



Red Hat OpenStack Certification 1.0

Red Hat OpenStack Certification Policy Guide

For Use with Red Hat OpenStack 16

Red Hat OpenStack Certification 1.0 Red Hat OpenStack Certification Policy Guide

For Use with Red Hat OpenStack 16

Legal Notice

Copyright © 2020 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

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.

Table of Contents

CHAPTER 1. INTRODUCTION	3
1.1. AUDIENCE	3
1.2. CREATING VALUE FOR OUR JOINT CUSTOMERS	3
1.3. RHOSP CERTIFICATION PREREQUISITES	3
1.4. RHOSP COMPONENT DISTRIBUTION	4
CHAPTER 2. RHOSP CERTIFICATION TARGETS	5
2.1. PRODUCTS IMPLEMENTING OPENSTACK APIS	5
2.2. PRODUCTS CONSUMING OPENSTACK APIS	5
2.3. SUPPORT FOR CERTIFIED OPENSTACK COMPONENT	5
CHAPTER 3. CERTIFICATION LIFECYCLE	7
3.1. PRODUCT CERTIFICATION LIFECYCLE	7
3.2. CONTINUAL TESTING	7
3.3. RECERTIFICATION	7
CHAPTER 4. SYSTEM REPORT	8
CHAPTER 5. TEST ENVIRONMENT SUPPORTABILITY	9
5.1. KERNEL	9
5.2. KERNEL MODULES	9
5.3. HARDWARE HEALTH	9
5.4. INSTALLED RPMS	10
5.5. SECURITY-ENHANCED LINUX (SELINUX)	11
CHAPTER 6. DIRECTOR	12
CHAPTER 7. CINDER	13
7.1. CINDER TEST	13
CHAPTER 8. MANILA	15
8.1. MANILA TEST	15
CHAPTER 9. NEUTRON	18
9.1. NEUTRON TEST	18
CHAPTER 10. OPENSTACK CONFIGURATION TEST	24
CHAPTER 11. TRUSTED CONTAINER TEST	25
CHAPTER 12. IN-PLACE UPGRADES	26

CHAPTER 1. INTRODUCTION

The Red Hat OpenStack Certification Policy guide delineate 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 RHOSP in a jointly supported customer environment.

1.2. CREATING VALUE FOR OUR JOINT 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.

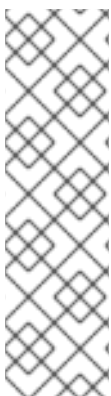
Logs from a single run with all of the mandatory tests and the test suite self-check test (rhcert/selfcheck) must be submitted to Red Hat for new certifications and for recertifications. For more information on running the tests, see [Red Hat OpenStack Certification Workflow Guide](#).

The certification tooling and workflow documented in the article will be supported for a period of 90 days to support certifications which are underway.

Partners 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 Partners 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 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 action(s) but do not prevent a certification from succeeding. Partners are recommended to review the output of such tests and perform appropriate action(s) based on the information contained within the warnings.

1.3. RHOSP CERTIFICATION PREREQUISITES

The following certification eligibility summarizes the prerequisites that Partners need in order to meet the OpenStack Certification requirements:

- 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
- Partners must have a support relationship with Red Hat. This can be fulfilled through:
 - custom support agreement, or
 - TSANet. See the [TSANet web page](#) for more information
- Partners must have a good working knowledge of RHOSP including installation and configuration of the product
- Partners 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.

To understand the product, you can use detailed product documentation on [Red Hat Customer Portal](#) or undertake the product training/certification on [Red Hat Training Page](#) .

1.4. RHOSP COMPONENT DISTRIBUTION

As part of 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. Partners 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.

CHAPTER 2. RHOSP CERTIFICATION TARGETS

Partners 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, Partners 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 vendor plugin controls.



IMPORTANT

It is the Partner's 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 of such applications include:

- 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, Partners 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.

For more information on supportability of RHOSP certified components, see to this [article](#).

CHAPTER 3. CERTIFICATION LIFECYCLE

3.1. PRODUCT CERTIFICATION LIFECYCLE

Starting with Red Hat OpenStack Platform 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 [Section 3.3, "Recertification"](#).

3.2. CONTINUAL TESTING

Partners are responsible for their own internal continual testing over the lifespan of their product and the Red Hat OpenStack Platform major version they are certified on. Partners 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.

Some partners 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 partners of and partners are requested to recertify their product in the following cases:

- A new major release of Red Hat OpenStack Platform.
- A new minor release of 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 Red Hat OpenStack Platform that updates the kernel and the partner product relies on kernel modules.

Partners will notify Red Hat of and partners 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.

CHAPTER 4. SYSTEM REPORT

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/system.

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

For more information about sosreport, see <https://access.redhat.com/solutions/3592>.

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.

CHAPTER 5. TEST ENVIRONMENT SUPPORTABILITY

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.

5.1. KERNEL

This subtest verifies that kernel is from Red Hat and inspect kernel is appropriate and supported for RHEL version included in the OpenStack deployment-under-test, and has not been modified. The kernel version may be the original General Availability (GA) version or any subsequent kernel released for the RHEL major releases.



NOTE

For more information on Red Hat Enterprise Linux Life Cycle and Kernel Versions, refer [Red Hat Enterprise Linux Life Cycle and Red Hat Enterprise Linux Release Dates](#) .

The kernel subtest also ensures that the kernel is not tainted when running in the environment. For more information about kernel tainting, see [Why is the kernel "tainted" and how are the taint values deciphered?](#)

Success Criteria

The running kernel is a Red Hat released kernel which should not be tainted and can be used with the RHEL version.

5.2. KERNEL MODULES

This subtest confirms the loaded kernel modules are from Red Hat, either from the running kernel's package or a Red Hat Driver Update (see [Where can I download Driver Update Program \(DUP\) disks](#)). This subtest also ensures the kernel modules do not identify as Technology Preview when running in the environment (see [What does a "Technology Preview" feature mean](#)).

Success Criteria

The kernel modules are from Red Hat and supported.

5.3. HARDWARE HEALTH

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 system under test (SUT) meets the minimum hardware requirements^[1] as follows:
 - RHEL 8: Minimum system RAM should be 1.5GB, per CPU logical core count^[2]
 - RHEL 7: Minimum system RAM should be 1GB, per CPU logical core count^[3]
- 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.

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 does 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.

5.4. INSTALLED RPMS

This subtest confirms that RPM packages installed on the system are from Red Hat and are not modified. This prevents potential significant risks to customer environments and ensures that customers make use of a supported environment.

Non Red Hat packages may be installed if they are necessary to enable the cloud environment but they are acceptable only when they are documented and do not modify or conflict with Red Hat packages/software. This subtest will require detailed review at Red Hat to confirm success or failure if Non Red Hat packages are installed.

For more information on Red Hat support policies on third-party software, see [Production Support Scope of Coverage](#).

Success Criteria

- The RMP packages installed are provided by Red Hat and is available in offering
- The installed Red Hat RPM packages are not modified

- The installed Non Red Hat RPM packages are necessary to enable the cloud environment, documented, and
- The installed Non Red Hat RPM packages do not conflict with available Red Hat provided packages/software and Red Hat products included in the offering.

5.5. SECURITY-ENHANCED LINUX (SELINUX)

This subtest confirms that 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..

For more information on SELinux in RHEL, see [SELinux Users and Administrators Guide](#).

Success Criteria

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

[1] [Minimum required memory](#)

[2] For more information about hardware support available in RHEL 7 but removed from RHEL 8, see [chapter Hardware Enablement](#) in the *Considerations in Adopting Red Hat Enterprise Linux 8* document.

[3] For more information about hardware support available in RHEL 6 but removed from RHEL 7, see [chapter Changes to Packages, Functionality, and Support](#) in the *RHEL 7 Migration Planning Guide* document.

CHAPTER 6. DIRECTOR

This test is 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 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.

For more information on installing RHOSP Director, see the [Director Installation and Usage Guide](#).

Success Criteria

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

CHAPTER 7. CINDER

The Cinder test is only applicable to OpenStack products/components that implement OpenStack features for the OpenStack Block Storage service. For more information, see [Section 2.1, "Products Implementing Openstack APIs"](#)

The test covers OpenStack Block Storage-component feature testing, which includes basic and operational functional testing using the [Tempest Framework](#) that is integrated in the RHOSP.

7.1. CINDER TEST

Based on the solutions provided by Partners, RHT will define a test plan in RH-cert web UI along with the test(s) that Partners needs to perform. The Cinder tests execute selected feature test(s) and, checks the plugin/driver functionality. The cinder test includes and supports the following features:

- **cinder_volumes**

It is imperative to perform cinder_volume test. The test will check if all the base functionality of associated actions with cinder such as volume actions, snapshots, boot, volume migrate, encryption, and clone are working, as well it will check plugin driver functionalities are working.

Success Criteria

All the driver functionalities mentioned are working for the tested drivers.

- **cinder_consistency_groups**

To ensure data consistency across different consistency group and disaster recovery, this test needs to be performed. This test needs to be effectuate only when consistency group is implemented in Partner solution. It will take multiple volume snapshot of the consistency group at the same time, and will also check all the actions related to consistency group work properly if vendor driver implements same features.

Following actions of consistency groups are tested:

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

Success Criteria

All the functions related to consistency groups are working which ensures data protection, data consistency and disaster recovery across consistency groups.

- **cinder_backups**

Performing this test will take backup of existing block storage volumes, by which backup can be restored, using the associated database information available in the Block Storage database.

Following actions of backup feature are tested:

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

Success Criteria

Performing backup for existing volume and creating snapshot of the complete backup volume.

- **cinder_multi-attach_volume**

The multi-attach_volume allows you to attach and access a single block storage volume to multiple hosts or servers. The cinder_multi-attach_volume test checks for the expected behavior of the multi-attach volume by incorporating the following tests:

- Boot from multi-attach volume
- Resize server with multi-attach volume
- List volume attachment for multi-attach volume
- Snapshot from backed multi-attach volume
- Attach/Detach volume served or offload server
- Delete attached multi-attach volume
- Attach multi-attach same or different server

Success Criteria

The ability to attach a volume to multiple hosts/servers simultaneously is expected in following scenarios:

- active/active mode
- active/standby mode

CHAPTER 8. MANILA

The openstack/manila test is only applicable to OpenStack products/components that implement OpenStack features for OpenStack file share service. For more information, see [Section 2.1, “Products Implementing Openstack APIs”](#).

The manila test cover OpenStack file share-component feature testing, which includes basic and operational functional testing using the [Tempest Framework](#) that is integrated in the RHOSP.

8.1. MANILA TEST

Based on the solutions provided by Partners, RHT will define a test plan in RH-cert web UI along with the test(s) that Partners needs to perform. The manila tests execute file share-component selected feature test(s) and, checks the plugin/driver functionality. The manila test includes and supports the following feature:

- **manila_shares**

Allows Partner to perform all the file operation and also provides NFS/CIFS protocol. This test will handle drivers with and without the driver handles share servers (DHSS) feature.

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:

- **manila_share_managed**

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

- **manila_share_shrink**

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

- **manila_share_extend**

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

- **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

- **manila_snapshot_managed**

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

- **manila_snapshot_share_from_snapshot**

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

- **manila_snapshot_revert_to_snapshot**

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

- **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

CHAPTER 9. NEUTRON

The openstack/neutron test is only applicable to OpenStack products/components that implement OpenStack features for the OpenStack networking Service. For more information, see [Section 2.1, “Products Implementing Openstack APIs”](#).

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.

9.1. NEUTRON TEST

Based on the solutions provided by Partners, RHT will define a test plan in RH-cert web UI along with the test(s) that Partners needs to perform. The neutron tests execute networking-component selected feature test(s) and, checks the plugin/driver functionality. The neutron test includes and supports the following features:

- **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.

- **neutron_agents**

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

Success Criteria

DHCP and L3 agent are operational.

- **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 associated.

- **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.

- **neutron_dhcp_extra**

The DHCP options extension allows adding DHCP options that are associated to a Neutron port. You can specify a 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` is tested to check if the new options can be applied.

Success Criteria

New dhcp options can successfully applied.

- **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 performed with the help of a third-party plugin/driver successfully.

- **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 successfully applied.

- **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.

- **neutron_ip_availability**

It allows a user or process to determine 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.

- **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.

- **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.

- **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.

- **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.

- **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 is performing successfully.

- **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.

- **neutron_lbaaSv2**

Load Balancing as a Service (LBaaS) v2 allows you to configure multiple listener ports on a single load balancer IP address. All the operation on load balancers are tested and checked to verify if LB's can be created, updated, deleted.

Success Criteria

Performs load balancing operations (creation, updation, deletion) successfully.

- **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.

- **neutron_mtu**

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

Success Criteria

MTU size is reflected.

- **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.

- **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.

- **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.

- **neutron_service_types**

Using this feature, Partners 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.

- **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.

- **neutron_subnet_default_pool**

This test checks the operations of default subnet pools.

Success Criteria

Default subnet pool operations performing successfully

- **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.

- **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.

- **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_gateway_protocol_vpn` test certifies following tempest test operations:

- create
- delete
- list
- show
- update

Success Criteria.

All the BGP VPN related operations performing successfully.

CHAPTER 10. OPENSTACK CONFIGURATION TEST

Starting 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. See, [OVN Architecture](#) for reference. 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.

CHAPTER 11. TRUSTED CONTAINER TEST

The Trusted Container test checks if Red Hat recognizes the RHOSP plugin/driver container. The test also verifies whether the container is provided by Red Hat or the Partners. 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 the Partner's 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 Partners 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. Partners 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 12. IN-PLACE UPGRADES

Starting with RHOSP 16.1.1, the Framework for Upgrades (FFU) is utilized in a new **In-place Upgrades** test. This test is available for Partners 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, for 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 the partners to confirm the same.

Success Criteria

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

- **Director Plugin Verification**

A self declaration prompt is added for the partners to declare if their plugin installation has been done with the director.

Success Criteria

- PASS: If the partners certify that the plugin installation is done with the director.
- FAIL: If the partners 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.