Red Hat OpenStack Red Hat OpenStack 2.1 (Folsom) Release Notes

Release Notes for Red Hat OpenStack 2.1 (Folsom)

Red Hat Engineering Content Services

Red Hat OpenStack Red Hat OpenStack 2.1 (Folsom) Release Notes

Release Notes for Red Hat OpenStack 2.1 (Folsom)

Red Hat Engineering Content Services

Legal Notice

Keywords

Abstract

The Release Notes document the major features, enhancements, and known issues of the Red Hat OpenStack 2.1 (Folsom) release.

Table of Contents

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the Liberation Fonts set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

> Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **gedit** file, choose **Applications** → **Accessories** →

> **Character Map** from the main menu bar. Next, choose **Search** → **Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.
>
> The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.
>
> To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[]) 
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. Getting Help and Giving Feedback

2.1. Do You Need Help?

If you experience difficulty with a procedure described in this documentation, visit the Red Hat Customer Portal at http://access.redhat.com. Through the customer portal, you can:

- search or browse through a knowledgebase of technical support articles about Red Hat products.
- submit a support case to Red Hat Global Support Services (GSS).
- access other product documentation.

Red Hat also hosts a large number of electronic mailing lists for discussion of Red Hat software and technology. You can find a list of publicly available mailing lists at https://www.redhat.com/mailman/listinfo. Click on the name of any mailing list to subscribe to that list or to access the list archives.

2.2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: http://bugzilla.redhat.com/ against the product **Red Hat OpenStack.**

When submitting a bug report, be sure to mention the manual's identifier: *doc-Release_Notes*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Chapter 1. Product Introduction

1.1. Overview

Red Hat OpenStack provides a cloud-based Infrastructure as a Service (IaaS), which is modular and scalable. The current Red Hat system is based on OpenStack Folsom, and packaged so that available physical hardware can be turned into a private, public, or hybrid cloud platform including:

- Fully distributed object storage
- Persistent block-level storage
- Virtual-machine provisioning engine and image storage
- Authentication and authorization mechanism
- Integrated networking
- Web browser-based GUI for both users and administration.

The Red Hat OpenStack IaaS cloud is implemented by a collection of interacting services that control its computing, storage, and networking resources. The cloud is managed using a web-based interface which allows administrators to control, provision, and automate Red Hat OpenStack resources. Additionally, the Red Hat OpenStack infrastructure is facilitated through an extensive API, which is also available to end users of the cloud.

1.2. Architecture

The following diagram provides a high-level overview of the Red Hat Openstack architecture.

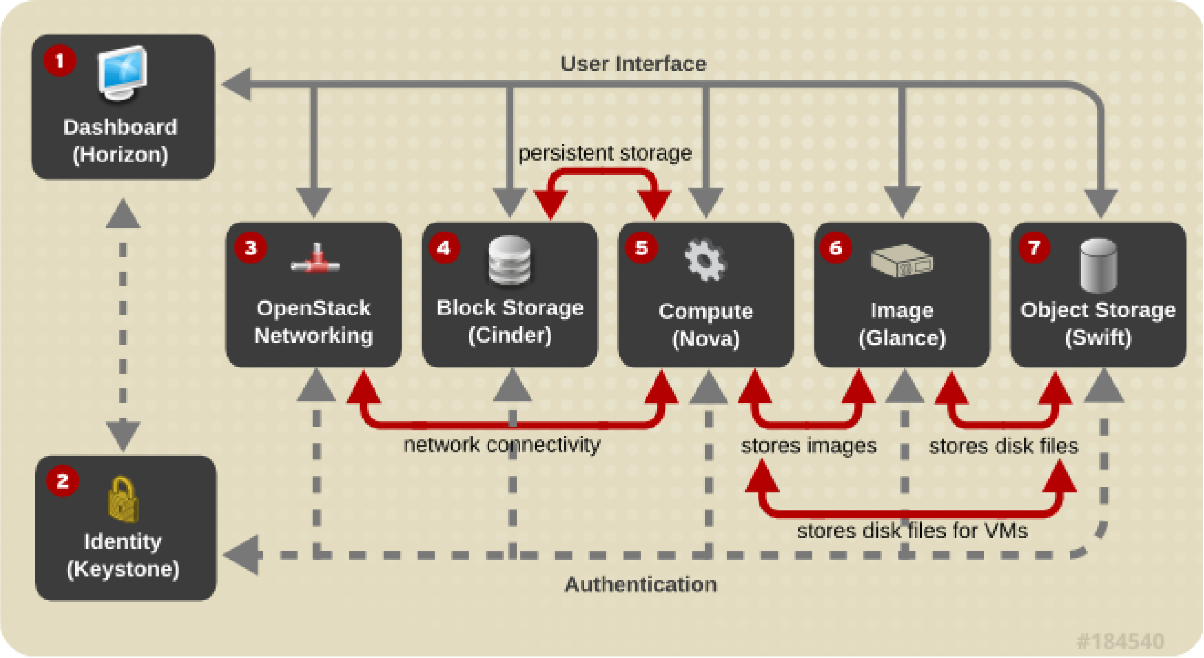


Table 1.1. Services

	Section	Description
❶	Section 1.3.1, “Dashboard - Horizon ”	A web-based dashboard for managing Red Hat OpenStack services.
❷	Section 1.3.2, “Identity - Keystone ”	A centralized identity service that provides authentication and authorization for other services, and manages users, tenants, and roles.
❸	Section 1.3.3, “OpenStack Networking ”	A networking service that provides connectivity between the interfaces of other Red Hat OpenStack services.
❹	Section 1.3.4, “Block Storage - Cinder”	A service that manages persistent block storage volumes for virtual machines.
❺	Section 1.3.5, “Compute - Nova”	A service that launches and schedules networks of machines running on nodes.
❻	Section 1.3.6, “Image - Glance”	A registry service for virtual machine images.
❼	Section 1.3.7, “Object Storage - Swift”	A service providing object storage which allows users to store and retrieve files (arbitrary data).

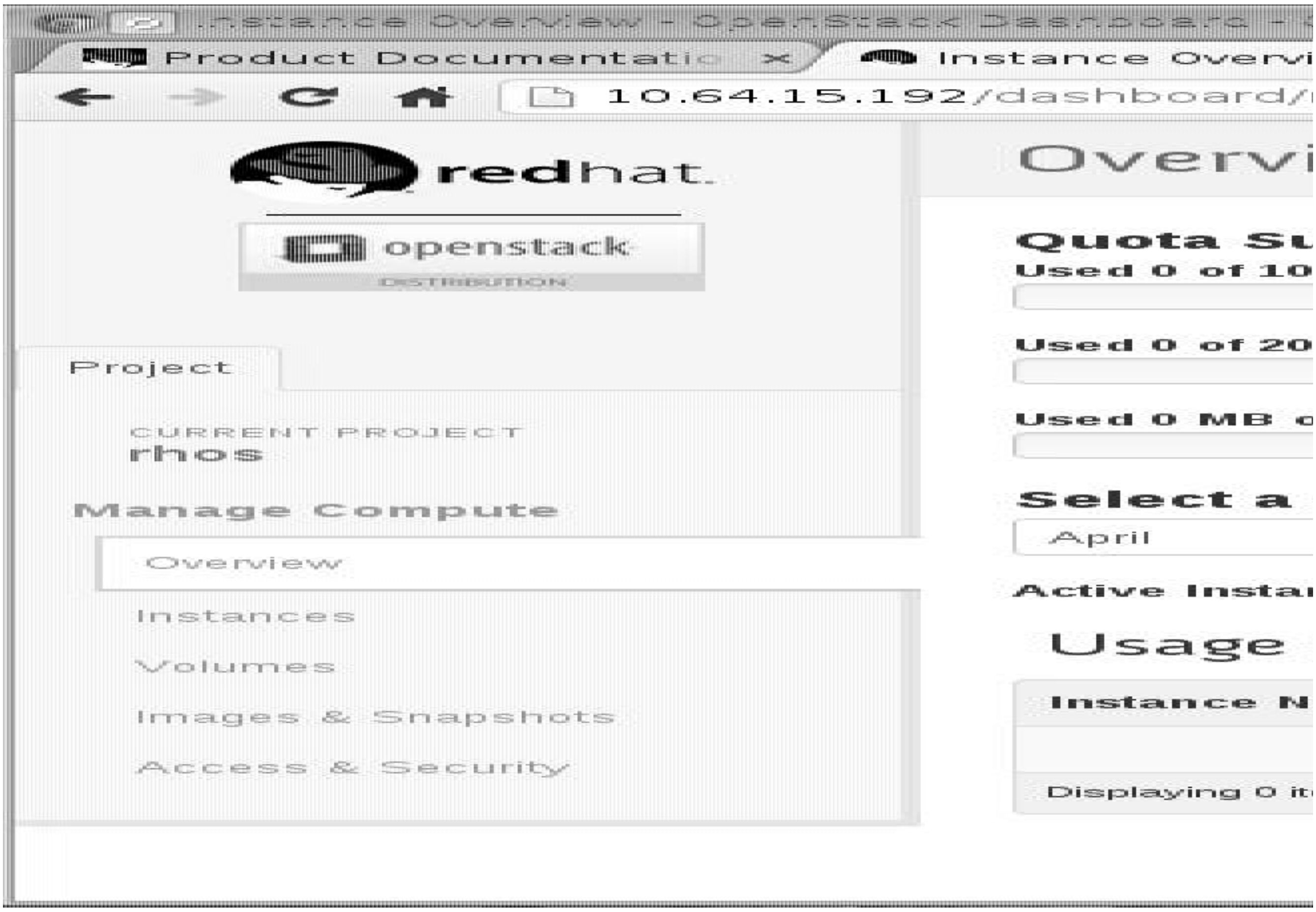
1.3. Service Details

This section provides more detailed information about the Red Hat Openstack service components. Each Red Hat OpenStack service is comprised of a collection of Linux services, MySQL databases, or other components, which together provide a functional group. For example, the **`glance-api`** and **`glance-registry`** Linux services, together with a MySQL database, implement the Glance service.

1.3.1. Dashboard - Horizon

Horizon provides a graphical user interface for end users and administrators, allowing operations such as creating and launching instances, managing networking, and setting access controls. Its modular design allows interfacing with other products such as billing, monitoring, and additional management tools. Horizon provides three basic dashboards: user, system, and settings.

The following screenshot displays a user's dashboard after Red Hat OpenStack is first installed:



The identity of the logged-in user determines the dashboards and panels that are visible within Horizon.

Horizon is composed of:

- **`openstack-dashboard`**, a Django (Python) web application, so that the dashboard can be easily accessed using any web browser.
- An Apache HTTP server (**`httpd`** service), to host the application.
- A MySQL database, for managing sessions.

1.3.2. Identity - Keystone

The Keystone service authenticates and authorizes OpenStack users (that is, keeps track of users and their permitted activities); the service is used by all OpenStack components. Keystone supports multiple forms of authentication including user name and password credentials, token-based systems, and AWS-style logins (Amazon Web Services).

Keystone also provides a central catalog of services and endpoints running in a particular Red Hat OpenStack cloud, which acts as a service directory for other Red Hat OpenStack systems.

The Keystone service uses the following concepts:

- Users, which have associated information (such as a name and password). In addition to custom users, a user is automatically defined for each cataloged service (for example, the 'glance' user for the Glance service), who belongs to the special tenant 'service'.
- Tenants, which are generally the user's group, project, or organization.
- Roles, which determine a user's permissions.

Keystone is composed of:

- **`keystone`** service, which provides the administrative and public APIs.
- MySQL databases for each of the internal services.

1.3.3. OpenStack Networking

The OpenStack Networking service provides a scalable and API-driven system for managing the Red Hat OpenStack network and IP addresses. Because the OpenStack network is software-defined, it can easily and quickly react to changing network needs (for example, creating and assigning new IP addresses).

Among its advantages:

- Users can create networks, control traffic, and connect servers and devices to one or more networks.
- OpenStack offers flexible networking models, so that administrators can change the networking model to adapt to their volume and tenancy.
- IPs can be dedicated or floating; floating IPs allow dynamic traffic rerouting.

The following networking models are currently available:

- Flat Network Manager - The network administrator specifies a subnet, and IP addresses for VM instances are taken from the subnet and injected into the image upon launch. The Linux networking bridge is manually configured.
- Flat DHCP Network Manager - IP addresses are taken from the DHCP server (dnsmasq). The networking bridge is still manually configured.
- Flat VLAN Network Manager (default mode)- A VLAN and bridge are automatically created for each project. The project is assigned a range of private IPs that are only accessible from inside the VLAN. Subnets can be defined by the network administrator, and assigned dynamically as required.

OpenStack Networking is composed of:

- **`quantum-server`** Python daemon, which manages user requests (and exposes the API)
- **`L3-agent`**, which provides L3/NAT forwarding.
- **`plugin-agent`**, which run on each hypervisor to perform local vswitch configuration.
- **`dhcp-agent`**, which provides DHCP services to tenant networks.
- MySQL database, for persistent storage.

1.3.4. Block Storage - Cinder

The Cinder service provides persistent block storage management for virtual hard drives. The block storage system manages the creation, attaching, and detaching of the block devices to servers. Block storage volumes are fully integrated into Nova and Horizon, which allows cloud users to manage their own storage needs.

Block storage is appropriate for performance-sensitive scenarios such as database storage, expandable file systems, or providing a server with access to raw block-level storage. Additionally, snapshots can be taken to either restore data or to create new block storage volumes (snapshots are dependent upon driver support).

Basic operations include:

- Create, list, and delete volumes.
- Create, list, and delete snapshots.
- Attach and detach volumes to running virtual machines.

Cinder is composed of the following:

- **`openstack-cinder-volume`**, which carves out storage for virtual machines on demand. A number of drivers are provided for interaction with storage providers.
- **`openstack-cinder-api`**, which responds to and handles requests, and places them in the message queue.
- **`openstack-cinder-scheduler`**, which assigns tasks to the queue and determines the provisioning volume server.
- MySQL database, for state information.

1.3.5. Compute - Nova

The Nova service is the heart of the Red Hat OpenStack cloud by providing virtual machines on demand. That is, Nova schedules virtual machines to run on a set of nodes (including virtual servers, logical containers, and GPUs). It does this by defining drivers that interact with underlying virtualization mechanisms, and exposing the functionality to the other OpenStack components.

Nova interacts with Keystone for authentication, Glance for images, and Horizon for the user and administrative interface. Access to images is limited by project and by user; quotas are limited per project (for example, the number of instances). Nova is designed to scale horizontally on standard hardware, and can download images to launch instances as required.

Nova is composed of the following:

- **`openstack-nova-api`** service, which handles requests and provides access to the Nova services (such as booting an instance).
- **`openstack-nova-cert`** service, which provides the certificate manager.
- **`openstack-nova-compute`** service, which creates and terminates the virtual instances. The service interacts with Hypervisor to bring up new instances, and ensures that the state is maintained in the Nova database.
- **`openstack-nova-consoleauth`** service, which handles console authorization.
- **`openstack-nova-network`** service, which handles Nova network traffic (both private and public access). This service handles such tasks as assigning an IP address to a new virtual instance, and implementing security group rules.
- **`openstack-nova-novncproxy`** service, which provides a VNC proxy for browsers (enabling VNC consoles to access virtual machines started by OpenStack).
- **`openstack-nova-scheduler`** service, which dispatches requests for new virtual machines to the correct node.
- Apache Qpid server (**`qpidd`** service), which provides the AMPQ message queue. This server (also used by Cinder) handles the OpenStack transaction management, including queuing, distribution, security, management, clustering, and federation. Messaging becomes especially important when a Red Hat OpenStack deployment is scaled and its services are running on multiple machines.
- **`libvirtd`** service, which enables the creation of virtual machines (that is, it is the driver for the hypervisor).
- KVM Linux hypervisor, which creates virtual machines and enables their live migration from node to node.
- MySQL database, for build-time and run-time infrastructure state.

1.3.6. Image - Glance

The Glance service acts as a registry for virtual disk images. Users can add new images or take a snapshot (copy) of an existing server for immediate storage. Snapshots can be used as back up or as templates for new servers. Registered images can be stored in the Swift service, as well as in other locations (for example, in simple file systems or external web servers).

The following image formats are supported:

- raw (unstructured format)
- aki/ami/ari (Amazon kernal, ramdisk, or machine image).
- iso (archive format for optical discs; for example, CDROM)
- qcow2 (Qemu/KVM, supports *Copy on Write*)
- vhd (Hyper-V, common for virtual machine monitors from VMWare, Xen, Microsoft, VirtualBox, and others)
- vdi (Qemu/VirtualBox)
- vmdk (VMWare)

Container formats can also be used by Glance; the format determines the type of metadata stored in the image about the actual virtual machine. The following formats are supported.

- bare (no metadata is included)
- ovf (OVF format)
- aki/ami/ari (Amazon kernel, ramdisk, or machine image)

Glance is composed of the following Linux services:

- **`openstack-glance-api`**, which handles requests, and image delivery (interacts with storage backends for retrieval and storage). This service uses the registry to retrieve image information (the registry service is never, and should never be, accessed directly).
- **`openstack-glance-registry`**, which manages all metadata associated with each image, and which requires a database.
- MySQL database, for image metadata.

1.3.7. Object Storage - Swift

The Swift service provides object storage in virtual containers, which allows users to store and retrieve files. The distributed Swift architecture supports horizontal scaling; redundancy as failure-proofing is provided through software-based data replication.

Because it supports asynchronous eventual consistency replication, it is well suited to multiple data-center deployment. Swift uses the concept of:

- Storage replicas, which are used to maintain the state of objects in the case of outage. A minimum of three replicas is recommended.
- Storage zones, which are used to host replicas. Zones ensure that each replica of a given object can be stored separately. A zone might represent an individual disk drive or array, a server, all the servers in a rack, or even an entire data center.

Swift is composed of the following:

- **`openstack-swift-proxy`** service, which exposes the public API, and is responsible for handling requests and routing them accordingly. Objects are streamed through the proxy server to the user (not spooled). Objects can also be served out via HTTP.
- **`openstack-swift-object`** blob server, which stores, retrieves, and deletes objects.
- **`openstack-swift-account`** server, which is responsible for listings of containers, using the MySQL account database.

- **`openstack-swift-container`** server, which handles listings of objects (what objects are in a specific container) using the MySQL container database.
- Ring files, which contain details of all the storage devices, and which are used to deduce where a particular piece of data is stored (maps the names of stored entities to their physical location). One file is created for each object, account, and container server.
- MySQL account database
- MySQL container database
- Ext4 (recommended) or XFS filesystem for object storage.
- Housekeeping processes, including replication and auditors.

Chapter 2. Release Introduction

2.1. About this Release

This release of Red Hat OpenStack is based on the OpenStack "Folsom" release. It includes updates made both in the initial "Folsom" release and subsequent updates to fix various security issues and bugs. It also includes additional features, known issues, and resolved issues specific to Red Hat OpenStack.

Only changes specific to Red Hat OpenStack are included in this release notes document. The release notes for the OpenStack "Folsom" release itself, and subsequent updates, are available at these locations:

OpenStack "Folsom" Release Notes
: https://wiki.openstack.org/wiki/ReleaseNotes/Folsom

OpenStack "Folsom" 2012.2.1 Release Notes
: https://wiki.openstack.org/wiki/ReleaseNotes/2012.2.1

OpenStack "Folsom" 2012.2.2 Release Notes
: https://wiki.openstack.org/wiki/ReleaseNotes/2012.2.2

OpenStack "Folsom" 2012.2.3 Release Notes
: https://wiki.openstack.org/wiki/ReleaseNotes/2012.2.3

OpenStack "Folsom" 2012.2.4 Release Notes
: https://wiki.openstack.org/wiki/ReleaseNotes/2012.2.4

2.2. Product Support

The Red Hat OpenStack preview is available free of charge, but does not come with any formal technical support from Red Hat at this time. Red Hat continues to actively work with and seek feedback from customers regarding the preview release to further improve the distribution.

To sign up for the Red Hat OpenStack preview visit http://www.redhat.com/openstack.

Even though the preview is not officially supported, our engineers and support staff want to hear your feedback and provide assistance. We have set up an email list and bug tracking system to allow users of the Red Hat OpenStack preview to interface with our engineering team.

Available resources include:

Customer Portal

The Red Hat Customer Portal offers a wide range of resources to help guide you through planning, deploying, and maintaining your OpenStack deployment. Facilities available via the Customer Portal include:

- Knowledge base articles and solutions.
- Reference architectures.

- Technical briefs.
- Product documentation.

Access the Customer Portal at https://access.redhat.com/.

Mailing Lists

The **rhsa-announce** mailing list provides notification of the release of security fixes for all Red Hat products, including Red Hat OpenStack.

Subscribe at https://www.redhat.com/mailman/listinfo/rhsa-announce.

> **Note**
>
> The full list of updates released for Red Hat OpenStack 2.0 and 2.1 (Folsom) is maintained at https://rhn.redhat.com/errata/rhel6-rhos-folsom-errata.html.

Community Documentation

Additional documentation provided by the wider OpenStack community is available at http://docs.openstack.org.

Chapter 3. Known Issues

Important

Due to dependencies amongst OpenStack's services, a package update will trigger a service restart if that service is running at the time of the update. However, to have tight control over how and when restarts are done, you should follow a process which implements a preparation specific to your site, then a yum update, and then a restart of services that were stopped during the preparation.

- OpenStack Networking requires some specific setup and configuration to be done for networking to function correctly. A knowledge base of relevant configuration and networking requirements can be found at https://access.redhat.com/knowledge/articles/339573
- Cinder is the supported method for providing block storage to a Nova deployment. The **`nova-volume`** service is no longer supported. If you have an older **`nova-volume`** installation, there are instructions for migrating to **`cinder-volume`** available at https://wiki.openstack.org/wiki/MigrateToCinder.
- Red Hat OpenStack does not yet fully support being used with ipv6 networking technologies. Only ipv4 is supported at this time.
- Red Hat OpenStack is only supported for use with the AMQP messaging provided by Apache Qpid. The RabbitMQ messaging service driver is included, but Red Hat cannot provide direct technical support for this driver.
- Red Hat OpenStack is only supported for use with the MySQL database driver. The PostgreSQL database driver is also included and although not yet supported, Red Hat would like feedback on any deployment success or problems, with a view to providing full support in a future release.

  For more information regarding database support refer to: https://access.redhat.com/knowledge/articles/340383.
- Red Hat OpenStack is only supported for use with the libvirt driver, using KVM as the hypervisor on Nova compute Nodes. Red Hat is unable to provide support for other Nova virtualization drivers, or non-KVM libvirt hypervisors.
- Some packages in the Red Hat OpenStack software repositories conflict with packages in the Extra Packages for Enterprise Linux (EPEL) software repositories.

  When installing Red Hat OpenStack on systems that are configured to retrieve packages from the EPEL software repositories you must ensure that the Red Hat OpenStack software repositories have the highest yum priority.

  Refer to the Red Hat OpenStack Getting Started Guide for more information on configuring yum priorities.
- Red Hat Support maintains a list of Network Interface Cards (NICs) supported for use with Open vSwitch in combination with VLAN tagging:

  https://access.redhat.com/knowledge/articles/289823

  If you wish to use Open vSwitch in combination with VLAN tagging and your NIC is not listed as supported then please contact Red Hat Support for more information.
- Note that the version of Horizon dashboard shipped with the OpenStack 2.1 (Folsom) release is not backwards compatible with the Essex release.
- When using the Horizon dashboard, images with these underlying disk formats are hidden from view:
  - Amazon kernel image (AKI).
  - Amazon RAMdisk image (ARI).

  It is not possible to launch instances from disk images that use these formats and they are not intended to be editable by regular users. To interact with ARI and AKI formatted disk images, access the dashboard as an administrative user or use the Glance command-line client.

- Red Hat OpenStack includes Puppet (http://www.puppetlabs.com/). Puppet is provided to support the rapid deployment of Red Hat OpenStack on existing servers using the **packstack** utility. Use of Puppet and the provided Puppet manifest files outside of this context is not currently supported by Red Hat.
- When using the NFS driver for the Cinder volume storage service, all compute nodes that will access volumes stored on NFS to host virtual machines must be configured to allow this access. To ensure this is the case log in to each node as the **root** user and set the **virt_use_nfs** SELinux boolean to **true**:

  ```
  # setsebool -P virt_use_nfs true
  ```

- Setting the configuration option **resume_guests_state_on_host_boot** to **True** is not recommended (The default value is **False**). Setting it to **True** causes problems with re-spawning instances when many services are being restarted simultaneously. This usually occurs when the services are running on the same host that is being restarted.
- Nova's **cloudpipe** VPN functionality is not supported. Red Hat is investing heavily in the development of OpenStack Networking Services to provide OpenStack networking. Support for equivalent functionality in Nova has thus been deferred.
- When running 2 or more active controller nodes, do not run **nova-consoleauth** on more than one node. Running more than one instance of **nova-consoleauth** causes conflicts between nodes with regard to token requests which may cause errors.
- It is recommended that you do not run sample scripts when installing production systems. Sample scripts are for demonstration and testing only. Specifically, the **openstack-keystone** package includes a script which can be executed by the **openstack-demo-install** script from the **openstack-utils** package. Hence, running **sample_data.sh** directly, or indirectly via **openstack-demo-install** script, will create OpenStack Keystone accounts with default credentials.
- Avoid using **nova-rootwrap**, because Nova attempts a **sudo chown** even if the instances directory is located on the NFS share. In order to use **nova-rootwrap**, you must be aware of the issues with using NFS and root owned files. The NFS share must be configured with **no_root_squash**.
- When the **openvswitch** quantum plugin is used, and Nova is configured with

  ```
  libvirt_vif_driver = nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
  ```

  the necessary forwarding rules are not created automatically and the Red Hat Enterprise Linux firewall blocks forwarding of network traffic. Hence traffic between VMs located on different compute nodes is blocked.

  Workarounds to avoid blocking traffic between VMs located on different compute nodes:

  1. If using nova security groups, add the following **iptables** rule on each compute node:

     ```
     #iptables -t filter -I FORWARD -i qbr+ -o qbr+ -j ACCEPT
     # service iptables save
     ```

     Either reboot, or restart **nova-compute** after adding this rule, since the rules **nova-compute** adds at startup must precede this rule.
  2. If not using Nova security groups, an alternative solution is to set:

```
libvirt_vif_driver =
nova.virt.libvirt.vif.LibvirtOpenVswitchVirtualPortDriver
```

- When using the following **nova-manage** commands you must restart all the networking services, including all **nova-network** and the **dnsmasq** processes, in order for the changes to be picked up:

  ```
  nova-manage network create
  nova-manage network delete
  ```

  These commands cannot be done via the API and need to be done through **nova-manage**.
- Glance does not fully support a graceful restart yet. Hence, image transfers that are still in progress will be lost when Glance services are restarted. This will occur when updating the *openstack-glance* package.

  The workaround to avoid losing images is to wait for image transfers that are in progress to complete, before updating the *openstack-glance* package or restarting Glance services.

  If there are no image transfers in progress during installation of a new version of the *openstack-glance* package or during a restart of the Glance services, then this problem will not occur.
- The OpenStack Networking DHCP and L3 agents must be deployed on separate hosts to ensure reliable networking.
- The **nova-manage** service list only lists services listening on the message bus, and hence should not be used to determine which OpenStack daemons are running on a host. Use the standard service interface to check the status instead.
- Attaching a volume stored on GlusterFS to a Nova compute instance is known to fail when the version of the selinux-policy package installed is less than 3.7.19-195.el6_4.2. Attaching such a volume with SELinux in Enforcing mode will result in AVC messages being generated.

  To work around this issue, update to selinux-policy-3.7.19-195.el6_4.2 or later. This package is available in the Red Hat Enterprise Linux 6.4.z channel. Users who have updated to the latest version of selinux-policy package are able to run in Enforcing mode.
- Red Hat Enterprise Linux 6.4 does not support network namespaces. Therefore the **use_namespaces** configuration variable in **/usr/share/quantum/quantum-dist.conf** is set to **False** by default. This has several consequences:

  1. When **use_namespaces** is **False** and **quantum-l3-agent** configures a virtual router with an external gateway, rather than setting the default route in a network namespace dedicated to that router, it sets the host's default route to be via the router's external gateway.

     As a consequence, changing the default route can break connectivity to the host's management interface, especially if the virtual router's external gateway and the host's management interface are in different network interfaces and/or different subnets. Thus, the recommended deployment architecture of having separate management, data, and external networks may not be achievable.

     Connectivity to the management interface should not be affected for access from within the management interface's subnet, so access to the host can be achieved by connecting from another host on the same subnet. However, access by the host to external services such as DNS servers, yum repositories, etc. that are not on the same subnet may not work.

     As a workaround, use the same interface and subnet for the management interface and for the external network.
  2. Attempting to create subnets on different networks with overlapping IP address ranges fails. This occurs because the IP address ranges available to one tenant are affected by the address ranges other tenants are using.

     A workaround is to place all subnets on non-overlapping IP address ranges.
  3. Without using network namespaces, a node running **quantum-l3-agent** can only provide a

single virtual router.

One way to work around this issue is to deploy a separate **quantum-l3-agent** on a separate node for each virtual router that is created. For each **quantum-l3-agent**, set **router_id** in **/etc/quantum/l3_agent.ini** to the UUID identifying the virtual router that it hosts. This allows multiple virtual routers to be created. However, since administrative action is needed for each, self-service deployment of routers by tenants is not possible.

- The **quantum-l3-agent** does not create firewall rules to allow traffic to be forwarded between its interface and gateway ports, and the Red Hat Enterprise Linux firewall's FORWARD chain rejects traffic by default. As a consequence, traffic between VMs and the external network is not forwarded.

  A workaround is to add the following iptable rules on each **quantum-l3-agent** node:

  ```
  iptables -t filter -I FORWARD -i qr-+ -o qg-+ -j ACCEPT
  iptables -t filter -I FORWARD -i qg-+ -o qr-+ -j ACCEPT
  iptables -t filter -I FORWARD -i qr-+ -o qr-+ -j ACCEPT
     # service iptables save
  ```

  This allows traffic to be forwarded as expected.

- Two different mechanisms are available for connecting a **quantum-l3-agent** to an external network. The first, using an external bridge (typically **br-ex**) only applies to the **openvswitch** plugin. The second, using a provider external network, applies to both the **openvswitch** and **linuxbridge** plugins.

  Using a provider external network involves creating a provider network with external connectivity, creating a subnet on that provider network describing the external network's IP addressing and gateway, and deploying the **quantum-l3-agent** with **external_network_bridge** turned off in **/etc/quantum/l3-agent.ini**. The following steps can be followed to configure a router with a provider VLAN as its external network:

  ```
   # quantum net-create external_network_name --router:external True \
       --provider:network_type vlan --provider:physical_network
  physical_network_name \
       --provider:segmentation_id external_VLAN_tag

   # quantum subnet-create --gateway external_gateway_IP \
       --allocation-pool start=external_IP_start,end=external_IP_end \
       --disable-dhcp external_network_name external_network_CIDR

   # quantum router-create router1

   # quantum router-gateway-set router1 external_network_UUID

   # quantum router-interface-add router1 private_subnet_UUID

   # quantum-l3-setup --plugin plugin_name

   # openstack-config --set /etc/quantum/l3_agent.ini DEFAULT
  external_network_bridge ""

   # openstack-config --set /etc/quantum/l3_agent.ini DEFAULT router_id
  router_UUID

   # service quantum-l3-agent start

   # chkconfig quantum-l3-agent on
  ```

  Note that ***external_gateway_IP*** must be within ***external_network_CIDR***, and that

external_IP_start and ***external_IP_end*** must specify a range for floating IPs within ***external_network_CIDR*** that does not include ***external_gateway_IP***.

A flat network can be used as the external provider network by substituting the following **net-create** command:

```
# quantum net-create external_network_name --router:external True \
    --provider:network_type flat \
    --provider:physical_network physical_network_name
```

- The limit set for Nova processes may be exceeded in very large deployments. Then a problem may occur where you get AVC denials for **sys_resource** and **sys_admin** while running Nova. For example:

```
avc:  denied  { sys_admin } for  pid=16497 comm="iptables-save" capability=21
scontext=unconfined_u:system_r:iptables_t:s0
tcontext=unconfined_u:system_r:iptables_t:s0 tclass=capability
```

Due to the way process inheritance is set up on Linux, calling sudo inherits the caller's ulimit. Processes owned by the new UID are counted against the inherited ulimit. Transitioning to the iptables domain drops the ability to break the soft ulimit for number of processes, which causes iptables commands to fail in certain cases. Currently the limit to the number of processes is set to 2048 for the Nova user.

While this limit should work for most installations, very large deployments may need a workaround. The workaround is to increase the limit by editing **/etc/security/limits.d/91-nova.conf**.

For example, change:

```
nova     soft    nproc     2048
```

to:

```
nova     soft    nproc     4096
```

Revision History

Revision 1.0-10	**Thu May 9 2013**	**Steve Gordon**	BZ#950454 - Added 2012.2.4 stable release..
Revision 1.0-9	**Wed April 24 2013**	**Steve Gordon**	BZ#950869 - Removed identity, token, and catalog descriptions under keystone service. BZ#950876 - Updated Nova compute introductory material.
Revision 1.0-8	**Mon April 22 2013**	**Steve Gordon**	Updated to provide a number of corrections to introductory text and add to known issues list.
Revision 1.0-7	**Wed April 10 2013**	**Bruce Reeler**	First General Availability release of Folsom. (Release 2.1)
Revision 1.0-6	**Tue April 9 2013**	**Summer Long**	New chapter containing product introduction.
Revision 1.0-5	**Tue Mar 19 2013**	**Stephen Gordon**	Restructured document.
Revision 1.0-4	**Thu Feb 7 2013**	**Bruce Reeler**	Renamed "Quantum" to "OpenStack Network".
Revision 1.0-3	**Tue Jan 22 2013**	**Stephen Gordon**	Edit OpenStack Network Known Issues.
Revision 1.0-2	**Wed Nov 14 2012**	**Alan Pevec**	Edit OpenStack Network Known Issues.
Revision 1.0-1	**Mon Nov 12 2012**	**Bruce Reeler**	Edits to Release notes for Folsom Preview release.
Revision 1.0-0	**Sun Nov 11 2012**	**Bruce Reeler**	First Folsom Preview release.