Red Hat Software Certification 2024

Red Hat OpenStack Certification Policy Guide

For Use with Red Hat OpenStack 17

Last Updated: 2024-05-30

Red Hat Software Certification 2024 Red Hat OpenStack Certification Policy Guide

For Use with Red Hat OpenStack 17

Legal Notice

Abstract

The Red Hat OpenStack Certification Policy Guide describes the procedural, technical and policy requirements for Partners who offer their own applications or infrastructure software (plug-in or driver) for use with Red Hat OpenStack Platform in a supported customer environment. Version 9.0 and 8.80 updated May 28, 2024.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code and documentation. We are beginning with these four terms: master, slave, blacklist, and whitelist. Due to the enormity of this endeavor, these changes will be gradually implemented over upcoming releases. For more details on making our language more inclusive, see our CTO Chris Wright's message .

CHAPTER 1. INTRODUCTION TO RED HAT OPENSTACK PLATFORM CERTIFICATION POLICY GUIDE

The Red Hat OpenStack certification policy guide describes the policy overview to certify third-party vendor solutions with Red Hat OpenStack Platform. Red Hat encourages Partners to test their plugins with pre-releases of both Red Hat builds and pre-releases of their own solutions.

1.1. AUDIENCE

This guide describes the technical certification requirements as implemented for software certification Partners who want to offer their own applications, management applications or plugin/driver software for use with Red Hat OpenStack Platform (RHOSP) in a jointly supported customer environment.

1.2. CREATING VALUE FOR CUSTOMERS

The certification process includes a series of tests that provides Red Hat Customers with the assurance that a certified solution meets all the requirements of an enterprise cloud. The certification process is jointly supported by Red Hat and Partner’s organization.

The Red Hat OpenStack Certification Workflow Guide includes multiple tests, each with a series of subtests and checks, which are explained in this guide. All tests are not required for each certification.

Logs from a single run with all of the mandatory tests and the test suite self-check test (rhcert/selfcheck) must be submitted to Red Hat for new certifications and for recertifications. The certification tooling and workflow documented in the article will be supported for a period of 90 days to support certifications which are underway.

You must complete all new certifications using the latest certification tooling and workflow document in the Red Hat OpenStack Certification Policy and Workflow guides.

Red Hat encourages you to install and use the latest version of the certification tooling and workflow for the certification process. A 90 day grace period is provided for the previous version of the tooling and workflow upon a new release of the certification tooling. This allows already in progress certifications to be completed without disruption. At the end of the grace period, test results generated using earlier versions of tooling will no longer be accepted.

The latest version of the certification tooling and workflow is available via Red Hat Subscription Management and documented in the Red Hat OpenStack Certification WorkFlow Guide.

NOTE

Most of the certification subtests provide an immediate return status (Pass/Fail), however, some subtests may require detailed review by Red Hat to confirm success. Such tests are marked with **Review** status in the Red Hat Certification application.

Some tests may also identify a potential issue and return a **Warn** status. This status indicates that best practices have not been followed. Tests marked with the **Warn** status warrant attention or actions but do not prevent a certification from succeeding. You are recommended to review the output of such tests and perform appropriate actions based on the information contained within the warnings.

Additional resources

- For more information on running the tests, see Red Hat OpenStack Certification Workflow Guide.

1.3. RED HAT OPENSTACK PLATFORM CERTIFICATION PREREQUISITES

You must meet the following requirements before applying for an OpenStack Certification.

- Companies must be Partners in Red Hat Connect for Technology Partners . This program enables an ecosystem for commercial OpenStack deployments and includes numerous technology companies.
- You must have a support relationship with Red Hat. This can be fulfilled by one of the following ways:
  - custom support agreement
  - TSANet
- You must have a good working knowledge of RHOSP including installation and configuration of the product
- You must provide a link to an installation guide for the OpenStack plugin being certified. This installation guide must indicate the usage of Red Hat Director for the OpenStack deployment.

Additional Resources

- For more information about the product, see detailed product documentation on Red Hat Customer Portal
- Undertake the product training or certification on Red Hat Training Page .
- For more information about TSANet, see TSANet web page.

1.4. RED HAT OPENSTACK PLATFORM COMPONENT DISTRIBUTION

As part of Red Hat OpenStack Platform (RHOSP), Red Hat distributes components that are committed in a release of the upstream OpenStack project such as Kilo, Liberty, and so on. These components are called **In tree** components.You are still responsible for certification and for distribution of all their dependencies that are not part of the upstream OpenStack project.

Distribution of products or components that are not committed in the upstream OpenStack project is the responsibility of the Partner. These components are also referred to as **Out of tree** components.

1.5. SUPPORTED RHEL VERSION AND ARCHITECTURE

The RHOSP certifications are supported on the following RHEL version and architecture.

RHOSP version	RHEL version	Architecture

RHOSP version	RHEL version	Architecture
16.0	RHEL 8.1	x86_64 ppc64le
16.1	RHEL 8.2	x86_64 ppc64le
16.2	RHEL 8.4	x86_64 ppc64le
17.0	RHEL 9.0	x86_64

CHAPTER 2. RED HAT OPENSTACK PLATFORM CERTIFICATION TARGETS

You are expected to implement the following targets achieve a certification:

2.1. PRODUCTS IMPLEMENTING OPENSTACK APIS

This category includes products that deliver an OpenStack service/functionality such as launching server instances, adding new routers, creating images, creating storage containers and objects, creating user profiles, etc. by implementing an API and/or an API Extension for Networking, Block Storage, or File Share services.

For products implementing OpenStack APIs, you need to successfully complete the relevant certification tests for the API group in addition to the OpenStack Director test (**openstack/director**) and OpenStack Supportability tests (**openstack/supportable**).

To ensure that the underlying platform is supported by Red Hat, run the OpenStack Director, Supportability, and sosreport tests on multiple overcloud nodes. The test results should be from a controller, and a compute or storage nodes implementing/consuming Openstack APIs that the vendor plugin controls.

IMPORTANT

It is your responsibility to ensure these tests are run on the correct nodes. Additional runs may be requested to fulfill these requirements.

In this case, the certification process verifies that the service delivers the API according to the platform specification and that the underlying OpenStack environment is configured correctly.

2.2. PRODUCTS CONSUMING OPENSTACK APIS

This category covers all products/applications that rely on OpenStack services (like Networking, Block Storage, File Share etc.) to operate.

Such products generally facilitate an OpenStack deployment or complement the Cloud Infrastructure with additional functionalities such as configuration, scaling, and management.

Examples

- OpenStack management and orchestration applications like Network Functions Virtualization Management and Orchestration (NFV MANO)
- OpenStack monitoring applications
- Other OpenStack-enabled applications such as virtual network functions (VNFs)

For such applications, you need to successfully complete the OpenStack Director test (**openstack/director**) and OpenStack Supportability tests (**openstack/supportable**).

2.3. SUPPORT FOR CERTIFIED OPENSTACK COMPONENT

Support for certified Openstack components such as plugins/drivers and customer assistance is derived from the vendor that is shipping the component.

- If Red Hat certifies and ships a third-party component as part of RHOSP and there is a question or issue with that component, the customers will contact Red Hat for assistance.

- If a third-party ships a RHOSP—certified component and there is a question or issue with that component, the third party will be fully responsible for assisting the customer and providing support for that component.

However, Red Hat certified Partners and Red Hat, maintain active engineering relationships that either party can leverage to ensure quick progress is made on customer issues.

Additional Resources

- For more information on supportability of RHOSP certified components, see article.

CHAPTER 3. CERTIFICATION LIFECYCLE

3.1. PRODUCT CERTIFICATION LIFECYCLE

Starting with Red Hat OpenStack Platform (RHOSP) 16, certification is granted on a specific major and minor release of RHOSP. For example, RHOSP 16.0 where 16 is the major release and 0 is the minor release.

While the certification remains valid for the life of the major release, in our example RHOSP 16, there are instances where recertification will be required. Those instances are described in Recertification.

3.2. CONTINUAL TESTING

You are responsible for your own internal continual testing over the lifespan of their product and the Red Hat OpenStack Platform major version they are certified on. You are encouraged to utilize a CI system such as DCI, that includes testing with the certification tests. Evaluating certification testing results from a CI system are not required to be submitted to Red Hat but should be monitored by the Partner for regressions and unexpected behaviors and to indicate when a recertification may be required.

You may have access to pre-released software builds of Red Hat OpenStack Platform and are encouraged to begin their initial and CI testing and engagement with the Red Hat Certification team prior to the Red Hat OpenStack Platform version being made generally available to customers. Final testing and container builds must be conducted on the generally available (GA) released containers for that major release.

3.3. RECERTIFICATION

Red Hat will notify you of, and you are requested to recertify your product in the following cases:

- A new major release of the Red Hat OpenStack Platform.
- A new minor release of the Red Hat OpenStack Platform that adds additional features or functionality not previously covered in an earlier certification that the partner desires to add to their certification.
- A new minor release of the Red Hat OpenStack Platform that updates the kernel and the partner product relies on kernel modules.

You will notify Red Hat of, and you are required to recertify their product in the following cases:

- A new major update of the partner’s product that invalidates the original testing conducted in the original certification.
- A new minor update of the partner’s product that would alter the original test plan of the certification.

A new certification should be submitted for each of these cases. Where possible, in minor release updates of Red Hat and Partner products, the certification efforts and test plan will focus on the new features and functionality not already tested in prior certification(s) as the established feature functionality is expected to be maintained through the required continual testing.

When a customized container image is provided as part of your OpenStack certification, it is important to rebuild this customized container image every time a Red Hat OpenStack Platform z-stream is released for a specific Major-Minor release. This will ensure that your image is taking advantage of the latest bug fixes and CVEs.

NOTE

For RHOSP container recertification, it is not required to revalidate the functionality of your product if it has not undergone any modification.

CHAPTER 4. SYSTEM REPORT TEST

The Red Hat system report test, is also known as **openstack/sosreport**, and captures the basic system report.

The system report test ensures that the SOS tool captures a basic report and performing operations as expected on the image or system.

The sosreport command is a tool that collects configuration details, system and diagnostic information from a RHEL system and assists you in troubleshooting their systems.

Success criteria

- A basic sosreport can be captured from the system under test.
- The test status will be PASS, if a valid rpm version captures and collects the openstack data and the openstack plugins (manila, cinder, neutron) are enabled.

Additional resources

- For more information about sosreport, see What is an sosreport and how to create one in Red Hat Enterprise Linux?.

CHAPTER 5. DESIGNATE TEST

The **designate** test uses the Tempest framework to verify whether the Designate DNS service works appropriately with the Neutron plugin that you are certifying.

The test performs the following actions:

- Ensures that CRUD operations are successful for the following objects :
    - Records
    - Domains
    - Servers
    - Blacklists
    - Pools
    - Quotas
    - Record sets
    - Top-level domains
    - Transaction signatures
- Ensures that only administrative users can perform operations on blacklists, quotas, pools, and transaction signatures.
- Checks that record sets not in compliance with the DNS RFC specification are rejected. For example, record sets containing trailing spaces or invalid characters are rejected.
- Checks that record sets that contain wildcard characters (*) match requests for non-existent domains.
- Ensures that zone transfers across DNS servers are possible.
- Ensures that zone creation and zone deletion propagates across DNS servers.

Success criteria

All operations and verifications are successful.

CHAPTER 6. TEST ENVIRONMENT SUPPORTABILITY TEST

The Supportability tests, also known as openstack/supportable, ensure that the test environment is compliant with Red Hat’s support policy. This test is required for all OpenStack software certifications. The test confirms that the test node (an OpenStack deployment-under-test) consists only of components supported by Red Hat (Red Hat OpenStack Platform, Red Hat Enterprise Linux) or supported by the Partner. An OpenStack deployment-under-test refers to the node where the plugin/application-under-test is installed and also the Undercloud Director node.

The openstack/supportable tests include the following subtests.

6.1. KERNEL SUBTEST

The **kernel** subtest checks the kernel module running on the test environment. The version of the kernel can be either the original General Availability (GA) version or any subsequent kernel update released for the RHEL major and minor releases.

The kernel subtest also ensures that the kernel is not tainted when running in the environment.

Success criteria

- The running kernel is a Red Hat kernel.
- The running kernel is released by Red Hat for use with the RHEL version.
- The running kernel is not tainted.
- The running kernel has not been modified.

Additional resources

- Red Hat Enterprise Linux Life Cycle
- Red Hat Enterprise Linux Release Dates
- Why is the kernel "tainted" and how are the taint values deciphered?

6.2. KERNEL MODULES SUBTEST

The **kernel modules** subtest verifies that loaded kernel modules are released by Red Hat, either as part of the kernel’s package or added through a Red Hat Driver Update. The kernel module subtest also ensures that kernel modules do not identify as Technology Preview.

Success criteria

- The kernel modules are released by Red Hat and supported.

Additional resources

- What does a "Technology Preview" feature mean?

6.3. HARDWARE HEALTH SUBTEST

The Hardware Health subtest checks the system's health by testing if the hardware is supported, meets the requirements, and has any known hardware vulnerabilities. The subtest does the following:

- Checks that the Red Hat Enterprise Linux (RHEL) kernel does not identify hardware as unsupported. When the kernel identifies unsupported hardware, it will display an unsupported hardware message in the system logs and/or trigger an unsupported kernel taint. This subtest prevents customers from possible production risks which may arise from running Red Hat products on unsupported configurations and environments.
  In hypervisor, partitioning, cloud instances, and other virtual machine situations, the kernel may trigger an unsupported hardware message or taint based on the hardware data presented to RHEL by the virtual machine (VM).
- Checks that the Host Under Test (HUT) meets the minimum hardware requirements.
  - RHEL 8 and 9: Minimum system RAM should be 1.5GB, per CPU logical core count.
  - RHEL 7: Minimum system RAM should be 1GB, per CPU logical core count.
- Checks if the kernel has reported any known hardware vulnerabilities, if those vulnerabilities have mitigations and if those mitigations have resolved the vulnerability. Many mitigations are automatic to ensure that customers do not need to take active steps to resolve vulnerabilities. In some cases this is not possible; where most of these remaining cases require changes to the configuration of the system BIOS/firmware which may not be modifiable by customers in all situations.
- Confirms the system does not have any offline CPUs.
- Confirms if Simultaneous Multithreading (SMT) is available, enabled, and active in the system.

Failing any of these tests will result in a WARN from the test suite and should be verified by the partner to have correct and intended behavior.

Success criteria

- The kernel does not have the UNSUPPORTEDHARDWARE taint bit set.
- The kernel does not report an unsupported hardware system message.
- The kernel should not report any vulnerabilities with mitigations as vulnerable.
- The kernel does not report the logic core to installed memory ratio as out of range.
- The kernel does not report CPUs in an offline state.

Additional resources

- Minimum required memory
- Hardware support available in RHEL 8 but removed from RHEL 9 .
- Hardware support available in RHEL 7 but removed from RHEL 8 .
- Hardware support available in RHEL 6 but removed from RHEL 7 .

6.4. INSTALLED RPMS SUBTEST

The **installed RPMs** subtest verifies that RPM packages installed on the system are released by Red Hat and not modified. Modified packages may introduce risks and impact the supportability of the customer's environment. You might install non-Red Hat packages if necessary, but you must add them to your product's documentation, and they must not modify or conflict with any Red Hat packages.

Red Hat will review the output of this test if you install non-Red Hat packages.

Success criteria

- The installed Red Hat RPMs are not modified.
- The installed non-Red Hat RPMs are necessary and documented.
- The installed non-Red Hat RPMs do not conflict with Red Hat RPMs or software.

Additional resources

- Production Support Scope of Coverage

6.5. SELINUX SUBTEST

This subtest confirms that Security-Enhanced Linux (SELinux) is running in enforcing mode on the OpenStack deployment-under test.

SELinux adds Mandatory Access Control (MAC) to the Linux kernel, and is enabled by default in Red Hat Enterprise Linux.

SELinux policy is administratively-defined, enforced system-wide, and is not set at user discretion reducing vulnerability to privilege escalation attacks helping limit the damage made by configuration mistakes. If a process becomes compromised, the attacker only has access to the normal functions of that process and to files the process has been configured to..

Success criteria

SELinux is configured and running in enforcing mode on the OpenStack deployment-under-test.

Additional Resources

For more information on SELinux in RHEL, see SELinux Users and Administrators Guide .

CHAPTER 7. DIRECTOR TEST

This test, also known as openstack/director, ensures that the deployment-under-test is originally installed using RHOSP Director. This test is required for all OpenStack software certifications.

Red Hat OpenStack Platform (RHOSP) Director is the supported toolset for installing and managing a RHOSP environment in production. It helps in easy installation of a lean and robust OpenStack cloud and is targeted specifically for enterprise cloud environments where updates, upgrades and infrastructure control are critical for underlying OpenStack operations.

Success criteria

The deployment under test is originally installed using Red Hat OpenStack Platform Director.

Additional resources

- For more information about installing RHOSP Director, see Director Installation and Usage Guide.

CHAPTER 8. CINDER TESTS

The **cinder test** applies to OpenStack products or components that implement features for the OpenStack Block Storage service only. The test uses Tempest Framework integrated with the Red Hat OpenStack Platform (RHOSP) to test both operational and functional features.

The cinder test verifies the functionality of the corresponding cinder driver that you are certifying by running the selected feature tests. The following features are currently supported:

- cinder_volumes
- cinder_consistency_groups
- cinder_backups
- cinder_multi-attach_volume

Prerequisites

1. When deploying the overcloud, ensure that:
   a. You have provided two instances of the backend you intend to certify.

      NOTE

      Having more than one hardware storage is not required; configuring both cinder backends to use the same hardware array is supported.

   b. You have enabled the *cinder-backup* service. To do this, add the **cinder-backup** environment file in the overcloud deployment command.

      ```
      -e /usr/share/openstack-tripleo-heat-templates/environments/cinder-backup.yaml
      ```

      For more information, see Configuring your overcloud by using environment files .

   c. You have disabled cinder's LVM/iSCSI backend, and configured Glance to use Cinder as its backend. To do this, create a new custom environment file, for example, *rhcert-overrides.yaml* and add the following lines:

      ```
      parameter_defaults:
        CinderEnableIscsiBackend: false
        GlanceBackend: cinder
      ```

      Then, add the *rhcert-overrides.yaml* file to the overcloud deployment command.

      ```
      -e /home/stack/rhcert-overrides.yaml
      ```

2. Before executing the **tempest_config** test, ensure that:
   a. You have enabled the following flags under their respective section header in the *tempest.conf* file.

NOTE

If your plugin supports the *consistency groups* and *multi-attach volumes* features, then ensure to enable their corresponding flags in the *tempest.conf* file.

For example, *consistency_group* and *volume_multiattach* flags in the *tempest.conf* file are enabled.

```
[volume-feature-enabled]
consistency_group = True
extend_attached_encrypted_volume = True
extend_attached_volume = True
manage_snapshot = True
manage_volume = True
volume_revert = True

[image-feature-enabled]
import_image = True

[compute-feature-enabled]
volume_multiattach = True
```

b. You have set the *tempest_roles* as follows in the *tempest.conf* file:

```
[auth]
tempest_roles = member,swiftoperator
```

3. If your driver supports multi-attach volume feature, perform the following steps before executing cinder's multi-attach volume tests:

   a. Create a multi-attach volume type by following the procedure explained in Creating a multi-attach volume type. Do not configure cinder's default volume type to be the multi-attach volume type.

   b. Add a reference to the multi-attach volume type in the *tempest.conf* file, as shown below:

```
[volume]
volume_type_multiattach = <multiattach volume type>
```

Additional resources

For more information, see Running tempest_config test.

8.1. CINDER_VOLUMES

The **cinder_volume** test checks if the driver functionalities and base functionalities of cinder, such as volume actions, snapshots, boot, volume migrate, encryption, and clone are working. This test is mandatory.

8.2. CINDER_CONSISTENCY_GROUPS

The **cinder_consistency_groups** test checks for disaster recovery and the following actions by taking multiple volume snapshots of the consistency group at the same time:

- Creating and deleting consistency groups
- Creating and deleting consistency group snapshots
- Creating a new consistency group from an existing consistency group snapshot

This test is mandatory if your driver implements the consistency groups feature.

8.3. CINDER_BACKUPS

The **cinder_backups** test verifies the driver's backup/restore functionality by testing the following:

- Creating and restoring a backup from existing volume
- Testing an incremental backup
- Taking a backup of a volume snapshot

8.4. CINDER_MULTI-ATTACH_VOLUME

The **cinder_multi-attach_volume** test checks whether it can attach and access a single block storage volume from multiple hosts or servers by running the following tests:

- Boot a VM from a multi-attach volume
- Resize a server with multi-attach volume
- List volume attachments for a multi-attach volume
- Snapshot from backed multi-attach volume
- Attach and detach a multi-attach volume from a shelved or offload server
- Delete an attached multi-attach volume
- Attach multi-attach volume to same or different server.

This test is mandatory if your driver implements the multi-attach feature.

Additional resources

- For more information about cinder tests, see Products implementing OpenStack APIs.

CHAPTER 9. MANILA TEST

Based on the solutions provided by Partners, Red Hat will define a test plan in RH-cert web UI along with the test(s) that Partners needs to perform. The manila test executes the selected file share-component feature test(s) and checks the plugin/driver functionality as chosen by the user during test run time. Manila testing must include the tests defined in the test plan which will include the mandatory base tests and any implemented additional features, one test run per supported filesystem backend and DHSS true or false mode supported as listed below:

9.1. MANILA_SHARES (BASE)

The manila_shares test will check the base file operations with either the NFS or CIFS protocols. This test also covers the base operations with and without the “driver handles share servers” (DHSS) feature enabled. It may be necessary to repeat this test in multiple subsequent runs in cases where the plug-in supports multiple protocols and DHSS being enabled and disabled.

In **manila.conf** file if **DHSS=true**, the networking plugin should either be **NeutronNetworkPlugin** or **NeutronBindNetworkPlugin**.

Success criteria

- If Manila is using NeutronNetworkPlugin and the tempest has multitenancy enabled the dhss test status will be PASS
- If Manila uses a standalone network dhss test status will be FAIL

Manila_shares has features like availability zones, consistency groups, extensions, limits, metadata, micro, versions, quotas, rules, security services, share networks, share actions, and share instances.

The plugin/driver functionalities that are tested as part of Manila_shares test are:

- create
- delete
- list
- snapshot
- modify
  If the vendor plugin implements manila_shares along with its feature they are also expected to perform the following subtest for manila_shares:

9.2. MANILA_SHARE_MANAGED

This test checks the driver ability to keep a share in managed/unmanaged state.

9.3. MANILA_SHARE_SHRINK

This test checks the drivers’ capability to shrink the manila shares.

9.4. MANILA_SHARE_EXTEND

This test checks the drivers’ capability to extend the manila shares.

9.5. MANILA_SNAPSHOT

A snapshot allows Customers to restore the data from a specific time they want to. A new share can be created only for the data that has its snapshot. The plugin/driver functionalities that are tested as part of manila_snapshot test are:

- reset snapshots
- force delete snapshot
- share snapshot instance
- deleting shares with existing snapshot
- create share with smaller size snapshot
- create share from snapshot with different share network
- delete snapshot with wrong id
- create snapshot with wrong id
- create access rule to snapshot
- list shares by snapshot id
- listing and renaming snapshots
- share snapshot instances
- snapshot rules

9.6. MANILA_SNAPSHOT_MANAGED

This test checks drivers’ capability to keep a snapshot and replicate share snapshot in managed or unmanaged state.

9.7. MANILA_SNAPSHOT_SHARE_FROM_SNAPSHOT

This test creates share snapshot from snapshot when the share network is not provided.

9.8. MANILA_SNAPSHOT_REVERT_TO_SNAPSHOT

This test checks the drivers’ capability to revert the share to snapshot.

9.9. MANILA_SNAPSHOT_MOUNTABLE

This test checks the drivers’ capability to create mountable snapshots rather than creating a whole share from the snapshot and then deleting the share.

Success criteria

Following are the individual Success Criteria for the Manila test and subtests:

- Manila test must be using NeutronNetworkPlugin and tempest must have multitenancy enabled

- manila_share_managed driver is available to manage manila share state
- manila_share_shrink driver carry out shrink operation of manila shares
- manila_share_extend is functional
- manila_snapshot is working with all its features
- All manila_snapshot subfeature tests are performed successfully

Additional resources

- For more information about manila tests, see Products implementing OpenStack APIs.

CHAPTER 10. NEUTRON TEST

The openstack/neutron test is only applicable to OpenStack products/components that implement OpenStack features for the OpenStack networking Service. These tests cover OpenStack networking-component feature testing, which includes basic and operational functional testing using the Tempest Framework that is integrated in the RHOSP. Neutron includes networking, IP address management (IPAM), and router support to enable routing between internal and external network.

Based on the solutions provided by you, Red Hat will define a test plan in RH-cert web UI along with the test(s) that You needs to perform. The neutron test executes the selected networking-component feature test(s) and, checks the plugin/driver functionality as chosen by the user during test run time. Neutroning must include the tests as defined in the test plan, which will include the mandatory base test(s) and any implemented additional features, one test run per support base protocol as listed below:

10.1. NEUTRON_IPV4 (BASE)

This test checks all the plugin/driver for neutron based capabilities like network, ports, routers, quotas, subnet pools, allowed_address_pair, external_networks and address_scope with respect to ipv4 address scheme.

Success criteria

Performs all the neutron based plugin/driver ipv4 functionality successfully.

10.2. NEUTRON_IPV6 (BASE)

This test checks all the plugin/driver for neutron based capabilities like network, ports, routers, quotas, subnet pools, allowed_address_pair, external_networks and address_scope with respect to ipv6 address scheme.

Success criteria

Performs all the neutron based plugin/driver ipv6 functionality successfully.

10.3. NEUTRON_ADDRESS_SCOPE

This test checks if all the operations available for address scope can be performed with the help of a vendor driver. Operations include:

- creation
- deletion
- updation
- how address scopes

Success criteria

All the address_scope operations are operational.

10.4. NEUTRON_AGENTS

This test checks if the DHCP and L3 agent operations can be performed successfully.

Success criteria

DHCP and L3 agent are operational.

10.5. NEUTRON_ATTRIBUTE_EXTENSIONS

This test checks if a timestamp can be associated to standard api extensions.

Success criteria

Neutron_attribute_extensions test status is Pass, and timestamp can successfully be associated.

10.6. NEUTRON_AVAILABILITY_ZONES

This test checks all the standard API operations that can be applied to availability zones.

Success criteria

Neutron_availability_zones is able to apply API operations to availability zones.

10.7. NEUTRON_DHCP_EXTRA

The DHCP options extension allows adding DHCP options that are associated to a Neutron port. You can specify DHCP options when defining or updating a port by specifying the extra_dhcp_opts tag and providing its options as name value pairs. All the port related operations with extra_dhcp_opts are tested to check if the new options can be applied.

Success criteria

New dhcp options can be successfully applied.

10.8. NEUTRON_FLAVOR

The purpose of a Flavor Framework is to provide an API that allows the user to choose the type of service by a set of advertised service capabilities rather than by a provider type or named vendor. This test checks if all the standard flavor operations can be done with the help of a third-party plugin/driver.

Success criteria

All the standard operations can be performed with the help of a third-party plugin/driver successfully.

10.9. NEUTRON_GATEWAY_EXTRA

This test checks if extra options related to gateways can be applied using the plugin/driver in use.

Success criteria

Extra gateway options can be successfully applied.

10.10. NEUTRON_GMAN

In some cloud deployments using Neutron, there is a need for each tenant to configure resources, like a network, subnet and router, before they are able to boot a VM. This test checks if as a tenant driver you can delete or get the allocated topologies.

Success criteria

Tenant driver delete and allocate topologies successfully.

10.11. NEUTRON_IP_AVAILABILITY

It allows a user or process to determine the number of IP addresses that are consumed across networks and the allocation pools of their subnets. The test checks the availability of network admin and network ip after performing operations on related resources like subnet and port addition and deletion.

Success criteria

Network IP and network admin are available.

10.12. NEUTRON_IPV4

This test checks all the plugin/driver for neutron based capabilities like network, ports, routers, quotas, subnet pools, allowed_address_pair, external_networks and address_scope with respect to ipv4 address scheme.

Success criteria

Performs all the neutron based plugin/driver ipv4 functionality successfully.

10.13. NEUTRON_IPV6

This test checks all the plugin/driver for neutron based capabilities like network, ports, routers, quotas, subnet pools, allowed_address_pair, external_networks and address_scope with respect to ipv6 address scheme.

Success criteria

Performs all the neutron based plugin/driver ipv6 functionality successfully.

10.14. NEUTRON_L2_MULTI_PROVIDER

The ml2 plugins database schema and driver APIs, support virtual L2 networks made up of multiple segments. Test differentiates the supported operations.

Success criteria

Supported operations are performing successfully.

10.15. NEUTRON_L3_EXTRA_ROUTE

This test checks the operations like updation and deletion of extra routes and if the plugin provide l3 functionality.

Success criteria

Is able to perform updation and deletion operations successfully.

10.16. NEUTRON_L3_FLAVORS

Flavors allows the running of multiple L3 drivers in the same deployment. This test checks the creation and deletion of routers with flavors.

Success criteria

Is able to perform creation and deletion operations successfully.

10.17. NEUTRON_L3_HA

High availability features are implemented as extensions and drivers. This test checks if high availability can be applied to routers.

Success criteria

High availability can be applied to routers successfully.

10.18. OCTAVIA_LOAD_BALANCER

LBaaS v2 supports Octavia plugins. If a Partner driver or a plugin supports this feature, the certification test runs will include results of octavia_load_balancer test. This test is implemented on Red Hat OpenStack Director based installation.

Octavia test checks the Load balancer creation flow with its following features:

- Health Manager
- Housekeeping Manager
- Loadbalancer
- Amphora
- Listener
- Pool
- Member

Success criteria

The test performs create, read, update, and delete operations on the Octavia load balancer features. Successful PASS operations signifies that all the Octavia related features are working for a Partner plugin.

10.19. NEUTRON_MTU

This test checks if the change in MTU size is reflected in the api.

Success criteria

MTU size is reflected.

10.20. NEUTRON_QOS

QoS is defined as the ability to guarantee certain network requirements like bandwidth, latency, jitter, and reliability in order to satisfy a Service Level Agreement (SLA) between an application provider and end users.This test checks if all the rules and policies related to QoS can be applied properly to neutron resources.

Success criteria

Rules and policies related to QoS applied to neutron resources successfully.

10.21. NEUTRON_RBAC

This test checks if all RBAC operations can be done on different neutron resources.

Success criteria

RBAC operation can be done on different neutron resources successfully.

10.22. NEUTRON_SECURITY_GROUPS

Security groups and security group rules allows administrators and tenants the ability to specify the type of traffic and direction (ingress/egress) that is allowed to pass through a port. A security group is a container for security group rules.This test checks all the security groups related operations that can be done if the driver/plugin implements the functionality.

Success criteria

Security group related operations performing successfully.

10.23. NEUTRON_SERVICE_TYPES

Using this feature, you can ensure that ports always use different subnets, for example instances and router interfaces,. This test checks if all the basic operations of subnet service types can be done properly.

Success criteria

Subnet service related operations performing successfully.

10.24. NEUTRON_SUBNET_ALLOCATION

It involves automatically allocating addresses for subnets instead of requesting subnet details at the time of creation. This test checks the testing of the subnet pool feature of neutron.

Success criteria

Subnet pool operation of neutron performing successfully.

10.25. NEUTRON_SUBNET_DEFAULT_POOL

This test checks the operations of default subnet pools.

Success criteria

Default subnet pool operations performing successfully

10.26. NEUTRON_TAGS

Various virtual networking resources support tags for use by external systems or any other clients of the Networking service API. This test checks if all the tag related operations can be performed.

Success criteria

Tag related operations performing successfully.

10.27. NEUTRON_TRUNK

The network trunk service allows multiple networks to be connected to an instance using a single virtual NIC (vNIC). Multiple networks can be presented to an instance by connecting it to a single port. This test checks if all the trunk related operations can be done.

Success criteria

Trunk related operations performing successfully.

10.28. NEUTRON_BORDER_GATEWAY_PROTOCOL_VPN

This is a new test introduced in RHOSP16 and corresponds to the new feature Border Gateway Protocol Virtual Private Network(BGP VPN).

BGP VPN supports inter-connection between L3VPNs and Neutron resources, such as Networks, Routers and Ports. To deliver an isolated connectivity between multiple sites, BGP-based VPNs allow a network operator to provide a VPN service to its customers.

BGP VPN allows your instances to connect to your existing layer 3 VPN services. Once a BGP VPN network is created, you can associate it with a project, allowing the project’s users to connect to the BGP VPN network.

The neutron_border_gateaway_protocol_vpn test certifies following tempest test operations:

- create
- delete
- list
- show
- update

Success criteria

All the BGP VPN related operations are performed successfully.

Additional resources

For more information about neutron test, see Products implementing OpenStack APIs.

CHAPTER 11. OPENSTACK CONFIGURATION TEST

Starting with Red Hat OpenStack Platform (RHOSP) 15, OpenStack Configuration test will be the new test available for Partners. This test is for the Manila and Neutron plugin certification test plan, and is planned only for controller node. The OpenStack Configuration test includes and supports Open Virtual Network subtest.

- **Open Virtual Network subtest**

Open Virtual Network (OVN) is an Open vSwitch-based software-defined networking (SDN) solution that supplies network services to instances. OVN is expected to be configured and operational. This subtest validates the status of OVN.

Success criteria

The status of the test depends on the OVN as follows:

- If OVN is ON, the test status is PASS.
- If OVN is OFF, the test status is FAIL.
- If OVN is unknown, the test status is REVIEW.

11.1. ADDITIONAL RESOURCES

For more information about OVN, see OVN Architecture.

CHAPTER 12. TRUSTED CONTAINER TEST

The Trusted Container test checks if Red Hat recognizes the Red Hat OpenStack Platform (RHOSP) plugin/driver container. The test also verifies whether the container is provided by Red Hat or you. The certified container image reduces the number of sources a customer must utilize for deployment, and it also ensures all the component included in solution stack are from a trusted source.

Working of RHOSP certification testing

During RHOSP certification testing, the Trusted Container Test captures information about the installed and running containers. After the information is captured, the test queries the Red Hat certification services to determine if the containers are recognized and certified.

Requirements for Partners

If your driver is shipped as part of RHOSP (In Tree) then they are expected to only run the Trusted Container Test because container image is already certified. However, if you ship their own container image (Out of Tree), as a prerequisite, the Partner must certify the container images with Red Hat Connect. For more information on container image certification, see Partner Integration guide.

In Red Hat connect when a Partner creates a new product request for RHOSP 13, they can only select the Release Category as Tech-Preview. You can execute the Trusted Container Test after the container image is certified on Red Hat Connect. After the Trusted Container Test is completed successfully, the Partner can choose the General Availability(GA) option.

Success criteria

You have received a container report that shows running and non-running containers on the overcloud controller node. The report shows that the RHOSP services like cinder, manila, and neutron are installed and running. Based on an RHOSP certification testing, the running container can either be an RHOSP certified or Red Hat Certified container.

CHAPTER 13. IN-PLACE UPGRADES

Starting with Red Hat OpenStack Platform (RHOSP) 16.1.1, the Framework for Upgrades (FFU) is utilized in a new **In-place Upgrades** test. This test is available during RHOSP plugin certification. It verifies the functionality of plugins to perform as expected during and after automated upgrades between long life releases of RHOSP. Successfully completing this test will add the corresponding feature to the Ecosystem Catalog entry for the certification.

Example

The “In-place upgrades from RHOSP 13 (Queens)” feature would appear as a line item in a RHOSP 16.1.x (Train) certification.

The in_place_upgrades test is available for all plugin types for which certification is available. It will appear on the certification test plan, and is only planned and executed on the Controller node. This feature and thus the test requires that your plugin be installed, managed, and upgradable via the RHOSP Director toolset. It is run after the upgrade is performed on the controller node and will introspect various log files in the upgraded environment to verify the upgrade conducted and was successful.

The In-place upgrades test includes the following subtests:

- **in_place_upgrades**
  This test validates that a RHOSP cloud has been upgraded from one RHOSP long life release to the next RHOSP long life release using the RHOSP Framework for Upgrades mechanism. And then prompts you to confirm the same.

  Success criteria

  - PASS: If the system is upgraded successfully and then you certify the same.
  - FAIL: If the system was not upgraded or you do not certify the same.

- **Director Plugin Verification**
  A self declaration prompt is added for you to declare if your plugin installation has been done with the director.

  Success criteria

  - PASS: If you certify that the plugin installation is done with the director.
  - FAIL: If you certify that the plugin installation is not done with the director.

Success criteria

The status of the In-place Upgrades test depends as follows:

- PASS: If both the subtests, in_place_upgrades and Director Plugin Verification pass.
- FAIL: If both or either subtest, in_place_upgrades and Director Plugin Verification fails.