Red Hat OpenStack Red Hat OpenStack 3.0 (Grizzly) Getting Started Guide

Getting Started with Red Hat Enterprise Linux OpenStack Platform 3 (Grizzly)

Edition 1.0

Steve Gordon
Deepti Navale
Steve Gordon
Bruce Reeler
Summer Long

Red Hat OpenStack Red Hat OpenStack 3.0 (Grizzly) Getting Started Guide

Getting Started with Red Hat Enterprise Linux OpenStack Platform 3 (Grizzly)

Edition 1.0

Steve Gordon
sgordon@redhat.com

Summer Long
slong@redhat.com

Deepti Navale
dnavale@redhat.com

Bruce Reeler
breeler@redhat.com

Legal Notice

Keywords

Abstract

This manual covers the basic getting started tasks for OpenStack Grizzly.

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 include 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, select 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:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                 struct kvm_assigned_pci_dev *assigned_dev)
{
         int r = 0;
         struct kvm_assigned_dev_kernel *match;

         mutex_lock(&kvm->lock);

         match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                       assigned_dev->assigned_dev_id);
         if (!match) {
                 printk(KERN_INFO "%s: device hasn't been assigned before, "
                   "so cannot be deassigned\n", __func__);
                 r = -EINVAL;
                 goto out;
         }

         kvm_deassign_device(kvm, match);

         kvm_free_assigned_device(kvm, match);

out:
         mutex_unlock(&kvm->lock);
         return r;
}
```

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 **OpenStack.**

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

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.

Part I. Introduction

Chapter 1. Product Introduction

1.1. Overview

Red Hat Enterprise Linux OpenStack Platform provides the foundation to build a private or public Infrastructure-as-a-Service (IaaS) cloud on top of Red Hat Enterprise Linux. It offers a massively scalable, fault-tolerant platform for the development of cloud-enabled workloads.

The current Red Hat system is based on OpenStack Grizzly, 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 Enterprise Linux OpenStack Platform 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 OpenStack resources. Additionally, the 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 OpenStack architecture.

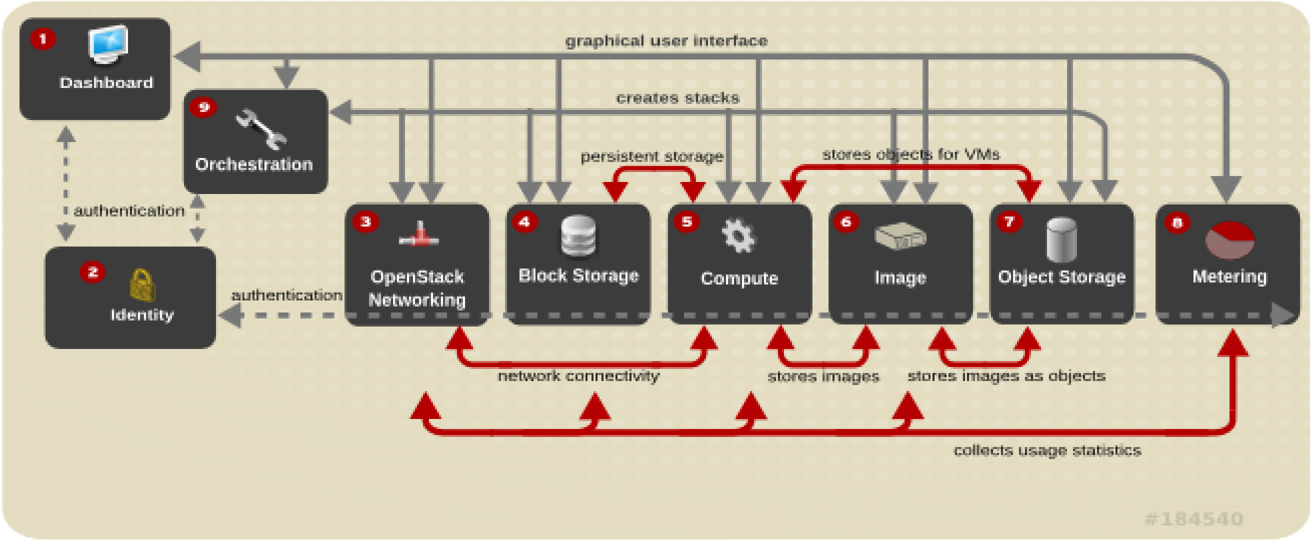


Figure 1.1. OpenStack Architecture

Each OpenStack service has a code name, which is reflected in the names of configuration files and command-line utility programs. For example, the Identity service has a configuration file called **`keystone.conf`**.

Table 1.1. Services

	Section	Code name	Description
1	Dashboard	Horizon	A web-based dashboard for managing OpenStack services.
2	Identity	Keystone	A centralized identity service that provides authentication and authorization for other services, and manages users, tenants, and roles.
3	OpenStack Networking	Quantum	A networking service that provides connectivity between the interfaces of other OpenStack services.
4	Block Storage	Cinder	A service that manages persistent block storage volumes for virtual machines.
5	Compute	Nova	A service that launches and schedules networks of machines running on nodes.
6	Image	Glance	A registry service for virtual machine images.
7	Object Storage	Swift	A service providing object storage which allows users to store and retrieve files (arbitrary data).
8	Metering (Technical Preview)	Ceilometer	A service providing measurements of cloud resources.
9	Orchestration (Technical Preview)	Heat	A service providing a template-based orchestration engine, which supports the automatic creation of resource stacks.

The Service Details section provides more detailed information about the OpenStack service components. Each OpenStack service is comprised of a collection of Linux services, MySQL databases, or other components. Together these provide a functional group. For example, the **`glance-api`** and **`glance-registry`** Linux services, together with a MySQL database, implement the Image service.

Important

For more information on the support scope for features marked as technical previews, refer to https://access.redhat.com/support/offerings/techpreview/

1.3. The PackStack Deployment Utility

PackStack is a command line utility that enables rapid deployment of OpenStack on existing servers over an SSH connection. PackStack is suitable for deploying both single-node proof-of-concept installations and more complex multi-node installations. Deployment options are provided either interactively, via the command line, or non-interactively by means of a text file containing a set of preconfigured values for OpenStack parameters.

1.4. OpenStack Service Details

1.4.1. Dashboard Service

The Dashboard service 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. The service provides three basic dashboards: user, system, and settings.

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

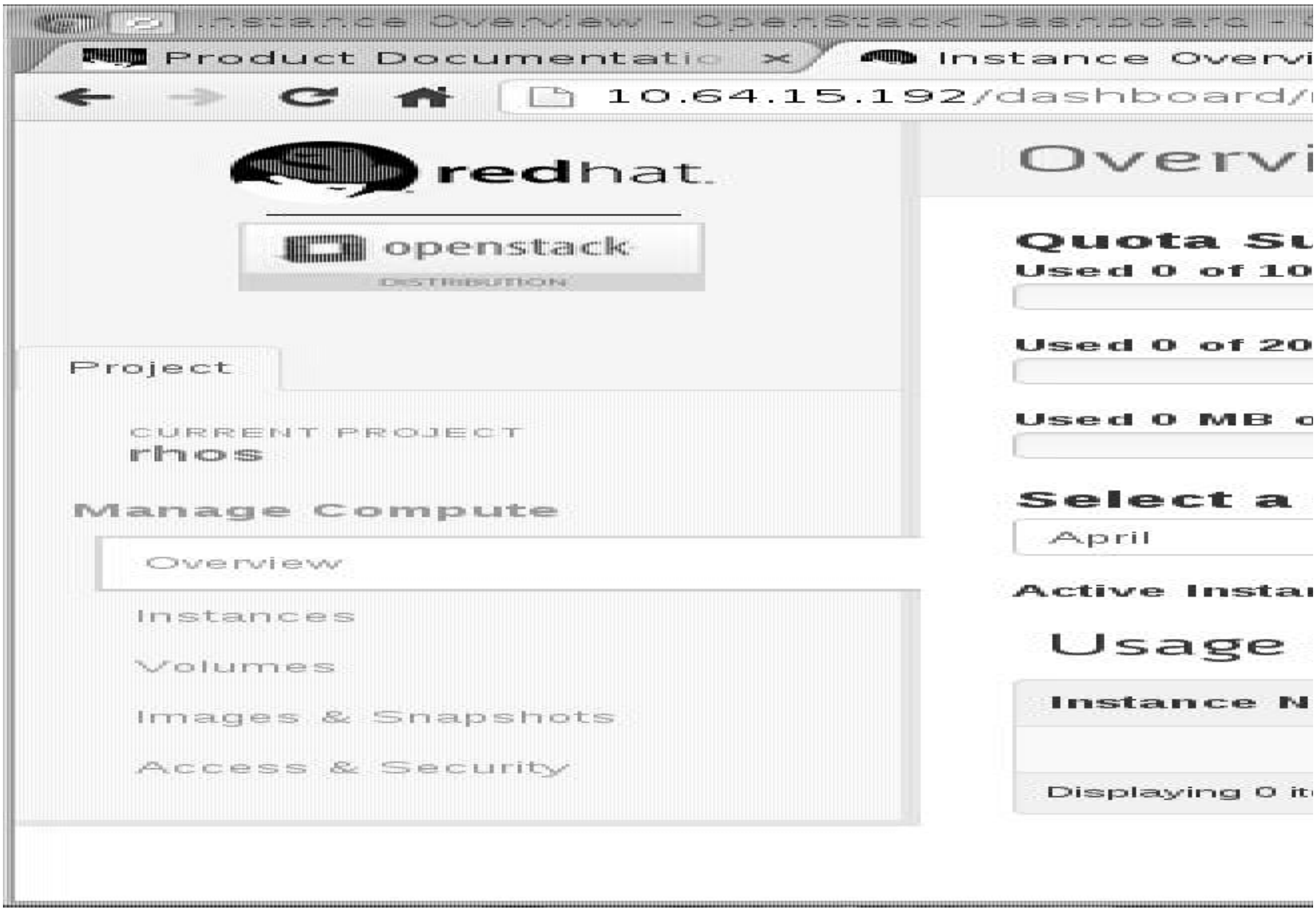


Figure 1.2. Dashboard Service Overview

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

The Dashboard service 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 database, for managing sessions.

1.4.2. Identity Service

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

The Identity service also provides a central catalog of services and endpoints running in a particular OpenStack cloud. This acts as a service directory for other OpenStack systems. Each endpoint is assigned:

- an **adminURL**, the URL for the administrative endpoint for the service. Only the Identity service might use a value here that is different from publicURL; all other services will use the same value.
- an **internalURL**, the URL of an internal-facing endpoint for the service (typically same as the publicURL).
- a **publicURL**, the URL of the public-facing endpoint for the service.
- a **region**, in which the service is located. By default, if a region is not specified, the 'RegionOne' location is used.

The Identity service uses the following concepts:

- Users, with 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 Image service), who belongs to the special tenant 'service'.
- Tenants, generally the user's group, project, or organization.
- Roles that determine a user's permissions.

The Identity service is composed of:

- **keystone** service that provides the administrative and public APIs.
- Databases for each of the internal services.

1.4.3. OpenStack Networking Service

The OpenStack Networking service provides a scalable and API-driven system for managing the network connectivity, addressing, and services within an OpenStack IaaS cloud deployment. 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).

Advantages include:

- 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.

OpenStack Networking is composed of:

- **quantum-server** Python daemon, which manages user requests (and exposes the API).

  The **quantum-server** daemon is configured with a plugin that implements the OpenStack Networking API operations using a specific set of networking mechanisms. A wide choice of plugins are also available. For example, the **openvswitch** and **linuxbridge** plugins utilize native Linux networking mechanisms, while other plugins interface with external devices or SDN controllers.
- **quantum-l3-agent**, an agent providing L3/NAT forwarding.
- **quantum-*-agent**, a plug-in agent that runs on each node to perform local networking configuration for the node's VMs and networking services.
- **quantum-dhcp-agent**, an agent providing DHCP services to tenant networks.
- Database, for persistent storage.

1.4.4. Block Storage Service

The Block Storage (or volume) service provides persistent block storage management for virtual hard drives. The block storage system manages the creation of block devices to servers. Block storage volumes are fully integrated into both the Compute and Dashboard services, which allows cloud users to manage their own storage needs (Compute handles the attaching and detaching of devices). For more information, see Section 1.4.5, “Compute Service”. Both regions and zones (for details, refer to Section 1.4.7, “Object Storage Service”) can be used to handle distributed block storage hosts.

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.

The Block Storage service is composed of the following:

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

1.4.5. Compute Service

The Compute service is the heart of the OpenStack cloud, providing virtual machines on demand. Compute schedules virtual machines to run on a set of nodes. It does this by defining drivers that interact with underlying virtualization mechanisms, and exposing the functionality to the other OpenStack components.

Compute interacts with the Identity service for authentication, Image service for images, and the Dashboard service 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). The Compute service is designed to scale horizontally on standard hardware, and can download images to launch instances as required.

Table 1.2. Ways to Segregate the Cloud

Concept	Description
Regions	Each service cataloged in the Identity service is identified by its region, which typically represents a geographical location, and its endpoint. In a cloud with multiple Compute deployments, regions allow for the discrete separation of services, and are a robust way to share some infrastructure between Compute installations, while allowing for a high degree of failure tolerance.
Cells (Technical Preview)	A cloud's Compute hosts can be partitioned into groups called cells (to handle large deployments or geographically separate installations). Cells are configured in a tree. The top-level cell ('API cell') runs the **`nova-api`** service, but no **`nova-compute`** services. In contrast, each child cell runs all of the other typical **`nova-*`** services found in a regular installation, except for the **`nova-api`** service. Each cell has its own message queue and database service, and also runs **`nova-cells`**, which manages the communication between the API cell and its child cells. This means that: » A single API server can be used to control access to multiple Compute installations. » A second level of scheduling at the cell level is available (versus host scheduling) that provides greater flexibility over the control of where virtual machines are run.
Host Aggregates and Availability Zones	A single Compute deployment can be partitioned into logical groups (for example, into multiple groups of hosts that share common resources like storage and network, or have a special property such as trusted computing hardware). If the user is: » An administrator, the group is presented as a Host Aggregate, which has assigned Compute hosts and associated metadata. An aggregate's metadata is commonly used to provide information for use with **`nova-scheduler`** (for example, limiting specific flavors or images to a subset of hosts). » A user, the group is presented as an Availability Zone. The user cannot view the group's metadata, nor which hosts make up the zone. Aggregates, or zones, can be used to: » Handle load balancing and instance distribution. » Provide some form of physical isolation and redundancy from other zones (such as by using a separate power supply or network equipment). » Identify a set of servers that have some common attribute. » Separate out different classes of hardware.

Important

For more information on the support scope for features marked as technical previews, refer to https://access.redhat.com/support/offerings/techpreview/

Compute is composed of the following:

- **`openstack-nova-api`** service, handles requests and provides access to the Compute services (such as booting an instance).
- **`openstack-nova-cert`** service, provides the certificate manager.
- **`openstack-nova-compute`** service, 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 Compute database.
- **`openstack-nova-conductor`** service provides database-access support for Compute nodes (thereby reducing security risks).
- **`openstack-nova-consoleauth`** service, manages console authorization.
- **`openstack-nova-network`** service, handles Compute 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, provides a VNC proxy for browsers (enabling VNC consoles to access virtual machines started by OpenStack).
- **`openstack-nova-scheduler`** service, dispatches requests for new virtual machines to the correct node.
- Apache Qpid server (**`qpidd`** service), provides the AMPQ message queue. This server (also used by Block Storage) handles the OpenStack transaction management, including queuing, distribution, security, management, clustering, and federation. Messaging becomes especially important when a OpenStack deployment is scaled and its services are running on multiple machines.
- **`libvirtd`** service, enables the creation of virtual machines (it is the driver for the hypervisor).
- KVM Linux hypervisor, creates virtual machines and enables their live migration from node to node.
- Database, for build-time and run-time infrastructure state.

1.4.6. Image Service

The Image 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 Object Storage 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 the Image service; 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)

The Image service is composed of the following:

- **`openstack-glance-api`**, 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`**, manages all metadata associated with each image, and which requires a database.
- Database, for image metadata.

1.4.7. Object Storage Service

The Object Storage service provides object storage in virtual containers, which allows users to store and retrieve files. The service's distributed 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. Object Storage uses the concept of:

- Storage replicas, used to maintain the state of objects in the case of outage. A minimum of three replicas is recommended.
- Storage zones, 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.
- Storage regions, essentially a group of zones sharing a location. Regions can be, for example, groups of servers or server farms, usually located in the same geographical area. Regions have a separate API endpoint per Object Storage service installation, which allows for a discrete separation of services.

The Object Storage service 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, responsible for listings containers, using the account database.
- **`openstack-swift-container`** server, handles listings of objects (what objects are in a specific container) using the container database.
- Ring files that contain details of all the storage devices, and 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.
- Account database.
- Container database.
- Ext4 (recommended) or XFS filesystem for object storage.
- Housekeeping processes, including replication and auditors.

1.4.8. Metering (Technical Preview)

The Metering service provides user-level usage data for OpenStack-based clouds that is used for customer billing, system monitoring, or alerts. Data can be collected by notifications sent by existing OpenStack components (for example, usage events emitted from Compute) or by polling the infrastructure (for example, libvirt).

Metering includes a storage daemon that communicates with authenticated agents via a trusted messaging system, to collect and aggregate data. Additionally, the service uses a plugin system, which makes it easy to add new monitors.

The Metering service is composed of the following:

- **`ceilometer-agent-compute`**, an agent that runs on each Compute node and polls for resource utilization statistics.
- **`ceilometer-agent-central`**, an agent that runs on a central management server to poll for utilization statistics about resources not tied to instances or Compute nodes.
- **`ceilometer-collector`**, an agent that runs on one or more central management servers to monitor the message queues. Notification messages are processed and turned into metering messages, and sent back out on to the message bus using the appropriate topic. Metering messages are written to the data store without modification.
- Mongo database, for collected usage sample data.
- API Server, which runs on one or more central management servers to provide access to the data store's data. Only the Collector and the API server have access to the data store.

1.4.9. Orchestration (Technical Preview)

The Orchestration service provides a template-based orchestration engine for the OpenStack cloud, which can be used to create and manage cloud infrastructure resources such as storage, networking, instances, and applications as a repeatable running environment.

Templates are used to create stacks, which are collections of resources (for example instances, floating IPs, volumes, security groups, or users). The service offers access to all OpenStack core services via a single modular template, with additional orchestration capabilities such as auto-scaling and basic high availability.

Features include:

- A single template provides access to all underlying service APIs.
- Templates are modular (resource orientated).
- Templates can be recursively defined, and therefore reusable (nested stacks). This means that the cloud infrastructure can be defined and reused in a modular way.
- Resource implementation is pluggable, which allows for custom resources.
- Autoscaling functionality (automatically adding or removing resources depending upon usage).
- Basic high availability functionality.

The Orchestration service is composed of the following:

- **`heat`**, a CLI tool that communicates with the heat-api to execute AWS CloudFormation APIs.
- **`heat-api`**, an OpenStack-native REST API that processes API requests by sending them to the heat-engine over RPC.
- **`heat-api-cfn`**, provides an AWS-Query API that is compatible with AWS CloudFormation and processes API requests by sending them to the heat-engine over RPC.
- **`heat-engine`**, orchestrates the launching of templates and provide events back to the API consumer.
- **`heat-api-cloudwatch`**, which provides monitoring (metrics collection) for the Orchestration service.
- **`heat-cfntools`**, a package of helper scripts (for example, cfn-hup, which handles updates to metadata and executes custom hooks).

Note

The **`heat-cfntools`** package is only installed on images that are launched by heat into Compute servers.

Chapter 2. Product Requirements

2.1. Software Requirements

2.1.1. Operating System Requirements

Red Hat Enterprise Linux OpenStack Platform requires Red Hat Enterprise Linux 6.4 Server. All systems in the environment must have Red Hat Enterprise Linux 6.4 Server installed and be subscribed to receive package updates from Red Hat Network or an equivalent source such as a Red Hat Network Satellite server.

Additionally all systems must be subscribed to receive software updates for both Red Hat Enterprise Linux 6.4 Server and Red Hat OpenStack.

- For further information on installing Red Hat Enterprise Linux 6.4 Server refer to the Red Hat Enterprise Linux 6 *Installation Guide*.
- For further information on managing Red Hat subscriptions refer to the Red Hat *Subscription Management Guide*.

Important

> RHN Classic is intended to be used with legacy systems (Red Hat Enterprise Linux 6.0 or Red Hat Enterprise Linux 5.6 and earlier releases). It is strongly recommended that Red Hat Enterprise Linux 6.1/5.7 and later systems use Customer Portal Subscription Management, Subscription Asset Manager, or similar certificate-based subscription management service. As such these instructions are **not** intended for use on systems which have been registered to Red Hat Network using RHN Classic.

2.1.2. Configuring Software Repositories

2.1.2.1. Register to Red Hat Network

Red Hat Enterprise Linux OpenStack Platform requires that each system in the OpenStack environment be running Red Hat Enterprise Linux Server and that all systems be signed up to receive updates from Red Hat Network.

- For further information on installing Red Hat Enterprise Linux Server refer to the Red Hat Enterprise Linux *Installation Guide*.
- For further information on managing Red Hat subscriptions refer to the Red Hat *Subscription Management Guide*.

All steps in this procedure must be executed while logged in to the account of the `root` user on the system being registered.

Important

RHN Classic is intended to be used with legacy systems (Red Hat Enterprise Linux 6.0 or Red Hat Enterprise Linux 5.6 and earlier releases). It is strongly recommended that Red Hat Enterprise Linux 6.1/5.7 and later systems use Customer Portal Subscription Management, Subscription Asset Manager, or similar certificate-based subscription management service. As such these instructions are **not** intended for use on systems which have been registered to Red Hat Network using RHN Classic.

1. Run the **`subscription-manager register`** command to register the system to Red Hat Network.

   ```
   # subscription-manager register
   ```

2. Enter your Red Hat Network user name when prompted.

   ```
   Username: admin@example.com
   ```

   Important

   Your Red Hat Network account must have Red Hat OpenStack entitlements. If your Red Hat Network account does not have Red Hat OpenStack entitlements then you may register for access to the evaluation program at http://www.redhat.com/openstack/.

3. Enter your Red Hat Network password when prompted.

   ```
   Password:
   ```

4. When registration completes successfully system is assigned a unique identifier.

   ```
   The system has been registered with id: IDENTIFIER
   ```

The system has been registered to Red Hat Network and is ready to be attached to specific software subscriptions.

2.1.2.2. Red Hat Enterprise Linux Repository Configuration

Follow the steps in this procedure to register a Red Hat Enterprise Linux system to receive updates from Red Hat Network. These steps must be run while logged in as the **`root`** user. Repeat these steps on each system in the OpenStack environment.

1. Use the **`subscription-manager list --`*`available`*** command to locate the pool identifier of the Red Hat Enterprise Linux subscription.

```
# subscription-manager list --available
+-------------------------------------------+
    Available Subscriptions
+-------------------------------------------+

Product Name:            Red Hat Enterprise Linux Server
Product Id:              69
Pool Id:                 POOLID
Quantity:                1
Service Level:           None
Service Type:            None
Multi-Entitlement:       No
Expires:                 01/01/2022
Machine Type:            physical
...
```

The pool identifier is indicated in the **Pool Id** field associated with the **Red Hat Enterprise Linux Server** product. The identifier will be unique to your subscription. Take note of this identifier as it will be required to perform the next step.

Note

The output displayed in this step has been truncated to conserve space. All other available subscriptions will also be listed in the output of the command.

2. Use the **subscription-manager attach** command to attach the subscription identified in the previous step.

```
# subscription-manager attach --pool=POOLID
Successfully attached a subscription for Red Hat Enterprise Linux Server.
```

Replace ***POOLID*** with the unique identifier associated with your Red Hat Enterprise Linux Server subscription. This is the identifier that was located in the previous step.

3. Run the **yum repolist** command. This command ensures that the repository configuration file **/etc/yum.repos.d/redhat.repo** exists and is up to date.

```
# yum repolist
```

Once repository metadata has been downloaded and examined, the list of repositories enabled will be displayed, along with the number of available packages.

```
repo id                   repo name
status
rhel-6-server-rpms        Red Hat Enterprise Linux 6 Server (RPMs)      8,816
repolist: 8,816
```

Note

The output displayed in this step may differ from that which appears when you run the **yum repolist** command on your system. In particular the number of packages listed will vary if or when additional packages are added to the **rhel-6-server-rpms** repository.

You have successfully configured your system to receive Red Hat Enterprise Linux updates from Red Hat Network.

2.1.2.3. Red Hat OpenStack Repository Configuration

Follow the steps in this procedure to configure a Red Hat Enterprise Linux system to receive Red Hat OpenStack packages and updates from Red Hat Network. Access to a Red Hat software entitlement that includes Red Hat OpenStack is required, such entitlements include:

- Red Hat Cloud Infrastructure
- Red Hat Cloud Infrastructure (without Guest OS)
- Red Hat Enterprise Linux OpenStack Platform
- Red Hat Enterprise Linux OpenStack Platform Preview
- Red Hat Enterprise Linux OpenStack Platform (without Guest OS)

These steps must be run while logged in as the **`root`** user. Repeat these steps on each system in the environment that will be used to host OpenStack services.

Note

Systems that will be used to host supporting software that is included in Red Hat Enterprise Linux such as MySQL and Qpid but will **not** host OpenStack services do not require access to Red Hat OpenStack subscription.

1. Use the **`subscription-manager list`** command to locate the pool identifier of the relevant Red Hat Cloud Infrastructure or Red Hat Enterprise Linux OpenStack Platform entitlement.

   ```
    # subscription-manager list --available
   +-------------------------------------------+
       Available Subscriptions
   +-------------------------------------------+
   ...
   Product Name:          ENTITLEMENT
   Product Id:            ID_1
   Pool Id:               POOLID_1
   Quantity:              3
   Service Level:         None
   Service Type:          None
   Multi-Entitlement:     No
   Expires:               02/14/2013
   Machine Type:          physical

   Product Name:          ENTITLEMENT
   Product Id:            ID_2
   Pool Id:               POOLID_2
   Quantity:              unlimited
   Service Level:         None
   Service Type:          None
   Multi-Entitlement:     No
   Expires:               02/14/2013
   Machine Type:          virtual
   ...
   ```

   Locate the entry in the list where the **`Product Name`** matches the name of the entitlement that will be used to access Red Hat OpenStack packages. Take note of the pool identifier associated with the entitlement, this value is indicated in the **Pool Id** field. The pool identifier is unique to your subscription and will be required to complete the next step.

> **Note**
>
> The output displayed in this step has been truncated to conserve space. All other available subscriptions will also be listed in the output of the command.

2. Use the **subscription-manager attach** command to attach the subscription identified in the previous step.

```
# subscription-manager attach --pool=POOLID
Successfully attached a subscription for ENTITLEMENT.
```

Replace ***POOLID*** with the unique identifier associated with your Red Hat Cloud Infrastructure or Red Hat Enterprise Linux OpenStack Platform entitlement. This is the identifier that was located in the previous step.

3. Install the *yum-utils* package. The *yum-utils* package is provided by the Red Hat Enterprise Linux subscription but provides the **yum-config-manager** utility required to complete configuration of the Red Hat OpenStack software repositories.

```
# yum install -y yum-utils
```

Note that depending on the options selected during Red Hat Enterprise Linux installation the *yum-utils* package may already be installed.

4. Use the **yum-config-manager** command to ensure that the correct software repositories are enabled. Each successful invocation of the command will display the updated repository configuration.

   a. Ensure that the repository for Red Hat OpenStack 1.0 (Essex) has been disabled.

```
# yum-config-manager --disable rhel-server-ost-6-preview-rpms
Loaded plugins: product-id
==== repo: rhel-server-ost-6-preview-rpms ====
[rhel-server-ost-6-preview-rpms]
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/6Server
baseurl =
https://cdn.redhat.com/content/beta/rhel/server/6/6Server/x86_64/opensta
ck/essex/os
cache = 0
cachedir = /var/cache/yum/x86_64/6Server/rhel-server-ost-6-preview-
rpms
cost = 1000
enabled = False
...
```

Note

Yum treats the values **False** and **0** as equivalent. As a result the output on your system may instead contain this string:

```
enabled = 0
```

Note

If you encounter this message in the output from **yum-config-manager** then the system has been registered to Red Hat Network using either RHN Classic or RHN Satellite.

```
This system is receiving updates from RHN Classic or RHN
Satellite.
```

Consult the Red Hat *Subscription Management Guide* for more information on managing subscriptions using RHN Classic or RHN Satellite.

b. Ensure that the repository for Red Hat OpenStack 2.1 (Folsom) is disabled.

```
# yum-config-manager --disable rhel-server-ost-6-folsom-rpms
Loaded plugins: product-id
==== repo: rhel-server-ost-6-folsom-rpms ====
[rhel-server-ost-6-folsom-rpms]
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/6Server
baseurl =
https://cdn.redhat.com/content/beta/rhel/server/6/6Server/x86_64/opensta
ck/folsom/os
cache = 0
cachedir = /var/cache/yum/x86_64/6Server/rhel-server-ost-6-folsom-rpms
cost = 1000
enabled = False
...
```

c. Ensure that the repository for Red Hat Enterprise Linux OpenStack Platform 3 (Grizzly) has been enabled.

```
# yum-config-manager --enable rhel-server-ost-6-3-rpms
Loaded plugins: product-id
==== repo: rhel-server-ost-6-3-rpms ====
[rhel-server-ost-6-3-rpms]
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/6Server
baseurl =
https://cdn.redhat.com/content/dist/rhel/server/6/6Server/x86_64/opensta
ck/3/os
cache = 0
cachedir = /var/cache/yum/x86_64/6Server/rhel-server-ost-6-3-rpms
cost = 1000
enabled = True
...
```

Note

Yum treats the values **True** and **1** as equivalent. As a result the output on your system may instead contain this string:

```
enabled = 1
```

5. Run the **yum repolist** command. This command ensures that the repository configuration file **/etc/yum.repos.d/redhat.repo** exists and is up to date.

```
# yum repolist
```

Once repository metadata has been downloaded and examined, the list of repositories enabled will be displayed, along with the number of available packages.

```
repo id                     repo name                                 status
rhel-6-server-rpms          Red Hat Enterprise Linux 6 Server (RPMs)   8,816
rhel-server-ost-6-3-rpms   Red Hat OpenStack 3 (RPMs)                   138
repolist: 10,058
```

Note

The output displayed in this step may differ from that which appears when you run the **yum repolist** command on your system. In particular the number of packages listed will vary if or when additional packages are added to the repositories.

6. Install the *yum-plugin-priorities* package. The *yum-plugin-priorities* package provides a **yum** plug-in allowing configuration of per-repository priorities.

```
# yum install -y yum-plugin-priorities
```

7. Use the **yum-config-manager** command to set the priority of the Red Hat OpenStack software repository to **1**. This is the highest priority value supported by the *yum-plugin-priorities* plug-in.

```
# yum-config-manager --enable rhel-server-ost-6-3-rpms \
        --setopt="rhel-server-ost-6-3-rpms.priority=1"
Loaded plugins: product-id
==== repo: rhel-server-ost-6-3-rpms ====
[rhel-server-ost-6-3-rpms]
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/6Server
baseurl =
https://cdn.redhat.com/content/dist/rhel/server/6/6Server/x86_64/openstack/3/
os
cache = 0
cachedir = /var/cache/yum/x86_64/6Server/rhel-server-ost-6-3-rpms
cost = 1000
enabled = True
...
priority = 1
...
```

8. Run the **yum *update*** command and reboot to ensure that the most up to date packages, including the kernel, are installed and running.

```
# yum update -y
```

```
# reboot
```

You have successfully configured your system to receive Red Hat OpenStack packages. You may use the **yum repolist** command to confirm the repository configuration again at any time.

2.1.3. Disabling Network Manager

Some installation methods of Red Hat Enterprise Linux, install **NetworkManager** which interfere with OpenStack Networking. If these installation methods are chosen, you must manually disable **NetworkManager**.

If you have chosen any of the following installation methods, no action is required.

- Basic Server
- Database Server
- Web Server
- Identity Management Server
- Virtualization Host
- Minimal Install

If you have chosen any of the following installation methods, then you will need to disable **NetworkManager**.

- Desktop
- Software Development Workstation

Procedure 2.1. Disabling NetworkManager

To verify if you need to disable NetworkManager, run the following command.

```
# chkconfig --list NetworkManager
```

No action is required if the result of is

```
error reading information on service NetworkManager: No such file or directory
```

You must disable the **NetworkManager** if the result is

```
NetworkManager  0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

1. Disable network manager by running the commands.

   ```
   # chkconfig NetworkManager off
   # service NetworkManager stop
   ```

2. Open each interface configuration file on the system in a text editor. Interface configuration files are found in the **/etc/sysconfig/network-scripts/** directory and have names of the form **ifcfg-*X*** where ***X*** is replaced by the name of the interface. Valid interface names include **eth0**, **p1p5**, and **em1**.

   In each file ensure that the **NM_CONTROLLED** configuration key is set to **no** and the **ON_BOOT** configuration key is set to **yes**.

   ```
   NM_CONTROLLED=no
   ONBOOT=yes
   ```

   This action ensures that the standard network service will take control of the interfaces and automatically activate them on boot.

3. Ensure that the network service is started using the **service** command.

   ```
   # service network start
   ```

4. Ensure that the network service is enabled using the **chkconfig** command.

   ```
   # chkconfig network on
   ```

   You have now successfully disabled the **NetworkManager**. The standard network service has been enabled and configured to control the required network interfaces.

2.2. Hardware Requirements

The system requirements for an OpenStack deployment vary based on the scale and workload of the environment being deployed.

This guide provides the recommended minimum system requirements for some common deployment scenarios.

Important

To verify that the processor of a system running Red Hat Enterprise Linux has the required CPU extensions and that they are enabled check the contents of the **`/proc/cpuinfo`** file:

```
# grep -E 'svm|vmx' /proc/cpuinfo | grep nx
```

If any output is shown, the processor is hardware virtualization capable. If no output is shown it is still possible that your processor supports hardware virtualization. In some circumstances manufacturers disable the virtualization extensions in the BIOS. Where you believe this to be the case consult the system's BIOS and the motherboard manual provided by the manufacturer.

2.2.1. Single Node ("All in One") Deployments

In this configuration all services are installed and run on a single system. This simplifies the deployment process and is suitable for evaluation purposes. Such a deployment is not however suitable for use in a production environment.

Processor

64-bit x86 processor with support for the Intel 64 or AMD64 CPU extensions, and the AMD-V or Intel VT hardware virtualization extensions enabled.

Memory

A minimum of 2 GB of RAM is recommended.

Add additional RAM to this requirement based on the amount of memory that you intend to make available to virtual machine instances.

Disk Space

A minimum of 50 GB of available disk space is recommended.

Add additional disk space to this requirement based on the amount of space that you intend to make available to virtual machine instances. This figure varies based on both the size of each disk image you intend to create and whether you intend to share one or more disk images between multiple instances.

1 TB of disk space is recommended for a realistic environment capable of hosting multiple instances of varying sizes.

Network Interface Cards

1 x 1 Gbps Network Interface Card.

2.2.2. Cloud Controller Deployment with One or More Compute Nodes

In this configuration one system acts as the cloud controller by hosting services including the compute database and API server.

Other available systems are used as compute nodes on which virtual machine instances are run. Support services such as image storage are provided on either the cloud controller or one or more of the compute nodes.

Cloud Controller

Processor

64-bit x86 processor with support for the Intel 64 or AMD64 CPU extensions, and the AMD-V or Intel VT hardware virtualization extensions enabled.

Memory

A minimum of 2 GB of RAM is recommended.

Disk Space

A minimum of 50 GB of available disk space is recommended.

Add additional disk space to this requirement based on the amount of space that you intend to make available to virtual machine instances. This figure varies based on both the size of each disk image you intend to create and whether you intend to share one or more disk images between multiple instances.

1 TB of disk space is recommended for a realistic environment capable of hosting multiple instances of varying sizes.

Network Interface Cards

2 x 1 Gbps Network Interface Cards.

Compute Nodes

Processor

64-bit x86 processor with support for the Intel 64 or AMD64 CPU extensions, and the AMD-V or Intel VT hardware virtualization extensions enabled.

Memory

A minimum of 2 GB of RAM is recommended.

Add additional RAM to this requirement based on the amount of memory that you intend to make available to virtual machine instances.

Disk Space

A minimum of 50 GB of available disk space is recommended.

Add additional disk space to this requirement based on the amount of space that you intend to make available to virtual machine instances. This figure varies based on both the size of each disk image you intend to create and whether you intend to share one or more disk images between multiple instances.

1 TB of disk space is recommended for a realistic environment capable of hosting multiple instances of varying sizes.

Network Interface Cards

2 x 1 Gbps Network Interface Cards.

2.2.3. Configuring Storage

PackStack installs Block Storage that uses a volume group. When the Block Storage Service starts, it looks for a specific volume group named **`cinder-volumes`**, the volume that PackStack can create should only be considered an example storage volume for testing. It is placed in **`/var/lib/cinder`** and installed as a loopback storage device on the host that the Block Storage Service is running on. To avoid the creation of loopback devices, you have to create volume groups manually for the Block Storage Service before installing and deploying OpenStack using PackStack.

> **Example 2.1. Creating Volume Groups**
>
> Initialize the volume manager as a physical volume and then create a volume group using the following commands.
>
> ```
> # pvcreate /dev/sdX
> # vgcreate cinder-volumes /dev/sdX
> ```

PackStack does not install a volume for Object Storage, instead it adds a device to a Swift ringfile. On the Swift storage host this is then represented by a directory in **`/srv/`**. Ideally the directory for the Swift device should be a separate filesystem. If you don't have one and just want to test Swift, then PackStack can create a small loopback storage device in place of a separate partition.

You can manually set **`CONFIG_SWIFT_STORAGE_HOSTS=`*`192.0.43.10/sdb1,192.0.43.10/sdc1`***. This would setup Swift with **`/dev/sdb1`**, **`/dev/sdc1`** and no testing loopback device.

Part II. Deploying OpenStack using PackStack

PackStack is a command line utility that uses Puppet (http://www.puppetlabs.com/) modules to support rapid deployment of OpenStack on existing servers over an SSH connection. PackStack is suitable for deploying both single node proof of concept installations and more complex multi-node installations.

Deployment options are provided either interactively, via the command line, or via a text file containing preconfigured answers to the questions PackStack asks.

Chapter 3. Installing PackStack

PackStack is provided by the *openstack-packstack* package. Follow this procedure to install the *openstack-packstack* package.

Procedure 3.1. Installing PackStack

1. Use the **yum** command to install the *openstack-packstack* package.

   ```
   # yum install -y openstack-packstack
   ```

2. Use the **which** command to verify that the PackStack utility is now available.

   ```
   # which packstack
   /usr/bin/packstack
   ```

The *openstack-packstack* package which provides the PackStack utility is now installed. Proceed to Chapter 4, *Running PackStack* for information on prerequisites and running PackStack for the first time.

Chapter 4. Running PackStack

PackStack supports a variety of different deployment modes:

Quick Start

When run with the **`--allinone`** or **`--install-hosts`** arguments, PackStack performs a single node or multiple node deployment respectively. These deployments are performed using default configuration values and are recommended for initial testing of Red Hat Enterprise Linux OpenStack Platform. Users requiring more customized deployments should consider the other deployment modes.

Refer to Section 4.1, “Quick Start Deployment using PackStack” for more information on running PackStack using the **`--allinone`** or **`--install-hosts`** options.

Interactively

When run interactively, PackStack provides prompts for entry of each configuration value required to complete deployment. Alternatively you may accept the provided default value.

Refer to Section 4.2, “Running PackStack Interactively” for more information on running PackStack interactively.

Non-interactively

When run non-interactively, PackStack expects an "answer" file to be provided as a command line option. This file contains the desired settings for all configuration values that are required to complete deployment.

Refer to Section 4.3, “Running PackStack Non-interactively” for more information on generating an answer file and using it to run PackStack non-interactively.

Important

To deploy OpenStack using PackStack each machine targeted for deployment must be configured to allow access using the account of the **`root`** user over SSH on port **`22`**.

Important

By default PackStack will configure a volume group named **`cinder-volumes`** on the system targeted for volume storage (Cinder) deployment if one does not already exist. This volume group will be backed by a loopback device and is not appropriate for production use.
If you intend to use physical storage for the **`cinder-volumes`** volume group then you must create the volume group in advance on the system to be used for Cinder.

Important

It is strongly recommended that each compute node has two network interfaces available. One for access to the public network and one for the internal Nova network. While it is possible to use a single interface for both purposes, this approach may result in virtual machine instances obtaining addresses from the wrong DHCP server.

4.1. Quick Start Deployment using PackStack

The quickest way to deploy an OpenStack environment using PackStack is to provide a host, or list of hosts, on the command line. The first host listed will be deployed as a compute controller node, subsequent hosts will be deployed as compute nodes.

When using this deployment method PackStack will use default values for all other deployment options unless they are overridden on the command line.

You can disable Quantum networking if you choose, see Procedure 4.1, “Quick Start Deployment using PackStack”

For a list of available command line options refer to Table 4.1, “PackStack Configuration Keys”.

Procedure 4.1. Quick Start Deployment using PackStack

1. A. **Single Node Deployment**

   Run PackStack with the ***--allinone*** parameter to perform an "all in one" deployment on the local host. You will be prompted to enter the password of the **root** user to facilitate SSH key installation.

   > **Example 4.1. Single Node Deployment using Quantum networking (default)**
   >
   > In this example PackStack is instructed to deploy an "all in one" installation to the local system.
   >
   > Quantum networking is enabled by default. A **demo** Keystone tenant is also created along with a **keystonerc_demo** file, which can be sourced like the existing **keystonerc_admin**. Hence, the ***--allinone*** option on its own automatically enables the keys **CONFIG_PROVISION_DEMO** and **CONFIG_PROVISION_ALL_IN_ONE_OVS_BRIDGE** in PackStack's answer file. This answer file will have a file name similar to **/root/packstack-answers-*20130306-051043*.txt**.
   >
   > When you run the Dashboard, you should log into Horizon using the **demo** account instead of the **admin** account due to the ownership of the private and public networks. The demo password is stored as **CONFIG_KEYSTONE_DEMO_PW** in PackStack's answer file.
   >
   > ```
   > # packstack --allinone
   > ```

   > **Example 4.2. Single Node Deployment without Quantum networking**
   >
   > In this example PackStack is instructed to deploy an "all in one" installation to the local system, but using only Nova networking.
   >
   > ```
   > # packstack --allinone --os-quantum-install=n
   > ```

B. **Multiple Node Deployment**

   Run PackStack with the ***--install-hosts*** parameter. The parameter expects a comma separated list of IP addresses. You will be prompted to enter the password of the **root** user of each system to facilitate SSH key installation.

   ```
   # packstack --install-hosts=CONTROLLER_ADDRESS,NODE_ADDRESSES
   ```

   Replace ***CONTROLLER_ADDRESS*** with the IP address of the system that you intend to use as a compute controller. Replace ***NODE_ADDRESSES*** with IP addresses of the systems that you intend to use as compute nodes.

   > **Example 4.3. Multiple Node Deployment**
   >
   > In this example PackStack is instructed to deploy a controller node on the system with IP address **192.168.43.10**.
   >
   > Additional compute nodes are deployed on the systems with IP addresses **192.168.43.11** and **192.168.43.12**.
   >
   > ```
   > # packstack --install-
   > hosts=192.168.43.10,192.168.43.11,192.168.43.12
   > ```

2. PackStack will prompt you to enter the password of the **root** user for each system in the deployment. This is required to connect to the system and install Puppet which is the tool used to facilitate the rest of the deployment.

   ```
   root@192.168.43.10's password:
   ```

3. The Puppet manifests used to deploy each component will be run on each of the target systems. The amount of time this takes to complete varies based on the hardware and existing workload of each system. It can be significant.

   When the deployment has successfully completed this message is displayed:

   ```
   **** Installation completed successfully ******
   ```

You have successfully deployed an OpenStack environment using PackStack. Please note that:

- An answer file containing all chosen configuration options is saved to disk on the system from which you ran PackStack. This file can be used to automate future deployments.

  ```
   * A new answerfile was created in: /root/packstack-answers-20130306-051043.txt
  ```

- A file containing the authentication details of the OpenStack **admin** user is saved to disk on the system to which the OpenStack client tools were deployed. You will need these details to manage the OpenStack environment.

  ```
   * To use the command line tools you need to source the file
  /root/keystonerc_admin created on 192.168.43.10
  ```

Refer to Part III, “Using OpenStack” to begin using your OpenStack environment.

4.2. Running PackStack Interactively

OpenStack can be deployed by running PackStack interactively. PackStack supports the creation of both single node and multiple node OpenStack deployments.

Note

The procedure below lists all the questions that PackStack prompts you to answer. Based on the choices you make, some of these options might be skipped during the setup.

Procedure 4.2. Running PackStack Interactively

1. **Running PackStack**

   Run the **`packstack`** command to commence the deployment process. Optionally append the **`--`*debug*** parameter to enable additional logging.

   ```
   # packstack
   ```

   Important

   You are not required to log in as the **`root`** user to run the **`packstack`** command itself. However you will be required to provide **`root`** credentials for each machine to which you choose to deploy services.

2. **Configuring the Public Key**

   Each server involved in the OpenStack deployment is configured for key-based authentication. If you already have a public key that you wish to use for this, enter the path to it. If you do not, then press **`Enter`** and the utility will generate one for you and save it to **`~/.ssh/id_rsa.pub`**.

   ```
   Enter the path to your ssh Public key to install on servers:
   ```

3. **Selecting the Services to Install**

   The PackStack script will prompt you to select the OpenStack services that you want to install and configure. At each prompt enter **`y`** to install the service, enter **`n`** to skip the service, or press **`Enter`** to select the default option listed in square brackets (**`[`**, **`]`**).

   ```
   Should Packstack install Glance image service [y|n] [y] :
   Should Packstack install Cinder volume service [y|n] [y] :
   Should Packstack install Nova compute service [y|n] [y] :
   Should Packstack install Quantum compute service [y|n] [y] :
   Should Packstack install Horizon dashboard [y|n] [y] :
   Should Packstack install Swift object storage [y|n] [n] :
   ```

   Each selected service can be deployed on either a local or remote system. Where each service deploys to will be determined based on the IP addresses you provide later in the deployment process.

4. OpenStack includes a number of client tools. Enter **`y`** to install the client tools. A file containing the authentication values of the administrative user will also be created.

   ```
   Should Packstack install OpenStack client tools [y|n] [y] :
   ```

5. Optionally, the PackStack script will configure all servers in the deployment to retrieve date and time information using Network Time Protocol (NTP). To use this facility enter a comma separated

pool of NTP servers.

```
Enter a comma separated list of NTP server(s). Leave plain if Packstack should
not install ntpd on instances.:
```

Example 4.4. Using the Default Red Hat Enterprise Linux NTP Servers

```
Enter list of NTP server(s). Leave plain if packstack should not install
ntpd on instances.: 0.rhel.pool.ntp.org, 1.rhel.pool.ntp.org
```

6. Optionally, the PackStack script will install and configure Nagios to provide advanced facilities for monitoring the nodes in the OpenStack environment.

```
Should Packstack install Nagios to monitor openstack hosts [y|n]  [n] :
```

7. **Configuring the MySQL Instance**

   OpenStack services require a MySQL database to store data in. To configure the database:

   a. Enter the IP address of the server to deploy the MySQL database server on.

   ```
   Enter the IP address of the MySQL server [192.0.43.10] :
   ```

   b. Enter the password to use for the MySQL administrative user. If you do not enter a value it will be randomly generated. The generated password will be available both in the **`~/.my.cnf`** file of the current user and in the answer file.

   ```
   Enter the password for the MySQL admin user :
   ```

8. **Configuring Qpid**

   OpenStack services use the Qpid (http://qpid.apache.org/) messaging system to communicate. Enter the IP address of the server to deploy Qpid on.

```
Enter the IP address of the QPID service  [192.0.43.10] :
```

9. **Configuring the Identity service**

   OpenStack uses the Identity service (**openstack-keystone**) for identity, token, catalog, and policy services. If Identity service installation was selected then enter the IP address of the server to deploy Identity on when prompted.

```
Enter the IP address of the Keystone server  [192.0.43.10] :
```

10. **Configuring the Image service**

    OpenStack uses the Image service (**openstack-glance-***) to store, discover, and retrieve virtual machine images. If Image service installation was selected then enter the IP address of the server to deploy Image service on when prompted.

```
Enter the IP address of the Glance server  [192.0.43.10] :
```

11. **Configuring the Volume service**

    OpenStack uses the Volume service (**openstack-cinder-***) to provide volume storage services. Enter the IP address of the server to deploy the Volume service on. If installation of the volume services was selected then these additional configuration prompts will be presented.

```
Enter the IP address of the Cinder server  [192.0.43.10] :
```

a. PackStack expects storage for use with Volume to be available on a volume group named **cinder-volumes**. If this volume group does not already exist then you will be asked if you want it to be created automatically.

   Answering yes means that PackStack will create a raw disk image in the **/var/lib/cinder** and mount it for use by Volume using a loopback device.

   ```
   Should Cinder's volumes group be created (for proof-of-concept
   installation)? [y|n] [y]:
   ```

b. If you elected to have PackStack create the **cinder-volumes** volume group for you then you will be prompted to enter the size of it in gigabytes (GB).

   ```
   Enter Cinder's volume group size  [20G] :
   ```

   Important

   The amount of space selected must be available on the device used for **/var/lib/cinder**.
   Remember that the size of the Volume service's volume group will restrict the amount of disk space that you can expose to compute instances.

12. **Configuring the Compute service**

    Compute is made up of a number of complementary services that must be deployed. If installation of the compute services was selected then these additional configuration prompts will be presented.

    a. The Compute API service (**openstack-nova-api**) provides web service endpoints for authenticating and interacting with the OpenStack environment over HTTP or HTTPS. Enter the IP address of the server to deploy the Compute API service on.

    ```
    Enter the IP address of the Nova API service  [192.0.43.10] :
    ```

    b. Compute includes a certificate management service (**openstack-nova-cert**). Enter the IP address of the server to deploy the Compute certificate management service on.

    ```
    Enter the IP address of the Nova Cert service  [192.0.43.10] :
    ```

    c. The Compute VNC proxy provides facilities for connecting users of the Compute service to their instances running in the OpenStack cloud. Enter the IP address for the server to deploy the Compute VNC proxy on.

    ```
    Enter the IP address of the Nova VNC proxy  [192.0.43.10] :
    ```

    d. The PackStack script is able to deploy one or more compute nodes. Enter a comma separated list containing the IP addresses or hostnames of all of the nodes that you wish to deploy compute services on.

    ```
    Enter a comma separated list of IP addresses on which to install the
    Nova Compute services  [192.0.43.10] :
    ```

e. A private interface must be configured to provide DHCP services on the Compute nodes. Enter the name of the private interface to use.

```
Enter the Private interface for Flat DHCP on the Nova compute servers
[eth1] :
```

f. The Compute network service (**openstack-nova-network**) provides network services for compute instances. Enter the IP address of the server to deploy the Compute Network service on.

```
Enter the IP address of the Nova Network service  [192.0.43.10] :
```

Important

The Compute networking service is incompatible with the OpenStack Network Service added since the Folsom release.

g. The Conductor service (**openstack-nova-conductor**) provides database query support to the compute service. Enter the IP address of the server to deploy the Conductor service on.

```
Enter the IP address of the Nova Conductor service  [192.0.43.10]:
```

h. A public interface must be configured to allow connections from other nodes and clients. Enter the name of the public interface to use.

```
Enter the Public interface on the Nova network server  [eth0] :
```

i. A private interface must be configured to provide DHCP services on the Nova network server. Enter the name of the private interface to use.

```
Enter the Private interface for Flat DHCP on the Nova network server
[eth1] :
```

j. All compute instances are automatically assigned a private IP address. Enter the range from which these private IP addresses must be assigned.

```
Enter the IP Range for Flat DHCP [192.168.32.0/22] :
```

k. Compute instances can optionally be assigned publicly accessible *floating* IP addresses. Enter the range from which floating IP addresses will be assigned.

```
Enter the IP Range for Floating IP's [10.3.4.0/22] :
```

l. The default floating pool needs to be named. Enter the name for the default floating pool

```
What should the default floating pool be called?  [nova] :
```

m. All compute instances are assigned a floating point IP. Enter **y** to automatically assign floating point IP address.

```
Should new instances automatically have a floating IP assigned? [y|n]
[n] :
```

n. The Compute scheduler (**openstack-nova-scheduler**) is used to map compute requests to compute resources. Enter the IP address of the server to deploy the Compute scheduler on.

```
Enter the IP address of the Nova Scheduler service  [192.0.43.10] :
```

o. In the default configuration, compute allows for overcommitment of physical CPU and memory resources. This means that more of these resources are made available for running instances than actually physically exist on the compute node.

The amount of overcommitment that is permitted is configurable.

a. The default level of CPU overcommitment allows 16 virtual CPUs to be allocated for each physical CPU socket or core that exists on the physical compute node. Press **Enter** to accept the default or enter a different value if desired.

```
Enter the CPU overcommitment ratio. Set to 1.0 to disable CPU
overcommitment  [16.0] :
```

b. The default level of memory overcommitment allows up to 50% more virtual memory to be allocated than exists on the physical compute node. Press **Enter** to accept the default or enter a different value if desired.

```
Enter the RAM overcommitment ratio. Set to 1.0 to disable RAM
overcommitment  [1.5] :
```

13. **Configuring OpenStack Networking**

OpenStack Networking service provides a scalable and API-driven system for managing the network connectivity, addressing, and services within an OpenStack IaaS cloud deployment.

a. Enter the IP address of the OpenStack Networking Server.

```
Enter the IP address of the Quantum server  [192.0.43.10] :
```

b. OpenStack Networking uses namespaces.

The OpenStack Networking namespaces virtualize access to network resources, giving each group of processes the network access it requires. The groups of processes are referred to as containers. Red Hat Enterprise Linux OpenStack Platform includes a custom Red Hat Enterprise Linux kernel that supports the use of network namespaces.

> **Important**
>
> This kernel must be installed on all OpenStack nodes. Additionally, the Open vSwitch plug-in will not work with kernels with versions lower than **2.6.32-343.el6.x86_64**.

Enter **y** to select the use of namespaces.

```
Should Quantum use network namespaces? [y|n]  [y] :
```

c. OpenStack Networking sets up the Quantum L3 agent.

The L3 agent acts as an abstract L3 router that can connect to and provide gateway services for multiple L2 networks. Usually the L3 agent will run on the network node. If there is no network node it should run on the controller node. The nodes on which the L3 agent will be hosted must have a range of IP addresses from the external network that are available for use by OpenStack Networking. These IP addresses will be assigned to the routers that provide the link between the internal and external networks.

Enter the IP addresses on which the Quantum L3 Agent should be set up.

> **Note**
>
> The range selected must be large enough to provide a unique IP address for each router in the deployment as well as each desired floating IP.

```
Enter a comma separated list of IP addresses on which to install the
Quantum L3 agent  [192.0.43.10]
```

d. In order to have OpenStack Networking set up a bridge for external traffic, you need to specify a name for this bridge. The Quantum L3 agent will use this bridge for external traffic, giving the node it is running on access to, for example, the internet. There is no specific naming convention but it is recommended to give the bridge a meaningful name. If you do not enter a name, the external bridge will by default be named **br-ex**. If you intend to use a provider network to handle external traffic, enter the special value **provider**.

```
Enter the name of the bridge that the Quantum L3 agent will use for
external traffic  [br-ex]
```

e. OpenStack Networking sets up the Quantum DHCP agent.

   This agent is capable of allocating IP addresses to virtual machines running on the network. The DHCP agent runs on the network node. If there is no network node the DHCP agent should run on the controller node. Enter the list of IP addresses on which you want Quantum DHCP set up.

```
Enter a comma separated list of IP addresses on which to install Quantum
DHCP agent  [192.0.43.10] :
```

f. Enter the name of the L2 plugin to be used with OpenStack Networking. Valid options are:

   - **linuxbridge**: Choose this option if you need a simple bridge and do not require support for VLANs or GRE.
   - **openvswitch**: Choose this option if you wish to have configurable ports on a managed switch or will require VLAN or GRE support.

```
Enter the name of the L2 plugin to be used with Quantum
[linuxbridge|openvswitch]  [openvswitch] :
```

g. The OpenStack Compute service (Nova) allows VMs to query metadata associated with a VM by making a web request to a special IP address. OpenStack Networking supports proxying those requests to **nova-api**, even when the requests are made from isolated networks, or from multiple networks that use overlapping IP addresses. In order to use this functionality, OpenStack Networking must install the metadata agent. Enter the IP addresses on which the metadata agent should be set up.

```
Enter a comma separated list of IP addresses on which to install the
Quantum metadata agent  [192.0.43.10] :
```

h. OpenStack Networking allocates tenant networks. Enter the type of network to allocate to the tenant networks.

   The use of **`local`** tenant networks is recommended for all-in-one deployments. The use of **`vlan`** tenant networks is recommended for multi-node deployments. The Open vSwitch Quantum plugin supports GRE tunneling, and you can select **`gre`** as long as the installed kernel (version 3.11 or later) and Open vSwitch userspace support GRE tunneling too.

```
Enter the type of network to allocate for tenant networks
[local|vlan|gre]  [local] :
```

i. Enter a list of VLAN ranges for use with the selected plug-in.

   Each tuple in the list is expected to be in the format ***PHYSICAL***:***START***:***END***. Note that ***PHYSICAL*** is just a user-provided label for a network name, not necessarily a physical device. Replace ***PHYSICAL*** with the name of a network, replace ***START*** with the start of the VLAN range to identify with it, and replace ***END*** with the end of the VLAN range to associate with it.

   For example, with a network called "physnet1" that has a VLAN range from 1 to 1000, you would specify "physnet1:1:1000".

```
Enter a comma separated list of VLAN ranges for the Quantum openvswitch
plugin:
```

j. Enter a list of bridge mappings for the OpenStack Networking Open vSwitch plugin.

   Each tuple in the list is expected to be in the format ***PHYSICAL***:***BRIDGE***. Replace ***PHYSICAL*** with the name of a network, and replace ***BRIDGE*** with the name of the Open vSwitch bridge that will be used to connect to the network.

   Continuing the example above, with physnet1 using the interface called "br-eth1", you could use the default option so physnet1 consists of VLANs 1 to 1000 on bridge br-eth1.

```
Enter a comma separated list of bridge mappings for the Quantum
openvswitch plugin  [physnet1:br-eth1] :
```

14. **Configuring Client Tools**

    If installation of the client tools was selected then enter the IP address of the server to install the client tools on when prompted.

```
Enter the IP address of the client server  [192.0.43.10] :
```

    An "rc" file containing administrative credentials will also be created on this host.

15. **Configuring the Dashboard**

    OpenStack uses the dashboard service (**openstack-dashboard**) to provide a web-based user interface or dashboard for accessing OpenStack services including Volume, Compute, Object Storage, and Identity. If installation of the Dashboard was selected then these additional configuration values will be requested.

    a. Enter the IP address of the server to deploy Dashboard on.

```
Enter the IP address of the Horizon server  [192.0.43.10] :
```

b. To enable HTTPS communication with the dashboard enter **y** when prompted. Enabling this option ensures that your access to the dashboard is encrypted.

```
Would you like to set up Horizon communication over https [y|n]  [n] :
```

16. **Configuring Object Storage**

    If installation of Object Storage was selected then these additional configuration values will be requested.

    a. Enter the IP address of the server that is to act as the Object Storage proxy. This server will act as the public link between clients and the Object Storage.

    ```
    Enter the IP address of the Swift proxy service  [192.0.43.10] :
    ```

    b. Enter a comma separated list of devices that Object Storage will use to store objects. Each entry must be specified in ***HOST/DEVICE*** format where ***HOST*** is replaced by the IP address of the host the device is attached to, and ***DEVICE*** is replaced by the path to the device.

    ```
    Enter the Swift Storage servers e.g. host/dev,host/dev  [192.0.43.10] :
    ```

    c. Object Storage uses zones to ensure that each replica of a given object is 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.

    When prompted enter the number of storage zones that must be defined. Note that the number provided must not be bigger than the number of individual devices specified.

    ```
    Enter the number of swift storage zones, MUST be no bigger than the
    number of storage devices configured  [1] :
    ```

    d. Object Storage relies on replication to maintain the state of objects even in the event of a storage outage in one or more of the configured storage zones. Enter the number of replicas that Object Storage must keep of each object when prompted.

    A minimum of three (3) replicas is recommended to ensure a reasonable degree of fault tolerance in the object store. Note however that the number of replicas specified must not be greater than the number of storage zones as this would result in one or more of the zones containing multiple replicas of the same object.

    ```
    Enter the number of swift storage replicas, MUST be no bigger than the
    number of storage zones configured  [1] :
    ```

    e. Currently PackStack supports the use of either Ext4 or XFS filesystems for object storage. The default and recommended choice is Ext4. Enter the desired value when prompted.

    ```
    Enter FileSystem type for storage nodes [xfs|ext4]  [ext4] :
    ```

17. **Configuring EPEL**

    PackStack allows you to subscribe each server to Extra Packages for Enterprise Linux (EPEL).

    ```
    To subscribe each server to EPEL enter "y" [y|n]  [n] :
    ```

18. **Configuring Software Sources**

    PackStack allows you to configure the target servers to retrieve software packages from a number of sources.

a. **Enabling Custom Software Repositories**

PackStack allows you to optionally configure each server to retrieve updates from additional custom software repositories. Enter the URL for the directory containing the **`repodata`** folder of each desired repository at the prompt, separated by a comma.

```
Enter a comma separated list of URLs to any additional yum repositories
to install:
```

b. **Enabling Red Hat Network Subscription**

Enter your Red Hat Network account details when prompted. This will ensure each server involved in the deployment is subscribed to receive updates from Red Hat Network.

```
To subscribe each server to Red Hat enter a username here:
```

```
To subscribe each server to Red Hat enter your password here:
```

Important

PackStack registers systems to Red Hat Network using Subscription Manager or Red Hat Network Satellite. You may encounter problems if your systems have already been registered and subscribed to the Red Hat OpenStack channels using RHN Classic.

c. **Enabling the Red Hat Enterprise Linux Beta Channel**

To enable the Red Hat Enterprise Linux Beta channel enter **y** when prompted. Note that selecting this option is not recommended at this time but may be required by future Red Hat Enterprise Linux OpenStack Platform preview releases.

```
To subscribe each server to Red Hat Enterprise Linux 6 Server Beta
channel (only needed for Preview versions of RHOS) enter y [y|n]  [n] :
```

d. **Enabling Red Hat Network Satellite**

PackStack allows you to optionally configure each server to retrieve updates from a Red Hat Network Satellite server.

Enter the URL of the Red Hat Network Satellite server that you wish to use when prompted. If you do not wish to use a Red Hat Satellite server then do not enter a value.

```
To subscribe each server with RHN Satellite enter RHN Satellite server
URL :
```

If an RHN Satellite URL is provided a number of follow up prompts will be displayed.

   a. Red Hat Network Satellite supports authentication using a user name and password or an activation key. If your Satellite administrator provided you with a user name and password enter them when prompted. If your Satellite administrator provided you with an access key then do not enter a value.

   ```
   Enter RHN Satellite username or leave plain if you will use
   activation key instead :
   ```

```
Enter RHN Satellite password or leave plain if you will use
activation key instead :
```

b. If your Satellite administrator provided you with an access key then enter it when prompted. Otherwise do not enter a value.

```
Enter RHN Satellite activation key or leave plain if you used
username/password instead :
```

c. Specify the path to the certificate of the certificate authority that is used to verify that the connection with the Satellite server is secure.

```
Specify a path or URL to a SSL CA certificate to use :
```

d. Specify the profile name that must be used to identify the system in Red Hat Network. This is optional.

```
If required specify the profile name that should be used as an
identifier for the system in RHN Satellite :
```

e. Specify the HTTP proxy that must be used when connecting to the Satellite server. If no proxy is required then do not enter a value.

```
Specify a HTTP proxy to use with RHN Satellite :
```

f. Specify the user name for authenticating with the HTTP proxy that must be used when connecting to the Satellite server. If no proxy is required or the chosen proxy does not require authentication then do not enter a value.

```
Specify a username to use with an authenticated HTTP proxy :
```

g. Specify the password for authenticating with the HTTP proxy server that must be used when connecting to the Satellite server. If no proxy is required or the chosen proxy does not require authentication then do not enter a value.

```
Specify a password to use with an authenticated HTTP proxy. :
```

h. Specify any additional Satellite flags that you need to be passed to the **rhnreg_ks** command when it is run on each system. This configuration key accepts a comma separated list of flags. Valid flags are **novirtinfo**, **norhnsd**, and **nopackages**.

Refer to the Red Hat Satellite documentation for more information. If unsure do not enter a value.

```
Enter comma separated list of flags passed to rhnreg_ks :
```

19. **Verify Parameters and Confirm**

At this point you will be asked to confirm the deployment details that you provided. Type **yes** and press **Enter** to continue with the deployment.

Depending on the options you chose, the following screen's content will vary.

```
Installer will be installed using the following configuration:
=================================================================
ssh-public-key:                 /root/.ssh/id_rsa.pub
os-glance-install:              y
os-cinder-install:              y
os-nova-install:                y
os-horizon-install:             y
os-swift-install:               n
os-client-install:              y
ntp-servers:
mysql-host:                     192.0.43.10
mysql-pw:                       ********
qpid-host:                      192.0.43.10
keystone-host:                  192.0.43.10
glance-host:                    192.0.43.10
cinder-host:                    192.0.43.10
cinder-volumes-create:          y
cinder-volumes-size:            20G
novaapi-host:                   192.0.43.10
novacert-host:                  192.0.43.10
novavncproxy-hosts:             192.0.43.10
novacompute-hosts:              192.0.43.10
novacompute-privif:             eth1
novanetwork-host:               192.0.43.10
novanetwork-pubif:              eth0
novanetwork-privif:             eth1
novanetwork-fixed-range:        192.168.32.0/22
novanetwork-floating-range:    10.3.4.0/22
novasched-host:                 192.0.43.10
novasched-cpu-allocation-ratio:16.0
novasched-ram-allocation-ratio:1.5
quantum-server-host:            192.0.43.10
quantum-use-namespaces:         y
quantum-l3-hosts:               192.0.43.10
quantum-l3-ext-bridge:          br-ex
quantum-dhcp-hosts:             192.0.43.10
quantum-l2-plugin:              openvswitch
quantum-metadata-hosts:        192.0.43.10
quantum-ovs-tenant-network-type:local
quantum-ovs-vlan-ranges:
quantum-ovs-bridge-mappings:    physnet1:1000:2000
osclient-host:                  192.0.43.10
os-horizon-host:                192.0.43.10
os-horizon-ssl:                 n
os-swift-proxy:                 192.0.43.10
os-swift-storage:               192.0.43.10
os-swift-storage-zones:         1
os-swift-storage-replicas:      1
os-swift-storage-fstype:        ext4
additional-repo:
rh-username:                    admin@example.com
rh-password:                    ********
rhn-satellite-server:
Proceed with the configuration listed above? (yes|no): yes
```

Important

At this stage, if you need to change any of the parameter values there are two ways to do so.

- Choose **`no`**, the installation then starts from **`Step 1`** prompting you to enter values from the beginning, but this time the default values displayed are the ones you had previously entered. You can now change the values of the parameters and complete the installation by choosing **`yes`** for **`Step 19`**.
- Choose **`yes`**, the installation begins and an answer file is created. But this could result in error if there is an issue with the parameters. You can then modify the parameters in the answer file (packstack-answers-xxxx.txt) and re-run with the following command.

```
# packstack --answer-file=packstack-answers-xxxx.txt
```

20. At this point PackStack will commence deployment. Note that when PackStack is setting up SSH keys it will prompt you to enter the **`root`** password to connect to machines that are not already configured to use key authentication.

    Depending on the options you chose, the following screen's content will vary.

```
Installing:
Clean Up...                                               [ DONE ]
Running Pre install scripts...                           [ DONE ]
Setting Up ssh keys...                                    [ DONE ]
Create MySQL manifest...                                  [ DONE ]
Creating QPID manifest...                                 [ DONE ]
Creating Keystone manifest...                            [ DONE ]
Adding Glance Keystone manifest entries...               [ DONE ]
Creating Galnce manifest...                               [ DONE ]
Adding Cinder Keystone manifest entries...               [ DONE ]
Checking if the Cinder server has a cinder-volumes vg... [ DONE ]
Creating Cinder manifest...                               [ DONE ]
Adding Nova API manifest entries...                      [ DONE ]
Adding Nova Keystone manifest entries...                 [ DONE ]
Adding Nova Cert manifest entries...                     [ DONE ]
Adding Nova Conductor manifest entries...                [ DONE ]
Adding Nova Compute manifest entries...                  [ DONE ]
Adding Nova Network manifest entries...                  [ DONE ]
Adding Nova Scheduler manifest entries...                [ DONE ]
Adding Nova VNC Proxy manifest entries...                 [ DONE ]
Adding Nova Common manifest entries...                    [ DONE ]
Adding Openstack Network-related Nova manifest entries...[ DONE ]
Adding Quantum API manifest entries...                   [ DONE ]
Adding Quantum Keystone manifest entries...              [ DONE ]
Adding Quantum L3 manifest entries...                    [ DONE ]
Adding Quantum L2 Agent manifest entries...              [ DONE ]
Adding Quantum DHCP Agent manifest entries...            [ DONE ]
Adding Quantum Metadata Agent manifest entries...       [ DONE ]
Adding OpenStack Client manifest entries...              [ DONE ]
Adding Horizon manifest entries...                        [ DONE ]
Preparing Servers...                                      [ DONE ]
Adding post install manifest enries...                   [ DONE ]
Installing Dependencies...                               [ DONE ]
Copying Puppet modules and manifests...                  [ DONE ]
Applying Puppet manifests...
```

21. Applying the Puppet manifests to all machines involved in the deployment takes a significant amount of time. PackStack provides continuous updates indicating which manifests are being deployed as it progresses through the deployment process. Once the process completes a confirmation message similar to the one shown below will be displayed.

    Depending on the options you chose, the following screen's content will vary.

    ```
     **** Installation completed successfully ******

         (Please allow Installer a few moments to start up.....)


    Additional information:
     * A new answerfile was created in: /root/packstack-answers-20130613-
    133303.txt
     * Time synchronization installation was skipped. Please note that
    unsynchronized time on server instances might be problem for some OpenStack
    components.
     * To use the command line tools you need to source the file
    /root/keystonerc_admin created on 192.0.43.10
     * To use the console, browse to http://192.0.43.10/dashboard
     * To use Nagios, browse to http://192.0.43.10/nagios username : nagiosadmin,
    password: abcdefgh12345678
     * Kernel package with netns support has been installed on host 192.0.43.10.
    Because of the kernel update host mentioned above requires reboot.
     * The installation log file is available at: /var/tmp/packstack/20130613-
    133302-5UY8KB/openstack-setup.log
    You have mail in /var/spool/mail/root
    ```

22. Reboot all the nodes in the environment to ensure that the kernel change takes effect.

    PackStack deploys a new kernel with network namespaces enabled for all the nodes. You must reboot the environment to ensure that the change takes effect.

    ```
    # reboot
    ```

You have successfully deployed OpenStack using PackStack.

The configuration details that you provided are also recorded in an answer file that can be used to recreate the deployment in future. The answer file is stored in **`~/answers.txt`** by default.

Refer to Section 4.3, “Running PackStack Non-interactively” for more information on using answer files to automate deployment.

> **Warning**
>
> The answer file also contains a number of required configuration values that are automatically generated if you choose not to provide them including the administrative password for MySQL. It is recommended that you store the answer file in a secure location.

4.3. Running PackStack Non-interactively

PackStack supports being run non-interactively. When you run the **`packstack`** command non-interactively you must provide your configuration options via a text file, referred to as an answer file, instead of via standard input.

To do this you must:

- Use PackStack to generate a default answer file.
- Edit the answer file inserting your desired configuration values.
- Run the **packstack** command providing the completed answer file as a command line argument.

PackStack will then attempt to complete the deployment using the configuration options provided in the answer file.

4.3.1. Generating a PackStack Answer File

PackStack is able to generate a generic answer file which you are then able to customize to suit your specific deployment needs.

Procedure 4.3. Generating a PackStack Answer File

- Run the **packstack** command with the ***--gen-answer-file=FILE*** argument to generate an answer file. Replace ***FILE*** with the name of the file you wish to use to store the answer file.

  ```
  # packstack --gen-answer-file=FILE
  ```

  > **Example 4.5. Generating a PackStack Answer File**
  >
  > In this example a PackStack answer file is generated and saved to the file **~/answers.cfg**.
  >
  > ```
  > # packstack --gen-answer-file=~/answers.cfg
  > ```

You have successfully generated an answer file and are ready to begin customizing it for your deployment.

4.3.2. Editing a PackStack Answer File

PackStack answer files are editable in any text editor. Lines preceded with a # character are treated as comments and are ignored.

The table presented here lists the configuration keys available. Configuration values are provided in the answer files as key-value pairs of the form:

```
KEY=VALUE
```

Where a key accepts multiple comma separated values, that is noted in the description of the configuration key. Some configuration keys also have command line equivalents, allowing them to be provided directly as arguments to the invocation of the **packstack** command. Where this is the case the command line argument is also listed in the table.

Table 4.1. PackStack Configuration Keys

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_SSH_KEY	**--sh-public-key**	***/root/.ssh/id_rsa.pub***	Path to a Public key to install on servers. If a usable key has not been installed on the remote servers you will be prompted for a password and this key will be installed so the password will not be required again.
CONFIG_GLANCE_INSTALL	**--os-glance-install**	***y***	Set to **y** if you would like PackStack to install the Image service.
CONFIG_CINDER_INSTALL	**--os-cinder-install**	***y***	Set to **y** if you would like PackStack to install the Volume service.
CONFIG_NOVA_INSTALL	**--os-nova-install**	***y***	Set to **y** if you would like PackStack to install the Compute service.
CONFIG_QUANTUM_INSTALL	**--os-quantum-install**	***y***	Set to **y** if you would like PackStack to install the OpenStack Networking service.
CONFIG_HORIZON_INSTALL	**--os-horizon-install**	***y***	Set to **y** if you would like PackStack to install the Dashboard service.
CONFIG_SWIFT_INSTALL	**--os-swift-install**	***n***	Set to **y** if you would like PackStack to install Object Storage.
CONFIG_CLIENT_INSTALL	**--os-client-install**	***y***	Set to **y** if you would like PackStack to install the OpenStack client packages. An admin "rc" file will also be installed.
CONFIG_NTP_SERVERS	**--ntp-servers**		Comma separated list of NTP servers. Leave plain if PackStack should not install ntpd on instances.
CONFIG_NAGIOS_INSTALL	**--nagios-install**	***n***	Set to **y** if you would like to install Nagios. Nagios provides additional tools for monitoring the OpenStack environment.
CONFIG_MYSQL_HOST	**--mysql-host**	***192.0.43.10***	The IP address of the server on which to install MySQL.
CONFIG_MYSQL_USER		***root***	User name for the MySQL

			administrative user.
CONFIG_MYSQL_PW	**--mysql-pw**		Password for the MySQL administrative user. This value is randomly generated if you do not provide it.
CONFIG_QPID_HOST	**--qpid-host**	***192.0.43.10***	The IP address of the server on which to install the QPID service.
CONFIG_KEYSTONE_HOST	**--keystone-host**	***192.0.43.10***	The IP address of the server on which to install the Identity service.
CONFIG_KEYSTONE_DB_PW	**--keystone-db-passwd**		The password to use for Identity to access the database. This value is randomly generated if you do not provide it.
CONFIG_KEYSTONE_ADMIN_TOKEN	**--keystone-admin-token**		The token to use for the Identity service API. This value is randomly generated if you do not provide it.
CONFIG_KEYSTONE_ADMIN_PW	**--keystone-admin-passwd**		The password to use for the Identity administrative user. This value is randomly generated if you do not provide it.
CONFIG_KEYSTONE_TOKEN_FORMAT	**--keystone-token-format**	***UUID***	PackStack allows a choice of the token format to be used by Keystone, either UUID or PKI. The current default is UUID, although the recommended format for new deployments is PKI, which will become the default in future.
CONFIG_KEYSTONE_DEMO_PW	**--keystone-demo-passwd**		The password to use for the demo tenant. This value is randomly generated if you do not provide it. Only used if CONFIG_PROVISION_DEMO=**y**
CONFIG_GLANCE_HOST	**--glance-host**	***192.0.43.10***	The IP address of the server on which to install the Image service.
CONFIG_GLANCE_DB_PW	**--glance-db-passwd**		The password to use for the Image service to access database. This value is randomly generated if you do not

			provide it.
CONFIG_GLANCE_KS_PW	**--glance-ks-passwd**		The password to use for the Image service to authenticate with Identity. This value is randomly generated if you do not provide it.
CONFIG_CINDER_HOST	**--cinder-host**	***192.0.43.10***	The IP address of the server on which to install the Volume service.
CONFIG_CINDER_DB_PW	**--cinder-db-passwd**		The password to use for the Volume service to access database. This value is randomly generated if you do not provide it.
CONFIG_CINDER_KS_PW	**--cinder-ks-passwd**		The password to use for the Volume service to authenticate with Identity. This value is randomly generated if you do not provide it.
CONFIG_CINDER_VOLUMES_CREATE	**--cinder-volumes-create**	***y***	PackStack expects storage for use with the Volume service to be available on a volume group named **cinder-volumes**. If this volume group does not already exist then PackStack is able to create it automatically. Selecting **y** means that PackStack will create raw disk image in the **/var/lib/cinder** and mount it for use by the Volume service using a loopback device.
CONFIG_CINDER_VOLUMES_SIZE	**--cinder-volumes-size**	***20G***	If you elected to have PackStack create the **cinder-volumes** volume group for you then you will need to provide the desired size of it in gigabytes (GB).
CONFIG_NOVA_API_HOST	**--novaapi-host**	***192.0.43.10***	The IP address of the server on which to install the Compute API service.
CONFIG_NOVA_CERT_HOST	**--novacert-host**	***192.0.43.10***	The IP address of the server on which to install the Compute Certificate

			service.
CONFIG_NOVA_VNCPROXY_HOST	**--novavncproxy-hosts**	***192.0.43.10***	The IP address of the server on which to install the Compute VNC proxy.
CONFIG_NOVA_COMPUTE_HOSTS	**--novacompute-hosts**	***192.0.43.10***	A comma separated list of IP addresses on which to install the Compute services.
CONFIG_NOVA_COMPUTE_PRIVIF	**--novacompute-privif**	***eth1***	Private interface for Flat DHCP on the Compute servers.
CONFIG_NOVA_NETWORK_HOST	**--novanetwork-host**	***192.0.43.10***	The IP address of the server on which to install the Compute Network service.
CONFIG_NOVA_CONDUCTOR_HOST	**--novaconductor-host**	***192.0.43.10***	The IP address of the server on which to install the Compute Network service.
CONFIG_NOVA_DB_PW	**--nova-db-passwd**		The password to use for Compute to access the database. This value is randomly generated if you do not provide it.
CONFIG_NOVA_KS_PW	**--nova-ks-passwd**		The password to use for Compute to authenticate with Identity. This value is randomly generated if you do not provide it.
CONFIG_NOVA_NETWORK_PUBIF	**--novanetwork-pubif**	***eth0***	Public interface on the Compute network server.
CONFIG_NOVA_NETWORK_PRIVIF	**--novanetwork-privif**	***eth1***	Private interface for Flat DHCP on the Compute network server.
CONFIG_NOVA_NETWORK_FIXEDRANGE	**--novanetwork-fixed-range**	***192.168.32.0/22***	IP Range for Flat DHCP.
CONFIG_NOVA_NETWORK_FLOATRANGE	**--nova-network-floating-range**	***10.3.4.0/22***	IP Range for Floating IP addresses.
CONFIG_NOVA_NETWORK_DEFAULTFLOATINGPOOL	**--novanetwork-default-floating-pool**	***nova***	Name of the default floating pool to which the specified floating ranges are added to.
CONFIG_NOVA_NETWORK_AUTOASSIGNFLOATINGIP	**--novanetwork-auto-assign-floating-ip**	***n***	Automatically assign a floating IP to new instances.
CONFIG_NOVA_SCHED_HOST	**--novasched-host**	***192.0.43.10***	The IP address of the server on which to install the Compute Scheduler service.
CONFIG_NOVA_SCHED_CP	**--novasched-cpu-**	***16.0***	The overcommitment ratio

U_ALLOC_RATIO	allocation-ratio		for virtual to physical CPUs. Set to 1.0 to disable CPU overcommitment.
CONFIG_NOVA_SCHED_RAM_ALLOC_RATIO	--novasched-ram-allocation-ratio	*1.5*	The overcommitment ratio for virtual to physical RAM. Set to 1.0 to disable RAM overcommitment.
CONFIG_QUANTUM_SERVER_HOST	--quantum-server-host	*192.0.43.10*	The IP addresses of the server on which to install the OpenStack Networking server.
CONFIG_QUANTUM_USE_NAMESPACES	--quantum-use-namespaces	*y*	Enable network namespaces for OpenStack Networking.
CONFIG_QUANTUM_KS_PW	--quantum-ks-password	*192.0.43.10*	The password to use for OpenStack Networking to authenticate with Keystone.
CONFIG_QUANTUM_DB_PW	--quantum-db-password		The password to use for OpenStack Networking to access database.
CONFIG_QUANTUM_L3_HOSTS	--quantum-l3-hosts	*192.0.43.10*	A comma separated list of IP addresses on which to install OpenStack Networking L3 agent.
CONFIG_QUANTUM_L3_EXT_BRIDGE	--quantum-l3-ext-bridge	*br-ex*	The name of the bridge that the OpenStack Networking L3 agent will use for external traffic. Leave this option blank if you intend to use a provider network to handle external traffic.
CONFIG_QUANTUM_DHCP_HOSTS	--quantum-dhcp-hosts	*192.0.43.10*	A comma separated list of IP addresses on which to install OpenStack Networking DHCP agent.
CONFIG_QUANTUM_L2_PLUGIN	--quantum-l2-plugin	*openvswitch*	The name of the L2 plugin to be used with OpenStack Networking.
CONFIG_QUANTUM_METADATA_HOSTS	--quantum-metadata-hosts	*192.0.43.10*	A comma separated list of IP addresses on which to install OpenStack Networking metadata agent.
CONFIG_QUANTUM_METADATA_PW	--quantum-metadata-pw		Password for OpenStack Networking metadata agent.
CONFIG_QUANTUM_LB_TENANT_NETWORK_TYPE	--quantum-lb-tenant-network-type	*local*	The type of network to allocate for tenant networks. Supported values are **local** and

			vlan. For multi-node deployments **vlan** is recommended.
CONFIG_QUANTUM_LB_VLAN_RANGES	**--quantum-lb-vlan-ranges**		A comma separated list of VLAN ranges for the OpenStack Networking linuxbridge plugin. Each tuple in the list is expected to be in the format ***PHYSICAL***:***START***:***END***. Replace ***PHYSICAL*** with the name of a physical network, replace ***START*** with the start of the VLAN range to identify with it, and replace ***END*** with the end of the VLAN range to associate with it.
CONFIG_QUANTUM_LB_INTERFACE_MAPPINGS	**--quantum-lb-interface-mappings**		A comma separated list of interface mappings for the OpenStack Networking linuxbridge plugin. Each tuple in the list is expected to be in the format ***PHYSICAL***:***INTERFACE***. Replace ***PHYSICAL*** with the name of a physical network, and replace ***INTERFACE*** with the name of the network interface that will be used to connect to the physical network.
CONFIG_QUANTUM_OVS_TENANT_NETWORK_TYPE	**--quantum-ovs-tenant-network-type**	***local***	The type of network to allocate for tenant networks. Supported values are **local** and **vlan**. For multi-node deployments **vlan** is recommended.
CONFIG_QUANTUM_OVS_VLAN_RANGES	**--quantum-ovs-vlan-ranges**		A comma separated list of VLAN ranges for the OpenStack Networking openvswitch plugin. Each tuple in the list is expected to be in the format ***PHYSICAL***:***START***:***END***. Replace ***PHYSICAL*** with the name of a physical network, replace ***START*** with the start of the VLAN range to identify with it, and replace ***END*** with the end of the

			VLAN range to associate with it.
CONFIG_QUANTUM_OVS_BRIDGE_MAPPINGS	**--quantum-ovs-bridge-mappings**	***physnet1:1000:2000***	A comma separated list of bridge mappings for the OpenStack Networking openvswitch plugin. Each tuple in the list is expected to be in the format ***PHYSICAL***:***BRIDGE***. Replace ***PHYSICAL*** with the name of a physical network, and replace ***BRIDGE*** with the name of the Open vSwitch bridge that will be used to connect to the physical network.
CONFIG_OSCLIENT_HOST	**--osclient-host**	***192.0.43.10***	The IP address of the server on which to install the OpenStack client packages. An admin "rc" file will also be installed.
CONFIG_HORIZON_HOST	**--os-horizon-host**	***192.0.43.10***	The IP address of the server on which to install Dashboard.
CONFIG_HORIZON_SSL	**--os-horizon-ssl**	***n***	To set up Dashboard communication over HTTPS set this parameter to **y**.
CONFIG_SSL_CERT	**--os-ssl-cert**		PEM encoded certificate to be used for SSL connections to the HTTPS server, leave blank if one should be generated. This certificate must not require a passphrase.
CONFIG_SSL_KEY	**--os-ssl-key**		Keyfile corresponding to the certificate if one was provided.
CONFIG_SWIFT_PROXY_HOSTS	**--os-swift-proxy**	***192.0.43.10***	The IP address on which to install the Object Storage proxy service.
CONFIG_SWIFT_KS_PW			The password to use for Object Storage to authenticate with Identity. This value is randomly generated if you do not provide it.
CONFIG_SWIFT_STORAGE_HOSTS	**--os-swift-storage**	***192.0.43.10***	A comma separated list of IP addresses on which to install the Object Storage services, each entry should

			take the format ***IP***[/***DEVICE***], for example **192.0.43.10/vdb** will install **/dev/vdb** on **192.0.43.10** as a swift storage device, if /***DEVICE*** is omitted PackStack will create a loopback device for a test setup.
CONFIG_SWIFT_STORAGE_ZONES	**--os-swift-storage-zones**	***1***	Number of swift storage zones, this number **must** be no bigger than the number of storage devices configured.
CONFIG_SWIFT_STORAGE_REPLICAS	**--os-swift-storage-replicas**	***1***	Number of swift storage replicas, this number **must** be no bigger than the number of storage zones configured.
CONFIG_SWIFT_STORAGE_FSTYPE	**--os-swift-storage-fstype**	***ext4***	FileSystem type for storage nodes. Supported values are **ext4** and **xfs** at this time.
CONFIG_PROVISION_DEMO	**--provision-demo**	***n (y for allinone)***	PackStack can provision for demo usage and testing. This key selects whether to provision demo quantum networks, subnets and routers. Set to **y** if you want to provision for demo usage and testing. It requires CONFIG_QUANTUM_INSTALL=**y** and CONFIG_QUANTUM_USE_NAMESPACES=**y**. CONFIG_PROVISION_DEMO will be enabled if you run **packstack --allinone** and CONFIG_QUANTUM_INSTALL=**y**, which it is by default.
CONFIG_PROVISION_TEMPEST	**--provision-tempest**	***n***	PackStack can configure Tempest (OpenStack test suite) for running tests against the OpenStack install. Set to **y** if you want to configure Tempest for testing. It requires CONFIG_QUANTUM_INSTALL=**y** and

			CONFIG_QUANTUM_USE_NAMESPACES=**y**
CONFIG_PROVISION_ALL_IN_ONE_OVS_BRIDGE	**--provision-all-in-one-ovs-bridge**	***n (y for allinone)***	PackStack allows you to configure the external OVS bridge in an all-in-one deployment. This sets up the L3 external bridge with the appropriate IP address to act as the gateway for Virtual Machines. Set to **y** if you want to configure the external OVS bridge. CONFIG_PROVISION_ALL_IN_ONE_OVS_BRIDGE will be enabled if you run **packstack --allinone** and CONFIG_QUANTUM_INSTALL=**y**, which it is by default.
CONFIG_REPO	**--additional-repo**		A comma separated list of URLs to any additional yum repositories to install.
CONFIG_RH_USER	**--rh-username**		To subscribe each server with Red Hat Subscription Manager, include this with **CONFIG_RH_PW**.
CONFIG_RH_PW	**--rh-password**		To subscribe each server with Red Hat Subscription Manager, include this with **CONFIG_RH_USER**.
CONFIG_RH_BETA_REPO	**--rh-beta-repo**	***n***	To subscribe each server to the Red Hat Enterprise Linux Beta repository set this configuration key to **y**. This is only required for preview releases of Red Hat Enterprise Linux OpenStack Platform.
CONFIG_SATELLITE_URL	**--rhn-satellite-server**		To subscribe each server to receive updates from a Satellite server provide the URL of the Satellite server. You must also provide a user name (**CONFIG_SATELLITE_USERNAME**) and password (**CONFIG_SATELLITE_PASSWORD**) or an access key (**CONFIG_SATELLITE_AKEY**) for authentication.

CONFIG_SATELLITE_USERNAME	**--rhn-satellite-username**	Satellite servers require a user name for authentication. If using Satellite to distribute packages to your systems then you must set this configuration key to your Satellite user name or provide an access key for authentication. If you intend to use an access key for Satellite authentication then leave this configuration key blank.
CONFIG_SATELLITE_PW	**--rhn-satellite-password**	Satellite servers require a password for authentication. If using Satellite to distribute packages to your systems then you must set this configuration key to your Satellite password or provide an access key for authentication. If you intend to use an access key for Satellite authentication then leave this configuration key blank.
CONFIG_SATELLITE_AKEY	**--rhn-satellite-activation-key**	Satellite servers are able to accept an access key for authentication. Set this configuration key to your Satellite access key if you have one. If you intend to use a user name and password for Satellite authentication then leave this configuration key blank.
CONFIG_SATELLITE_CACERT	**--rhn-satellite-cacert**	Specify the path to the certificate of the certificate authority that is used to verify that the connection with the Satellite server is secure. Leave this configuration key blank if you are not using Satellite in your deployment.
CONFIG_SATELLITE_PRO	**--rhn-satellite-**	Specify the profile name

FILE	**profile**	that must be used to identify the system in Red Hat Network, if you require one.
CONFIG_SATELLITE_FLAGS	**--rhn-satellite-flags**	Specify any additional Satellite flags that you need to be passed to the **rhnreg_ks** command. This configuration key accepts a comma separated list of flags. Valid flags are **novirtinfo**, **norhnsd**, and **nopackages**. Refer to the Red Hat Satellite documentation for more information.
CONFIG_SATELLITE_PROXY	**--rhn-satellite-proxy-host**	Specify the HTTP proxy that must be used when connecting to the Satellite server, if required.
CONFIG_SATELLITE_PROXY_USER	**--rhn-satellite-proxy-username**	Specify the user name for authenticating with the HTTP proxy that must be used when connecting to the Satellite server, if required.
CONFIG_SATELLITE_PROXY_PW	**--rhn-satellite-proxy-password**	Specify the password for authenticating with the HTTP proxy server that must be used when connecting to the Satellite server, if required.
CONFIG_NAGIOS_HOST	**--nagios-host**	The IP address of the server on which to install Nagios.
CONFIG_NAGIOS_PW	**--nagios-passwd**	The password of the **nagiosadmin** user on the Nagios server. This value will be randomly generated if it is not provided.

Important

The amount of space selected for **CINDER_VOLUMES_SIZE** must be available on the device used for **/var/lib/cinder**.

Important

Remember that the size of the volume group will restrict the amount of disk space that you can expose to compute instances.

Important

PackStack registers systems to Red Hat Network using Subscription Manager. You may encounter problems if your systems have already been registered and subscribed to the Red Hat OpenStack channels using RHN Classic.

4.3.3. Running PackStack with an Answer File

Once an answer file has been created and customized it can be used to run the **`packstack`** command non-interactively.

Procedure 4.4. Running PackStack with an Answer File

1. Run the **`packstack`** command with the ***`--answer-file=FILE`*** parameter to specify an answer file. Replace ***`FILE`*** with the path to the answer file.

   ```
   # packstack --answer-file=FILE
   ```

   > **Example 4.6. Running PackStack with an Answer File**
   >
   > In this example PackStack is run using an answer file stored in **`~/answers.cfg`**.
   >
   > ```
   > # packstack --answer-file=~/answers.cfg
   > ```

2. PackStack will attempt to deploy OpenStack using Puppet manifests. This process may take a significant amount of time depending on the deployment options selected. When the deployment is complete PackStack will display this message:

   ```
   **** Installation completed successfully ******
   ```

   Additional information about your environment including the location of the **`keystonerc`** containing your OpenStack administrator authentication token and the URL of the dashboard, if configured, will also be displayed.

3. Reboot all the nodes in the environment to ensure that the kernel change takes effect.

   PackStack deploys a new kernel with network namespaces enabled for all the nodes. You must reboot the environment to ensure that the change takes effect.

   ```
   # reboot
   ```

You have successfully deployed OpenStack using a PackStack answer file.

Note

A log file containing the details of each PackStack run is stored in a uniquely named folder in the **`/var/tmp/packstack/`** directory.

Chapter 5. PackStack and Passwords

When PackStack deploys OpenStack, it generates passwords for each of the services. You will be using a subset of these passwords for authentication. This chapter describers the location of the passwords and also the steps to be followed in order to change them.

5.1. Password Locations

This section describes the location of the passwords for each service.

Table 5.1. PackStack and Passwords

Service	Location of the Passwords
Identity	`~/keystonerc_admin`
Compute	`/etc/nova/nova.conf`
OpenStack Networking	`/etc/quantum/quantum.conf`
Image	`/etc/glance/glance-api.conf`
Block Storage	`/etc/cinder/cinder.conf`
Object Storage	`/etc/swift/proxy-server.conf`
MySQL Database	`~/.my.cnf`
Nagios	`/etc/nagios/passwd`

Note

Most of the config files also contain the MySQL passwords for the service in the following format. For example, for Glance, **`sql_connection = mysql://glance:12345678abcdefgh@192.0.43.10/glance`** where,

- ***12345678abcdefgh*** is the MySQL password for glance
- first instance of **`glance`** is the user name
- the second instance of **`glance`** is the database name

5.2. Commands to Change Passwords

This section describes the commands that you can use to update the passwords for the services.

- Dashboard Login

```
# keystone user-password-update admin
```

- MySQL

```
# /usr/bin/mysqladmin -u root -p OLDPASS NEWPASS
```

Replace the ***OLDPASS*** with the existing password and the ***NEWPASS*** with the new password.

> **Note**
>
> This command can be use to change the 'root' password and not that of any of the other MySQL users.

- Nagios

  ```
  # htpasswd /etc/nagios/passwd nagiosadmin
  ```

  Replace the **`nagiosadmin`** by the non-admin user name to change the password for a user.

The passwords in the Table 5.1, “PackStack and Passwords” table for Compute, OpenStack Networking, Image, Block Storage and Object Storage are the keystone authentication passwords.

To change these passwords, use the following command.

```
# keystone user-password-update USERNAME
```

Replace ***USERNAME*** with the name of the service you want to change the password to. You will have to enter the new password when the machine prompts you to do so.

> **Important**
>
> Make sure to update the config files for the services after changing the passwords.

Part III. Using OpenStack

Once OpenStack is deployed, interacting with the environment is primarily done using either the dashboard or the command line interface. This part of the guide provides procedures for performing some common tasks using either of these interfaces.

> **Note**
>
> Commands that begin with **`vm$`** as opposed to just **`$`** are commands that should be run inside a virtual machine.

Chapter 6. Using OpenStack With the Dashboard

6.1. Accessing the Dashboard

To access the dashboard you must have first:

- Installed the dashboard.

Procedure 6.1. Accessing the Dashboard

- Log in to the dashboard.

  In your browser, open the link for your configuration to access the dashboard for the first time.

  ```
  http://HOSTNAME/dashboard/
  ```

  Replace **HOSTNAME** with the host name or IP address of the server on which you installed the dashboard.

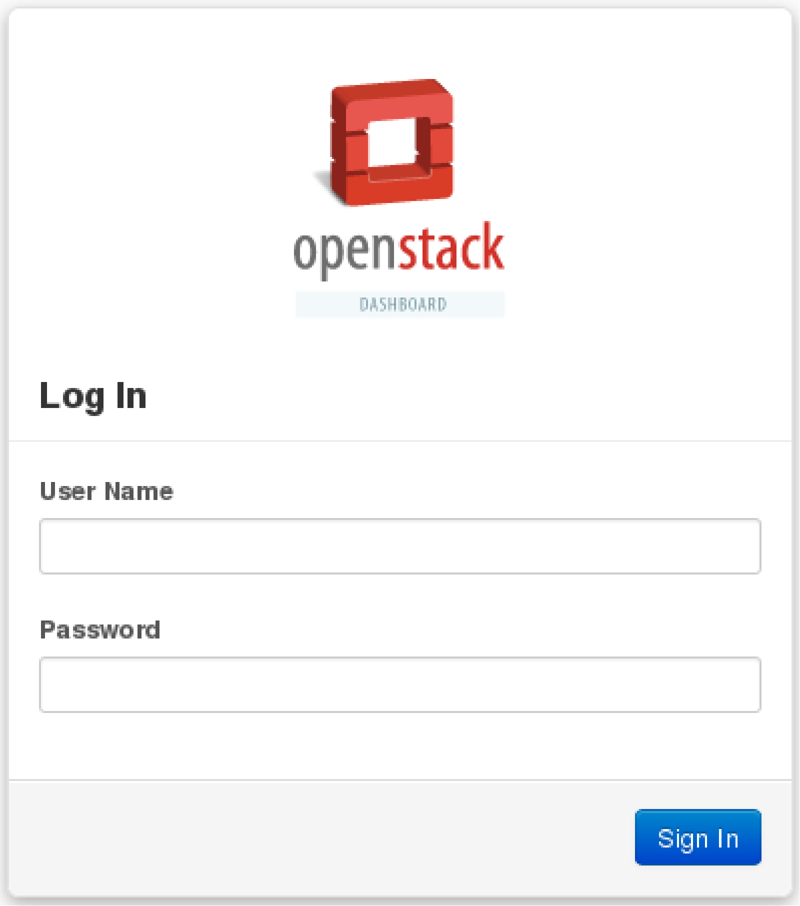
  

  Figure 6.1. Log In Screen

  Enter the user name and password and then click **Sign In**.

  The user name is **admin** and the password is stored as **export OS_PASSWORD=** in the **~/keystonerc_admin** file. If you have enabled the **demo** Keystone tenant, for example by running **packstack --allinone**, you should log into Horizon using the **demo** account instead of the **admin** account due to the ownership of the private and public networks. The **demo** password is stored as **CONFIG_KEYSTONE_DEMO_PW** in PackStack's answer file.

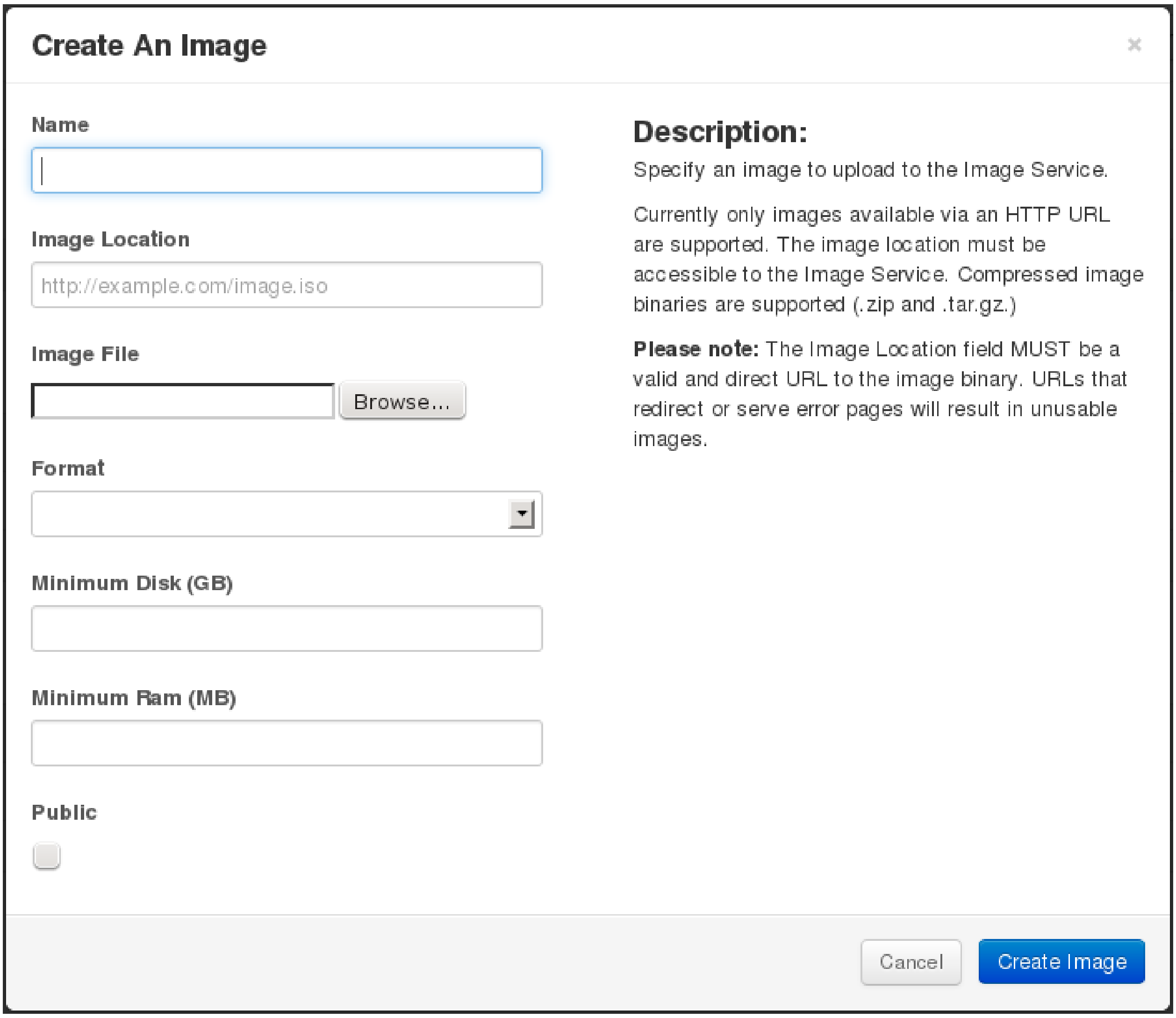
6.2. Uploading a Disk Image

To upload an image using the dashboard you must have first:

- Installed the dashboard.

Procedure 6.2. Uploading an Image using the Dashboard

1. Log in to the dashboard.
2. In the **`Project`** tab, click on **Images & Snapshots** under the **Manage Compute** menu.
3. Click the **`Create Image`** button. The **`Create An Image`** dialog is displayed.



Create An Image

Description:

Specify an image to upload to the Image Service.

Currently only images available via an HTTP URL are supported. The image location must be accessible to the Image Service. Compressed image binaries are supported (.zip and .tar.gz.)

Please note: The Image Location field MUST be a valid and direct URL to the image binary. URLs that redirect or serve error pages will result in unusable images.

Figure 6.2. Create An Image Dialog

4. Configure the settings that define your instance on the **`Details`** tab.
   a. Enter a name for the image.
   b. Include the location URL of the image in the **`Image Location`** field, or save the image file to your machine and use this location in the **`Image File`** field.

      For example, log in to https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=16952 with your Red Hat Customer Portal user name and password. Download the 'KVM Guest Image' and use the location of the saved file in the **`Image File`** field.

c. Select the correct type from the drop down menu for the **Format** field (for example, **QCOW2**).
d. Leave the **Minimum Disk (GB)** and **Minimum RAM (MB)** fields empty.
e. Check the **Public** box.
f. Click the **Create Image** button.

You have successfully uploaded an image.

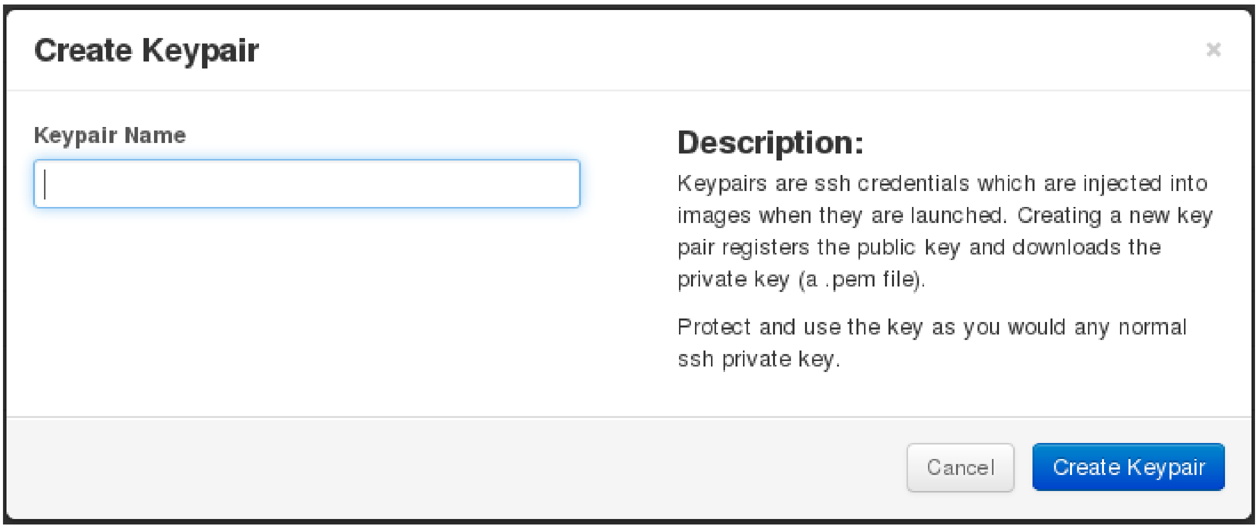
> **Note**
>
> As a result of this procedure, the image is placed in a queue to be uploaded. It may take some time before the **Status** of the image changes from **Queued** to **Active**.

6.3. Creating a Keypair

When a Compute instance is launched, a keypair must be specified, which allows the secure logging in of users into the instance. This section details the steps to create a keypair using the dashboard; this means you must have first installed the dashboard.

Procedure 6.3. Creating a Keypair Using the Dashboard

1. Log in to the dashboard.
2. In the **Project** tab, click on **Access & Security** under the **Manage Compute** menu.
3. On the **Keypairs** tab, click the **Create Keypair** button. The **Create Keypair** dialog is displayed.



Create Keypair

Keypair Name

Description:

Keypairs are ssh credentials which are injected into images when they are launched. Creating a new key pair registers the public key and downloads the private key (a .pem file).

Protect and use the key as you would any normal ssh private key.

Figure 6.3. Create Keypair

4. Specify a name in the **Keypair Name** field, and click the **Create Keypair** button.

   This creates the keypair, which can be used when launching an instance.

Note

- When a keypair is created, a keypair file is automatically downloaded through the browser. You can optionally load this file into ssh, for command-line ssh connections, by executing:

```
# ssh-add ~/.ssh/NAME.pem
```

- To delete an existing keypair, click the keypair's **Delete Keypair** button on the **Keypairs** tab.

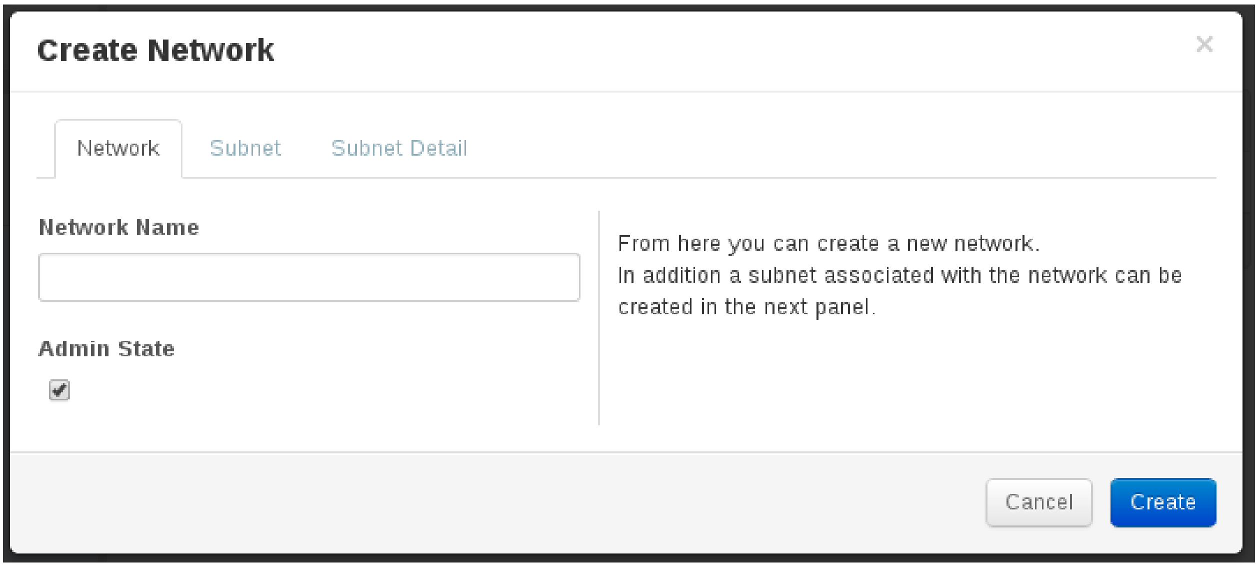
6.4. Creating a Network

To create a network from the dashboard, you must have first:

- Installed the dashboard.
- Installed OpenStack Networking Services.

Procedure 6.4. Creating a Network Using the Dashboard

1. Log in to the dashboard.
2. In the **Project** tab, click on **Networks** under the **Manage Network** menu.
3. Click the **Create Network** button. The **Create Network** dialog is displayed.



Create Network

Network | Subnet | Subnet Detail

Network Name

Admin State

☑

From here you can create a new network.
In addition a subnet associated with the network can be created in the next panel.

Figure 6.4. Create Network: Network Tab

4. By default, the dialog opens to the **Network** tab. You have the option of specifying a network name.
5. To define the network's subnet, click on the **Subnet** and **Subnet Detail** tabs. Click into each field for field tips.

> **Note**
>
> You do not have to initially specify a subnet (although this will result in any attached instance having the status of 'error'). If you do not define a specific subnet, clear the **`Create Subnet`** check box.

6. Click the **`Create`** button.

You have successfully created a new network.

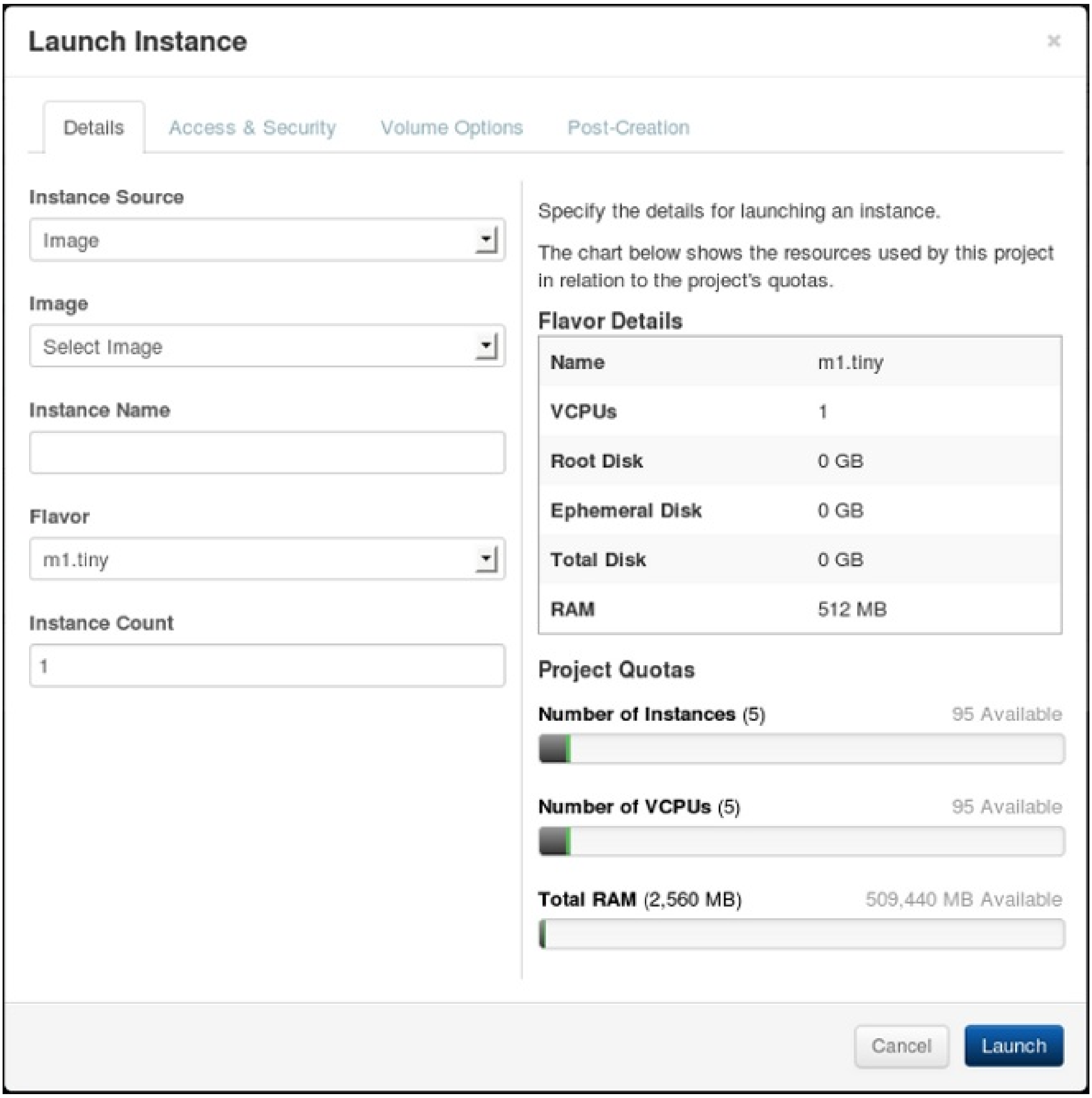
6.5. Launching an Instance

To launch an instance from the dashboard you must have first:

- Installed the dashboard.
- Uploaded an image to use as the basis for your instances.
- Created a network.

Procedure 6.5. Launching an Instance using the Dashboard

1. Log in to the dashboard.
2. In the **`Project`** tab, click on **Instances** under the **Manage Compute** menu.
3. Click the **`Launch Instance`** button. The **`Launch Instance`** dialog is displayed.



Instance Source: Image

Image: Select Image

Instance Name:

Flavor: m1.tiny

Instance Count: 1

Specify the details for launching an instance.

The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

Name	m1.tiny
VCPUs	1
Root Disk	0 GB
Ephemeral Disk	0 GB
Total Disk	0 GB
RAM	512 MB

Project Quotas

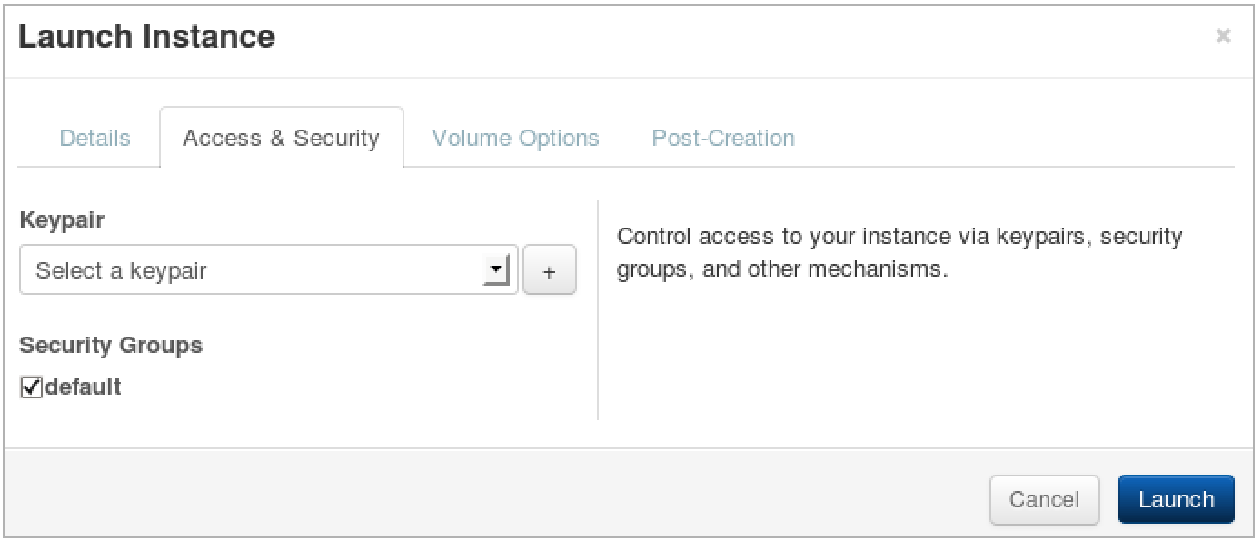
Number of Instances (5) — 95 Available

Number of VCPUs (5) — 95 Available

Total RAM (2,560 MB) — 509,440 MB Available

Figure 6.5. Launch Instance Dialog

4. By default, the dialog opens to the **`Details`** tab.
   a. Select an **`Instance Source`** for your instance. Available values are:
      - **`Image`**
      - **`Snapshot`**
   b. Select an **`Image`** or **`Snapshot`** to use when launching your instance. The image selected defines the operating system and architecture of your instance.
   c. Enter an **`Instance Name`** to identify your instance.
   d. Select a **`Flavor`** for your instance. The flavor selected determines the compute resources available to your instance. The specific resources for the flavor selected are displayed in the **`Flavor Details`** pane for you to preview.
   e. Enter an **`Instance Count`**. This determines how many instances to launch using the selected options.
5. Click the **`Access & Security`** tab and configure the security settings for your instance.



☑ default

Control access to your instance via keypairs, security groups, and other mechanisms.

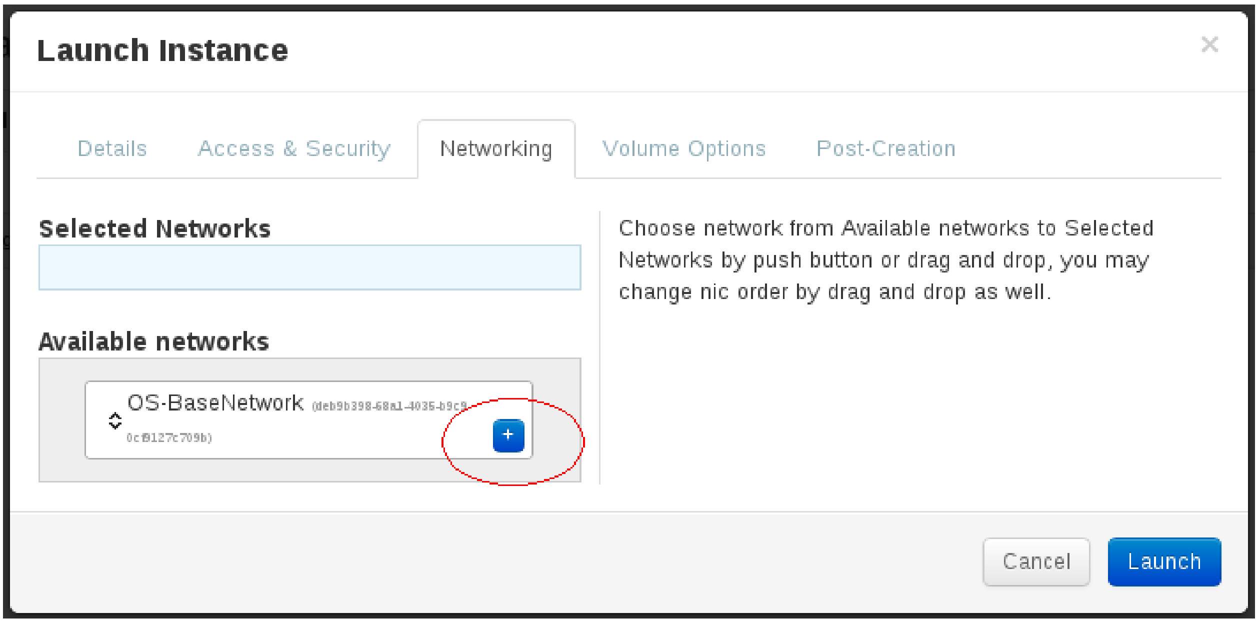
Figure 6.6. Launch Instance: Access & Security Tab

a. Select an existing keypair from the **`Keypair`** drop down box or click the **+** button to upload a new keypair

b. Select the **`Security Groups`** that you wish to apply to your instances. By default only the **`default`** security group will be available.

6. Click the **`Networking`** tab and select the network for the instance by clicking on the network's **+** sign.

   If you have logged in as the **`demo`** Keystone tenant, choose the **`private`** network, due to the ownership of the private and public networks.



OS-BaseNetwork (deb9b398-68a1-4035-b9c9-0cf8127c709b)

Choose network from Available networks to Selected Networks by push button or drag and drop, you may change nic order by drag and drop as well.

Figure 6.7. Launch Instance: Networking Tab

7. Click the **`Launch`** button.

You have just created a Compute instance.

Note

To launch the instance console from the Dashboard:

1. On the **Instances** tab, click the name of your instance. The **Instance Detail** page is displayed.
2. Click the **Console** tab on the resultant page.

An instance of the VNC console is run within the browser.

6.6. Creating a Volume

Compute instances support the attachment and detachment of block storage volumes. This procedure details the steps involved in creating a logical volume using the dashboard.

To create a volume from the dashboard, you must have first:

- Installed the dashboard.
- Installed the Block Storage service.

Procedure 6.6. Creating a Volume using the Dashboard

1. Log in to the dashboard.
2. In the **Project** tab, click on **Volumes** under the **Manage Compute** menu.
3. Click the **Create Volume** button. The **Create Volume** dialog is displayed.

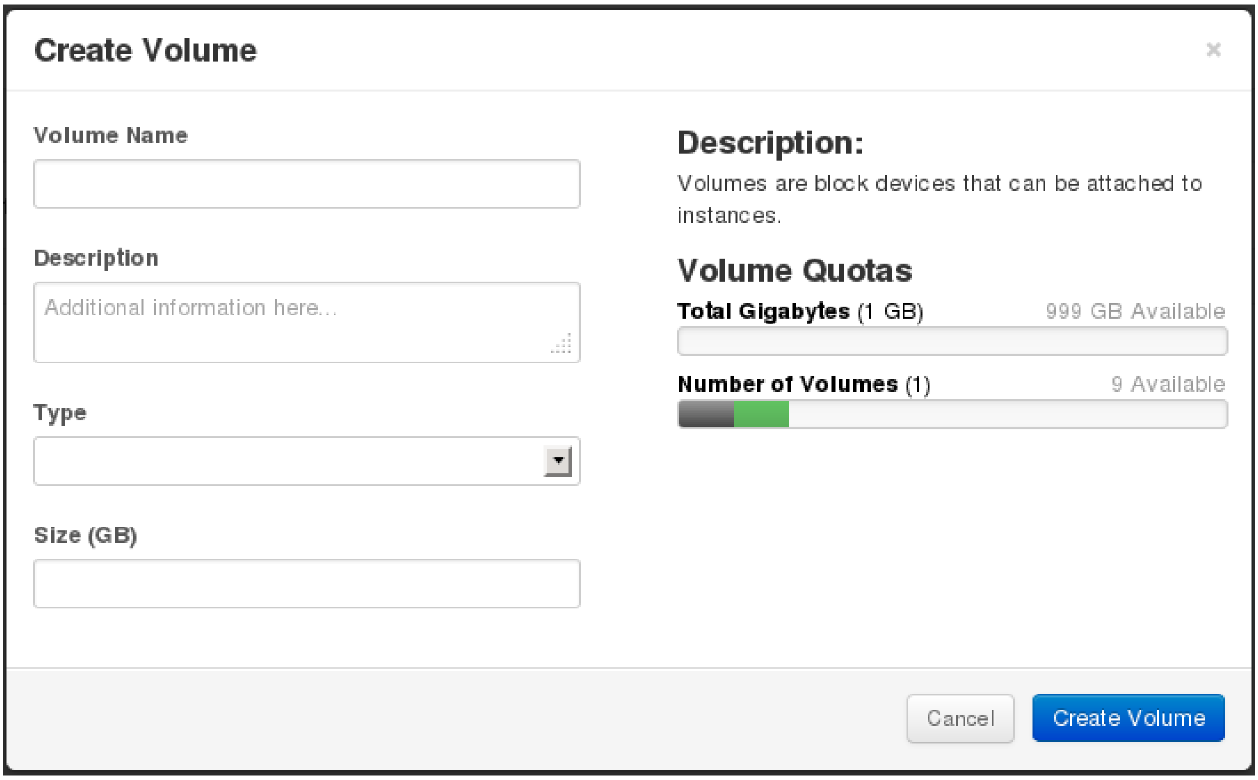


Figure 6.8. Create Volume Dialog

4. Configure the values that will define your new volume.

   a. Enter a **`Volume Name`** to identify your new volume by.
   b. Enter a **`Description`** to further describe your new volume.
   c. Enter the **`Size`** of your new volume in gigabytes (GB).

> **Important**
>
> Your new volume will be allocated from the **`cinder-volumes`** volume group. There must be enough free disk space in the **`cinder-volumes`** volume group for your new volume to be allocated.

5. Click the **`Create Volume`** button to create the new volume.

You have successfully created a Cinder volume using the dashboard.

6.7. Attaching a Volume

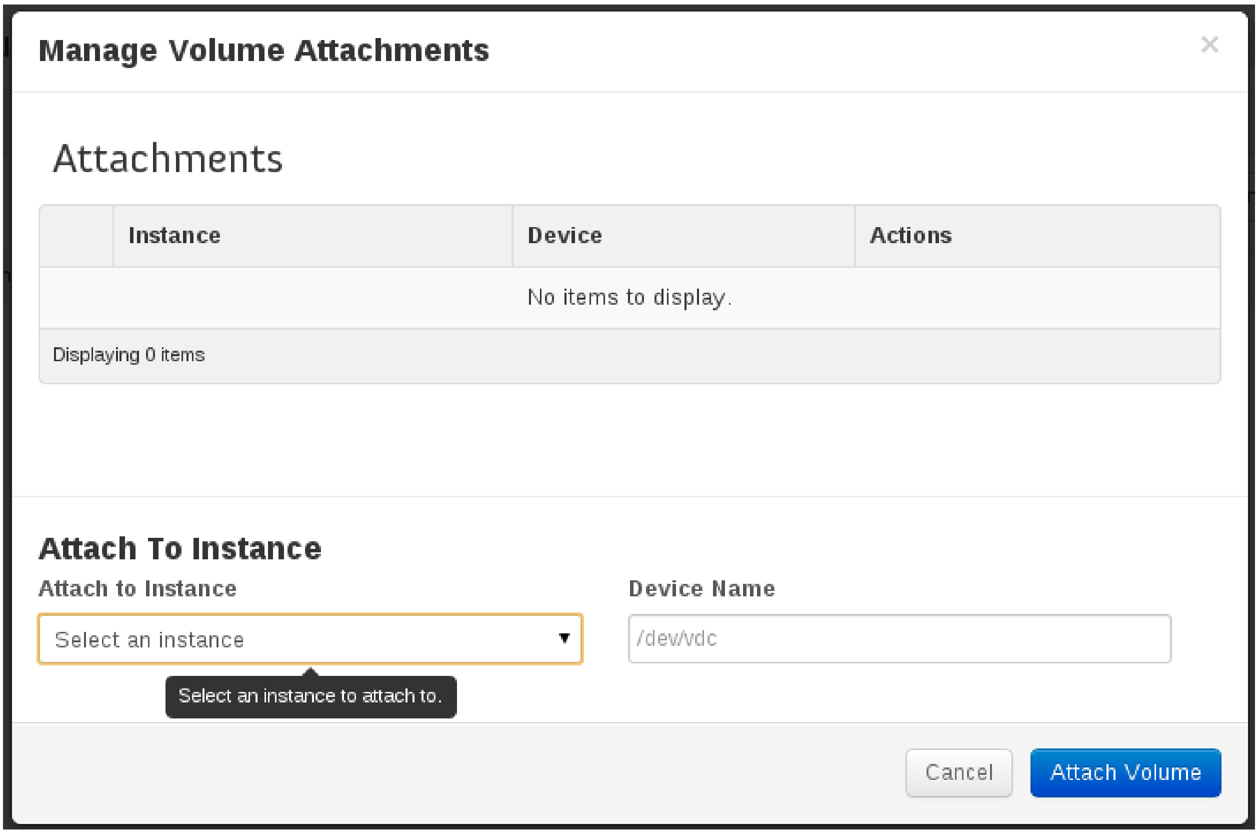
This procedure details the steps involved in attaching a Cinder volume to an existing compute instance using the dashboard.

To create a volume from the dashboard, you must have first:

- Installed the dashboard.
- Launched an instance.
- Created a volume.

Procedure 6.7. Attaching a Volume using the Dashboard

1. Log in to the dashboard.
2. In the **`Project`** tab, click on **Volumes** under the **Manage Compute** menu.
3. Click the **`Edit Attachments`** button on the row associated with the volume that you want to attach to an instance. The **`Manage Volume Attachments`** dialog is displayed.



Manage Volume Attachments

Attachments

	Instance	Device	Actions
		No items to display.	

Displaying 0 items

Attach To Instance

Attach to Instance: Select an instance

Device Name: /dev/vdc

Select an instance to attach to.

Figure 6.9. Manage Volume Attachments dialog

4. Select the instance to attach the volume to from the **Attach to Instance** box.
5. Click the **Attach Volume** button to attach the volume to the selected instance.

You have successfully attached a Cinder volume to an instance using the dashboard. The volume will appear as a physical hard disk drive to the guest operating system.

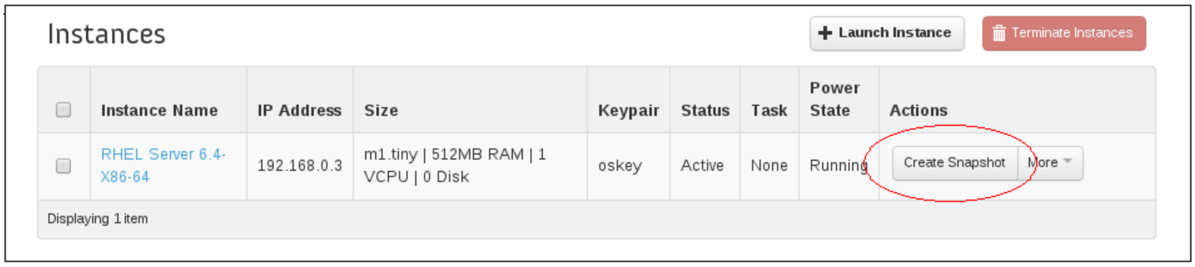
6.8. Creating an Instance Snapshot

This procedure details the steps involved in creating a snapshot based on a running instance using the dashboard. This may be done for backup purposes or for creating a base image to create other instances from after applying some customization to it.

To create a snapshot, a running instance must be available.

Procedure 6.8. Creating a Snapshot using the Dashboard

1. Log in to the dashboard.
2. In the **Project** tab, click on **Instances** under the **Manage Compute** menu.
3. Click the **Create Snapshot** button on the row associated with the instance that you want to create a snapshot. The **Create Snapshot** dialog is displayed.



Instances

□	Instance Name	IP Address	Size	Keypair	Status	Task	Power State	Actions
□	RHEL Server 6.4-X86-64	192.168.0.3	m1.tiny | 512MB RAM | 1 VCPU | 0 Disk	oskey	Active	None	Running	Create Snapshot More

Displaying 1 item

Figure 6.10. Instances: Create Snapshot

4. Enter a descriptive name for your snapshot in the **`Snapshot Name`** field.
5. Click the **`Create Snapshot`** to create the snapshot.
6. The **`Images & Snapshots`** screen is displayed. Your new snapshot will appear in the **`Image Snapshots`** table.

You have successfully created a snapshot of your instance which can be used to restore instance state or as a basis for spawning new instances.

6.9. Adding a Rule to a Security Group

Security groups are used to specify what IP traffic is allowed to reach an instance on its public IP address. The rules defined by security groups are processed before network traffic reaches any firewall rules defined within the guest itself.

Note

In the default configuration, the 'default' security group accepts all connections from the 'default' source; all instances with the 'default' group can talk to each other on any port.

Procedure 6.9. Adding a Rule to a Security Group using the Dashboard

1. Log in to the dashboard.
2. In the **`Project`** tab, click on **Access & Security** under the **Manage Compute** menu.
3. In the **`Security Groups`** tab click the **`Edit Rules`** button on the row for the **`default`** security group.

   The **`Edit Security Group Rules`** page is displayed.
4. Click the **`Add Rule`** button. The Add Rule window is displayed.
5. Configure the rule.
   a. Select the protocol that the rule must apply to from the **`IP Protocol`** list.
   b. Define the port or ports to which the rule will apply using the **`Open`** field:
      - Port- Define a specific port in the **`Port`** field
      - Port Range- Define the port range using the **`From Port`** and **`To Port`** fields.
   c. Define the IP address form which connections should be accepted on the defined post using the **`Source`** field:
      - CIDR- Enter the IP address to accept connections from using Classless Inter-Domain Routing (CIDR) notation. A value of **`0.0.0.0/0`** allows connections from all IP addresses.

    - Security Group- Alternatively select an existing security group from the **Source Group** list to use the same IP address range selection for this entry.
6. Click the **Add** button to add the new rule to the security group.

You have successfully added a rule to a security group using the dashboard. It is now possible to connect to instances that use the altered security group from the specified IP address block and using the specified ports and protocol.

6.10. Adding Floating IP Addresses

When an instance is created in OpenStack, it is automatically assigned a fixed IP address in the network to which the instance is assigned. This IP address is permanently associated with the instance until the instance is terminated.

However, a floating IP address can also be attached to an instance (in addition to their fixed IP address). Unlike fixed IP addresses, floating IP addresses are able to have their associations modified at any time, regardless of the state of the instances involved. This procedure details the reservation of a floating IP address from an existing pool of addresses and the association of that address with a specific instance.

To associate floating IP addresses, you must have first:

- Created a pool of floating IP addresses
- Launched an instance.

Defining a pool of floating IP addresses is currently only possible using the command line interface. Reserving addresses and associating addresses with specific instances is possible using both the command line interface and the dashboard.

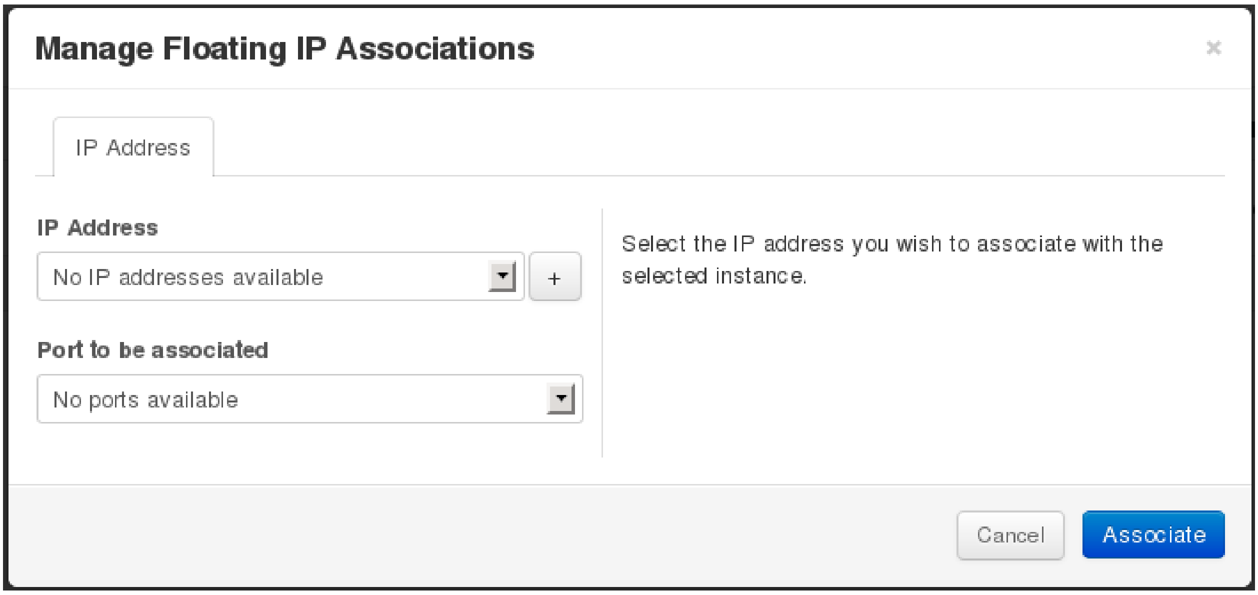
This procedure details the reservation of a floating IP address from an existing pool of addresses and the association of that address with a specific compute instance. This assumes that a pool of floating IP addresses has already been defined.

Refer to Section 7.7.2, “Adding Floating IP Addresses” for information about defining a pool of floating IP addresses.

Procedure 6.10. Adding Floating IP Addresses using the Dashboard

1. Log in to the dashboard.
2. In the **Project** tab, click on **Access & Security** under the **Manage Compute** menu.
3. Click the **Allocate IP To Project** button. The **Allocate Floating IP** window is displayed.
4. Select a pool of addresses from the **Pool** list.
5. In the **Floating IPs** tab, click on the **Allocate IP to Project** button. The allocated IP address will appear in the **Floating IPs** table.
6. Locate the newly allocated IP address in the **Floating IPs** table. On the same row click the **Associate Floating IP** button to assign the IP address to a specific instance.

   The **Manage Floating IP Associations** window is displayed.



IP Address: No IP addresses available

Port to be associated: No ports available

Select the IP address you wish to associate with the selected instance.

Figure 6.11. Manage Floating IP Addresses Dialog

7. The **`IP Address`** field is automatically set to the selected floating IP address.

   Select the instance to associate the floating IP address with from the **`Instance`** list.
8. Click the **`Associate`** button to associate the IP address with the selected instance.

> **Note**
>
> To disassociate a floating IP address from an instance when it is no longer required use the **`Disassociate Floating IP`** button.

You have successfully associated a floating IP address with an instance using the dashboard.

6.11. Creating a Router

This section details the step to create a router using the dashboard, which connects an internal network to an external one. You must first have:

- Installed the dashboard.
- Created an external network by *Defining a Floating IP-Address Pool*.
- Created an internal network.

Procedure 6.11. Creating a Router Using the Dashboard

1. Log in to the dashboard.
2. In the **`Project`** tab, click on **Routers** under the **Manage Network** menu.
3. Click on the **`Create Router`** button. The **`Create Router`** window is displayed:

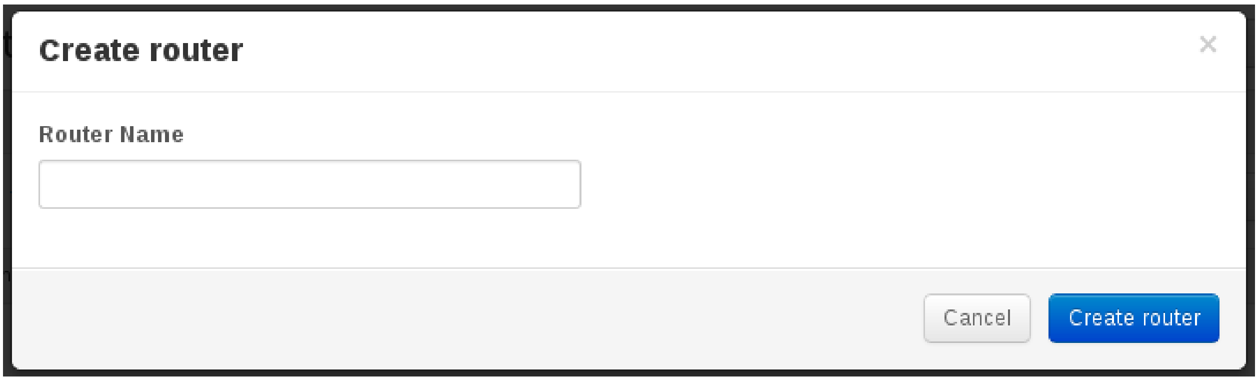
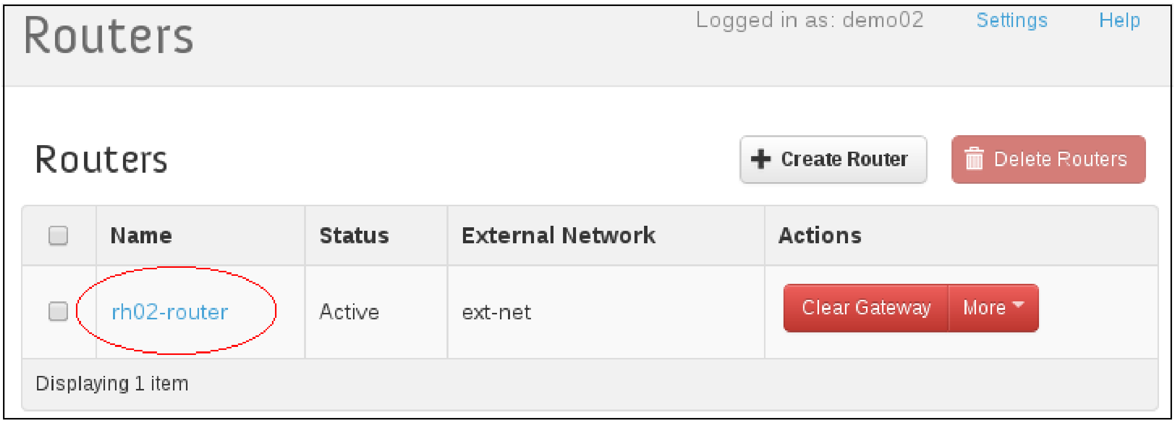


Figure 6.12. Create Router

The new router is now displayed in the router list.

4. Click the new router's **`Set Gateway`** button.
5. Specify the network to which the router will connect in the **`External Network`** field, and click the **`Set Gateway`** button.
6. To connect a private network to the newly created router:
    a. Click on the router name in the router list:

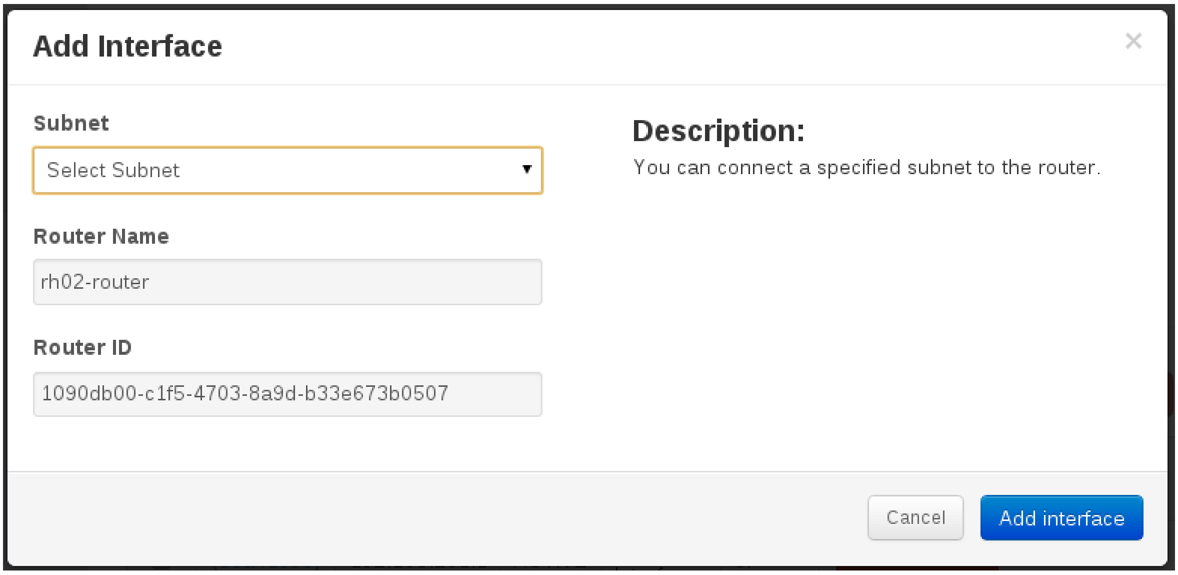


☐	Name	Status	External Network	Actions
☐	rh02-router	Active	ext-net	Clear Gateway More

Displaying 1 item

Figure 6.13. Select the router

    b. Click the **`Add Interface`** button. The **`Add Interface`** window is displayed:



Router Name: rh02-router

Router ID: 1090db00-c1f5-4703-8a9d-b33e673b0507

Description:

You can connect a specified subnet to the router.

Figure 6.14. Add Interface

c. Specify the new subnet to which the router should be attached in the **`Subnet`** field, and click **`Add Interface`**.

You have successfully created the router; you can view the new topology by clicking on **`Network Topology`** in the **`Manage Network`** menu.

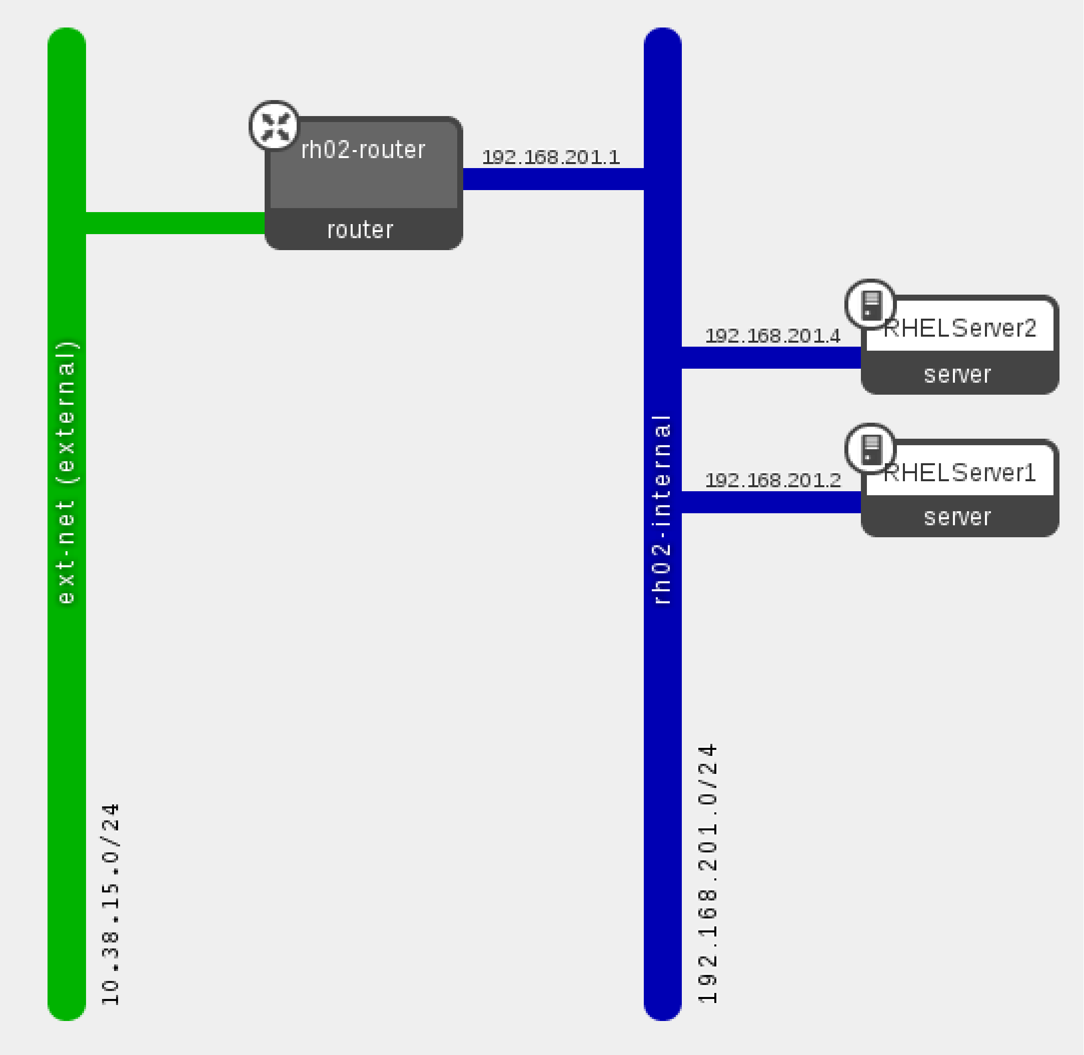


Figure 6.15. Network Topology

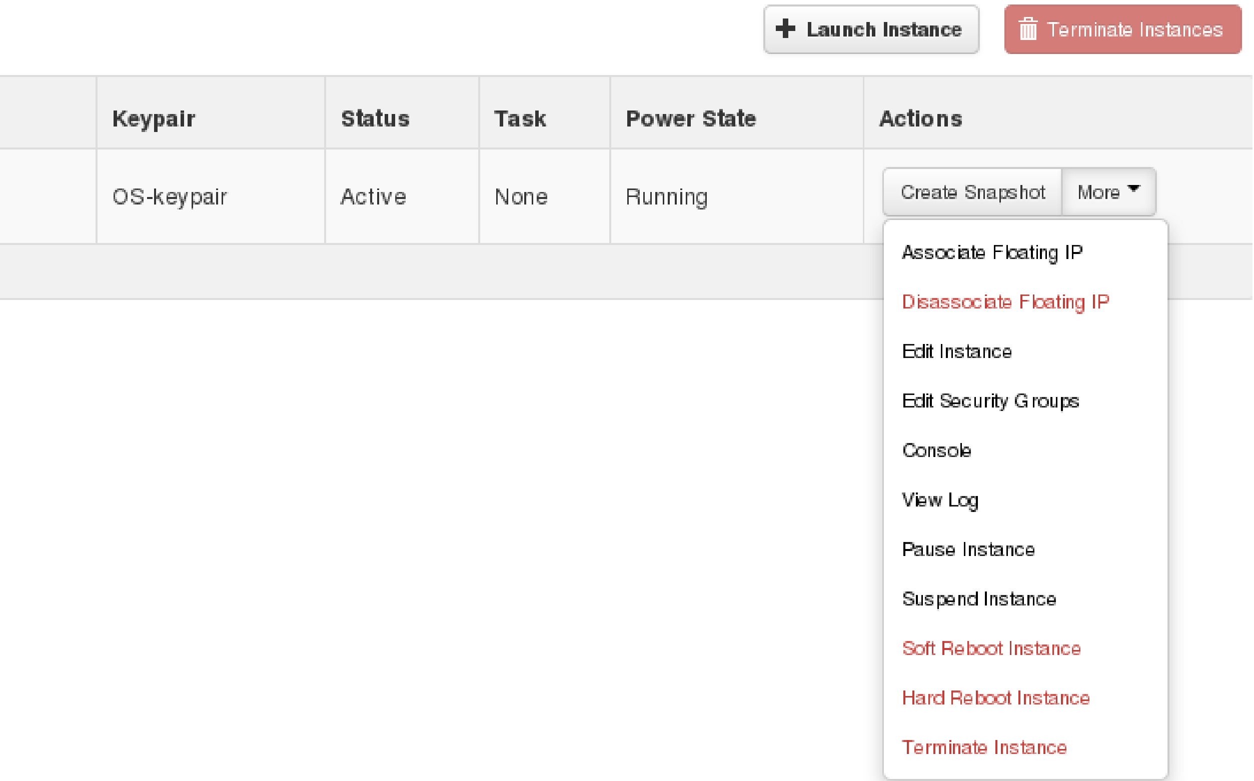
6.12. Controlling the State of an Instance (Pause, Suspend, Reboot)

To change the state of an instance using the dashboard you must have first:

- Installed the dashboard.
- Uploaded an image to use as the basis for your instances.
- Launched an instance and associated a floating IP address with it.

Procedure 6.12. Controlling the State of an Instance using the Dashboard

1. Log in to the dashboard.
2. In the **`Project`** tab, click on **Instances** under the **Manage Compute** menu.
3. Select the instance for which you want to change the state and click on the **`More`** dropdown button. The dropdown list is displayed.



	Keypair	Status	Task	Power State	Actions
	OS-keypair	Active	None	Running	Create Snapshot More

Figure 6.16. Control the State of an Instance

4. Select the state that you want to change the instance into.

> **Warning**
>
> Be careful while using the options in **`red`**. Using these could affect the working of your OpenStack setup.

You have successfully changed the state of the instance.

6.13. Deleting an Instance

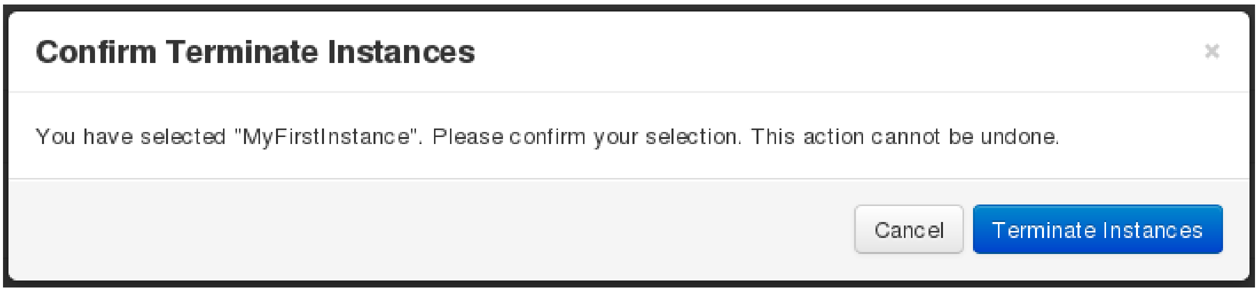
To delete an instance using the dashboard you must have first:

- Installed the dashboard.
- Uploaded an image to use as the basis for your instances.
- Launched an instance and associated a floating IP address with it.

Procedure 6.13. Deleting an Instance using the Dashboard

1. Log in to the dashboard.

2. In the **`Project`** tab, click on **Instances** under the **Manage Compute** menu.
3. Select the instance or instances that you want to delete and click on the **`Terminate Instances`** button. The **`Terminate Instances`** dialog is displayed.



Confirm Terminate Instances

You have selected "MyFirstInstance". Please confirm your selection. This action cannot be undone.

Figure 6.17. Terminate an Instance

4. Click on the **`Terminate Instances`** button to confirm deletion of the instance or instances.

You have now deleted the instances.

Chapter 7. Using OpenStack With the Command Line Interface

7.1. Uploading an Image

To launch instances based on images stored in the OpenStack Image storage service, you must first have added some images. You must either have downloaded or created suitable images to use in an OpenStack environment.

The simplest way is to download an image. Log in to https://rhn.redhat.com/rhn/software/channel/downloads/Download.do?cid=16952 with your Customer Portal user name and password. Download the 'KVM Guest Image' and use the file with the **--file** parameter.

If you want to create an image, refer to the section on *Building Images using Oz* in the Red Hat Enterprise Linux OpenStack Platform *Installation and Configuration Guide*. You can also refer to the Red Hat Enterprise Linux *Virtualization Host Configuration and Guest Installation Guide* for more information.

Important

It is recommended that the **virt-sysprep** command is run on Linux-based virtual machine images prior to uploading them to Image service. The **virt-sysprep** command re-initializes a disk image in preparation for use in a virtual environment. Operations it performs by default include removal of SSH keys, removal of persistent MAC addresses, and removal of user accounts.
The **virt-sysprep** command is available in the *libguestfs-tools* package in Red Hat Enterprise Linux.

```
$ yum install -y libguestfs-tools
$ virt-sysprep --add FILE
```

Refer to the **virt-sysprep** manual page by running the **man virt-sysprep** command for information on enabling and disabling specific operations.

Procedure 7.1. Uploading an Image Using the Command Line Interface

1. Ensure that you have set the environment variables used for authenticating with OpenStack Identity service by loading them from the **keystonerc** file associated with your user. Note that an administrative account is not required.

   ```
   $ source ~/keystonerc_user
   ```

2. Use the **glance image-create** command to import your disk image:

   ```
   $ glance image-create --name "NAME" \
           --is-public IS_PUBLIC \
           --disk-format DISK_FORMAT \
           --container-format CONTAINER_FORMAT \
           --file IMAGE
   ```

   Replace the command line arguments in **glance image-create** with the appropriate values for your environment and disk image:

- Replace ***NAME*** with the name that users will refer to the disk image by.
- Replace ***IS_PUBLIC*** with **`true`** or **`false`**. Setting this value to **`true`** means that all users will be able to view and use the image.
- Replace ***DISK_FORMAT*** with the format of the virtual machine disk image. Valid values include **`raw`**, **`vhd`**, **`vmdk`**, **`vdi`**, **`iso`**, **`qcow2`**, **`aki`**, and **`ami`**.

  If the format of the virtual machine disk image is unknown then use the **`qemu-img info`** command to try and identify it.

  Example 7.1. Using `qemu-img info`

  In this example the **`qemu-img info`** command is used to determine the format of a disk image stored in the file **`./RHEL64.img`**.

  ```
   $ qemu-img info ./RHEL64.img
  image: ./RHEL64.img
  file format: qcow2
  virtual size: 5.0G (5368709120 bytes)
  disk size: 136K
  cluster_size: 65536
  ```

- Replace ***CONTAINER_FORMAT*** with the container format of the image. The container format is **`bare`** unless the image is packaged in a file format such as **`ovf`** or **`ami`** that includes additional metadata related to the image.
- Replace ***IMAGE*** with the local path to the image file to upload.

Refer to the output of the **`glance help image-create`** command for more information about supported arguments to **`glance image-create`**.

Note

If the image being uploaded is not locally accessible but is available using a remote URL then provide it using the **`--location`** parameter instead of using the **`--file`** parameter. Note however that unless you also specify the **`--copy-from`** argument, the Image service will not copy the image into the object store but instead it will be accessed remotely each time it is required.

Example 7.2. Uploading an Image

In this example the **`qcow2`** format image in the file named **`rhel-64.qcow2`** is uploaded to the Image service. It is created in the Image service as a publicly accessible image named **`RHEL 6.4`**.

```
 $ glance image-create --name "RHEL 6.4" --is-public true --disk-format
qcow2 \
          --container-format bare \
          --file rhel-64.qcow2
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | 2f81976cae15c16ef0010c51e3a6c163     |
| container_format | bare                                 |
| created_at       | 2013-01-25T14:45:48                  |
| deleted          | False                                |
| deleted_at       | None                                 |
| disk_format      | qcow2                                |
| id               | 0ce782c6-0d3e-41df-8fd5-39cd80b31cd9 |
| is_public        | True                                 |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | RHEL 6.4                             |
| owner            | b1414433c021436f97e9e1e4c214a710     |
| protected        | False                                |
| size             | 25165824                             |
| status           | active                               |
| updated_at       | 2013-01-25T14:45:50                  |
+------------------+--------------------------------------+
```

3. Use the **`glance image-list`** command to verify that your image was successfully uploaded.

```
 $ glance image-list
+---------------+-----------+-------------+------------------+----------+----
----+
| ID            | Name      | Disk Format | Container Format |Size      |
Status |
+---------------+-----------+-------------+------------------+----------+----
----+
| 0ce782c6-... | RHEL 6.4 | qcow2       | bare             |213581824 |
active |
+---------------+-----------+-------------+------------------+----------+----
----+
```

Use the **`glance image-show`** command to view more detailed information about an image. Use the identifier of the image to specify the image that you wish to view.

```
$ glance image-show 0ce782c6-0d3e-41df-8fd5-39cd80b31cd9
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | 2f81976cae15c16ef0010c51e3a6c163     |
| container_format | bare                                 |
| created_at       | 2013-01-25T14:45:48                  |
| deleted          | False                                |
| disk_format      | qcow2                                |
| id               | 0ce782c6-0d3e-41df-8fd5-39cd80b31cd9 |
| is_public        | True                                 |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | RHEL 6.4                             |
| owner            | b1414433c021436f97e9e1e4c214a710     |
| protected        | False                                |
| size             | 25165824                             |
| status           | active                               |
| updated_at       | 2013-01-25T14:45:50                  |
+------------------+--------------------------------------+
```

You have successfully uploaded a disk image to the OpenStack Image storage service. This disk image can now be used as the basis for launching virtual machine instances in your OpenStack environment.

7.2. Launching an Instance

When launching an instance using OpenStack, you must specify the ID for the flavor you want to use for the instance. A flavor is a resource allocation profile. For example, it specifies how many virtual CPUs and how much RAM your instance will get. To see a list of the available profiles, run the **nova flavor-list** command.

```
$ nova flavor-list
+----+-----------+-----------+------+-----------+------+-------+-------------+
| ID |    Name   | Memory_MB | Disk | Ephemeral | Swap | VCPUs | RXTX_Factor |
+----+-----------+-----------+------+-----------+------+-------+-------------+
| 1  | m1.tiny   | 512       | 0    | 0         |      | 1     | 1.0         |
| 2  | m1.small  | 2048      | 10   | 20        |      | 1     | 1.0         |
| 3  | m1.medium | 4096      | 10   | 40        |      | 2     | 1.0         |
| 4  | m1.large  | 8192      | 10   | 80        |      | 4     | 1.0         |
| 5  | m1.xlarge | 16384     | 10   | 160       |      | 8     | 1.0         |
+----+-----------+-----------+------+-----------+------+-------+-------------+
```

Get the ID of the image you would like to use for the instance using the **nova image-list** command. Create the instance using **nova boot**. If there is not enough RAM available to start an instance, Nova will not do so. Create an instance using flavor **1** or **2**. Once the instance has been created, it will show up in the output of **nova list**. You can also retrieve additional details about the specific instance using the **nova show** command.

```
$ nova image-list
+--------------------------------------+-------------+--------+--------+
|                  ID                  |    Name     | Status | Server |
+--------------------------------------+-------------+--------+--------+
| 17a34b8e-c573-48d6-920c-b4b450172b41 | RHEL 6.2    | ACTIVE |        |
+--------------------------------------+-------------+--------+--------+
$ nova boot --flavor 2 --key_name oskey --image 17a34b8e-c573-48d6-920c-
b4b450172b41 rhel
+------------------------+--------------------------------------+
|        Property        |                Value                 |
+------------------------+--------------------------------------+
| OS-DCF:diskConfig      | MANUAL                               |
| OS-EXT-STS:power_state | 0                                    |
| OS-EXT-STS:task_state  | scheduling                           |
| OS-EXT-STS:vm_state    | building                             |
| accessIPv4             |                                      |
| accessIPv6             |                                      |
| adminPass              | QVAmyS5i5etE                         |
| config_drive           |                                      |
| created                | 2012-05-18T13:41:40Z                 |
| flavor                 | m1.small                             |
| hostId                 |                                      |
| id                     | 0e4011a4-3128-4674-ab16-dd1b7ecc126e |
| image                  | RHEL 6.2                             |
| key_name               | oskey                                |
| metadata               | {}                                   |
| name                   | rhel                                 |
| progress               | 0                                    |
| status                 | BUILD                                |
| tenant_id              | 05816b0106994f95a83b913d4ff995eb     |
| updated                | 2012-05-18T13:41:40Z                 |
| user_id                | 1d59c0bfef9b4ea9ab63e2a058e68ae0     |
+------------------------+--------------------------------------+
$ nova list
+--------------------------------------+--------+--------+------------------+
|                  ID                  |  Name  | Status |     Networks     |
+--------------------------------------+--------+--------+------------------+
| 0e4011a4-3128-4674-ab16-dd1b7ecc126e | rhel   | BUILD  | demonet=10.0.0.2 |
+--------------------------------------+--------+--------+------------------+
$ nova list
+--------------------------------------+--------+--------+------------------+
|                  ID                  |  Name  | Status |     Networks     |
+--------------------------------------+--------+--------+------------------+
| 0e4011a4-3128-4674-ab16-dd1b7ecc126e | rhel   | ACTIVE | demonet=10.0.0.2 |
+--------------------------------------+--------+--------+------------------+
$ nova show 0e4011a4-3128-4674-ab16-dd1b7ecc126e
```

Once enough time has passed so that the instance is fully booted and initialized, you can ssh into the instance. You can obtain the IP address of the instance from the output of **`nova list`**.

```
$ ssh -i oskey.priv root@10.0.0.2
```

7.3. Creating a Volume

Nova compute instances support the attachment and detachment of Cinder storage volumes. This procedure details the steps involved in creating a logical volume in the **`cinder-volumes`** volume group using the **`cinder`** command line interface.

Procedure 7.2. Creating a Volume using the Command Line Interface

1. Use the **`keystonerc_admin`** file to authenticate with Keystone.

   ```
   $ source ~/keystonerc_admin
   ```

2. Use the **`cinder create`** command to create a new volume.

   ```
   $ cinder create --display_name NAME SIZE
   ```

   Replace ***NAME*** with a name to identify your new volume and ***SIZE*** with the desired size for the new volume in gigabytes (GB).

You have successfully created a Cinder volume using the command line interface.

7.4. Attaching a Volume

This procedure details the steps involved in attaching a Cinder volume to an existing compute instance using the **`cinder`** and **`nova`** command line interfaces.

Procedure 7.3. Attaching a Volume using the Command Line Interface

1. Use the **`keystonerc_admin`** file to authenticate with Keystone.

   ```
   $ source ~/keystonerc_admin
   ```

2. Use the **`cinder list`** command to find available volumes.

   ```
   $ cinder list
   +------------------------------------+---------+------------+----+---------
   --+
   |                 ID                 | Status  |Display Name|Size|Volume
   Type|
   +------------------------------------+---------+------------+----+---------
   --+
   |15a9f901-ba9d-45e1-8622-a5438473ae76|available|    NAME    | 1  |
   |
   +------------------------------------+---------+------------+----+---------
   --+
   ```

   Take note of the **`ID`** of the volume you wish to use. You will need it when attaching the volume to an instance.

   Note

   The **`Attached to`** column has been intentionally omitted from this example output.

3. Use the **`nova list`** command to find running instances.

```
$ nova list
+--------------------------------------+------+--------+-------------------
---+
|                  ID                  | Name | Status |      Networks
|
+--------------------------------------+------+--------+-------------------
---+
| 6842461c-973d-f91b-170a-07324034fbb9 | NAME | ACTIVE |
private=192.0.43.10  |
+--------------------------------------+------+--------+-------------------
---+
```

Take note of the **ID** of the instance you wish to use. You will need it when attaching the volume.

4. Use the **nova *volume-attach*** command to attach the volume to the instance. Replace ***INSTANCE_ID*** with the identifier of the instance and replace ***VOLUME_ID*** with the identifier of the volume.

```
$ nova volume-attach INSTANCE_ID VOLUME_ID auto
```

The ***auto*** parameter indicates that Nova must attempt to automatically assign a device identifier to the volume within the guest. Manual allocation of specific device identifiers within the guest is not supported on KVM hypervisors at this time.

You have successfully attached a Cinder volume to an instance using the command line interface. The volume will appear as a physical hard disk drive to the guest operating system.

7.5. Accessing a Volume from a Running Instance

Once you attach a volume to an instance a new device will appear to the guest operating system. The device is accessible both using a traditional device label such as **/dev/vdc** and in the **/dev/disk/by-id/** tree.

Volumes appear in **/dev/disk/by-id/** with identifiers of the form **virtio-*ID*** where ***ID*** is a subset of the volume identifier assigned when the volume was defined in Cinder.

For example a disk with the identifier **15a9f901-ba9d-45e1-8622-a5438473ae76** in Cinder appears as **/dev/disk/by-id/virtio-15a9f901-ba9d-45e1-8** when viewed from within a virtual machine instance that it is attached to.

```
vm$ ls /dev/disk/by-id/
virtio-15a9f901-ba9d-45e1-8
```

Create a filesystem on the device and mount it in the virtual machine:

```
vm$ mkfs.ext4 /dev/disk/by-id/virtio-15a9f901-ba9d-45e1-8
vm$ mkdir -p /mnt/volume
vm$ mount /dev/disk/by-id/virtio-15a9f901-ba9d-45e1-8 /mnt/volume
```

Write some data to the mounted volume:

```
vm$ echo "Red Hat OpenStack" > /mnt/volume/test.txt
```

Unmount the volume inside the virtual machine.

```
vm$ umount /mnt/volume
```

From the host running Nova, detach the volume from the instance. The **volume-detach** command requires an instance ID and the volume ID you would like to detach from that instance:

```
$ nova volume-detach <instanceid> <volumeid>
```

To verify that the data written to the volume has persisted, you can start up a new instance. Once the new instance is in the **ACTIVE** state, attach the volume to that instance, and then mount the volume in the instance:

```
$ nova boot --image <imageid> --flavor 2 --key_name oskey rhel2
+------------------------+--------------------------------------+
|        Property        |                Value                 |
+------------------------+--------------------------------------+
| OS-DCF:diskConfig      | MANUAL                               |
| OS-EXT-STS:power_state | 0                                    |
| OS-EXT-STS:task_state  | scheduling                           |
| OS-EXT-STS:vm_state    | building                             |
| accessIPv4             |                                      |
| accessIPv6             |                                      |
| adminPass              | uPnzQhpdZZf9                         |
| config_drive           |                                      |
| created                | 2012-05-18T13:45:56Z                 |
| flavor                 | m1.small                             |
| hostId                 |                                      |
| id                     | b8d5c952-f2fc-4556-83f2-57c79378d867 |
| image                  | RHEL 6.2                             |
| key_name               | oskey                                |
| metadata               | {}                                   |
| name                   | rhel2                                |
| progress               | 0                                    |
| status                 | BUILD                                |
| tenant_id              | 05816b0106994f95a83b913d4ff995eb     |
| updated                | 2012-05-18T13:45:56Z                 |
| user_id                | 1d59c0bfef9b4ea9ab63e2a058e68ae0     |
+------------------------+--------------------------------------+
$ nova list
+--------------------------------------+---------+--------+------------------+
|                  ID                  |   Name  | Status |     Networks     |
+--------------------------------------+---------+--------+------------------+
| 0e4011a4-3128-4674-ab16-dd1b7ecc126e | rhel    | ACTIVE | demonet=10.0.0.2 |
| b8d5c952-f2fc-4556-83f2-57c79378d867 | rhel2   | BUILD  | demonet=10.0.0.3 |
+--------------------------------------+---------+--------+------------------+
$ nova list
+--------------------------------------+---------+--------+------------------+
|                  ID                  |   Name  | Status |     Networks     |
+--------------------------------------+---------+--------+------------------+
| 0e4011a4-3128-4674-ab16-dd1b7ecc126e | rhel    | ACTIVE | demonet=10.0.0.2 |
| b8d5c952-f2fc-4556-83f2-57c79378d867 | rhel2   | ACTIVE | demonet=10.0.0.3 |
+--------------------------------------+---------+--------+------------------+
$ nova volume-attach b8d5c952-f2fc-4556-83f2-57c79378d867 15a9f901-ba9d-45e1-
8622-a5438473ae76 auto
$ ssh -i oskey.priv root@10.0.0.3
```

```
vm2$ mkdir -p /mnt/volume
vm2$ mount /dev/disk/by-id/virtio-15a9f901-ba9d-45e1-8 /mnt/volume
vm2$ cat /mnt/volume/test.txt
Red Hat OpenStack
vm2$ umount /mnt/volume
```

Now detach the volume, where the first id is the instance id (Nova) and the second id is the volume id (Cinder):

```
$ nova volume-detach b8d5c952-f2fc-4556-83f2-57c79378d867 15a9f901-ba9d-45e1-
8622-a5438473ae76
```

7.6. Creating a Snapshot

It is possible to create a snapshot of a running instance. This may be done for backup purposes or for creating a base image to create other instances from after applying some customization to it.

As an example, you may want every instance to have a user called **`projectuser`**. Create that user in the virtual machine and then create a snapshot. That snapshot can be used as the base for new instances.

Start by applying some sort of customization to the virtual machine. These commands could be used to create a user and set its password:

```
vm$ useradd projectuser
vm$ passwd projectuser
```

Now create a snapshot of that running instance:

```
$ nova image-create <instanceid> "snapshot 1"
```

The snapshot is complete when its status in **`nova image-list`** changes from **`SAVING`** to **`ACTIVE`**.

```
$ nova image-list
+--------------------------------------+------------+--------+--------+
|                  ID                  |    Name    | Status | Server |
+--------------------------------------+------------+--------+--------+
| 17a34b8e-c573-48d6-920c-b4b450172b41 | RHEL 6.2   | ACTIVE |        |
| 84420f08-1e4b-499a-837a-5c6c1b9903d0 | snapshot 1 | SAVING | ...... |
+--------------------------------------+------------+--------+--------+
$ nova image-list
+--------------------------------------+------------+--------+--------+
|                  ID                  |    Name    | Status | Server |
+--------------------------------------+------------+--------+--------+
| 17a34b8e-c573-48d6-920c-b4b450172b41 | RHEL 6.2   | ACTIVE |        |
| 84420f08-1e4b-499a-837a-5c6c1b9903d0 | snapshot 1 | ACTIVE | ...... |
+--------------------------------------+------------+--------+--------+
```

Once the snapshot's status is active, you can start up a new instance using this image:

```
$ nova boot --image 84420f08-1e4b-499a-837a-5c6c1b9903d0 --flavor 2 --key_name
oskey \
  snapshot-instance
+-------------------------+--------------------------------------+
|        Property         |                Value                 |
+-------------------------+--------------------------------------+
| OS-DCF:diskConfig       | MANUAL                               |
| OS-EXT-STS:power_state  | 0                                    |
| OS-EXT-STS:task_state   | scheduling                           |
| OS-EXT-STS:vm_state     | building                             |
| accessIPv4              |                                      |
| accessIPv6              |                                      |
| adminPass               | QASX3r8jKzVd                         |
| config_drive            |                                      |
| created                 | 2012-05-18T13:49:07Z                 |
| flavor                  | m1.small                             |
| hostId                  |                                      |
| id                      | ac9e6a9f-58c3-47c3-9b4c-485aa421b8a8 |
| image                   | snapshot 1                           |
| key_name                | oskey                                |
| metadata                | {}                                   |
| name                    | snapshot-instance                    |
| progress                | 0                                    |
| status                  | BUILD                                |
| tenant_id               | 05816b0106994f95a83b913d4ff995eb     |
| updated                 | 2012-05-18T13:49:08Z                 |
| user_id                 | 1d59c0bfef9b4ea9ab63e2a058e68ae0     |
+-------------------------+--------------------------------------+
 $ nova list
+--------------------------------------+-------------------+--------+----------
-------+
|                  ID                  |        Name       | Status |
Networks     |
+--------------------------------------+-------------------+--------+----------
-------+
| 0e4011a4-3128-4674-ab16-dd1b7ecc126e | rhel              | ACTIVE |
demonet=10.0.0.2 |
| ac9e6a9f-58c3-47c3-9b4c-485aa421b8a8 | snapshot-instance | BUILD  |
demonet=10.0.0.4 |
| b8d5c952-f2fc-4556-83f2-57c79378d867 | rhel2             | ACTIVE |
demonet=10.0.0.3 |
+--------------------------------------+-------------------+--------+----------
-------+
```

Finally, test that the new instance contains the expected customizations made earlier in this exercise. If you followed the example, the instance should have a user named **projectuser**.

```
$ ssh -i oskey.priv root@10.0.0.4
```

```
vm$ su projectuser
```

7.7. Working with Nova Networking

This section describes the procedures for 'Adding a Rule to Security Group' and 'Adding Floating IP Addresses' using Nova Networking.

Note

If you have opted for OpenStack Networking by setting the **`CONFIG_QUANTUM_INSTALL`** to **y**, then refer to Section 7.8, “Working with OpenStack Networking”.

7.7.1. Adding a Rule to a Security Group

The **`nova`** command line interface provides facilities for adding rules to security groups.

Procedure 7.4. Adding a Rule to a Security Group using the Command Line Interface

1. Use the **`nova secgroup-list`** command to list the security groups that have been defined.

   ```
   $ nova secgroup-list
   +---------+-------------+
   |   Name  | Description |
   +---------+-------------+
   | default | default     |
   +---------+-------------+
   ```

   On an installation where no security groups have been created yet only the **`default`** security group will be defined.

2. Use the **`nova secgroup-add-rule`** command to add a new rule to a security group. The syntax of the **`nova secgroup-add-rule`** command is:

   ```
   $ nova secgroup-add-rule GROUP \
          PROTOCOL \
          FROM \
          TO \
          CIDR
   ```

   The arguments that the **`nova secgroup-add-rule`** command expects represent:

   - Replace ***`GROUP`*** with the identifier of the security group to add the rule to.
   - Replace ***`PROTOCOL`*** with the IP protocol that the group applies to. Valid values are **`icmp`**, **`tcp`**, and **`udp`**.
   - Replace ***`FROM`*** with the port that defines the start of the range of ports to allow network traffic on. Valid values are in the range -1 to 65535 for TCP and UDP, -1 to 255 for ICMP.
   - Replace ***`TO`*** with the port that defines the end of the range of ports to allow network traffic on. Valid values are in the range -1 to 65535 for TCP and UDP, -1 to 255 for ICMP.
   - Replace ***`CIDR`*** with the Classless Inter-Domain Routing (CIDR) notation defining the IP addresses to accept connections from. A value of **`0.0.0.0/0`** allows connections from any IP address.

   Note

   A port range of **`-1`** to **`-1`** is taken to mean that all valid ports are included.

3. Use the **`nova secgroup-list-rules`** command to verify that your new rule has been added to the selected security group.

```
$ nova secgroup-list-rules GROUP
```

Replace ***GROUP*** with the identifier of the security group that you added the rule to.

You have successfully added a rule to a security group using the command line interface. It is now possible to connect to instances that use the altered security group from the specified IP address block and using the specified ports and protocol.

> **Example 7.3. Adding a Security Rule to Allow SSH Connections**
>
> In this example a rule is added to the **default** security group to allow SSH access from machines in the IP address block **172.31.0.224/28**.
>
> ```
> $ nova secgroup-add-rule default tcp 22 22 172.31.0.224/28
> +-------------+-----------+---------+-----------------+--------------+
> | IP Protocol | From Port | To Port |     IP Range    | Source Group |
> +-------------+-----------+---------+-----------------+--------------+
> | tcp         | 22        | 22      | 172.31.0.224/28 |              |
> +-------------+-----------+---------+-----------------+--------------+
> ```

7.7.2. Adding Floating IP Addresses

This procedure details the definition of a pool of floating IP addresses. It also covers the reservation of a floating IP address from the pool and the association of that address with a specific compute instance.

Procedure 7.5. Adding Floating IP Addresses using the Command Line Interface

1. Use the **nova-manage floating create** command to define a pool of floating IP addresses.

   ```
   $ nova-manage floating create IP_BLOCK
   ```

   Replace ***IP_BLOCK*** with the block of IP addresses to use. This value is expressed using CIDR notation.

   > **Example 7.4. Defining a Pool of Floating IP Addresses**
   >
   > ```
   > $ nova-manage floating create 172.31.0.224/28
   > ```

2. Use the **nova floating-ip-create** command to reserve a floating IP address from the available blocks of public IP addresses.

   ```
   $ nova floating-ip-create
   +--------------+-------------+----------+------+
   |      Ip      | Instance Id | Fixed Ip | Pool |
   +--------------+-------------+----------+------+
   | 172.31.0.225 |             |          | nova |
   +--------------+-------------+----------+------+
   ```

3. Use the **nova list** command to identify running instances and select an instance to assign the floating IP address to.

```
$ nova list
+--------------------------------------+------------------+--------+------
------------+
|                  ID                  |       Name       | Status |
Networks     |
+--------------------------------------+------------------+--------+------
------------+
| 0e4011a4-3128-4674-ab16-dd1b7ecc126e | rhel             | ACTIVE |
demonet=10.0.0.1 |
+--------------------------------------+------------------+--------+------
------------+
```

4. Use the **`nova add-floating-ip`** command to assign the floating IP address that reserved in an earlier step to the selected instance.

   ```
   $ nova add-floating-ip INSTANCE IP
   ```

   Replace ***INSTANCE*** with the identifier of the selected instance and replace ***IP*** with the floating IP address being assigned to it.

   Example 7.5. Assigning a Floating IP Address to an Instance

   ```
   $ nova add-floating-ip 0e4011a4-3128-4674-ab16-dd1b7ecc126e
   172.31.0.225
   ```

5. Periodically check the output of the **`nova list`** command until the floating IP address appears in the output for the selected instance. Once this occurs the instance is accessible using the floating IP address.

   Note

   To disassociate a floating IP address from an instance when it is no longer required, use the **`nova remove-floating-ip`** command.

   ```
   $ nova remove-floating-ip INSTANCE IP
   ```

   Replace ***INSTANCE*** with the identifier of the instance and replace ***IP*** with the floating IP address to remove from it.

You have successfully associated a floating IP address with an instance using the command line interface.

7.8. Working with OpenStack Networking

This section describes the different procedures using OpenStack Networking.

7.8.1. Creating a Network

This section describes the steps to be followed for creating a network.

Procedure 7.6. Creating a Network Using the Command Line Interface

1. Use the **`source`** command to load the credentials of the administrative user.

```
$ source ~/keystonerc_admin
```

2. Use the **net-create** action of the **quantum** command line client to create a new provider network.

   ```
   $ quantum net-create EXTERNAL_NAME \
     --router:external True \
     --provider:network_type TYPE \
     --provider:physical_network PHYSICAL_NAME \
     --provider:segmentation_id VLAN_TAG
   ```

   Replace these strings with the appropriate values for your environment:

   - Replace ***EXTERNAL_NAME*** with a name for the new external network provider.
   - Replace ***PHYSICAL_NAME*** with a name for the physical network. This is not applicable if you intend to use a local network type.
   - Replace ***TYPE*** with the type of provider network you wish to use. Supported values are **flat** (for flat networks), **vlan** (for VLAN networks), and **local** (for local networks).
   - Replace ***VLAN_TAG*** with the VLAN tag that will be used to identify network traffic. The VLAN tag specified must have been defined by the network administrator.

     If the **network_type** was set to a value other than **vlan** then this parameter is not required.

   Take note of the unique external network identifier returned, this will be required in subsequent steps.

3. Use the **subnet-create** action of the command line client to create a new subnet for the new external provider network.

   ```
   $ quantum subnet-create --gateway GATEWAY \
     --allocation-pool start=IP_RANGE_START,end=IP_RANGE_END \
     --disable-dhcp EXTERNAL_NAME EXTERNAL_CIDR
   ```

   Replace these strings with the appropriate values for your environment:

   - Replace ***GATEWAY*** with the IP address or hostname of the system that is to act as the gateway for the new subnet.
   - Replace ***IP_RANGE_START*** with the IP address that denotes the start of the range of IP addresses within the new subnet that floating IP addresses will be allocated from.
   - Replace ***IP_RANGE_END*** with the IP address that denotes the end of the range of IP addresses within the new subnet that floating IP addresses will be allocated from.
   - Replace ***EXTERNAL_NAME*** with the name of the external network the subnet is to be associated with. This must match the name that was provided to the **net-create** action in the previous step.
   - Replace ***EXTERNAL_CIDR*** with the Classless Inter-Domain Routing (CIDR) representation of the block of IP addresses the subnet represents. An example would be **192.168.100.0/24**.

   Take note of the unique subnet identifier returned, this will be required in subsequent steps.

Important

The IP address used to replace the string ***GATEWAY*** **must** be within the block of IP addresses specified in place of the ***EXTERNAL_CIDR*** string but outside of the block of IP addresses specified by the range started by ***IP_RANGE_START*** and ended by ***IP_RANGE_END***.

The block of IP addresses specifed by the range started by ***IP_RANGE_START*** and ended by ***IP_RANGE_END*** **must** also fall within the block of IP addresses specified by ***EXTERNAL_CIDR***.

7.8.2. Creating a Router

This procedure describes the steps for creating a router.

Procedure 7.7. Creating a Router using the Command Line Interface

1. Use the **router-create** action of the **quantum** command line client to create a new router.

   ```
   $ quantum router-create NAME
   ```

   Replace ***NAME*** with the name to give the new router. Take note of the unique router identifier returned, this will be required in subsequent steps.

2. Use the **router-gateway-set** action of the **quantum** command line client to link the newly created router to the external provider network.

   ```
   $ quantum router-gateway-set ROUTER NETWORK
   ```

   Replace ***ROUTER*** with the unique identifier of the router, replace ***NETWORK*** with the unique identifier of the external provider network.

3. Use the **router-interface-add** action of the **quantum** command line client to link the newly created router to the subnet.

   ```
   $ quantum router-interface-add ROUTER SUBNET
   ```

   Replace ***ROUTER*** with the unique identifier of the router, replace ***SUBNET*** with the unique identifier of the subnet.

An external provider network has been created. Use the unique identifier of the router when configuring the L3 agent.

7.8.3. Adding a Rule to a Security Group

This procedure describes the procedure for adding a rule to the security group.

Procedure 7.8. Adding a Rule to a Security Group Using the Command Line Interface

- You can configure security group rules directly by using **quantum security-group-rule-create** command to enable access to your VMs.

```
# quantum security-group-rule-create --protocol PROTOCOL --direction
DIRECTION --port_range_min MAX_PORT --port_range_max MIN_PORT
SECURITY_GROUP_ID
```

7.8.4. Defining a Floating IP-Address Pool

By default, each virtual instance is automatically assigned a private IP address in the network to which it is assigned. You may optionally assign public IP addresses to instances.

OpenStack uses the term "floating IP" to refer to an IP address that can be dynamically added to a running virtual instance. In OpenStack Networking, a floating IP pool is represented as an external network and a floating IP is allocated from a subnet associated with the external network.

For this procedure, you must first have installed OpenStack Networking.

To define a pool of floating IP addresses:

Procedure 7.9. Defining a Floating IP-Address Pool Using the Command Line INterface

1. Create an external network for the pool:

```
# quantum net-create networkName --router:external=True
```

Example 7.6. Defining an External Network

```
 # quantum net-create ext-net --router:external=True
Created a new network:
+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | True                                 |
| id                        | 3a53e3be-bd0e-4c05-880d-2b11aa618aff |
| name                      | ext-net                              |
| provider:network_type     | local                                |
| provider:physical_network |                                      |
| provider:segmentation_id  |                                      |
| router:external           | True                                 |
| shared                    | False                                |
| status                    | ACTIVE                               |
| subnets                   |                                      |
| tenant_id                 | 6b406408dff14a2ebf6d2cd7418510b2     |
+---------------------------+--------------------------------------+
```

2. Create the pool of floating IP addresses:

```
$ quantum subnet-create  --allocation-pool start=IPStart,end=IPStart --
gateway GatewayIP --disable-dhcp networkName CIDR
```

Example 7.7. Defining a Pool of Floating IP Addresses

```
$ quantum subnet-create --allocation-pool
start=10.38.15.128,end=10.38.15.159 --gateway 10.38.15.254 --disable-dhcp
ext-net 10.38.15.0/24
Created a new subnet:
+------------------+--------------------------------------------------+
| Field            | Value                                            |
+------------------+--------------------------------------------------+
| allocation_pools | {"start": "10.38.15.128", "end": "10.38.15.159"} |
| cidr             | 10.38.15.0/24                                    |
| dns_nameservers  |                                                  |
| enable_dhcp      | False                                            |
| gateway_ip       | 10.38.15.254                                     |
| host_routes      |                                                  |
| id               | 6a15f954-935c-490f-a1ab-c2a1c1b1529d             |
| ip_version       | 4                                                |
| name             |                                                  |
| network_id       | 4ad5e73b-c575-4e32-b581-f9207a22eb09             |
| tenant_id        | e5be83dc0a474eeb92ad2cda4a5b94d5                 |
+------------------+--------------------------------------------------+
```

You have successfully created a pool of floating IP addresses.

7.8.5. Associating the Floating IP Addresses

This procedure describes the procedure for associating floating point IP addresses.

Procedure 7.10. Associating the Floating IP Address using the Command Line Interface

1. Retreive the pool id for the VM.

   After a VM is deployed a floating IP address can be associated to the VM. A VM that is created will be allocated an OpenStack Networking port ($PORT_ID).

   ```
   # nova list
   +--------------------------------------+--------+--------+---------------+
   |                  ID                  |  Name  | Status |    Networks   |
   +--------------------------------------+--------+--------+---------------+
   |              DEVICE_ID               | testvm | ACTIVE |    net1=IP    |
   +--------------------------------------+--------+--------+---------------+
   ```

   ```
   # quantum port-list -- --device_id DEVICE_ID
   +----------+--------+-----------------+-----------------------------------
   ------------+
   | ID | Name | MAC_Address       |                         Networks
   |
   +--------------------------------------+--------+--------+----------------
   ------------+
   | ID |       | fa:16:3e:b4:d6:6c| {"subnet_id": "SUBNET_ID", "ip_address":
   "IP_ADDRESS"}|
   +--------------------------------------+--------+--------+----------------
   ------------+
   ```

2. Allocate a floating IP

```
# quantum floatingip-create ext_net
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| fixed_ip_address    |                                      |
| floating_ip_address | 7.7.7.131                            |
| floating_network_id | 8858732b-0400-41f6-8e5c-25590e67ffeb |
| id                  | 40952c83-2541-4d0c-b58e-812c835079a5 |
| port_id             |                                      |
| router_id           |                                      |
| tenant_id           | e40fa60181524f9f9ee7aa1038748f08     |
+---------------------+--------------------------------------+
```

3. Associate a floating IP to a VM

```
# quantum floatingip-associate FLOATING_ID PORT_ID
```

4. Show the floating IP

```
# quantum floatingip-show FLOATING_ID
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| fixed_ip_address    | 10.5.5.3                             |
| floating_ip_address | 7.7.7.131                            |
| floating_network_id | 8858732b-0400-41f6-8e5c-25590e67ffeb |
| id                  | 40952c83-2541-4d0c-b58e-812c835079a5 |
| port_id             | 9aa47099-b87b-488c-8c1d-32f993626a30 |
| router_id           | 685f64e7-a020-4fdf-a8ad-e41194ae124b |
| tenant_id           | e40fa60181524f9f9ee7aa1038748f08     |
+---------------------+--------------------------------------+
```

7.9. Controlling Instance State (Suspend, Resume, Reboot, Terminate)

Up to this point you have only booted up instances. There are some other commands that can be used to adjust instance state. You can suspend, resume, reboot, and terminate an instance. The following commands show some examples of doing a suspend, resume, and reboot. Terminating instances is covered in Section 7.10, “Deleting Instances”.

```
$ nova list
+--------------------------------------+-------------------+--------+-----------
-------+
|                  ID                  |        Name       | Status |
Networks     |
+--------------------------------------+-------------------+--------+-----------
-------+
| 0e4011a4-3128-4674-ab16-dd1b7ecc126e | rhel              | ACTIVE |
demonet=10.0.0.2 |
| ac9e6a9f-58c3-47c3-9b4c-485aa421b8a8 | snapshot-instance | ACTIVE |
demonet=10.0.0.4 |
| b8d5c952-f2fc-4556-83f2-57c79378d867 | rhel2             | ACTIVE |
demonet=10.0.0.3 |
+--------------------------------------+-------------------+--------+-----------
-------+
$ nova suspend ac9e6a9f-58c3-47c3-9b4c-485aa421b8a8
$ ping 10.0.0.4   # should not get a response
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
Ctrl+c
--- 10.0.0.4 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2879ms
$ nova resume ac9e6a9f-58c3-47c3-9b4c-485aa421b8a8
$ ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=1685 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=685 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.451 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.394 ms
Ctrl+c
--- 10.0.0.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3607ms
$ nova reboot ac9e6a9f-58c3-47c3-9b4c-485aa421b8a8
$ ssh -i oskey.priv root@10.0.0.4
Last login: Fri May 18 09:50:38 2012 from 10.0.0.1
vm$ uptime
 09:59:09 up 0 min,  1 user,  load average: 0.15, 0.03, 0.01
```

7.10. Deleting Instances

A running instance can be deleted using **`nova delete`**. The following example shows how to delete all running instances:

```
$ nova list
+--------------------------------------+-------------------+--------+-----------
-------+
|                  ID                  |        Name       | Status |
Networks     |
+--------------------------------------+-------------------+--------+-----------
-------+
| 0e4011a4-3128-4674-ab16-dd1b7ecc126e | rhel              | ACTIVE |
demonet=10.0.0.2 |
| ac9e6a9f-58c3-47c3-9b4c-485aa421b8a8 | snapshot-instance | ACTIVE |
demonet=10.0.0.4 |
| b8d5c952-f2fc-4556-83f2-57c79378d867 | rhel2             | ACTIVE |
demonet=10.0.0.3 |
+--------------------------------------+-------------------+--------+-----------
-------+
$ nova delete 0e4011a4-3128-4674-ab16-dd1b7ecc126e
$ nova delete ac9e6a9f-58c3-47c3-9b4c-485aa421b8a8
$ nova delete b8d5c952-f2fc-4556-83f2-57c79378d867
$ nova list
+----+------+--------+----------+
| ID | Name | Status | Networks |
+----+------+--------+----------+
+----+------+--------+----------+
```

Part IV. Monitoring OpenStack PackStack Deployments

There are two types of monitoring for OpenStack.

- Process Monitoring: A basic type of alert monitoring to check and see if a required process is running.
- Resource Alerting: Provides notifications when one or more resources are critically low. While the monitoring thresholds should be tuned to your specific OpenStack environment, monitoring resource usage is not specific to OpenStack and any generic type of alert will work fine.

  Some of the resources that you may want to monitor include:

  - Disk Usage
  - Server Load
  - Memory Usage
  - Network IO
  - Available vCPUs

Chapter 8. Monitoring OpenStack Using Nagios

Nagios is a system, network, and infrastructure monitoring software application. Nagios offers monitoring and alerting for servers, switches, applications, and services. It alerts users when things go wrong and alerts them again when the problem has been resolved.

> **Note**
>
> By default, PackStack does not install Nagios. To install Nagios, you must set the **`CONFIG_NAGIOS_INSTALL`** parameter to **`y`** in the answer file or during the interactive set up.

8.1. Accessing the Nagios Dashboard

To access the dashboard you must have first:

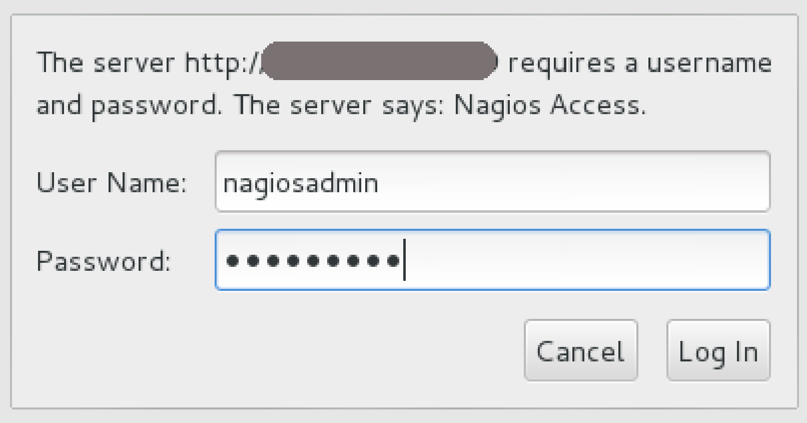
- Installed the Nagios dashboard.

Procedure 8.1. Accessing the Nagios Dashboard

- **Log In**

  In your browser, open the link for your configuration to access the dashboard for the first time.

  ```
  http://HOSTNAME/nagios/
  ```

  The server http:/[redacted] requires a username and password. The server says: Nagios Access.

  User Name: nagiosadmin

  Password: ●●●●●●●●●

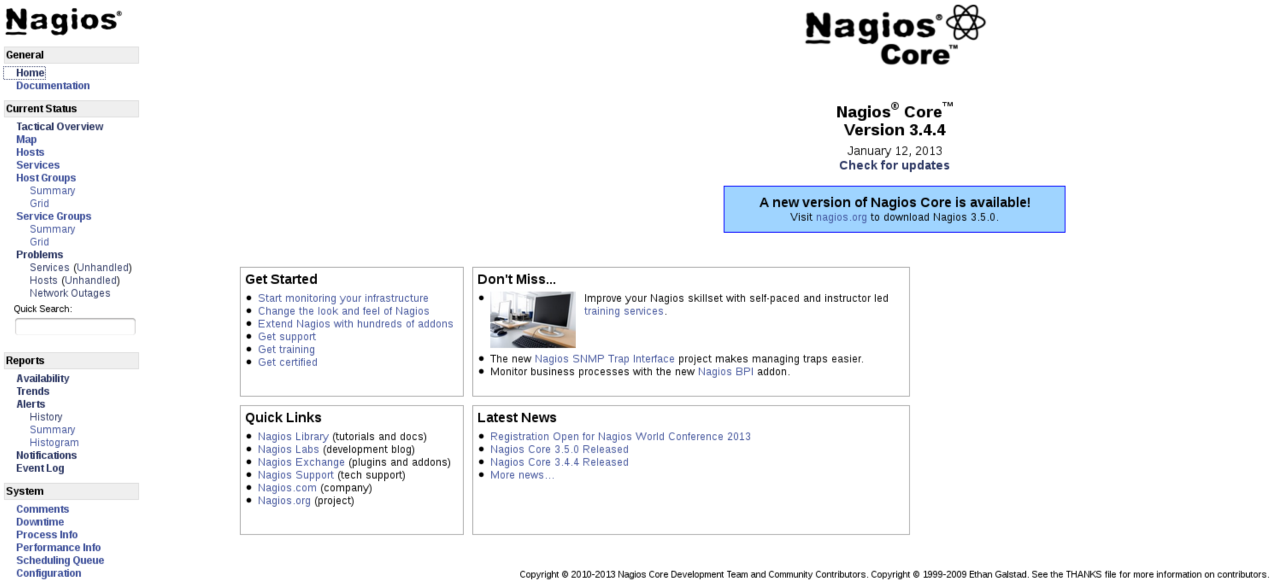
  Figure 8.1. Log In Screen

  Replace **`HOSTNAME`** with the host name or IP address of the server on which you installed the dashboard. The user name and password are available in the 'Additional information:' as shown below:

```
**** Installation completed successfully ******

Additional information:
 * A new answerfile was created in: /root/packstack-answers-x.txt
 * To use the command line tools you need to source the file
/root/keystonerc_admin created on 192.0.43.10
 * To use the console, browse to http://192.0.43.10/dashboard
 * To use Nagios, browse to http://192.0.43.10/nagios username: nagiosadmin,
password : abcdefgh12345678
 * The installation log file is available at: /var/tmp/packstack/x-y/openstack-
setup.log
```

Once you are logged in to Nagios, you can see the browser setup with a left panel that displays different options to monitor the OpenStack service running on your system.



Nagios® Core™
Version 3.4.4
January 12, 2013
Check for updates

A new version of Nagios Core is available!
Visit nagios.org to download Nagios 3.5.0.

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

Don't Miss...

- Improve your Nagios skillset with self-paced and instructor led training services.
- The new Nagios SNMP Trap Interface project makes managing traps easier.
- Monitor business processes with the new Nagios BPI addon.

Quick Links

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Latest News

- Registration Open for Nagios World Conference 2013
- Nagios Core 3.5.0 Released
- Nagios Core 3.4.4 Released
- More news...

Copyright © 2010-2013 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Figure 8.2. Nagios Home

8.2. Default Nagios Configuration

The configuration file for Nagios is available at **`/etc/nagios/nagios.cfg`**. You can modify the parameters by changing the default values in the configuration file.

Table 8.1, “Nagios Configuration Parameters” lists the default values for the configuration parameters with OpenStack PackStack deployment.

Note

The default values that are common to Nagios configuration without PackStack are not listed below.
For more details on Nagios configuration parameters, refer to the Red Hat Enterprise Linux OpenStack Platform *Installation and Configuration Guide*.

Table 8.1. Nagios Configuration Parameters

Configuration Parameter	Default Value/Location	Description
Nagios Log File	`/var/log/nagios/nagios.log`	Log file that contains all the services and host events.
Object Configuration Files	`/etc/nagios/objects/*.cfg`	Configuration files in which you can define hosts, host groups, contacts, contact groups, services, etc.
Object Cache File	`/var/log/nagios/objects.cache`	Location of cached object definitions when Nagios starts/restarts.
Pre-Cached File	`/var/log/nagios/objects.precache`	Location of the precached object file.
Temporary File	`/var/log/nagios/nagios.tmp`	Temporary file used as scratch space when Nagios updates the status log, cleans the comment file, etc. This file is created, used, and deleted throughout the time that Nagios is running.
Status File	`/var/log/nagios/status.dat`	File which stores the location of the current status of all monitored services and hosts. Its contents are read and processed by the CGIs. The contents of the status file are deleted every time Nagios restarts.
Resource File	`/etc/nagios/private/resource.cfg`	Optional resource file that contains $USERx$ macro definitions. Multiple resource files can be specified by using multiple resource_file definitions.
Status File Update Interval	`status_update_interval=10`	Determines the frequency (in seconds) that Nagios will periodically dump program, host, and service status data.
External Command File	`/var/spool/nagios/cmd/nagios.cmd`	File that Nagios checks for external command requests.
External Command Buffer Slots	`external_command_buffer_slots=4096`	Parameter that can be tweaked so that the Nagios daemon can allocate the number of items or "slots" to the buffer that holds incoming external commands before they are processed. As external commands are processed by the daemon, they are removed from the buffer.
Lock File	`/var/run/nagios.pid`	Lock file that Nagios uses to store its PID number in when it

		is running in daemon mode.
Log Archive Path	**`/var/log/nagios/archives`**	Directory where archived log files are saved.
Initial Logging States Option	**`log_initial_states=0`**	If you want Nagios to log all initial host and service states to the main log file (the first time the service or host is checked) you can enable this option by setting this value to 1. If you are not using an external application that does long term state statistics reporting, you do not need to enable this option. In this case, set the value to 0.
Maximum Concurrent Service Checks	**`max_concurrent_checks=0`**	Allows you to specify the maximum number of service checks that can be run in parallel at any given time. Specifying a value of 1 for this variable essentially prevents any service checks from being parallelized. A value of 0 will not restrict the number of concurrent checks that are being executed.
Host and Service Check Reaper Frequency	**`check_result_reaper_frequency=10`**	Frequency (in seconds) that Nagios will process the results of host and service checks.
Check Result Path	**`/var/log/nagios/spool/checkresults`**	Directory where Nagios stores the results of host and service checks that have not yet been processed. **Note** Make sure that only one instance of Nagios has access to this directory.
Time Change Adjustment Thresholds	**`time_change_threshold=900`**	Determines when Nagios will react to detected changes in system time (either forward or backwards).
Auto-Rescheduling Option	**`auto_reschedule_checks=0`**	Determines whether or not Nagios will attempt to automatically reschedule active host and service checks to "smooth" them out over time. This can help balance the load on the monitoring server.
Sleep Time	**`sleep_time=0.25`**	Time (in seconds) to sleep

		between checking for system events and service checks that need to be run.
Timeout Values	**service_check_timeout=60**, **host_check_timeout=30**, **event_handler_timeout=30**, **notification_timeout=30**, **ocsp_timeout=5**, **perfdata_timeout=5**	Option to control how much time Nagios will allow various types of commands to execute before killing them off. Options are available for controlling maximum time allotted for service checks, host checks, event handlers, notifications, the ocsp command, and performance data commands. All values are in seconds.
State Retention File	**/var/log/nagios/retention.dat**	File that Nagios uses to store host and service state information before it shuts down. The state information in this file is read prior to monitoring the network when Nagios is restarted. This file is used only if the **retain_state_information** variable is set to 1.
Process Performance Data Option	**process_performance_data=0**	Determines whether or not Nagios will process performance data returned from service and host checks. If this option is enabled, host performance data will be processed using the **host_perfdata_command** and service performance data will be processed using the **service_perfdata_command**. Values: 1 = process performance data, 0 = do not process performance data.
Host and Service Performance Data Files	**/tmp/host-perfdata**, **/tmp/service-perfdata**	Files used to store host and service performance data. Performance data is only written to these files if the **enable_performance_data** option is set to 1.
Host and Service Performance Data Process Empty Results	**host_perfdata_process_empty_results=1**, **service_perfdata_process_empty_results=1**	Determine whether the core will process empty performance data results or not. This is needed for distributed monitoring, and intentionally turned on by default. Values: 1 = enable, 0 = disable.
Obsess Over Service Checks	**obsess_over_services=0**	Determines whether Nagios

Option		runs the predefined **`ocsp_command`** command after every service check (that is, whether Nagios *obsesses* over these services). Unless you are planning on implementing distributed monitoring, do not enable this option. Values: 1 = obsess over services, 0 = do not obsess (default).
Obsess Over Host Checks Option	**`obsess_over_hosts=0`**	Determines whether Nagios runs the predefined **`ocsp_command`** command after every host check (that is, whether Nagios *obsesses* over these hosts). Unless you are planning on implementing distributed monitoring, do not enable this option. Values: 1 = obsess over hosts, 0 = do not obsess (default).
Translate Passive Host Checks Option	**`translate_passive_host_checks=0`**	Determines whether or not Nagios will translate DOWN/UNREACHABLE passive host check results into their proper state for this instance of Nagios. Values: 1 = perform translation, 0 = do not translate (default).
Passive Host Checks Are SOFT Option	**`passive_host_checks_are_soft=0`**	Determines whether or not Nagios will treat passive host checks as being HARD or SOFT. By default, a passive host check result will put a host into a HARD state type. This can be changed by enabling this option. Values: 0 = passive checks are HARD, 1 = passive checks are SOFT.
Service Freshness Check Option	**`check_service_freshness=1`**	Determines whether or not Nagios will periodically check the "freshness" of service results. Enabling this option is useful for ensuring passive checks are received in a timely manner. Values: 1 = enabled freshness checking, 0 = disable freshness checking.
Service Check Timeout State	**`service_check_timeout_state=c`**	Determines the state Nagios will report when a service check times out, that is when a service does not respond within

		`service_check_timeout` seconds. This can be useful if a machine is running at too high a load and you do not want to consider a failed service check to be critical (the default). Valid settings are: c - Critical (default), u - Unknown, w - Warning, o - OK
Flap Detection Option	**`enable_flap_detection=1`**	Determines whether or not Nagios will try and detect hosts and services that are "flapping". Flapping occurs when a host or service changes between states too frequently. When Nagios detects that a host or service is flapping, it will temporarily suppress notifications for that host/service until it stops flapping. Values: 1 = enable flap detection, 0 = disable flap detection (default)
Flap Detection Thresholds for Hosts and Services	**`low_service_flap_threshold=5.0`**, **`high_service_flap_threshold=20.0`**, **`low_host_flap_threshold=5.0`**, **`high_host_flap_threshold=20.0`**	This option has no effect if flap detection is disabled.
P1.pl File Location	**`/usr/sbin/p1.pl`**	Determines the location of the p1.pl perl script (used by the embedded Perl interpreter). If you didn't compile Nagios with embedded Perl support, this option has no effect.
Administrator Email/Pager Address	**`admin_email=nagios@localhost`**, **`admin_pager=pagenagios@localhost`**	The email and pager address of a global administrator. Nagios never uses these values itself, but you can access them by using the $ADMINEMAIL$ and $ADMINPAGER$ macros in your notification commands.
Daemon Core Dump Option	**`daemon_dumps_core=0`**	Determines whether or not Nagios is allowed to create a core dump when it runs as a daemon. Values: 1 - Allow core dumps, 0 - Do not allow core dumps (default).

		Note Generally, setting this option is not recommended, but it may be useful for debugging purposes. Enabling this option does not guarantee that a core file will be created.
Debug Level	**`debug_level=0`**	Determines how much (if any) debugging information will be written to the debug file. OR values together to log multiple types of information. Values: -1 = Everything, 0 = Nothing, 1 = Functions, 2 = Configuration, 4 = Process information, 8 = Scheduled events, 16 = Host/service checks, 32 = Notifications, 64 = Event broker, 128 = External commands, 256 = Commands, 512 = Scheduled downtime, 1024 = Comments, 2048 = Macros
Debug file	**`/var/log/nagios/nagios.debug`**	Location of the file where Nagios writes debugging information.

8.3. Starting, Stopping and Restarting Nagios

To start the Nagios monitoring service, log in to the Dashboard as shown in Section 8.1, “Accessing the Nagios Dashboard”.

Procedure 8.2. Stopping Nagios

1. To stop Nagios using the Dashboard, click on **Process Info** on the menu list.

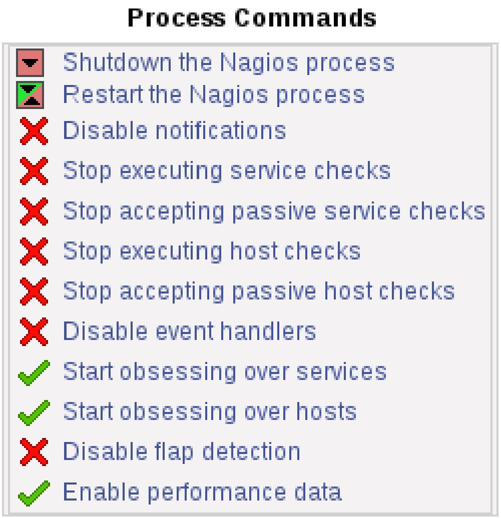

Process Commands
Shutdown the Nagios process
Restart the Nagios process
Disable notifications
Stop executing service checks
Stop accepting passive service checks
Stop executing host checks
Stop accepting passive host checks
Disable event handlers
Start obsessing over services
Start obsessing over hosts
Disable flap detection
Enable performance data

Figure 8.3. Restarting and Stopping Nagios

2. Select the **Shutdown the Nagios Process** option.
3. Select **`Commit`** to confirm the shutdown on Nagios service.

Procedure 8.3. Restarting Nagios

1. To restart Nagios using the Dashboard, click on **Process Info** on the menu list.
2. Select the **Restart the Nagios Process** option.
3. Select **`Commit`** to confirm the shutdown on Nagios service.

Chapter 9. Service Log Files

9.1. Block Storage Service Log Files

The log files of the block storage service are stored in the **/var/log/cinder/** directory of the host on which each service runs.

Table 9.1. Block Storage Log Files

File name	Description
api.log	The log of the API service (**openstack-cinder-api**).
cinder-manage.log	The log of the management interface (**cinder-manage**).
scheduler.log	The log of the scheduler service (**openstack-cinder-scheduler**).
volume.log	The log of the volume service (**openstack-cinder-volume**).

9.2. Compute Service Log Files

The log files of the compute service are stored in the **/var/log/nova/** directory of the host on which each service runs.

Table 9.2. Compute Service Log Files

File name	Description
api.log	The log of the API service (**openstack-nova-api**).
cert.log	The log of the X509 certificate service (**openstack-nova-cert**). This service is only required by the EC2 API to the compute service.
compute.log	The log file of the compute service itself (**openstack-nova-compute**).
conductor.log	The log file of the conductor (**openstack-nova-conductor**) that provides database query support to the compute service.
consoleauth.log	The log file of the console authentication service (**openstack-nova-consoleauth**).
network.log	The log of the network service (**openstack-nova-network**). Note that this service only runs in environments that are *not* configured to use OpenStack networking.
nova-manage.log	The log of the management interface (**nova-manage**).
scheduler.log	The log of the scheduler service (**openstack-nova-scheduler**).

9.3. Dashboard Log Files

The dashboard is served to users using the Apache web server (**httpd**). As a result the log files for the dashboard are stored in the **/var/log/httpd** directory.

Table 9.3. Dashboard Log Files

File name	Description
access_log	The access log details all attempts to access the web server.
error_log	The error log details all unsuccessful attempts to access the web server and the reason for each failure.

9.4. Identity Service Log Files

The log files of the identity service are stored in the **/var/log/keystone/** directory of the host on which it runs.

Table 9.4. Identity Service Log Files

File name	Description
keystone.log	The log of the identity service (**openstack-keystone**).

9.5. Image Service Log Files

The log files of the image service are stored in the **/var/log/glance/** directory of the host on which each service runs.

Table 9.5. Image Service Log Files

File name	Description
api.log	The log of the API service (**openstack-glance-api**).
registry.log	The log of the image registry service (**openstack-glance-registry**).

9.6. Monitoring Service Log File

The log files of the monitoring service are stored in the **/var/log/nagios/** directory of the host on which each service runs.

Table 9.6. Monitoring Service Log Files

File name	Description
`nagios.log`	The log for the monitoring service (**`nagios`**).

9.7. Networking Service Log Files

The log files of the networking service are stored in the **`/var/log/quantum/`** directory of the host on which each service runs.

Table 9.7. Networking Service Log Files

File name	Description
`dhcp-agent.log`	The log for the DHCP agent (**`quantum-dhcp-agent`**).
`l3-agent.log`	The log for the L3 agent (**`quantum-l3-agent`**).
`lbaas-agent.log`	The log for the Load Balancer as a Service (LBaaS) agent (**`quantum-lbaas-agent`**).
`linuxbridge-agent.log`	The log for the Linux Bridge agent (**`quantum-linuxbridge-agent`**).
`metadata-agent.log`	The log for the metadata agent (**`quantum-metadata-agent`**).
`openvswitch-agent.log`	The log for the Open vSwitch agent (**`quantum-openvswitch-agent`**).
`server.log`	The log for the OpenStack networking service itself (**`quantum-server`**).

Removing PackStack Deployments

Important

There is no automated uninstall process for undoing a PackStack install. If you have a previously installed version of OpenStack, you will need to uninstall it first, before installing with PackStack.

You can follow either of two procedures. Both are scripts. The first procedure below removes OpenStack, all application data and all packages installed on a base system. The second procedure removes only OpenStack specific application data and packages, although it may also leave some OpenStack related data behind.

Note

These procedures need to be carried out on all OpenStack hosts. Also, some of the commands may give errors if the information which the script is attempting to delete was not created in the first place.

A.1. Completely removing OpenStack, application data and all packages

To completely uninstall a deployment made using PackStack, including all application data and all packages which are installed on a base system, run the script in the following procedure.

Procedure A.1. Removing OpenStack, all application data and all packages installed on a base system

Warning

This script will remove packages including Puppet, httpd, Nagios and others which you may require for other packages. The script will also delete all MySQL databases and Nagios application data.

- Copy the following script into a file and then run it.

```
# Warning! Dangerous step! Destroys VMs
for x in $(virsh list --all | grep instance- | awk '{print $2}') ; do
    virsh destroy $x ;
    virsh undefine $x ;
done ;

# Warning! Dangerous step! Removes lots of packages
yum remove -y nrpe "*nagios*" puppet "*ntp*" "*openstack*" \
"*nova*" "*keystone*" "*glance*" "*cinder*" "*swift*" \
mysql mysql-server httpd "*memcache*" scsi-target-utils \
iscsi-initiator-utils perl-DBI perl-DBD-MySQL ;

# Warning! Dangerous step! Deletes local application data
rm -rf /etc/nagios /etc/yum.repos.d/packstack_* /root/.my.cnf \
/var/lib/mysql/ /var/lib/glance /var/lib/nova /etc/nova /etc/swift \
/srv/node/device*/* /var/lib/cinder/ /etc/rsync.d/frag* \
/var/cache/swift /var/log/keystone /var/log/cinder/ /var/log/nova/ \
/var/log/httpd /var/log/glance/ /var/log/nagios/ /var/log/quantum/ ;

umount /srv/node/device* ;
killall -9 dnsmasq tgtd httpd ;

vgremove -f cinder-volumes ;
losetup -a | sed -e 's/:.*//g' | xargs losetup -d ;
find /etc/pki/tls -name "ssl_ps*" | xargs rm -rf ;
for x in $(df | grep "/lib/" | sed -e 's/.* //g') ; do
    umount $x ;
done
```

You have now completely uninstalled the OpenStack deployment made using PackStack, including all application data and all packages.

A.2. Removing only OpenStack specific application data and packages

To uninstall only OpenStack specific application data and packages, run the script in the following procedure.

Procedure A.2. Removing OpenStack specific application data and packages

Important

After running this script, there will still be some OpenStack related data left behind.

- Copy the following script into a file and then run it.

```
# Warning! Dangerous step! Destroys VMs
for x in $(virsh list --all | grep instance- | awk '{print $2}') ; do
    virsh destroy $x ;
    virsh undefine $x ;
done ;

yum remove -y "*openstack*" "*nova*" "*keystone*" "*glance*" "*cinder*"
"*swift*";

ps -ef | grep -i repli | grep swift | awk '{print $2}' | xargs kill ;

# Warning! Dangerous step! Deletes local OpenStack application data
rm -rf  /etc/yum.repos.d/packstack_* /var/lib/glance /var/lib/nova /etc/nova
/etc/swift \
/srv/node/device*/* /var/lib/cinder/ /etc/rsync.d/frag* \
/var/cache/swift /var/log/keystone /var/log/cinder/ /var/log/nova/ \
/var/log/glance/ /var/log/quantum/ ;

# Ensure there is a root user and that we know the password
service mysql stop
cat > /tmp/set_mysql_root_pwd < < EOF
FLUSH PRIVILEGES;
EOF

/usr/bin/mysqld_safe --init-file=/tmp/set_mysql_root_pwd &
rm /tmp/set_mysql_root_pwd

mysql -uroot -pMyNewPass -e "drop database nova; drop database cinder; drop
database keystone; drop database glance;"

umount /srv/node/device* ;
vgremove -f cinder-volumes ;
losetup -a | sed -e 's/:.*//g' | xargs losetup -d ;
find /etc/pki/tls -name "ssl_ps*" | xargs rm -rf ;
for x in $(df | grep "/lib/" | sed -e 's/.* //g') ; do
    umount $x ;
done
```

You have now uninstalled only OpenStack specific application data and packages.

Revision History

Revision 1.0-42	Fri September 27, 2013	Bruce Reeler

BZ#986036 - Added information on setting up quantum networks/subnets/routers for all-in-one to be similar to Nova Networking.

BZ#973346 - Expanded information on OpenStack Networking Configuration.

Revision 1.0-41	Fri September 06, 2013	Bruce Reeler

BZ#982717 - Removed references to adminshell.

Revision 1.0-40	Mon August 19, 2013	Stephen Gordon

BZ#978670 - Replaced 192.168.1.0 with a valid IP address in examples.

BZ#984683 - Added token format configuration key to PackStack reference material.

BZ#982712 - Fixed typographical error.

BZ#973346 - Add networking configuration examples.

Revision 1.0-39	Thu July 04, 2013	Bruce Reeler

BZ#974357 - Added explanation to Running PackStack Interactively about what to do if there is an error in configuration.

BZ#980661 - Updated image of Network Topology to reflect the correct IP address.

BZ#974344 - Typos and corrections in 4.2. Running PackStack Interactively.

BZ#979149 - Updated software repository configuration to use correct entitlements.

Revision 1.0-38	Fri June 21, 2013	Deepti Navale

BZ#970812 - Included Validating the Installation section from Installation and Configuration Guide.

BZ#973327 - Updated Introduction to introduce PackStack.

BZ#974289 - Included information on Nagios and Kernel update in PackStack deployment output.

BZ#974390, BZ#974395, BZ#974399, BZ#974401, BZ#974405, BZ#974406, BZ#974410, BZ#974415, BZ#974416, BZ#974418 - Updates to Guide after QE

BZ#976117 - Added nova-conductor component to Compute introduction.

Revision 1.0-37	Fri June 14, 2013	Bruce Reeler

BZ#958293 - Updated OpenStack Networking information.

BZ#960344 - Updated Monitoring OpenStack with Nagios.

BZ#960354 - Changed appendix: Removing OpenStack PackStack Deployments.

BZ#967366 - Edits to Disabling NetworkManager section.

BZ#972940 - Included steps to reboot nodes which receive network namespaces to ensure kernel changes take effect.

BZ#973301 - Included tech review information in OpenStack Networking deployment with PackStack.

BZ#973710 - Removed OpenStack 'Preview' references.

Revision 1.0-36	Thu June 6, 2013	Deepti Navale

BZ#958293 - Added PackStack support for OpenStack networking.

Revision 1.0-35	Thu June 6, 2013	Bruce Reeler

BZ#958820 - Added 'Configuring Storage' describing manual definition of storage for Cinder and Swift instead of relying on loopback devices.

BZ#960341 - Included procedures for 'Accessing the Dashboard' and 'Uploading an Image' sections in the guide.

BZ#960354 - Added appendix: Removing OpenStack PackStack Deployments.

BZ#966617 - Provided section detailing generated passwords and where they may be changed.

BZ#967366 - Added instructions for disabling NetworkManager to avoid interference with OpenStack Networking in some installation methods.

Revision 1.0-34 **Tue May 21, 2013** **Bruce Reeler**

BZ#964246 - Removed OpenStack Networking (Quantum) attribution from Legal Notice.
BZ#960344 - Added chapter 'Monitoring OpenStack Using Nagios'.
BZ#960353 - Added chapter 'Service Log Files'.
BZ#962808 - Updated repository configuration information for 3.0.
BZ#965352 - Updated Introduction to conform with Installation and Configuration Guide Introduction.

Revision 1.0-33 **Sat May 18, 2013** **Bruce Reeler**

BZ#963051 - Removed the 'Deploying OpenStack Manually' part from Getting Started Guide.
BZ#918623 - Added controlling an instance's state and instance deletion via dashboard to 'Using OpenStack'.
BZ#960336 - Updated Software requirements (Operating system requirements) for OpenStack and removed "sudo" from commands as root login is recommended.
BZ#960341 - Updated to include sections for 'Accessing the Dashboard' and 'Uploading an Image'.

Revision 1.0-32 **Mon May 14, 2013** **Bruce Reeler**

BZ#877820 - Modified **/etc/libvirt/qemu.conf** to start VMs with OpenStack compute.
BZ#920457 - Updated figure 8.1 Keystone and Glance installed and configured.
BZ#958257 - Removed upgrade instructions (moved to Release Notes).
BZ#923263 - Tech Review: Split "Using OpenStack" into two based on using the Dashboard vs using the CLI.
BZ#921371 - Docs QE - General, Preface, Chapter 1 [4 issues].
BZ#928612 - Docs QE - Chapter 2 and 6 [6 issues].
BZ#928613 - Docs QE - Chapter 8 and 10 [2 issues].
BZ#928617 - Docs QE - Chapter 11 [12 issues].
BZ#928619 - Docs QE - Chapter 12, 13,16, 17, 19 -21 [9 issues].
BZ#929188 - Typo in Deploying Identity Service chapter.
BZ#959010 - Removed project names in chapter headings in GSG.

Revision 1.0-31 **Wed Apr 24, 2013** **Steve Gordon**

BZ#927526 - Replaced **admin** user with **openstack_network** user for consistency in the OpenStack Networking content.
BZ#950350 - Added additional upgrade step detailing the comparison and cleanup of **.rpmnew** and **.rpmsave** files.
BZ#911459 - Cleaned up unnecessary and insecure use of **/tmp** directory for temporary file storage.
BZ#921395 - Updated PackStack documentation to include Red Hat Enterprise Linux beta options.
BZ#923017 - Added explicit installation of *python-cinderclient* package for Nova compute nodes.

Revision 1.0-30 **Tue Apr 09, 2013** **Steve Gordon**

Updated subtitle to reflect status of release.

Revision 1.0-29 **Tue Apr 09, 2013** **Steve Gordon**

BZ#923845 - Added steps for using a Swift storage backend to Glance chapter.
BZ#915788 - Added basic firewall configuration information to each chapter.
BZ#927063 - Updated to remove reference to **./answers.txt** file.
BZ#928348 - Added informational and warning text around Nova to OpenStack Networking conversion.

Revision 1.0-28 **Wed Mar 27, 2013** **Steve Gordon**

BZ#895699 - Added additional documentation covering the creation of images using Oz.
BZ#923021 - Moved QPID configuration from the Nova chapter to the Cinder chapter.
BZ#896197 - Updated **packstack** answer file variables and configuration options.
BZ#927526 - Added missing configuration keys to **/etc/quantum/quantum.conf** configuration steps.
BZ#927520 - Added missing **sudo** call to **openstack-config** calls.
BZ#920397 - Updated permissions applied to **/etc/swift/**.
BZ#921395 - Updated **packstack** installation material in response to Quality Engineering review.
BZ#923423 - Removed explicit enablement of Red Hat Enterprise Linux 6 repository.
BZ#922787 - Added missing call to **restorecon** in Swift chapter.

Revision 1.0-27 Fri Mar 22, 2013 Steve Gordon

BZ#921782 - Removed API version from Glance endpoint URLs.
BZ#903271 - Removed errant trademark symbols.
BZ#907990 - Corrected errors in volume attachment procedure in response to QE review.
BZ#920457 - Added a caption to Figure 8.1 clarifying the links between the services illustrated.
BZ#923017 - Removed references to the use of **nova-volume**.
BZ#923020 - Corrected ordering of starting the **libvirt** and **messagebus** services as a result of dependencies.
BZ#920456 - Updated to consistently use the **source** instead of the period shortcut.

Revision 1.0-26 Thu Mar 14, 2013 Steve Gordon

BZ#920466 - Reduced size of table displayed in Glance chapter to avoid overflow of page.
BZ#918809 - Updated to use **/dev/disk/by-id/** structure to access attached Cinder volumes.
BZ#911101 - Added steps required to switch from Nova Networking to OpenStack Networking.
BZ#892383 - Corrected explanation of ***--location*** parameter to **glance**.
BZ#888773 - Added pointer to create an entry in **/etc/hosts** file before running **quantum-server-setup**.

Revision 1.0-25 Wed Mar 13, 2013 Steve Gordon

BZ#920456 - Updated Glance and Cinder chapters to use ***PASSWORD*** placeholder for consistency with other chapters..
BZ#918809 - Updated output of all Glance examples.
BZ#918809 - Removed statements about manual device assignment when attaching volumes. This feature does not work with KVM at this time.
BZ#918615 - Added an overview of what the "all in one" installation actually entails.
BZ#918582 - Added a note detailing the need to install *mod_ssl* for a secure dashboard installation.
BZ#920351 - Added missing **chkconfig** commands to Swift chapter.
BZ#920283 - Removed redundant *python-keystone-auth-token* package installation from Swift chapter.
BZ#903321 - Added a note explaining that Red Hat OpenStack has a different default network manager to the OpenStack upstream releases.
BZ#892383 - Updated Glance chapter to provide more detailed information regarding uploading images.
BZ#915461 - Updated all Keystone endpoints to use a "real" IP address instead of loopback.
BZ#917645 - Added admonitions explaining that the **br-ex** and **br-int** network bridges must not be manually removed or modified.
BZ#888045 - Updated **subscription-manager** output to include a valid Red Hat Enterprise Linux subscription (not a beta subscription).
BZ#917326 - Removed email address from OpenStack Networking service definition.
BZ#918992 - Updated **sudo** configuration instructions to correctly add the **wheel** group to the user.

Revision 1.0-24 Wed Mar 6, 2013 Steve Gordon

BZ#903271 - Added system requirements.
BZ#910873 - Moved Keystone service and endpoint creation to correct location in procedure.
BZ#916066 - Removed systemd specific service suffix from volume storage configuration procedure.
BZ#913138 - Updated object storage instructions for using a loopback device.
BZ#903843 - Added additional materials to introduction.
BZ#905944 - Added initial RHOS 2.0 to RHOS 2.1 upgrade instructions.
BZ#889581, BZ#917466 - Updated diagrams for Glance and Keystone services.
BZ#907990 - Expanded "Using OpenStack" material to include more examples of using the dashboard.
BZ#896197 - Documented the ***--install-hosts*** argument for **packstack**.

Revision 1.0-23 **Tue Feb 26, 2013** **Steve Gordon**

BZ#910717 - Added warning regarding nova-manager usage.
BZ#905160 - Updated PackStack material to note automatic creation of answer file.
BZ#889113 - Added step for sourcing keystonerc_admin file to all procedures that require it.
BZ#911194 - Added parameters for Cinder volume creation.
BZ#911349 - Added PackStack options for Satellite based deployments.
BZ#913283 - Changed chapter titles in manual deployment flow to better illustrate contents.

Revision 1.0-22 **Fri Feb 15, 2013** **Bruce Reeler**

BZ#888402 - Restructured Nova VNC proxy configuration to make it clear where each step needs to be applied.
BZ#876763 - Updated openstack-config commands for Glance and Cinder configuration.
BZ#889613 - Improved commands in Ch 2 Upgrading from Essex to Folsom.

Revision 1.0-20 **Tue Feb 13, 2013** **Steve Gordon**

BZ#910444, BZ#910447 - Added instructions to work around temporary issue with Nova and Cinder management utilities.

Revision 1.0-18 **Mon Feb 12, 2013** **Steve Gordon**

BZ#876763 - Updated openstack-config commands for Glance configuration.
BZ#902469 - Added informational message instructing users to install *python-cinderclient* on their Nova compute nodes if using the Cinder volume service.
BZ#888812 - Added warning message instructing users wishing to use the OpenStack Network Service to skip network creation using the **nova-manage** utility.

Revision 1.0-16 **Thu Feb 7, 2013** **Bruce Reeler**

BZ#906081 - Updated Figure 1.1. Relationships between OpenStack services.

Revision 1.0-15 **Wed Feb 6, 2013** **Bruce Reeler**

BZ#906081 - Renamed "Quantum" to "OpenStack Network".

Revision 1.0-14 **Fri Feb 1, 2013** **Stephen Gordon**

BZ#896197 - Added documentation of PackStack non-interactive use case.

Revision 1.0-13 **Tue Jan 29, 2013** **Stephen Gordon**

BZ#876763 - Updated authtoken configuration for Nova, Glance and Cinder.
BZ#888343 - Fixed various issues in the OpenStack Network chapter
BZ#888496 - Updated to use Keystone regions consistently.
BZ#891407 - Added information about kernel requirements for Open vSwitch.

Revision 1.0-12 **Tue Jan 29, 2013** **Bruce Reeler**

BZ#889306 Fixed typo.

BZ#881869 Replaced deprecated commands 'glance add' and 'glance show'.

Revision 1.0-11 Fri Jan 25, 2013 Stephen Gordon

BZ#886178 - Added steps to configure yum-plugin-priorities.

BZ#888073 - Replaced usage of "glance index" which is deprecated.

BZ#888336 - Updated instructions for configuring network interfaces to be more generic.

BZ#888553 - Updated OpenStack Network instructions to start correct services and note need for root access where required.

BZ#889105 - Expanded warning associated with temporary cinder-volumes volume group creation.

BZ#889106 - sections 7.2.2 and 7.2.3 should become subtopics of 7.2.1

BZ#889118 - Added step to Horizon instructions detailing required firewall rules.

BZ#889224 - Added step to Horizon instructions detailing need to persist SELinux change.

BZ#890510 - Added step to Horizon instructions detailing need to add "Member" role to Keystone.

BZ#903920 - Corrected argument list for "glance image-create" example.

Revision 1.0-10 Thu Jan 24, 2013 Bruce Reeler

Updated architecture section and replaced architecture diagram.

Revision 1.0-9 Wed Jan 23, 2013 Stephen Gordon

BZ#889306 - Updated repository configuration information.

BZ#889526 - Appended "-y" argument to all yum install commands.

BZ#888045 - Updated subscription manager output examples.

BZ#888060 - Updated keystone output examples.

BZ#888087 - Updated description of expected "cinder list" output.

BZ#891370 - Added section detailing expected sudo configuration.

BZ#888064 - Updated example keystonerc files to use correct PS1 values.

Revision 1.0-8 Tue Jan 22, 2013 Stephen Gordon

Updated web_version_label.

Revision 1.0-7 Tue Jan 22, 2013 Bruce Reeler

BZ888061 RH Summit namings removed, RHEL spelled out, minor edits.

BZ895236 OpenStack Network description added to Intro.

Revision 1.0-6 Fri Dec 21, 2012 Bruce Reeler

BZ885070 Missing packages in Folsom added.

BZ889160 Old Essex URL in Nova chapter replaced with Folsom URL.

BZ888061 RH summit refs in example tenant names removed.

Revision 1.0-5 Wed Dec 12, 2012 Bruce Reeler

BZ884766 Several commands in OpenStack Network packages installation section replaced.

Revision 1.0-4 Tue Dec 11, 2012 Bruce Reeler

BZ884932 Command to subscribe to RHEL beta added.

BZ871703 Broken hyperlink in Horizon chapter to reach dashboard replaced with example URL.

Revision 1.0-3 Fri Dec 7, 2012 Bruce Reeler

BZ884762 Cinder Keystone service-create cmd screen output example corrected.

BZ881844 & BZ876775 typos.

BZ877289 subscription update Essex to Folsom.

Revision 1.0-2	**Thu Nov 29, 2012**	**Bruce Reeler**

Information on subscription added to Section 1.1 Repository Config and section 1.2 Getting Started.

Revision 1.0-1	**Mon Nov 12, 2012**	**Bruce Reeler**

Edits to Guide for Folsom Preview release.

Revision 1.0-0	**Sat Nov 10, 2012**	**Bruce Reeler**

First version of the Folsom Preview guide.