port must be identical to the port on which the RDP HTML5 console proxy service is listening. Related options: • rdp.enabled: Must be set to True for html5_proxy_base_url to be effective.

9.1.45. remote_debug

The following table outlines the options available under the **[remote_debug]** group in the **/etc/nova/nova.conf** file.

Table 9.44. remote_debug

Configuration option = Default value	Type	Description
host = None	host address value	Debug host (IP or name) to connect to. This command line parameter is used when you want to connect to a nova service via a debugger running on a different host. Note that using the remote debug option changes how nova uses the eventlet library to support async IO. This could result in failures that do not occur under normal operation. Use at your own risk. Possible Values: ● IP address of a remote host as a command line parameter to a nova service. For example <pre>nova-compute --config-file /etc/nova/nova.conf --remote_debug- host <IP address of the debugger></pre>
port = None	port value	Debug port to connect to. This command line parameter allows you to specify the port you want to use to connect to a nova service via a debugger running on different host. Note that using the remote debug option changes how nova uses the eventlet library to support async IO. This could result in failures that do not occur under normal operation. Use at your own risk. Possible Values: ● Port number you want to use as a command line parameter to a nova service. For example <pre>nova-compute --config-file /etc/nova/nova.conf --remote_debug- host <IP address of the debugger> -- remote_debug-port <port debugger is listening on>.</pre>

9.1.46. scheduler

The following table outlines the options available under the **[scheduler]** group in the **/etc/nova/nova.conf** file.

Table 9.45. scheduler

Configuration option = Default value	Type	Description
discover_hosts_in_cells_interval = -1	integer value	Periodic task interval. This value controls how often (in seconds) the scheduler should attempt to discover new hosts that have been added to cells. If negative (the default), no automatic discovery will occur. Deployments where compute nodes come and go frequently may want this enabled, where others may prefer to manually discover hosts when one is added to avoid any overhead from constantly checking. If enabled, every time this runs, we will select any unmapped hosts out of each cell database on every run. Possible values: ● An integer, where the integer corresponds to periodic task interval in seconds. 0 uses the default interval (60 seconds). A negative value disables periodic tasks.
enable_isolated_aggregate_filtering = False	boolean value	Restrict use of aggregates to instances with matching metadata. This setting allows the scheduler to restrict hosts in aggregates based on matching required traits in the aggregate metadata and the instance flavor/image. If an aggregate is configured with a property with key trait:\$TRAIT_NAME and value required, the instance flavor extra_specs and/or image metadata must also contain trait:\$TRAIT_NAME=required to be eligible to be scheduled to hosts in that aggregate. More technical details at https://docs.openstack.org/nova/latest/reference/isolate-aggregates.html Possible values: ● A boolean value.

Configuration option = Default value	Type	Description
image_metadata_prefilter = False	boolean value	Use placement to filter hosts based on image metadata. This setting causes the scheduler to transform well known image metadata properties into placement required traits to filter host based on image metadata. This feature requires host support and is currently supported by the following compute drivers: ● libvirt.LibvirtDriver (since Ussuri (21.0.0)) Possible values: ● A boolean value. Related options: ● [compute] compute_driver
limit_tenants_to_placement_aggregate = False	boolean value	Restrict tenants to specific placement aggregates. This setting causes the scheduler to look up a host aggregate with the metadata key of filter_tenant_id set to the project of an incoming request, and request results from placement be limited to that aggregate. Multiple tenants may be added to a single aggregate by appending a serial number to the key, such as filter_tenant_id:123. The matching aggregate UUID must be mirrored in placement for proper operation. If no host aggregate with the tenant id is found, or that aggregate does not match one in placement, the result will be the same as not finding any suitable hosts for the request. Possible values: ● A boolean value. Related options: ● [scheduler] placement_aggregate_required_for_tenants

Configuration option = Default value	Type	Description
max_attempts = 3	integer value	The maximum number of schedule attempts. This is the maximum number of attempts that will be made for a given instance build/move operation. It limits the number of alternate hosts returned by the scheduler. When that list of hosts is exhausted, a MaxRetriesExceeded exception is raised and the instance is set to an error state. Possible values: • A positive integer, where the integer corresponds to the max number of attempts that can be made when building or moving an instance.
max_placement_results = 1000	integer value	The maximum number of placement results to request. This setting determines the maximum limit on results received from the placement service during a scheduling operation. It effectively limits the number of hosts that may be considered for scheduling requests that match a large number of candidates. A value of 1 (the minimum) will effectively defer scheduling to the placement service strictly on "will it fit" grounds. A higher value will put an upper cap on the number of results the scheduler will consider during the filtering and weighing process. Large deployments may need to set this lower than the total number of hosts available to limit memory consumption, network traffic, etc. of the scheduler. Possible values: • An integer, where the integer corresponds to the number of placement results to return.

Configuration option = Default value	Type	Description
placement_aggregate_required_for_tenants = False	boolean value	Require a placement aggregate association for all tenants. This setting, when <code>limit_tenants_to_placement_aggregate=True</code>, will control whether or not a tenant with no aggregate affinity will be allowed to schedule to any available node. If aggregates are used to limit some tenants but not all, then this should be <code>False</code>. If all tenants should be confined via aggregate, then this should be <code>True</code> to prevent them from receiving unrestricted scheduling to any available node. Possible values: • A boolean value. Related options: • [scheduler] placement_aggregate_required_for_tenants
query_placement_for_availability_zone = False	boolean value	Use placement to determine availability zones. This setting causes the scheduler to look up a host aggregate with the metadata key of availability_zone set to the value provided by an incoming request, and request results from placement be limited to that aggregate. The matching aggregate UUID must be mirrored in placement for proper operation. If no host aggregate with the availability_zone key is found, or that aggregate does not match one in placement, the result will be the same as not finding any suitable hosts. Note that if you enable this flag, you can disable the (less efficient) <code>AvailabilityZoneFilter</code> in the scheduler. Possible values: • A boolean value. Related options: • [filter_scheduler] enabled_filters

Configuration option = Default value	Type	Description
query_placement_for_image_type_support = False	boolean value	Use placement to determine host support for the instance's image type. This setting causes the scheduler to ask placement only for compute hosts that support the disk_format of the image used in the request. Possible values: • A boolean value.
query_placement_for_routed_network_aggregates = False	boolean value	Enable the scheduler to filter compute hosts affined to routed network segment aggregates. See https://docs.openstack.org/neutron/latest/admin/config-routed-networks.html for details.
workers = None	integer value	Number of workers for the nova-scheduler service. Defaults to the number of CPUs available. Possible values: • An integer, where the integer corresponds to the number of worker processes.

9.1.47. serial_console

The following table outlines the options available under the **[serial_console]** group in the **/etc/nova/nova.conf** file.

Table 9.46. serial_console

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
base_url = ws://127.0.0.1:6083/	uri value	The URL an end user would use to connect to the nova-serialproxy service. The nova-serialproxy service is called with this token enriched URL and establishes the connection to the proper instance. Related options: • The IP address must be identical to the address to which the nova-serialproxy service is listening (see option serialproxy_host in this section). • The port must be the same as in the option serialproxy_port of this section. • If you choose to use a secured websocket connection, then start this option with wss:// instead of the unsecured ws://. The options cert and key in the [DEFAULT] section have to be set for that.
enabled = False	boolean value	Enable the serial console feature. In order to use this feature, the service nova-serialproxy needs to run. This service is typically executed on the controller node.
port_range = 10000:20000	string value	A range of TCP ports a guest can use for its backend. Each instance which gets created will use one port out of this range. If the range is not big enough to provide another port for an new instance, this instance won't get launched. Possible values: • Each string which passes the regex ^\d+:\d+\$ For example 10000:20000. Be sure that the first port number is lower than the second port number and that both are in range from 0 to 65535.
proxycient_address = 127.0.0.1	string value	The IP address to which proxy clients (like nova-serialproxy) should connect to get the serial console of an instance. This is typically the IP address of the host of a nova-compute service.

Configuration option = Default value	Type	Description
serialproxy_host = 0.0.0.0	string value	The IP address which is used by the nova-serialproxy service to listen for incoming requests. The nova-serialproxy service listens on this IP address for incoming connection requests to instances which expose serial console. Related options: Ensure that this is the same IP address which is defined in the option base_url of this section or use 0.0.0.0 to listen on all addresses.
serialproxy_port = 6083	port value	The port number which is used by the nova-serialproxy service to listen for incoming requests. The nova-serialproxy service listens on this port number for incoming connection requests to instances which expose serial console. Related options: Ensure that this is the same port number which is defined in the option base_url of this section.

9.1.48. service_user

The following table outlines the options available under the **[service_user]** group in the **/etc/nova/nova.conf** file.

Table 9.47. service_user

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.

Configuration option = Default value	Type	Description
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
send_service_user_token = False	boolean value	When True, if sending a user token to a REST API, also send a service token. Nova often reuses the user token provided to the nova-api to talk to other REST APIs, such as Cinder, Glance and Neutron. It is possible that while the user token was valid when the request was made to Nova, the token may expire before it reaches the other service. To avoid any failures, and to make it clear it is Nova calling the service on the user's behalf, we include a service token along with the user token. Should the user's token have expired, a valid service token ensures the REST API request will still be accepted by the keystone middleware.

Configuration option = Default value	Type	Description
split-loggers = False	boolean value	Log requests to multiple loggers.
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User ID
username = None	string value	Username

9.1.49. spice

The following table outlines the options available under the **[spice]** group in the `/etc/nova/nova.conf` file.

Table 9.48. spice

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
agent_enabled = True	boolean value	Enable the SPICE guest agent support on the instances. The Spice agent works with the Spice protocol to offer a better guest console experience. However, the Spice console can still be used without the Spice Agent. With the Spice agent installed the following features are enabled: ● Copy & Paste of text and images between the guest and client machine ● Automatic adjustment of resolution when the client screen changes – e.g. if you make the Spice console full screen the guest resolution will adjust to match it rather than letterboxing. ● Better mouse integration – The mouse can be captured and released without needing to click inside the console or press keys to release it. The performance of mouse movement is also improved.
enabled = False	boolean value	Enable SPICE related features. Related options: ● VNC must be explicitly disabled to get access to the SPICE console. Set the enabled option to False in the [vnc] section to disable the VNC console.

Configuration option = Default value	Type	Description
html5proxy_base_url = http://127.0.0.1:6082/spice_auto.html	uri value	Location of the SPICE HTML5 console proxy. End user would use this URL to connect to the nova-spicehtml5proxy service. This service will forward request to the console of an instance. In order to use SPICE console, the service nova-spicehtml5proxy should be running. This service is typically launched on the controller node. Possible values: • Must be a valid URL of the form: http://host:port/spice_auto.html where host is the node running nova-spicehtml5proxy and the port is typically 6082. Consider not using default value as it is not well defined for any real deployment. Related options: • This option depends on html5proxy_host and html5proxy_port options. The access URL returned by the compute node must have the host and port where the nova-spicehtml5proxy service is listening.
html5proxy_host = 0.0.0.0	host address value	IP address or a hostname on which the nova-spicehtml5proxy service listens for incoming requests. Related options: • This option depends on the html5proxy_base_url option. The nova-spicehtml5proxy service must be listening on a host that is accessible from the HTML5 client.
html5proxy_port = 6082	port value	Port on which the nova-spicehtml5proxy service listens for incoming requests. Related options: • This option depends on the html5proxy_base_url option. The nova-spicehtml5proxy service must be listening on a port that is accessible from the HTML5 client.

Configuration option = Default value	Type	Description
server_listen = 127.0.0.1	string value	The address where the SPICE server running on the instances should listen. Typically, the nova-spicehtml5proxy proxy client runs on the controller node and connects over the private network to this address on the compute node(s). Possible values: ● IP address to listen on.
server_proxyclient_address = 127.0.0.1	string value	The address used by nova-spicehtml5proxy client to connect to instance console. Typically, the nova-spicehtml5proxy proxy client runs on the controller node and connects over the private network to this address on the compute node(s). Possible values: ● Any valid IP address on the compute node. Related options: ● This option depends on the server_listen option. The proxy client must be able to access the address specified in server_listen using the value of this option.

9.1.50. upgrade_levels

The following table outlines the options available under the **[upgrade_levels]** group in the **/etc/nova/nova.conf** file.

Table 9.49. upgrade_levels

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
baseapi = None	string value	Base API RPC API version cap. Possible values: ● By default send the latest version the client knows about ● A string representing a version number in the format <i>N.N</i>; for example, possible values might be <i>1.12</i> or <i>2.0</i>. ● An OpenStack release name, in lower case, such as <i>mitaka</i> or <i>liberty</i>.
cert = None	string value	Cert RPC API version cap. Possible values: ● By default send the latest version the client knows about ● A string representing a version number in the format <i>N.N</i>; for example, possible values might be <i>1.12</i> or <i>2.0</i>. ● An OpenStack release name, in lower case, such as <i>mitaka</i> or <i>liberty</i>. Deprecated since: 18.0.0 Reason: The nova-cert service was removed in 16.0.0 (Pike) so this option is no longer used.

Configuration option = Default value	Type	Description
compute = None	string value	Compute RPC API version cap. By default, we always send messages using the most recent version the client knows about. Where you have old and new compute services running, you should set this to the lowest deployed version. This is to guarantee that all services never send messages that one of the compute nodes can't understand. Note that we only support upgrading from release N to release N+1. Set this option to "auto" if you want to let the compute RPC module automatically determine what version to use based on the service versions in the deployment. Possible values: ● By default send the latest version the client knows about ● <i>auto</i>: Automatically determines what version to use based on the service versions in the deployment. ● A string representing a version number in the format <i>N.N</i>; for example, possible values might be <i>1.12</i> or <i>2.0</i>. ● An OpenStack release name, in lower case, such as <i>mitaka</i> or <i>liberty</i>.
conductor = None	string value	Conductor RPC API version cap. Possible values: ● By default send the latest version the client knows about ● A string representing a version number in the format <i>N.N</i>; for example, possible values might be <i>1.12</i> or <i>2.0</i>. ● An OpenStack release name, in lower case, such as <i>mitaka</i> or <i>liberty</i>.

Configuration option = Default value	Type	Description
scheduler = None	string value	Scheduler RPC API version cap. Possible values: ● By default send the latest version the client knows about ● A string representing a version number in the format <i>N.N</i>; for example, possible values might be <i>1.12</i> or <i>2.0</i>. ● An OpenStack release name, in lower case, such as <i>mitaka</i> or <i>liberty</i>.

9.1.51. vault

The following table outlines the options available under the **[vault]** group in the `/etc/nova/nova.conf` file.

Table 9.50. vault

Configuration option = Default value	Type	Description
approle_role_id = None	string value	AppRole role_id for authentication with vault
approle_secret_id = None	string value	AppRole secret_id for authentication with vault
kv_mountpoint = secret	string value	Mountpoint of KV store in Vault to use, for example: secret
kv_version = 2	integer value	Version of KV store in Vault to use, for example: 2
root_token_id = None	string value	root token for vault
ssl_ca_cert_file = None	string value	Absolute path to ca cert file
use_ssl = False	boolean value	SSL Enabled/Disabled
vault_url = http://127.0.0.1:8200	string value	Use this endpoint to connect to Vault, for example: "http://127.0.0.1:8200"

9.1.52. vendordata_dynamic_auth

The following table outlines the options available under the **[vendordata_dynamic_auth]** group in the `/etc/nova/nova.conf` file.

Table 9.51. vendordata_dynamic_auth

Configuration option = Default value	Type	Description
auth-url = None	string value	Authentication URL
auth_section = None	string value	Config Section from which to load plugin specific options
auth_type = None	string value	Authentication type to load
cafile = None	string value	PEM encoded Certificate Authority to use when verifying HTTPs connections.
certfile = None	string value	PEM encoded client certificate cert file
collect-timing = False	boolean value	Collect per-API call timing information.
default-domain-id = None	string value	Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
default-domain-name = None	string value	Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.
domain-id = None	string value	Domain ID to scope to
domain-name = None	string value	Domain name to scope to
insecure = False	boolean value	Verify HTTPS connections.
keyfile = None	string value	PEM encoded client certificate key file
password = None	string value	User's password
project-domain-id = None	string value	Domain ID containing project
project-domain-name = None	string value	Domain name containing project
project-id = None	string value	Project ID to scope to
project-name = None	string value	Project name to scope to
split-loggers = False	boolean value	Log requests to multiple loggers.

Configuration option = Default value	Type	Description
system-scope = None	string value	Scope for system operations
tenant-id = None	string value	Tenant ID
tenant-name = None	string value	Tenant Name
timeout = None	integer value	Timeout value for http requests
trust-id = None	string value	Trust ID
user-domain-id = None	string value	User's domain id
user-domain-name = None	string value	User's domain name
user-id = None	string value	User ID
username = None	string value	Username

9.1.53. vmware

The following table outlines the options available under the **[vmware]** group in the **/etc/nova/nova.conf** file.

Table 9.52. vmware

Configuration option = Default value	Type	Description
api_retry_count = 10	integer value	Number of times VMware vCenter server API must be retried on connection failures, e.g. socket error, etc.
ca_file = None	string value	Specifies the CA bundle file to be used in verifying the vCenter server certificate.

Configuration option = Default value	Type	Description
cache_prefix = None	string value	This option adds a prefix to the folder where cached images are stored This is not the full path - just a folder prefix. This should only be used when a datastore cache is shared between compute nodes. Note: This should only be used when the compute nodes are running on same host or they have a shared file system. Possible values: Any string representing the cache prefix to the folder
cluster_name = None	string value	Name of a VMware Cluster ComputeResource.
connection_pool_size = 10	integer value	This option sets the http connection pool size The connection pool size is the maximum number of connections from nova to vSphere. It should only be increased if there are warnings indicating that the connection pool is full, otherwise, the default should suffice.
console_delay_seconds = None	integer value	Set this value if affected by an increased network latency causing repeated characters when typing in a remote console.
datastore_regex = None	string value	Regular expression pattern to match the name of datastore. The datastore_regex setting specifies the datastores to use with Compute. For example, datastore_regex="nas.*" selects all the data stores that have a name starting with "nas".  NOTE If no regex is given, it just picks the datastore with the most freespace. Possible values: Any matching regular expression to a datastore must be given
host_ip = None	host address value	Hostname or IP address for connection to VMware vCenter host.

Configuration option = Default value	Type	Description
host_password = None	string value	Password for connection to VMware vCenter host.
host_port = 443	port value	Port for connection to VMware vCenter host.
host_username = None	string value	Username for connection to VMware vCenter host.
insecure = False	boolean value	If true, the vCenter server certificate is not verified. If false, then the default CA truststore is used for verification. Related options: * ca_file: This option is ignored if "ca_file" is set.
integration_bridge = None	string value	This option should be configured only when using the NSX-MH Neutron plugin. This is the name of the integration bridge on the ESXi server or host. This should not be set for any other Neutron plugin. Hence the default value is not set. Possible values: Any valid string representing the name of the integration bridge
maximum_objects = 100	integer value	This option specifies the limit on the maximum number of objects to return in a single result. A positive value will cause the operation to suspend the retrieval when the count of objects reaches the specified limit. 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	string value	This option specifies the default policy to be used. If pbm_enabled is set and there is no defined storage policy for the specific request, then this policy will be used. Possible values: - Any valid storage policy such as VSAN default storage policy Related options: - pbm_enabled

Configuration option = Default value	Type	Description
pbm_enabled = False	boolean value	This option enables or disables storage policy based placement of instances. Related options: ● <code>pbm_default_policy</code>
pbm_wsdl_location = None	string value	This option specifies the PBM service WSDL file location URL. Setting this will disable storage policy based placement of instances. Possible values: ● Any valid file path e.g <code>file:///opt/SDK/spbm/wsdl/pbmService.wsdl</code>
serial_log_dir = /opt/vmware/vspc	string value	Specifies the directory where the Virtual Serial Port Concentrator is storing console log files. It should match the <code>serial_log_dir</code> config value of VSPC.
serial_port_proxy_uri = None	uri value	Identifies a proxy service that provides network access to the <code>serial_port_service_uri</code>. Possible values: ● Any valid URI (The scheme is <code>telnet</code> or <code>telnets</code>.) Related options: This option is ignored if <code>serial_port_service_uri</code> is not specified. *
serial_port_service_uri = None	string value	Identifies the remote system where the serial port traffic will be sent. This option adds a virtual serial port which sends console output to a configurable service URI. At the service URI address there will be virtual serial port concentrator that will collect console logs. If this is not set, no serial ports will be added to the created VMs. Possible values: ● Any valid URI

Configuration option = Default value	Type	Description
task_poll_interval = 0.5	floating point value	Time interval in seconds to poll remote tasks invoked on VMware VC server.
use_linked_clone = True	boolean value	This option enables/disables the use of linked clone. The ESX hypervisor requires a copy of the VMDK file in order to boot up a virtual machine. The compute driver must download the VMDK via HTTP from the OpenStack Image service to a datastore that is visible to the hypervisor and cache it. 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. If set to false, even with a cached VMDK, there is still a copy operation from the cache location to the hypervisor file directory in the shared datastore. If set to true, the above copy operation is avoided as it creates copy of the virtual machine that shares virtual disks with its parent VM.
vnc_keymap = en-us	string value	Keymap for VNC. The keyboard mapping (keymap) determines which keyboard layout a VNC session should use by default. Possible values: ● A keyboard layout which is supported by the underlying hypervisor on this node. This is usually an <i>IETF language tag</i> (for example <i>en-us</i>).
vnc_port = 5900	port value	This option specifies VNC starting port. Every VM created by ESX host has an option of enabling VNC client for remote connection. Above option <i>vnc_port</i> helps you to set default starting port for the VNC client. Possible values: ● Any valid port number within 5900 -(5900 + vnc_port_total) Related options: Below options should be set to enable VNC client. * vnc.enabled = True * vnc_port_total

Configuration option = Default value	Type	Description
vnc_port_total = 10000	integer value	Total number of VNC ports.

9.1.54. vnc

The following table outlines the options available under the **[vnc]** group in the `/etc/nova/nova.conf` file.

Table 9.53. vnc

Configuration option = Default value	Type	Description
auth_schemes = ['none']	list value	The authentication schemes to use with the compute node. Control what RFB authentication schemes are permitted for connections between the proxy and the compute host. If multiple schemes are enabled, the first matching scheme will be used, thus the strongest schemes should be listed first. Related options: • [vnc]vencrypt_client_key, [vnc]vencrypt_client_cert: must also be set
enabled = True	boolean value	Enable VNC related features. Guests will get created with graphical devices to support this. Clients (for example Horizon) can then establish a VNC connection to the guest.

Configuration option = Default value	Type	Description
novncproxy_base_url = http://127.0.0.1:6080/vnc_ auto.html	uri value	Public address of noVNC VNC console proxy. The VNC proxy is an OpenStack component that enables compute service users to access their instances through VNC clients. noVNC provides VNC support through a websocket-based client. This option sets the public base URL to which client systems will connect. noVNC clients can use this address to connect to the noVNC instance and, by extension, the VNC sessions. If using noVNC >= 1.0.0, you should use vnc_lite.html instead of vnc_auto.html. Related options: ● novncproxy_host ● novncproxy_port
novncproxy_host = 0.0.0.0	string value	IP address that the noVNC console proxy should bind to. The VNC proxy is an OpenStack component that enables compute service users to access their instances through VNC clients. noVNC provides VNC support through a websocket-based client. This option sets the private address to which the noVNC console proxy service should bind to. Related options: ● novncproxy_port ● novncproxy_base_url

Configuration option = Default value	Type	Description
novncproxy_port = 6080	port value	Port that the noVNC console proxy should bind to. The VNC proxy is an OpenStack component that enables compute service users to access their instances through VNC clients. noVNC provides VNC support through a websocket-based client. This option sets the private port to which the noVNC console proxy service should bind to. Related options: ● novncproxy_host ● novncproxy_base_url
server_listen = 127.0.0.1	host address value	The IP address or hostname on which an instance should listen to for incoming VNC connection requests on this node.
server_proxyclient_address = 127.0.0.1	host address value	Private, internal IP address or hostname of VNC console proxy. The VNC proxy is an OpenStack component that enables compute service users to access their instances through VNC clients. This option sets the private address to which proxy clients, such as nova-novncproxy, should connect to.
vencrypt_ca_certs = None	string value	The path to the CA certificate PEM file The fully qualified path to a PEM file containing one or more x509 certificates for the certificate authorities used by the compute node VNC server. Related options: ● vnc.auth_schemes: must include vencrypt

Configuration option = Default value	Type	Description
vencrypt_client_cert = None	string value	The path to the client key file (for x509) The fully qualified path to a PEM file containing the x509 certificate which the VNC proxy server presents to the compute node during VNC authentication. Related options: ● vnc.auth_schemes: must include vencrypt ● vnc.vencrypt_client_key: must also be set
vencrypt_client_key = None	string value	The path to the client certificate PEM file (for x509) The fully qualified path to a PEM file containing the private key which the VNC proxy server presents to the compute node during VNC authentication. Related options: ● vnc.auth_schemes: must include vencrypt ● vnc.vencrypt_client_cert: must also be set

9.1.55. workarounds

The following table outlines the options available under the **[workarounds]** group in the **/etc/nova/nova.conf** file.

Table 9.54. workarounds

Configuration option = Default value	Type	Description
disable_compute_service_check_for_ffu = False	boolean value	If this is set, the normal safety check for old compute services will be treated as a warning instead of an error. This is only to be enabled to facilitate a Fast-Forward upgrade where new control services are being started before compute nodes have been able to update their service record. In an FFU, the service records in the database will be more than one version old until the compute nodes start up, but control services need to be online first.

Configuration option = Default value	Type	Description
disable_fallback_pcpu_query = False	boolean value	Disable fallback request for VCPU allocations when using pinned instances. Starting in Train, compute nodes using the libvirt virt driver can report PCPU inventory and will use this for pinned instances. The scheduler will automatically translate requests using the legacy CPU pinning-related flavor extra specs, hw:cpu_policy and hw:cpu_thread_policy, their image metadata property equivalents, and the emulator threads pinning flavor extra spec, hw:emulator_threads_policy, to new placement requests. However, compute nodes require additional configuration in order to report PCPU inventory and this configuration may not be present immediately after an upgrade. To ensure pinned instances can be created without this additional configuration, the scheduler will make a second request to placement for old-style VCPU-based allocations and fallback to these allocation candidates if necessary. This has a slight performance impact and is not necessary on new or upgraded deployments where the new configuration has been set on all hosts. By setting this option, the second lookup is disabled and the scheduler will only request PCPU-based allocations. Deprecated since: 20.0.0 *Reason:*None

Configuration option = Default value	Type	Description
disable_group_policy_check_upcall = False	boolean value	Disable the server group policy check upcall in compute. In order to detect races with server group affinity policy, the compute service attempts to validate that the policy was not violated by the scheduler. It does this by making an upcall to the API database to list the instances in the server group for one that it is booting, which violates our api/cell isolation goals. Eventually this will be solved by proper affinity guarantees in the scheduler and placement service, but until then, this late check is needed to ensure proper affinity policy. Operators that desire api/cell isolation over this check should enable this flag, which will avoid making that upcall from compute. Related options: • [filter_scheduler]/track_instance_changes also relies on upcalls from the compute service to the scheduler service.

Configuration option = Default value	Type	Description
disable_libvirt_livesnaps hot = False	boolean value	Disable live snapshots when using the libvirt driver. Live snapshots allow the snapshot of the disk to happen without an interruption to the guest, using coordination with a guest agent to quiesce the filesystem. When using libvirt 1.2.2 live snapshots fail intermittently under load (likely related to concurrent libvirt/qemu operations). This config option provides a mechanism to disable live snapshot, in favor of cold snapshot, while this is resolved. Cold snapshot causes an instance outage while the guest is going through the snapshotting process. For more information, refer to the bug report: https://bugs.launchpad.net/nova/+bug/1334398 Possible values: • True: Live snapshot is disabled when using libvirt • False: Live snapshots are always used when snapshotting (as long as there is a new enough libvirt and the backend storage supports it) Deprecated since: 19.0.0 Reason: This option was added to work around issues with libvirt 1.2.2. We no longer support this version of libvirt, which means this workaround is no longer necessary. It will be removed in a future release.

Configuration option = Default value	Type	Description
disable_native_luksv1 = False	boolean value	When attaching encrypted LUKSv1 Cinder volumes to instances the Libvirt driver configures the encrypted disks to be natively decrypted by QEMU. A performance issue has been discovered in the libgcrypt library used by QEMU that serverly limits the I/O performance in this scenario. For more information please refer to the following bug report: RFE: hardware accelerated AES-XTS mode https://bugzilla.redhat.com/show_bug.cgi?id=1762765 Enabling this workaround option will cause Nova to use the legacy dm-crypt based os-brick encryptor to decrypt the LUKSv1 volume. Note that enabling this option while using volumes that do not provide a host block device such as Ceph will result in a failure to boot from or attach the volume to an instance. See the [workarounds]/rbd_block_device option for a way to avoid this for RBD. Related options: ● compute_driver (libvirt) ● rbd_block_device (workarounds) Deprecated since: 23.0.0 Reason: The underlying performance regression within libgcrypt that prompted this workaround has been resolved as of 1.8.5

Configuration option = Default value	Type	Description
disable_rootwrap = False	boolean value	Use sudo instead of rootwrap. Allow fallback to sudo for performance reasons. For more information, refer to the bug report: https://bugs.launchpad.net/nova/+bug/1415106 Possible values: • True: Use sudo instead of rootwrap • False: Use rootwrap as usual Interdependencies to other options: • Any options that affect <i>rootwrap</i> will be ignored.

Configuration option = Default value	Type	Description
enable_numa_live_migration = False	boolean value	Enable live migration of instances with NUMA topologies. Live migration of instances with NUMA topologies when using the libvirt driver is only supported in deployments that have been fully upgraded to Train. In previous versions, or in mixed Stein/Train deployments with a rolling upgrade in progress, live migration of instances with NUMA topologies is disabled by default when using the libvirt driver. This includes live migration of instances with CPU pinning or hugepages. CPU pinning and huge page information for such instances is not currently re-calculated, as noted in <code>`bug #1289064`_</code>. This means that if instances were already present on the destination host, the migrated instance could be placed on the same dedicated cores as these instances or use hugepages allocated for another instance. Alternately, if the host platforms were not homogeneous, the instance could be assigned to non-existent cores or be inadvertently split across host NUMA nodes. Despite these known issues, there may be cases where live migration is necessary. By enabling this option, operators that are aware of the issues and are willing to manually work around them can enable live migration support for these instances. Related options: ● compute_driver: Only the libvirt driver is affected. a. <code>_bug #1289064</code>: https://bugs.launchpad.net/nova/+bug/1289064 Deprecated since: 20.0.0 *Reason:*This option was added to mitigate known issues when live migrating instances with a NUMA topology with the libvirt driver. Those issues are resolved in Train. Clouds using the libvirt driver and fully upgraded to Train support NUMA-aware live migration. This option will be removed in a future release.

Configuration option = Default value	Type	Description
enable_gemu_monitor_announce_self = False	boolean value	If it is set to True the libvirt driver will try as a best effort to send the announce-self command to the QEMU monitor so that it generates RARP frames to update network switches in the post live migration phase on the destination. Please note that this causes the domain to be considered tainted by libvirt. Related options: • :oslo.config.option:DEFAULT.compute_driver (libvirt)
ensure_libvirt_rbd_instance_dir_cleanup = False	boolean value	Ensure the instance directory is removed during clean up when using rbd. When enabled this workaround will ensure that the instance directory is always removed during cleanup on hosts using [libvirt]/images_type=rbd. This avoids the following bugs with evacuation and revert resize clean up that lead to the instance directory remaining on the host: https://bugs.launchpad.net/nova/+bug/1414895 https://bugs.launchpad.net/nova/+bug/1761062 Both of these bugs can then result in DestinationDiskExists errors being raised if the instances ever attempt to return to the host. a. warning:: Operators will need to ensure that the instance directory itself, specified by [DEFAULT]/instances_path, is not shared between computes before enabling this workaround otherwise the console.log, kernels, ramdisks and any additional files being used by the running instance will be lost. Related options: • compute_driver (libvirt) • [libvirt]/images_type (rbd) • instances_path

Configuration option = Default value	Type	Description
handle_virt_lifecycle_events = True	boolean value	Enable handling of events emitted from compute drivers. Many compute drivers emit lifecycle events, which are events that occur when, for example, an instance is starting or stopping. If the instance is going through task state changes due to an API operation, like resize, the events are ignored. This is an advanced feature which allows the hypervisor to signal to the compute service that an unexpected state change has occurred in an instance and that the instance can be shutdown automatically. Unfortunately, this can race in some conditions, for example in reboot operations or when the compute service or when host is rebooted (planned or due to an outage). If such races are common, then it is advisable to disable this feature. Care should be taken when this feature is disabled and <i>sync_power_state_interval</i> is set to a negative value. In this case, any instances that get out of sync between the hypervisor and the Nova database will have to be synchronized manually. For more information, refer to the bug report: https://bugs.launchpad.net/bugs/1444630 Interdependencies to other options: ● If sync_power_state_interval is negative and this feature is disabled, then instances that get out of sync between the hypervisor and the Nova database will have to be synchronized manually.
libvirt_disable_apic = False	boolean value	With some kernels initializing the guest apic can result in a kernel hang that renders the guest unusable. This happens as a result of a kernel bug. In most cases the correct fix is to update the guest image kernel to one that is patched however in some cases this is not possible. This workaround allows the emulation of an apic to be disabled per host however it is not recommended to use outside of a CI or developer cloud.

Configuration option = Default value	Type	Description
never_download_image_if_on_rbd = False	boolean value	When booting from an image on a ceph-backed compute node, if the image does not already reside on the ceph cluster (as would be the case if glance is also using the same cluster), nova will download the image from glance and upload it to ceph itself. If using multiple ceph clusters, this may cause nova to unintentionally duplicate the image in a non-COW-able way in the local ceph deployment, wasting space. For more information, refer to the bug report: https://bugs.launchpad.net/nova/+bug/1858877 Enabling this option will cause nova to refuse to boot an instance if it would require downloading the image from glance and uploading it to ceph itself. Related options: • compute_driver (libvirt) • [libvirt]/images_type (rbd)

Configuration option = Default value	Type	Description
rbd_volume_local_attach = False	boolean value	Attach RBD Cinder volumes to the compute as host block devices. When enabled this option instructs os-brick to connect RBD volumes locally on the compute host as block devices instead of natively through QEMU. This workaround does not currently support extending attached volumes. This can be used with the <code>disable_native_luksv1</code> workaround configuration option to avoid the recently discovered performance issues found within the libgcrypt library. This workaround is temporary and will be removed during the W release once all impacted distributions have been able to update their versions of the libgcrypt library. Related options: • compute_driver (libvirt) • disable_qemu_native_luksv1 (workarounds) Deprecated since: 23.0.0 Reason: The underlying performance regression within libgcrypt that prompted this workaround has been resolved as of 1.8.5

Configuration option = Default value	Type	Description
reserve_disk_resource_for_image_cache = False	boolean value	If it is set to True then the libvirt driver will reserve DISK_GB resource for the images stored in the image cache. If the :oslo.config.option:DEFAULT.instances_path is on different disk partition than the image cache directory then the driver will not reserve resource for the cache. Such disk reservation is done by a periodic task in the resource tracker that runs every :oslo.config.option:update_resources_interval seconds. So the reservation is not updated immediately when an image is cached. Related options: • :oslo.config.option:DEFAULT.instances_path • :oslo.config.option:image_cache.subdirectory_name • :oslo.config.option:update_resources_interval
skip_cpu_compare_on_dest = False	boolean value	When this is enabled, it will skip CPU comparison on the destination host. When using QEMU >= 2.9 and libvirt >= 4.4.0, libvirt will do the correct thing with respect to checking CPU compatibility on the destination host during live migration.
skip_hypervisor_version_check_on_lm = False	boolean value	When this is enabled, it will skip version-checking of hypervisors during live migration.
wait_for_vif_plugged_event_during_hard_reboot = []	list value	The libvirt virt driver implements power on and hard reboot by tearing down every vif of the instance being rebooted then plug them again. By default nova does not wait for network-vif-plugged event from neutron before it lets the instance run. This can cause the instance to requests the IP via DHCP before the neutron backend has a chance to set up the networking backend after the vif plug. This flag defines which vifs nova expects network-vif-plugged events from during hard reboot. The possible values are neutron port vnic types: • normal • direct • baremetal

Configuration option = Default value	Type	• direct-physical • virtio-forwarder Description
		• smart-nic • vdpa • accelerator-direct • accelerator-direct-physical Adding a vnic_type to this configuration makes Nova wait for a network-vif-plugged event for each of the instance's vifs having the specific vnic_type before unpausing the instance, similarly to how new instance creation works. Please note that not all neutron networking backends send plug time events, for certain vnic_type therefore this config is empty by default. The ml2/ovs and the networking-odl backends are known to send plug time events for ports with normal vnic_type so it is safe to add normal to this config if you are using only those backends in the compute host. The neutron in-tree SRIOV backend does not reliably send network-vif-plugged event during plug time for ports with direct vnic_type and never sends that event for port with direct-physical vnic_type during plug time. For other vnic_type and backend pairs, please consult the developers of the backend. Related options: • :oslo.config:option:DEFAULT.vif_plugging_timeout

9.1.56. wsgi

The following table outlines the options available under the **[wsgi]** group in the `/etc/nova/nova.conf` file.

Table 9.55. wsgi

Configuration option = Default value	Type	Description

Configuration option = Default value	Type	Description
api_paste_config = api-paste.ini	string value	This option represents a file name for the paste.deploy config for nova-api. Possible values: • A string representing file name for the paste.deploy config.
client_socket_timeout = 900	integer value	This option specifies the timeout for client connections' socket operations. If an incoming connection is idle for this number of seconds it will be closed. It indicates timeout on individual read/writes on the socket connection. To wait forever set to 0.
default_pool_size = 1000	integer value	This option specifies the size of the pool of greenthreads used by wsgi. It is possible to limit the number of concurrent connections using this option.
keep_alive = True	boolean value	This option allows using the same TCP connection to send and receive multiple HTTP requests/responses, as opposed to opening a new one for every single request/response pair. HTTP keep-alive indicates HTTP connection reuse. Possible values: • True : reuse HTTP connection. • False : closes the client socket connection explicitly. Related options: • tcp_keepidle
max_header_line = 16384	integer value	This option specifies the 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). Since TCP is a stream based protocol, in order to reuse a connection, the HTTP has to have a way to indicate the end of the previous response and beginning of the next. Hence, in a keep_alive case, all messages must have a self-defined message length.

Configuration option = Default value	Type	Description
secure_proxy_ssl_header = None	string value	This option specifies the HTTP header used to determine the protocol scheme for the original request, even if it was removed by a SSL terminating proxy. Possible values: • None (default) - the request scheme is not influenced by any HTTP headers • Valid HTTP header, like HTTP_X_FORWARDED_PROTO  WARNING Do not set this unless you know what you are doing. Make sure ALL of the following are true before setting this (assuming the values from the example above): • Your API is behind a proxy. • Your proxy strips the X-Forwarded-Proto header from all incoming requests. In other words, if end users include that header in their requests, the proxy will discard it. • Your proxy sets the X-Forwarded-Proto header and sends it to API, but only for requests that originally come in via HTTPS. If any of those are not true, you should keep this setting set to None.

Configuration option = Default value	Type	Description
ssl_ca_file = None	string value	This option allows setting path to the CA certificate file that should be used to verify connecting clients. Possible values: ● String representing path to the CA certificate file. Related options: ● enabled_ssl_apis
ssl_cert_file = None	string value	This option allows setting path to the SSL certificate of API server. Possible values: ● String representing path to the SSL certificate. Related options: ● enabled_ssl_apis
ssl_key_file = None	string value	This option specifies the path to the file where SSL private key of API server is stored when SSL is in effect. Possible values: ● String representing path to the SSL private key. Related options: ● enabled_ssl_apis
tcp_keepidle = 600	integer value	This option sets the value of TCP_KEEPIDLE in seconds for each server socket. It specifies the duration of time to keep connection active. TCP generates a KEEPALIVE transmission for an application that requests to keep connection active. Not supported on OS X. Related options: ● keep_alive

Configuration option = Default value	Type	Description
wsgi_log_format = % (client_ip)s "% (request_line)s" status: % (status_code)s len: % (body_length)s time: % (wall_seconds).7f	string value	It represents 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. This option is used for building custom request loglines when running nova-api under eventlet. If used under uwsgi or apache, this option has no effect. Possible values: • <code>%(client_ip)s "%(request_line)s" status: %(status_code)s 'len: %(body_length)s time: %(wall_seconds).7f</code> (default) • Any formatted string formed by specific values. Deprecated since: 16.0.0 Reason: This option only works when running nova-api under eventlet, and encodes very eventlet specific pieces of information. Starting in Pike the preferred model for running nova-api is under uwsgi or apache mod_wsgi.

9.1.57. zvm

The following table outlines the options available under the **[zvm]** group in the `/etc/nova/nova.conf` file.

Table 9.56. zvm

Configuration option = Default value	Type	Description
ca_file = None	string value	CA certificate file to be verified in httpd server with TLS enabled A string, it must be a path to a CA bundle to use.
cloud_connector_url = None	uri value	URL to be used to communicate with z/VM Cloud Connector.

Configuration option = Default value	Type	Description
image_tmp_path = \$state_path/images	string value	The path at which images will be stored (snapshot, deploy, etc). Images used for deploy and images captured via snapshot need to be stored on the local disk of the compute host. This configuration identifies the directory location. Possible values: A file system path on the host running the compute service.
reachable_timeout = 300	integer value	Timeout (seconds) to wait for an instance to start. The z/VM driver relies on communication between the instance and cloud connector. After an instance is created, it must have enough time to wait for all the network info to be written into the user directory. The driver will keep rechecking network status to the instance with the timeout value. If setting network failed, it will notify the user that starting the instance failed and put the instance in ERROR state. The underlying z/VM guest will then be deleted. Possible Values: Any positive integer. Recommended to be at least 300 seconds (5 minutes), but it will vary depending on instance and system load. A value of 0 is used for debug. In this case the underlying z/VM guest will not be deleted when the instance is marked in ERROR state.