Red Hat Enterprise Linux OpenStack Platform 5
Deploying OpenStack: Proof-of-Concept Environments (PackStack)

Getting Started with Red Hat Enterprise Linux OpenStack Platform

OpenStack Documentation Team

Red Hat Enterprise Linux OpenStack Platform 5 Deploying OpenStack: Proof-of-Concept Environments (PackStack)

Getting Started with Red Hat Enterprise Linux OpenStack Platform

OpenStack Documentation Team
Red Hat Customer Content Services
rhos-docs@redhat.com

Legal Notice

Abstract

This guide covers the basic getting started tasks for Red Hat Enterprise Linux OpenStack Platform 5. The deployment method described here uses PackStack for proof of concept deployments, and is not meant for production environments. PackStack can be used for single node, proof-of-concept deployments. On such deployments, you can use the Dashboard to load disk images, create a network, launch instances, or perform other basic OpenStack tasks.

Table of Contents

Preface

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 Icehouse, 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.

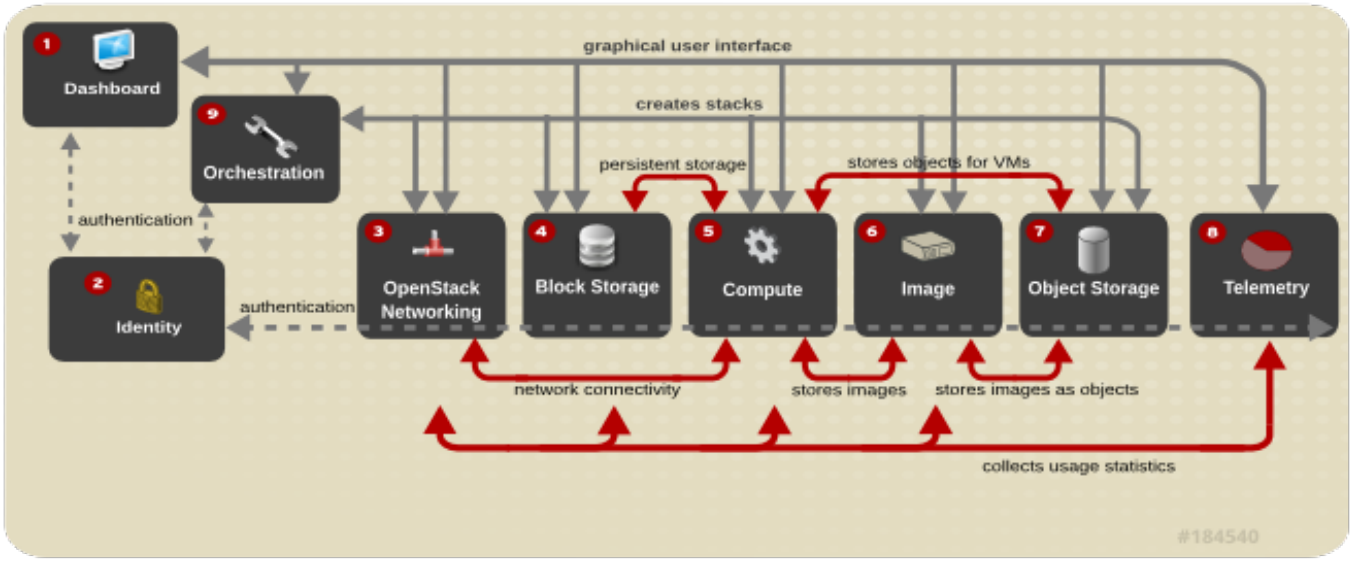


Figure 1.1. OpenStack Architecture

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

	Service	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	Neutron	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 manages and provisions virtual machines running on hypervisor nodes.
6	Image	Glance	A registry service for storing resources such as virtual machine images and Cinder snapshots.
7	Object Storage	Swift	A service providing object storage which allows users to store and retrieve files (arbitrary data).
8	Telemetry	Ceilometer	A service providing measurements of cloud resources.
9	Orchestration	Heat	A service providing a template-based orchestration engine, which supports the automatic creation of resource stacks.

Each OpenStack service is comprised of a collection of Linux services, MariaDB databases, or other components, which together provide a functional group. For example, the **glance-api** and **glance-registry** Linux services, together with a MariaDB database, implement the Image service.

For a more detailed overview of each OpenStack service, see the relevant section on the service in *Deploying OpenStack: Learning Environments (Manual Setup)* at:

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/

1.3. The Packstack Deployment Utility

Packstack is a command-line utility that uses Puppet modules to enable rapid deployment of OpenStack on existing servers over an SSH connection. 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.

Packstack is suitable for deploying the following types of configurations:

- Single-node proof-of-concept installations, where all controller services and your virtual machines run on a single physical host. This is referred to as an all-in-one install.
- Proof-of-concept installations where there is a single controller node and multiple compute nodes. This is similar to the all-in-one install above, except you may use one or more additional hardware nodes for running virtual machines.

Important

Packstack is designed for proof-of-concept deployments, and is not suitable as a production deployment tool. Packstack makes many assumptions in its configuration to simplify the installation process, and cannot deploy services in a highly available (HA) or load balanced configuration, nor provide the flexibility required for configuring complex networking. See *Deploying OpenStack: Enterprise Environments (Red Hat Enterprise Linux OpenStack Platform Installer)* for more information on using the Red Hat Enterprise Linux OpenStack Platform installer to deploy production environments.

1.4. Supported Virtual Machine Operating Systems

All guest operating systems that are certified with KVM in Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7 are supported by RHEL OpenStack Platform 5. A detailed list of the supported guest operating systems can be found here: Certified Guest Operating Systems in Red Hat Enterprise Linux OpenStack Platform and Red Hat Enterprise Virtualization.

Chapter 2. Product Requirements

2.1. Software Requirements

2.1.1. Operating System Requirements

This version of Red Hat Enterprise Linux OpenStack Platform is supported on:

- Red Hat Enterprise Linux 7

For further information on configuring your subscription and the required repositories, see:

- Section 2.1.2.2, “Customer Portal Subscription Management”
- Section 2.1.2.4, “Red Hat Enterprise Linux Repository Configuration”
- Section 2.1.2.5, “Red Hat Enterprise Linux OpenStack Platform Repository Configuration”

For detailed information on installing Red Hat Enterprise Linux, see the corresponding installation guide at:

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/

2.1.2. Configure Software Repositories

2.1.2.1. Assigning Fully Qualified Domain Names

It is recommended that systems used to host OpenStack API endpoints are assigned fixed IP addresses or fully qualified domain names (FQDN).

If OpenStack API endpoints are hosted on systems that have their IP addresses dynamically assigned by a DHCP server, there is a loss of service if the assigned address changes. When this occurs, you would have to manually update the endpoint definitions stored in the database of the Identity service.

- To avoid this problem, before registering to Red Hat Network, set the **HOSTNAME** in the **/etc/sysconfig/network** file on each system that will host an OpenStack API endpoint.

```
HOSTNAME=myhost.parentdomain
```

2.1.2.2. Customer Portal Subscription Management

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 the Customer Portal Subscription Management using Subscription Manager. For further information on managing Red Hat subscriptions, see the Red Hat Subscription Management documentation at the following link:

https://access.redhat.com/site/documentation/en-US/Red_Hat_Subscription_Management/

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

Procedure 2.1. Registering a Red Hat Enterprise Linux system using Subscription Management

1. Run the **subscription-manager register** command to register the system with Customer Portal Subscription Management.

```
# subscription-manager register
```

2. Enter your Red Hat Customer Portal user name when prompted.

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

Important

Your Red Hat Subscription must have Red Hat Enterprise Linux OpenStack Platform entitlements. If your subscription does not have Red Hat Enterprise Linux OpenStack entitlements then you may register for access to the evaluation program at http://www.redhat.com/products/enterprise-linux/openstack-platform/.

3. Enter your Red Hat Customer Portal password when prompted.

```
Password:
```

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

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

The system has been registered with Customer Portal Subscription Management and is ready to be attached to specific software subscriptions.

2.1.2.3. RHN/CDN Channels

This section discusses channel and repository settings required for deploying Red Hat Enterprise Linux OpenStack Platform 5.

Warning

Although older Red Hat OpenStack repositories are available, you must ensure that your system can no longer access them before installing Red Hat Enterprise Linux OpenStack Platform 5. For example, for CDN, unsubscribe from or disable the following:

- Red Hat OpenStack 1.0 (Essex) -- rhel-server-ost-6-preview-rpms
- Red Hat OpenStack 2.1 (Folsom) -- rhel-server-ost-6-folsom-rpms
- Red Hat Enterprise Linux OpenStack Platform 3 (Grizzly) -- rhel-server-ost-6-3-rpms
- Red Hat Enterprise Linux OpenStack Platform 4 Beta (Havana) -- rhel-6-server-openstack-beta-rpms
- Red Hat Enterprise Linux OpenStack Platform 4 (Havana) -- rhel-6-server-openstack-4.0-rpms

Note

The Red Hat Common for RHEL Server channel is recommended for use if creating custom Red Hat Enterprise Linux guest images that require cloud-init.

For Red Hat Enterprise Linux 7, run:

```
# subscription-manager repos \
    --enable=rhel-7-server-rh-common-rpms
```

2.1.2.3.1. Content Delivery Network Channels

You can install Red Hat Enterprise Linux OpenStack Platform 5 through the Content Delivery Network (CDN). To do so, configure **subscription-manager** to use the correct channels.

Run the following command to enable a CDN channel:

```
# subscription-manager repos --enable=[reponame]
```

Run the following command to disable a CDN channel:

```
# subscription-manager repos --disable=[reponame]
```

Red Hat Enterprise Linux 7

The following tables outline the channels for Red Hat Enterprise Linux 7.

Table 2.1. Required Channels

Channel	Repository Name
Red Hat Enterprise Linux 7 Server (RPMS)	**rhel-7-server-rpms**
Red Hat OpenStack 5.0 for Server 7 (RPMS)	**rhel-7-server-openstack-5.0-rpms**

Table 2.2. Optional Channels

Channel	Repository Name
Red Hat Enterprise Linux 7 Server - Optional	**rhel-7-server-optional-rpms**

Red Hat Enterprise Linux OpenStack Platform Installer

The following tables outline the channels for the Red Hat Enterprise Linux OpenStack Platform installer.

Table 2.3. Required Channels

Channel	Repository Name
Foreman-RHEL-6	**rhel-6-server-openstack-foreman-rpms**
Red Hat Enterprise Linux 6 Server (RPMS)	**rhel-6-server-rpms**

Channel	Repository Name
Red Hat Software Collections RPMs for Red Hat Enterprise Linux 6 Server	**rhel-server-rhscl-6-rpms**

Disable Channels

The following table outlines the channels you must disable to ensure Red Hat Enterprise Linux OpenStack Platform 5 functions correctly.

Table 2.4. Disable Channels

Channel	Repository Name
Red Hat CloudForms Management Engine	**"cf-me-*"**
Red Hat CloudForms Tools for RHEL 6	**"rhel-6-server-cf-*"**
Red Hat Enterprise Virtualization	**"rhel-6-server-rhev*"**
Red Hat Enterprise Linux 6 Server - Extended Update Support	**"*-eus-rpms"**

2.1.2.3.2. Red Hat Network (RHN) Channels

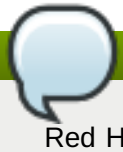
You can install Red Hat Enterprise Linux OpenStack Platform 5 through Red Hat Network (RHN).

Run the following to add a channel via RHN:

```
# rhn-channel --add --channel=[reponame]
```

Run the following to remove a channel via RHN:

```
# rhn-channel --remove --channel=[reponame]
```



Note

Red Hat Network is only available via Red Hat Satellite on Red Hat Enterprise Linux 7.1. For more information on this product, see:

https://access.redhat.com/site/documentation/en-US/Red_Hat_Satellite/6.0/html/User_Guide/index.html

Red Hat Enterprise Linux 7

The following tables outline the channels for Red Hat Enterprise Linux 7.1.

Table 2.5. Required Channels

Channel	Repository Name
Red Hat OpenStack 5.0 for Server 7.1	**rhel-x86_64-server-7-ost-5**
Red Hat Enterprise Linux Server (v7 for 64-bit AMD64 / Intel64)	**rhel-x86_64-server-7**

Table 2.6. Optional Channels

Channel	Repository Name
Red Hat Enterprise MRG Messaging v. 2 (for RHEL 7 Server x86_64)	**rhel-x86_64-server-7-mrg-messaging-2**

2.1.2.4. Red Hat Enterprise Linux Repository Configuration

Log in as the **root** user and follow the steps in this procedure to configure a Red Hat Enterprise Linux system to receive updates from Red Hat Network. Repeat these steps on each system in the OpenStack environment.

Procedure 2.2. Attaching Pool IDs to your subscription

1. Use the **subscription-manager list** 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.
```

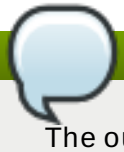
   Replace *POOLID* with the unique identifier associated with your Red Hat Enterprise Linux Server subscription. This is the identifier that was located in the previous step.

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-7-server-rpms         Red Hat Enterprise Linux 7 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-7-server-rpms** repository.

4. For deployments based on Red Hat Enterprise Linux 6, add the **MRG Messaging v. 2 (for RHEL 6 Server x86_64)** repository to receive the updated Qpid version:

    a. Enable the repository using the **subscription-manager** command:

    ```
    # subscription-manager repos --enable=rhel-6-server-mrg-
    messaging-2-rpms
    ```

    b. 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
    ```

    c. Update the Qpid packages:

    ```
    # yum update python-qpid qpid-cpp-server qpid-tools
    ```

    d. Restart the Red Hat Enterprise Linux OpenStack Platform services:

    ```
    # openstack-service restart
    ```

    e. For controller nodes, the **qpidd** service must also be restarted:

    ```
    # service qpidd restart
    ```

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

2.1.2.5. Red Hat Enterprise Linux OpenStack Platform Repository Configuration

Follow the steps in this procedure to configure a Red Hat Enterprise Linux system to receive OpenStack packages and updates from Content Delivery Network or Red Hat Network. Access to a Red Hat software entitlement that includes Red Hat Enterprise Linux OpenStack Platform 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)

Important

Required and optional repository names for each version are listed in Section 2.1.2.3, "RHN/CDN Channels".

These steps must be run while logged in as the **root** user. Repeat these steps on each system in the environment.

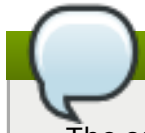
Procedure 2.3. Attaching Pool IDs to your subscription

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:               DATE
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:               DATE
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 Enterprise Linux OpenStack Platform 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.

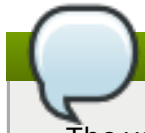
```
# subscription-manager attach --pool=POOLID
Successfully attached a subscription for ENTITLEMENT.
```

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. Use either the **subscription-manager** or **yum-config-manager** commands to enable or disable the appropriate software repositories (channels).

For example, to ensure that the repository for Red Hat Enterprise Linux OpenStack Platform 3 (Grizzly) has been disabled, run:

```
# subscription-manager repos --disable 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/op
enstack/3/os
cache = 0
cachedir = /var/cache/yum/x86_64/6Server/rhel-server-ost-6-3-rpms
cost = 1000
enabled = False
...
```



Note

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

```
enabled = 1
```

4. 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 current list of enabled repositories will be displayed, along with the number of available packages. For example:

```
repo id                repo name
status
rhel-7-server-rpms     Red Hat Enterprise Linux 7 Server (RPMs)
11,610+460
rhel-7-server-openstack-5.0-rpms  \
Red Hat OpenStack 5.0 for Server 7 (RPMS)   487+143
```

5. Use the **subscription-manager** command to enable the Red Hat Enterprise Linux OpenStack Platform repository.

```
# subscription-manager repos --enable=[reponame]
```

Replace *[reponame]* with the appropriate repository name. For details, see Section 2.1.2.3, “RHN/CDN Channels”.

6. Install the *yum-plugin-priorities* package provided by the **rhel-6-server-optional-rpms** channel for Red Hat Enterprise Linux 6.6 or the **rhel-7-server-optional-rpms** channel for Red Hat Enterprise Linux 7.1:

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

7. If not yet installed, use the following to install yum-config-manager:

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

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

```
# yum-config-manager --enable [reponame] --
setopt="[reponame].priority=1"
```

For example:

```
# yum-config-manager --enable rhel-7-server-openstack-5.0-rpms \
        --setopt="rhel-7-server-openstack-5.0-rpms.priority=1"
Loaded plugins: product-id
==== repo: rhel-7-server-openstack-5.0-rpms ====
[rhel-7-server-openstack-5.0-rpms]
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/6Server
...
cost = 1000
```

```
enabled = True
...
priority = 1
...
```

9. 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
# reboot
```

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

2.1.3. Disable Network Manager

OpenStack Networking currently does not work on systems that have the Network Manager (**NetworkManager**) service enabled.

Follow the steps listed in this procedure while logged in as the **root** user on each system in the environment that will handle network traffic. This includes the system that will host the OpenStack Networking service, all network nodes, and all compute nodes.

These steps determine the state of the **NetworkManager** service, disable it from running, and replace it with the standard network service:

Procedure 2.4. Disabling the Network Manager service

1. Verify Network Manager is currently enabled using the **systemctl** command.

```
# systemctl status NetworkManager.service | grep Active:
```

The output indicates whether or not the Network Manager service is enabled.

A. The system displays an error if the Network Manager service is not currently installed:

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

If this error is displayed then no further action is required to disable the Network Manager service.

B. The system displays **Active: active (running)** if Network Manager is running, or **Active: inactive (dead)** if it is not.

For example, if Network Manager is active:

```
Active: active (running) since Thu 2014-06-26 19:34:00 EDT; 2s
ago
```

If Network Manager is inactive, then no further action is required.

2. If Network Manager is running, then you must first stop it:

```
# systemctl stop NetworkManager.service
```

3. Then disable Network Manager:

```
# systemctl disable NetworkManager.service
```

4. 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 **ONBOOT** configuration key is set to **yes**. Add these keys manually if they do not already exist in each file.

```
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.

5. Ensure that the standard network service is started using the **systemctl** command:

```
# systemctl start network.service
```

6. Ensure that the network service is enabled:

```
# systemctl enable network.service
```

The Network Manager service has been disabled. 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 only proof-of-concept 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.

2.3. Configuring Storage

Block Storage

Block Storage uses volume groups to identify attached volumes. By default, PackStack creates:

- An example storage volume for testing. It is placed in **/var/lib/cinder** and installed as a loopback storage device on the host for the Block Storage service.
- The **cinder-volumes** volume group (configured in **volume_group** in **/etc/cinder/cinder.conf**).

To avoid the creation of loopback devices, you must initialize your volume manually for the Block Storage service before installing and deploying OpenStack using PackStack.

Example 2.1. Create Volume Group

Initialize the volume manager as a physical volume, and then use it to create the **cinder-volumes** volume group:

```
# pvcreate /dev/sdX
# vgcreate cinder-volumes /dev/sdX
```

Object Storage

Instead of installing a volume for Object Storage, PackStack adds a device to an Object Storage ringfile. On the Object Storage host, the device is represented by a directory in **/srv/**. Ideally, the directory for the Object Storage device should be a separate file system.

If you do not have a separate file system, or just want to test Object Storage, then PackStack creates a small loopback storage device in place of a separate partition. Otherwise, you must manually configure your system using PackStack's answer file (see Section 5.3.2, “Editing a PackStack Answer File”).

Example 2.2. Configure Storage Devices using PackStack Answer File

Configure Object Storage with **/dev/sdb1** and **/dev/sdc1** (no testing loopback device):

```
CONFIG_SWIFT_STORAGE_HOSTS=192.0.43.10/sdb1,192.0.43.10/sdc1
```

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 single-node proof-of-concept installations. Such deployments are **not suitable for production environments**.

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. Selecting Components to Deploy

In OpenStack, there are different options available to achieve similar results. This chapter discusses your networking deployment options.

- *OpenStack Networking versus Compute networking*

  OpenStack networks can be deployed using either OpenStack Networking (Neutron) or Compute networking:

  - OpenStack Networking is a service which provides Networking-as-a-Service functionality in OpenStack. It can be configured for advanced virtual network topologies, such as per-tenant private networks and more. It is able to virtualize and manage both Layer 2 (logical) and Layer 3 (network) of the OSI network model. Each tenant has a virtual neutron router with one or more private networks, which can communicate with the outside world. This allows full routing isolation for each tenant private network. OpenStack Networking functions can be performed using either the dashboard or the command line interface.

    OpenStack Networking uses namespaces (netns), which virtualize access to network resources, giving each group of processes the network access it requires. 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 requires a kernel with the version **`2.6.32-431.el6.x86_64`** or later.

    To check if you have the required netns enabled kernel installed, you can install the package *iproute-2.6.32-130.el6ost.netns.2.x86_64* and run the command:

    ```
    # ip netns
    ```

  - Compute networking accepts networking tasks such as setting up bridging interfaces or changing iptables rules from the queue and performs them to manipulate the network. It is able to virtualize and manage simple Layer 3 virtualization. When you deploy OpenStack using nova-network, all network related commands must be done using the command line interface.

Chapter 4. Installing PackStack

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

Procedure 4.1. 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 5, *Running PackStack* for information on prerequisites and running PackStack for the first time.

Chapter 5. 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.

See Section 5.1, “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.

See Section 5.2, “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.

See Section 5.3, “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 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 Volume Storage.

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 Compute 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.

Note

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. For more information, see Appendix A, *Removing PackStack Deployments*

5.1. 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 is deployed as a Controller node; subsequent hosts are deployed as Compute nodes.

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

For a list of available command-line options, see Section 5.3.2, “Editing a PackStack Answer File”.

Procedure 5.1. 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 are prompted to enter the password of the **`root`** user to facilitate SSH key installation.

   Example 5.1. Single-node Deployment using OpenStack Networking (default)

   In this example, PackStack is instructed to deploy an "all in one" installation to the local system, which results in the following:

   - OpenStack Networking is enabled by default.
   - An **`admin`** Identity service tenant is created along with a **`keystonerc_admin`** file. The admin password and other information are stored in the **`keystonerc_admin`** file, which is located in the **`/root`** directory, and also in the directory from which PackStack is run, if different from **`/root`**. The admin password is also stored as **`CONFIG_KEYSTONE_ADMIN_PW`** in PackStack's answer file.
   - A **`demo`** Identity service tenant is created, with its associated **`keystonerc_demo`** file, which can be sourced like the **`keystonerc_admin`** file. The demo password is also stored as **`CONFIG_KEYSTONE_DEMO_PW`** in PackStack's answer file.

- The keys **CONFIG_PROVISION_DEMO** and **CONFIG_PROVISION_ALL_IN_ONE_OVS_BRIDGE** are automatically enabled in PackStack's answer file. This answer file will have a file name similar to **/root/packstack-answers-*20130306-051043*.txt**.
- The OpenStack Dashboard is automatcially installed. After the installation is finished, you should log into the Dashboard using the **demo** account instead of the **admin** account due to the ownership of the private and public networks.

```
# packstack --allinone
```

Example 5.2. Single-Node Deployment without OpenStack Networking

In this example, PackStack is instructed to deploy an "all in one" installation to the local system, but using only Compute networking.

```
# packstack --allinone --os-neutron-install=n
```

B. **Multiple-node Deployment**

Important

Red Hat Enterprise Linux OpenStack Platform requires that each system in the OpenStack environment is running Red Hat Enterprise Linux Server, and that all systems are signed up to receive updates from Red Hat Network using Subscription Manager.

See Section 2.1.2.2, “Customer Portal Subscription Management” for information on registering to Red Hat Network.

- You can ssh into each node and register it, attach an OpenStack subscription, and enable the required OpenStack repository. This allows PackStack to install OpenStack on each node.
- Alternatively, you can get PackStack to generate an answer file, then edit the answer file to contain your Red Hat subscription username and password, using the parameters **CONFIG_RH_USER** and **CONFIG_RH_PW**. Running PackStack with this answer file configures each node to receive updates from Red Hat Network using Subscription Manager. This allows PackStack to install OpenStack on each node.

  For instructions on generating and editing an answer file, see Section 5.3.1, “Generating a PackStack Answer File”.

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 Controller node. Replace *NODE_ADDRESSES* with IP addresses of the systems that you intend to use as Compute nodes.

> **Example 5.3. 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. The time can be significant.

   PackStack provides continuous updates indicating which manifests are being deployed as it progresses through the deployment process. Once the process is completed, a confirmation message similar to the one shown below will be displayed:

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

   Additional information regarding the locations of the answer file and other files is given. Depending on the options you chose, the following screen's content will vary:

   ```
   Additional information:
    * A new answerfile was created in: /root/packstack-answers-20131205-155916.txt
    * Time synchronization installation was skipped. Please note that unsynchronized time on server instances might be problem for some OpenStack components.
    * File /root/keystonerc_admin has been created on OpenStack client host 192.168.43.10. To use the command line tools you need to source the file.
    * To use the console, browse to http://192.168.43.10/dashboard
    * To use Nagios, browse to http://192.168.43.10/nagios username : nagiosadmin, password : abcdefgh12345678
    * The installation log file is available at: /var/tmp/packstack/20131205-155915-tZ0BTD/openstack-setup.log
    * The generated manifests are available at: /var/tmp/packstack/20131205-155915-tZ0BTD/manifests
   ```

Procedure 5.2. Viewing the created network

1. Retrieve the URL shown in the "Additional information" output shown above, and the demo tenant password in **/root/keystonerc_demo**.
2. Type the URL into a browser to access the Dashboard:

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

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

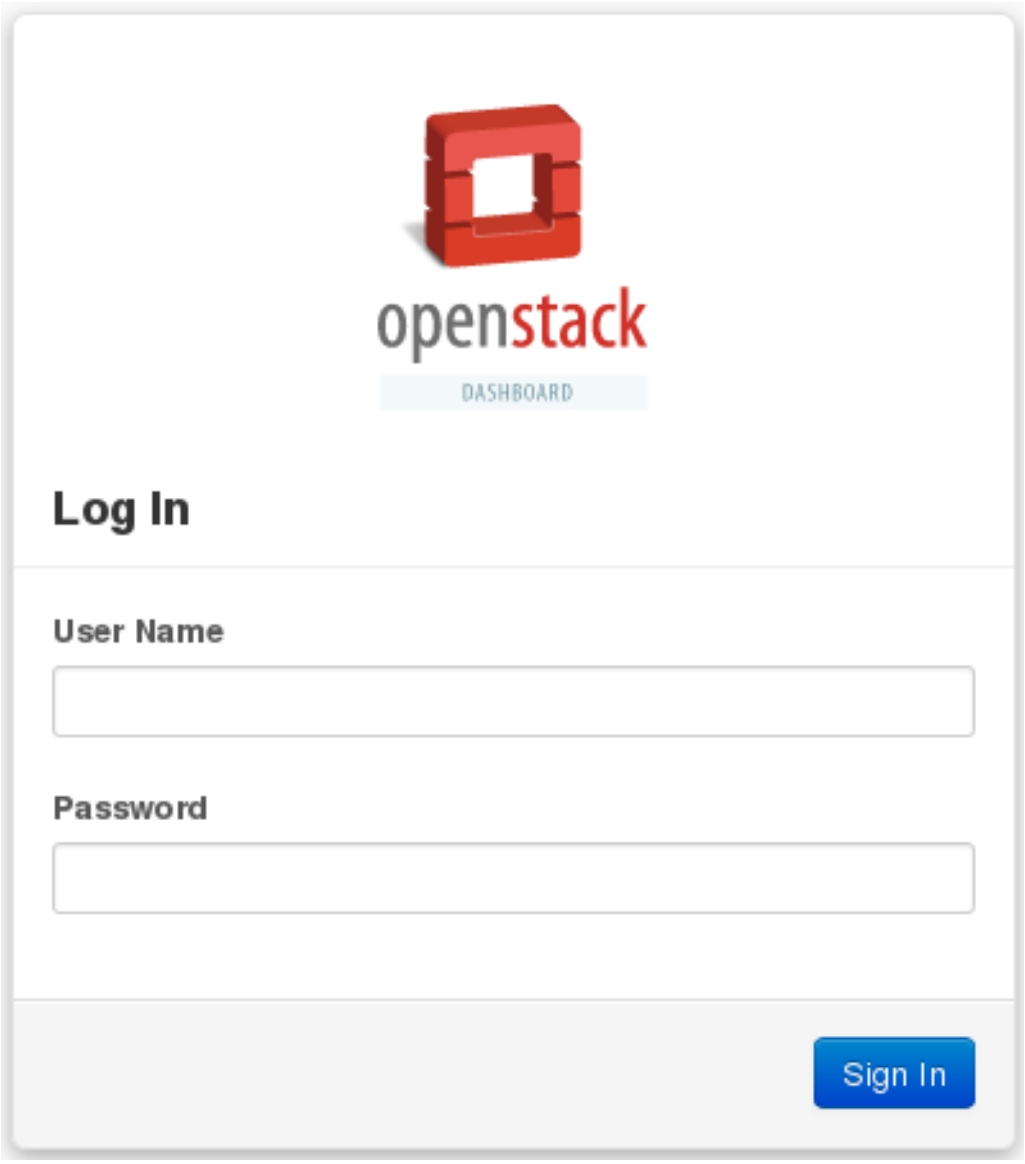


Figure 5.1. Log In Screen

3. Type **demo** as the **User Name** and the password in **/root/keystonerc_demo** as the **Password**.
4. Click **Sign In**.
5. After signing in to the Dashboard, you can see details of the "all in one" deployment. For example, to see the basic network set-up, select the **Project** tab and choose **Network Topology**.

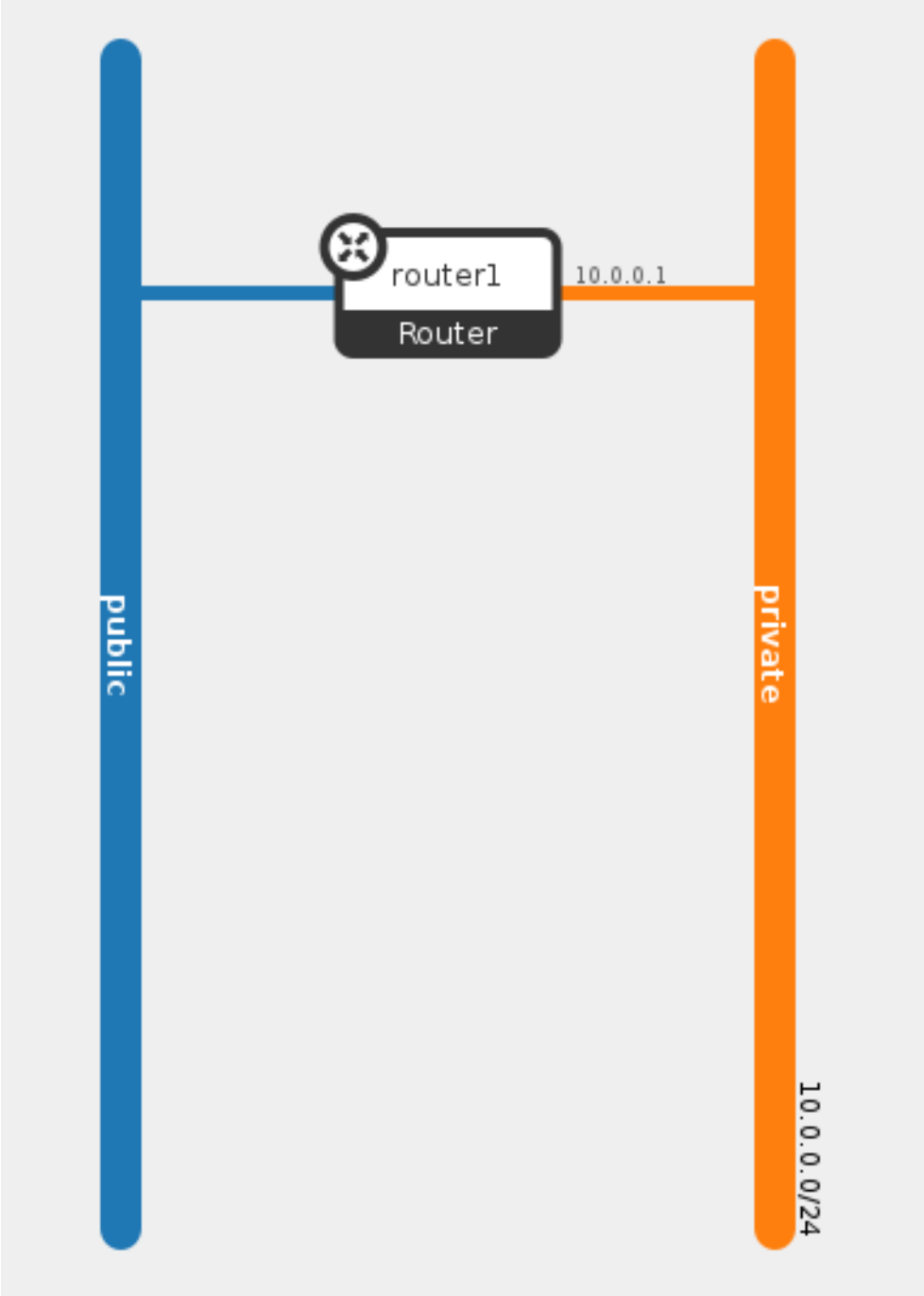


Figure 5.2. Default deployed network

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. (That is, on the system CONFIG_OSCLIENT_HOST) 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
```

This is just the basic setup. You will now need to configure the network and run up instances, using either the Dashboard or the command-line interface.

See Part III, “Using OpenStack” for further instructions to begin using your OpenStack environment.

5.2. 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 5.3. 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 MySQL Database**

   PackStack will prompt you to choose if you want to install MySQL database on the same host or on a remote host.

   ```
   Should Packstack install MySQL DB [y|n]  [y] :
   ```

Note

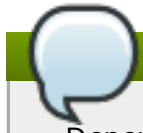
PackStack is capable of installing a single MySQL database node. On the other hand, PackStack does not handle MySQL cluster installation, but it allows you to work with a MySQL cluster that you have set up separately.

If you choose **n**, PackStack asks you for credentials and uses **CONFIG_MYSQL_HOST** as a given remote database.

4. **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 OpenStack Image service (Glance) [y|n] [y]
:
Should Packstack install OpenStack Block Storage service (Cinder)
[y|n] [y] :
Should Packstack install OpenStack Compute service (Nova) [y|n] [y]
:
Should Packstack install OpenStack Networking service (Neutron)
[y|n] [y] :
Should Packstack install OpenStack Dashboard service (Horizon) [y|n]
[y] :
Should Packstack install OpenStack Object Storage service (Swift)
[y|n] [n] :
Should Packstack install OpenStack Telemetry service (Ceilometer)
[y|n] [y] :
Should Packstack install OpenStack Orchestration (Heat) [y|n] [n] :
```



Note

Depending on which services you selected for installation, the ensuing prompts in this procedure will vary.

   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.

5. 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] :
```

6. 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 5.4. 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
```

7. 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] :
```

8. If you have existing servers that have been configured previously, and you do not want PackStack to overwrite their configurations, you can specify the IP addresses of the servers to be excluded.

```
Enter a comma separated list of server(s) to be excluded. Leave
plain if you don't need to exclude any server.:
```

9. Select whether you want to run OpenStack services in debug mode.

```
Do you want to run OpenStack services in debug mode [y|n] [n]:
```

10. Select whether you want to use VMware vCenter as hypervisor and datastore.

```
Do you want to use VMware vCenter as hypervisor and datastore [y|n]
[n]:
```

11. **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 :
    ```

12. **Configuring the AMQP service**

    OpenStack services by default use the RabbitMQ messaging system to communicate. RabbitMQ is the recommended messaging system.

    a. You can choose between RabbitMQ or Qpid.

    ```
    Set the server for the AMQP service (qpid, rabbitmq)?
    [qpid|rabbitmq]  [rabbitmq] :
    ```

b. Enter the IP address of the AMQP service.

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

c. Choose whether to enable SSL for the AMQP service.

```
Enable SSL for the AMQP service? [y|n]  [n] :
```

d. Choose whether to enable authentication for the AMQP service.

```
Enable Authentication for the AMQP service? [y|n]  [n] :
```

13. **Configuring the Identity service**

OpenStack uses the Identity service (**openstack-keystone**) for identity, token, catalog, and policy services.

a. If Identity service installation was selected, then you must enter the IP address of the server to deploy Identity on.

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

b. A Keystone **admin** user is created when the Identity service is installed. This user requires a password. The password and other information is stored in the **keystonerc_admin** file, which is located in the **/root** directory. In case of multi-host installation, **keystonerc_admin** is located only in **/root** on the **CONFIG_OSCLIENT_HOST** host.The admin password is also stored in PackStack's answer file.

Enter the password for the Keystone **admin user**.

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

c. A **demo** Keystone tenant is also created along with a **keystonerc_demo** file, which can be sourced like the existing **keystonerc_admin** file. Creating this user 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**.

Enter the password for the Keystone **demo user**.

```
Enter the password for the Keystone demo user :
```

14. **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] :
```

15. **Configuring the Volume service**

OpenStack uses the Volume service (**openstack-cinder-***) to provide volume storage services.

a. If Volume service installation was selected, enter the IP address of the server to deploy the Volume service on.

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

b. OpenStack Block Storage requires some back-end storage that the service is built on. The default implementation is to use Logical Volume Management (LVM) to create a Logical Volume Group called **cinder-volumes**. Alternatives are Red Hat Storage (gluster) or Network File System (NFS).

```
Enter the Cinder backend to be configured [lvm|gluster|nfs]
[lvm]:
```

c. If you chose LVM, PackStack expects storage for use with Volume to be available on a volume group named **cinder-volumes**.

i. 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]:
```

ii. 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 volumes 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.

d. If you chose gluster, you do not need to create a local volume. Instead you need to choose the gluster volume to mount.

i. Enter a gluster volume for use with Cinder, for example *ip-address:/vol-name*.

```
Enter a single or comma separated list of gluster volume
shares to use with Cinder [^'([\d]{1,3}\.){3}[\d]
{1,3}:/.*']:
```

16. **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. 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]:
```

f. The Compute scheduler (**openstack-nova-scheduler**) is used to map Compute's 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] :
```

g. 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.

   i. 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] :
```

ii. 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] :
```

h. 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] :
```

i. The Compute networking service (**openstack-nova-network**) provides network services for Compute instances. Enter the IP address of the server to deploy the Compute networking 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.

j. The Compute network manager can be selected to be VLAN Manager, Flat Manager or Flat DHCP manager. Type *VlanManager*, *FlatManager*, or *FlatDHCPManager* to replace the final term *Manager* in the expression **nova.network.manager.*Manager*** as required. Flat DHCP is the default.

```
Enter the Nova network manager
[nova.network.manager.FlatDHCPManager] :
```

k. A public interface must be configured to allow connections from other nodes and clients. Enter the name of the public interface to use. Examples are eth0, eth1, and so on, or em1, em2 and so on, or p1p1, p1p2, p2p1 and so on.

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

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

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

m. 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 network manager [192.168.32.0/22] :
```

n. 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] :
```

o. 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] :
```

p. 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] :
```

17. **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 Neutron server  [192.0.43.10] :
```

b. OpenStack Networking uses namespaces (netns).

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 requires a kernel with the version **`2.6.32-431.el6.x86_64`** or later.

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

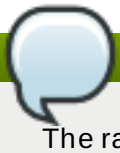
```
Should Neutron use network namespaces? [y|n]  [y] :
```

c. OpenStack Networking sets up the Neutron 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 Neutron 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 Neutron 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 Neutron 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, such as **br-ex**. 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 Neutron L3 agent will
use for external traffic  [br-ex]
```

e. OpenStack Networking sets up the Neutron 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 Neutron DHCP set up.

```
Enter a comma separated list of IP addresses on which to
install Neutron 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. Example linuxbridge names would be 'br1' or 'br100'.
- **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 Neutron
[linuxbridge|openvswitch]  [openvswitch] :
```

g. The OpenStack Compute service allows virtual machines 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 Neutron 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 Neutron plugin supports GRE tunneling, and you can select **gre** as long as the installed kernel (version 2.6.32-431.el6.x86_64 or later) and Open vSwitch userspace support GRE tunneling too.

```
Enter the type of network to allocate for tenant networks
[local|vlan|gre|vxlan]  [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 Neutron
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
Neutron openvswitch plugin  [physnet1:br-eth1] :
```

18. **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.

19. **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] :
    ```

20. **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] :
```

21. **Configuring Demo a User and Testing**

PackStack allows you to provision a demo user and a testing suite.

a. PackStack allows you to optionally configure a demo user and testing. Select *y* or *n* to make your choice.

```
Would you like to provision for demo usage and testing? [y|n]
[n] :
```

b. If you choose to provision a demo user, then enter a network address for the floating IP subnet.

```
Enter the network address for the floating IP subnet:
[192.168.32.0/22] :
```

c. Tempest is the OpenStack Integration test suite. It runs tests using a simple configuration file that describes the test environment. The tests are run against all OpenStack service endpoints by exercising API calls and validates the responses.

If you choose to configure the demo user and testing, would you like to configure the OpenStack test suite.

```
Would you like to configure Tempest (OpenStack test suite)?
[y|n]  [n] :
```

d. For the demo user and testing, would you like to configure the external OVS bridge.

```
Would you like to configure the external ovs bridge? [y|n]
[n] :
```

22. **Configuring the Orchestration Service**

The Orchestration service provides a template-based orchestration for describing a cloud application by running OpenStack API calls to generate running cloud applications. The software integrates other core components of OpenStack into a one-file template system.

a. The CloudWatch API can be used to do the following functionality: list alarms, list metrics etc. Enter *y* to install the Heat CloudWatch API.

```
Should Packstack install Heat CloudWatch API [y|n]  [n] :
```

b. Heat endeavors to provide compatibility with the AWS CloudFormation template format, so that many existing CloudFormation templates can be launched on OpenStack. Heat provides both an OpenStack-native REST API and a CloudFormation-compatible Query API. Enter *y* to install Heat CloudFormation API.

```
Should Packstack install the Heat CloudFormation API [y|n]
[n] :
```

Note

For more information on how to use the Dashboard and CLI commands to create and manage stacks using templates, see the section "Launch and manage stacks" in the *Red Hat Enterprise Linux OpenStack Platform End User Guide*, at this link: Launch and manage stacks.

23. **Configuring the Telemetry Service**

The Telemetry service is the unique point of contact for billing systems to acquire all of the measurements they need to establish customer billing, across all current OpenStack core components. Enter the IP address of the server on which the Telemetry service has to be installed.

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

Note

For more information on the CLI commands required to use the Telemetry commands, see the section "Measure cloud resources" in the *Red Hat Enterprise Linux OpenStack Platform End User Guide*, at this link: Measure cloud resources.

24. **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] :
```

25. **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.

i. 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 :
```

ii. 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 :
```

iii. 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 :
```

iv. 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 :
```

v. 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 :
```

vi. 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 :
```

vii. 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. :
```

viii. 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**.

See 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
:
```

26. **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-mysql-install:               y
os-glance-install:              y
os-cinder-install:              y
os-nova-install:                y
os-neutron-install:             y
os-horizon-install:             y
os-swift-install:               n
os-ceilometer-install:          y
os-heat-install:                n
os-client-install:              y
ntp-servers:
nagios-install:                 n
exclude-servers:
os-debug-mode:                  n
os-vmware:                      n
mysql-host:                     192.0.43.10
mysql-pw:                       ********
amqp-server           rabbitmq
amqp-host:                      192.0.43.10
amqp-enable-ssl:                n
amqp-enable-auth:               n
keystone-host:                  192.0.43.10
keystone-admin-passwd:          ********
keystone-demo-passwd:           ********
glance-host:                    192.0.43.10
cinder-host:                    192.0.43.10
cinder-backend:                 lvm
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
novaconductor-host:             192.0.43.10
nova-db-passwd:                 ********
nova-ks-passwd:                 ********
novasched-host:                 192.0.43.10
novasched-cpu-allocation-ratio:16.0
novasched-ram-allocation-ratio:1.5
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
```

```
neutron-server-host:            192.0.43.10
neutron-use-namespaces:         y
neutron-l3-hosts:               192.0.43.10
neutron-l3-ext-bridge:          br-ex
neutron-dhcp-hosts:             192.0.43.10
neutron-l2-plugin:              openvswitch
neutron-metadata-hosts:         192.0.43.10
neutron-l2-agent:               openvswitch
neutron-ovs-tenant-network-type:local
neutron-ovs-vlan-ranges:
neutron-ovs-bridge-mappings:    physnet1:1000:2000
neutron-ovs-bridge-interfaces:
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
os-swift-storage-size:          2G
provision-demo:                 n
provision-tempest:              n
provision-demo-floatrange:      192.168.32.0/22
provision-tempest-repo-uri:
https://github.com/openstack/tempest.git
provision-tempest-repo-revision:master
provision-all-in-one-ovs-bridge:n
os-heat-cloudwatch-install:     n
os-heat-cfn-install:            n
ceilometer-host:                192.0.43.10
mongodb-host:                   192.0.43.10
use-epel:                       n
additional-repo:
rh-username:                    admin@example.com
rh-password:                    ********
rh-beta-repo:                   n
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** when prompted, after the screen of new parameters is displayed.
- 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
```

27. 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 ]
Setting up ssh keys...                                  [ DONE ]
Discovering hosts' details...                           [ DONE ]
Disabling NetworkManager...                             [ DONE ]
Adding pre install manifest entries...                  [ DONE ]
Adding MySQL manifest entries...                        [ DONE ]
Adding AMQP manifest entries...                         [ DONE ]
Adding Keystone manifest entries...                     [ DONE ]
Adding Glance Keystone manifest entries...              [ DONE ]
Adding Glance manifest entries...                       [ DONE ]
Installing dependencies for Cinder...                   [ DONE ]
Adding Cinder Keystone manifest entries...              [ DONE ]
Adding Cinder 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 Neutron API manifest entries...                  [ DONE ]
Adding Neutron Keystone manifest entries...             [ DONE ]
Adding Neutron L3 manifest entries...                   [ DONE ]
Adding Neutron L2 Agent manifest entries...             [ DONE ]
Adding Neutron DHCP Agent manifest entries...           [ DONE ]
```

```
Adding Neutron Metadata Agent manifest entries...        [ DONE ]
Adding OpenStack Client manifest entries...              [ DONE ]
Adding Horizon manifest entries...                       [ DONE ]
Adding Swift Keystone manifest entries                  [ DONE ]
Adding Swift builder manifest entries                    [ DONE ]
Adding Swift proxy manifest entries                      [ DONE ]
Adding Swift storage manifest entries                   [ DONE ]
Adding Swift common manifest entries                     [ DONE ]
Adding Provisioning manifest entries                     [ DONE ]
Adding MongoDB manifest entries                          [ DONE ]
Adding Ceilometer manifest entries                       [ DONE ]
Adding Ceilometer Keystone manifest entries              [ DONE ]
Adding post install manifest entries                    [ DONE ]
Preparing servers                                        [ DONE ]
Installing Dependencies                                  [ DONE ]
Copying Puppet modules and manifests                     [ DONE ]
Applying Puppet manifests...
```

28. 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
```

29. 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 the user's home directory, and is given a file name containing the date and time, for example **~/packstack-answers-20140214-133303.txt**.

See Section 5.3, “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.

5.3. 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.

Important

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

See Section 2.1.2.2, “Customer Portal Subscription Management” for information on registering to Red Hat Network.

- You can ssh into each node and register it, attach an OpenStack subscription, and enable the required OpenStack repository. This will allow PackStack to install OpenStack on each node.
- Alternatively, you can get PackStack to generate an answer file, then edit the answer file to contain your Red Hat subscription username and password, using the parameters **CONFIG_RH_USER** and **CONFIG_RH_PW**. Running PackStack with this answer file configures each node to receive updates from Red Hat Network using Subscription Manager. This will allow PackStack to install OpenStack on each node.

5.3.1. 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 5.4. 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 5.5. 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.

5.3.2. 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 5.1. PackStack Answer File Configuration Keys

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_SSH_KEY	**--ssh-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.

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_MYSQL_INSTALL	**--mysql-install**	*y*	Set to **y** if you would like Packstack to install MySQL. **Note** PackStack is capable of installing a single MySQL database node. On the other hand, PackStack does not handle MySQL cluster installation, but it allows you to work with a MySQL cluster that you have set up separately. If you choose **n**, PackStack asks you for credentials and uses **CONFIG_MYSQL_HOST** as a given remote database.
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_NEUTRON_INSTALL	**--os-neutron-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**	*y*	Set to **y** if you would like PackStack to install Object Storage.

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_CEILOMETER_INSTALL	**--os-ceilometer-install**	*y*	Set to **y** if you would like PackStack to install the Telemetry service.
CONFIG_HEAT_INSTALL	**--os-heat-install**	*n*	Set to **y** if you would like PackStack to install the Orchestration service.
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**	*y*	Set to **y** if you would like to install Nagios. Nagios provides additional tools for monitoring the OpenStack environment.
EXCLUDE_SERVERS	**--exclude-servers**		If you have existing servers that have been configured previously, and you do not want PackStack to overwrite their configurations, you can specify a list of IP addresses of the servers to be excluded. Leave plain if you do not need to exclude any server.
CONFIG_DEBUG_MODE	**--os-debug-mode**	*n*	Set to **y** if you want to run OpenStack services in debug mode. Otherwise set to **n**.
CONFIG_CONTROLLER_HOST	**--os-controller-host**	*192.0.43.10*	The IP address of the server on which to install OpenStack services specific to the controller role, such as API servers, Horizon, etc.
CONFIG_COMPUTE_HOSTS	**--os-compute-hosts**	*192.0.43.10*	The list of IP addresses of the servers on which to install the Nova compute service.
CONFIG_NETWORK_HOSTS	**--os-network-hosts**	*192.0.43.10*	The list of IP addresses of the servers on which to install the network service such as Nova network or Neutron.

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_VMWARE_BACKEND	**--os-vmware**	*n*	Set to **y** if you want to use VMware vCenter as hypervisor and storage. Otherwise set to **n**.
CONFIG_VCENTER_HOST	**--vcenter-host**	*192.0.43.10*	The IP address of the VMware vCenter server.
CONFIG_VCENTER_USER	**--vcenter-username**		The username to authenticate to VMware vCenter server.
CONFIG_VCENTER_PASSWORD	**--vcenter-password**		The password to authenticate to VMware vCenter server.
CONFIG_VCENTER_CLUSTER_NAME	**--vcenter-cluster**		The name of the vCenter cluster.
CONFIG_USE_EPEL	**--use-epel**	*n*	To subscribe each server to EPEL enter *y*.
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_OPTIONAL	**--rhn-enable-optional**	*y*	To enable RHEL optional repos enter **y**.
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.

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_SATELLITE_USER	**--rhn-satellite-user**		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.

Configuration Key	Command Line Argument	Default Value	Description
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_PROFILE	**--rhn-satellite-profile**		Specify the profile name 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**. See 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_AMQP_BACKEND	**--amqp-backend**	*rabbitmq*	Set the AMQP service backend. Allowed values are: qpid, rabbitmq.
CONFIG_AMQP_HOST	**--amqp-host**	*192.0.43.10*	The IP address of the server on which to install the AMQP service.

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_AMQP_ENABLE_SSL	**--amqp-enable-ssl**	*n*	Enable SSL for the AMQP service. Set to **y** to enable SSL.
CONFIG_AMQP_ENABLE_AUTH	**--amqp-enable-auth**	*n*	Enable authentication for the AMQP service. Set to **y** to enable authentication.
CONFIG_AMQP_NSS_CERTDB_PW	**--amqp-nss-certdb-pw**		The password for the NSS certificate database of the AMQP service.
CONFIG_AMQP_SSL_PORT	**--amqp-ssl-port**	*5671*	The port on which the AMQP service listens for SSL connections.
CONFIG_AMQP_SSL_CERT_FILE	**--amqp-ssl-cert-file**	*/etc/pki/tls/certs/amqp_selfcert.pem*	The filename of the certificate that the AMQP service is going to use.
CONFIG_AMQP_SSL_KEY_FILE	**--amqp-ssl-key-file**	*/etc/pki/tls/private/amqp_selfkey.pem*	The filename of the private key that the AMQP service is going to use.
CONFIG_AMQP_SSL_SELF_SIGNED	**--amqp-ssl-self-signed**	*y*	Automatically generates a self signed SSL certificate and key. Set to **y** to automatically generate an SSL certificate and key.
CONFIG_AMQP_AUTH_USER	**--amqp-auth-user**	*amqp_user*	The user for AMQP authentication.
CONFIG_AMQP_AUTH_PASSWORD	**--amqp-auth-password**		The password used for AMQP authentication.
CONFIG_MYSQL_HOST	**--mysql-host**	*192.0.43.10*	The IP address of the server on which to install MySQL, or alternatively the IP address of the DB server to use if MySQL installation was not selected.
CONFIG_MYSQL_USER	**--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_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.

Configuration Key	Command Line Argument	Default Value	Description
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_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_KEYSTONE_TOKEN_FORMAT	**--keystone-token-format**	*PKI*	PackStack allows a choice of the token format to be used by Identity, either PKI or UUID. The recommended format for new deployments is PKI.
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_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_BACKEND	**--cinder-backend**	*lvm*	Cinder backend to use, valid options are: lvm, gluster, nfs

Configuration Key	Command Line Argument	Default Value	Description
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. **Note** This should only be done for testing on a proof-of-concept installation of Volume service. This will create a file-backed volume group and is not suitable for production usage. 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). Actual volume size will be extended with 3% more space for VG metadata.
CONFIG_CINDER_GLUSTER_MOUNTS	**--cinder-gluster-mounts**		A single or comma separated list of gluster volume shares to mount, for example: ip-address:/vol-name, domain:/vol-name

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_CINDER_NFS_MOUNTS	**--cinder-nfs-mounts**	*20G*	A single or comma separated list of NFS exports to mount, eg: ip-address:/export-name
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_SCHED_CPU_ALLOC_RATIO	**--novasched-cpu-allocation-ratio**	*16.0*	The overcommitment 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_NOVA_COMPUTE_PRIVIF	**--novacompute-privif**	*eth1*	Private interface for Flat DHCP on the Compute servers.
CONFIG_NOVA_NETWORK_MANAGER	**--novanetwork-manager**		Compute Network Manager.
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_NETWORK_VLAN_START	**--novanetwork-vlan-start**		First VLAN for private networks.
CONFIG_NOVA_NETWORK_NUMBER	**--novanetwork-num-networks**	*n*	Number of networks to support.
CONFIG_NOVA_NETWORK_SIZE	**--novanetwork-network-size**	*n*	Number of addresses in each private subnet.

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_NEUTRON_USE_NAMESPACES	**--os-neutron-use-namespaces**	*y*	Enable network namespaces for OpenStack Networking.
CONFIG_NEUTRON_KS_PW	**--os-neutron-ks-password**		The password to use for OpenStack Networking to authenticate with Identity.
CONFIG_NEUTRON_DB_PW	**--os-neutron-db-password**		The password to use for OpenStack Networking to access database.
CONFIG_NEUTRON_L3_EXT_BRIDGE	**--os-neutron-l3-ext-bridge**	*br-ex*	The name of the bridge that the OpenStack Networking L3 agent will use for external traffic. Use **provider** if you intend to use a provider network to handle external traffic.
CONFIG_NEUTRON_L2_PLUGIN	**--os-neutron-l2-plugin**	*openvswitch*	The name of the L2 plugin to be used with OpenStack Networking.
CONFIG_NEUTRON_METADATA_PW	**--os-neutron-metadata-pw**		Password for OpenStack Networking metadata agent.
CONFIG_LBAAS_INSTALL	**--os-neutron-lbaas-install**	*n*	Set to **y** if you would like PackStack to install OpenStack Networking LBaaS.
CONFIG_NEUTRON_METERING_AGENT_INSTALL	**--os-neutron-metering-agent-install**	*n*	Set to **y** if you would like PackStack to install OpenStack Networking Metering agent.
CONFIG_NEUTRON_FWAAS	**--neutron-fwaas**	*n*	Set to **y** if you would like to configure Neutron firewall as a service.
CONFIG_NEUTRON_ML2_TYPE_DRIVERS	**--os-neutron-ml2-type-drivers**	*vxlan*	A comma separated list of network type driver entry points to be loaded from the neutron.ml2.type_drivers namespace.
CONFIG_NEUTRON_ML2_TENANT_NETWORK_TYPES	**--os-neutron-ml2-tenant-network-types**	*vxlan*	A comma separated ordered list of network types to allocate as tenant networks. The value **local** is only useful for single-box testing but provides no connectivity between hosts.

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_NEUTRON_ML2_MECHANISM_DRIVERS	**--os-neutron-ml2-mechanism-drivers**	*openvswitch*	A comma separated ordered list of networking mechanism driver entry points to be loaded from the neutron.ml2.mechanism_drivers namespace.
CONFIG_NEUTRON_ML2_FLAT_NETWORKS	**--os-neutron-ml2-flat-networks**	*	A comma separated list of physical_network names with which flat networks can be created. Use ***** to allow flat networks with arbitrary physical network names.
CONFIG_NEUTRON_ML2_VLAN_RANGES	**--os-neutron-ml2-vlan-ranges**		A comma separated list of **physical_network:vlan_min:vlan_max** or **physical_network** specifying physical_network names usable for VLAN provider and tenant networks, as well as ranges of VLAN tags on each available for allocation to tenant networks.
CONFIG_NEUTRON_ML2_TUNNEL_ID_RANGES	**--os-neutron-ml2-tunnel-id-ranges**		A comma separated list of **tun_min:tun_max** tuples enumerating ranges of GRE tunnel IDs that are available for tenant network allocation. It must be an array with tun_max +1 - tun_min > 1,000,000
CONFIG_NEUTRON_ML2_VXLAN_GROUP	**--os-neutron-ml2-vxlan-group**		Multicast group for VXLAN. If not set, disables VXLAN from sending allocated broadcast traffic to this multicast group. When left unconfigured, will disable multicast VXLAN mode. Must be a Multicast IP (v4 or v6) address.
CONFIG_NEUTRON_ML2_VNI_RANGES	**--os-neutron-ml2-vni-ranges**	*n:m*	A comma separated list of **vni_min:vni_max** tuples enumerating ranges of VXLAN VNI IDs that are available for tenant network allocation. Minimum value is 0 and maximum value is 16777215.

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_NEUTRON_L2_AGENT	**--os-neutron-l2-agent**	*openvswitch*	The name of the L2 agent to be used with OpenStack Networking.
CONFIG_NEUTRON_LB_TENANT_NETWORK_TYPE	**--os-neutron-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_NEUTRON_LB_VLAN_RANGES	**--os-neutron-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. (For example: physnet1:1:4094,physnet2, physnet3:3000:3999)
CONFIG_NEUTRON_LB_INTERFACE_MAPPINGS	**--os-neutron-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_NEUTRON_OVS_TENANT_NETWORK_TYPE	**--os-neutron-ovs-tenant-network-type**	*vxlan*	The type of network to allocate for tenant networks. Supported values are **vlan, local, gre, vxlan**.

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_NEUTRON_OVS_VLAN_RANGES	**--os-neutron-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. (For example physnet1:1:4094,physnet2, physnet3:3000:3999)
CONFIG_NEUTRON_OVS_BRIDGE_MAPPINGS	**--os-neutron-ovs-bridge-mappings**	*physnet1:br-eth1*	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. (For example physnet1:br-eth1,physnet2:br-eth2,physnet3:br-eth3)
CONFIG_NEUTRON_OVS_BRIDGE_IFACES	**--os-neutron-ovs-bridge-interfaces**		A comma separated list of colon-separated OVS bridge:interface pairs. The interface will be added to the associated bridge.
CONFIG_NEUTRON_OVS_TUNNEL_RANGES	**--os-neutron-ovs-tunnel-ranges**		A comma separated list of tunnel ranges for the Neutron openvswitch plugin.
CONFIG_NEUTRON_OVS_TUNNEL_IF	**--os-neutron-ovs-tunnel-if**		The interface for the OVS tunnel. Packstack will override the IP address used for tunnels on this hypervisor to the IP found on the specified interface. (eg. eth1).

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_NEUTRON_OVS_VXLAN_UDP_PORT	**--os-neutron-ovs-vxlan-udp-port**	*4789*	The vxlan UDP port.
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_SSL_CACHAIN	**--os-ssl-key**		PEM encoded CA certificates from which the certificate chain of the server certificate can be assembled.
CONFIG_SWIFT_KS_PW	**--os-swift-ks-passwd**		The password to use for Object Storage to authenticate with Identity. This value is randomly generated if you do not provide it.
CONFIG_SWIFT_STORAGES	**--os-swift-storages**		A comma separated list of devices which to use as Object Storage device. Each entry must take the format /path/to/dev, for example /dev/vdb will install /dev/vdb as Object Storage device (PackStack does not create the filesystem, you must do this first). If the value is omitted PackStack will create a loopback device for a test setup.
CONFIG_SWIFT_STORAGE_ZONES	**--os-swift-storage-zones**	*1*	Number of Object 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 Object Storage replicas, this number **must** be no bigger than the number of storage zones configured.

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_SWIFT_STORAGE_FSTYPE	**--os-swift-storage-fstype**	*ext4*	FileSystem type for storage nodes. Supported values are **ext4** and **xfs** at this time.
CONFIG_SWIFT_HASH	**--os-swift-hash**		The custom seed number following this command will be used to set swift_hash_suffix in swift.conf.
CONFIG_SWIFT_STORAGE_SIZE	**--os-swift-storage-size**	*2G*	The size of the Object Storage loopback file storage device.
CONFIG_PROVISION_DEMO	**--provision-demo**	*n (y for allinone)*	PackStack can provision for demo usage and testing. This key selects whether to provision demo OpenStack Networking networks, subnets and routers. Set to **y** if you want to provision for demo usage and testing. It requires CONFIG_NEUTRON_INSTALL=**y** and CONFIG_NEUTRON_USE_NAMESPACES=**y**. CONFIG_PROVISION_DEMO will be enabled if you run **packstack --allinone** and CONFIG_NEUTRON_INSTALL=**y**, which it is by default.
CONFIG_PROVISION_DEMO_FLOATRANGE	**--provision-demo-floatrange**		The CIDR network address for the floating IP subnet.
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_NEUTRON_INSTALL=**y** and CONFIG_NEUTRON_USE_NAMESPACES=**y**
CONFIG_PROVISION_TEMPEST_USER	**--provision-tempest-user**		The name of the Tempest Provisioning user. If you do not provide a user name, Tempest will be configured in a standalone mode.

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_PROVISION_TEMPEST_USER_PW	**--provision-tempest-user-passwd**		The password to use for the Tempest Provisioning user.
CONFIG_PROVISION_TEMPEST_REPO_URI	**--provision-tempest-repo-uri**		The uri of the tempest git repository to use.
CONFIG_PROVISION_TEMPEST_REPO_REVISION	**--provision-tempest-repo-revision**		The revision of the tempest git repository to use.
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_NEUTRON_INSTALL=**y**, which it is by default.
CONFIG_HEAT_DB_PW	**--os-heat-mysql-password**		The password used by Heat user to authenticate against MySQL.
CONFIG_HEAT_AUTH_ENC_KEY	**--heat-auth-encryption-key**		The encryption key to use for authentication info in database.
CONFIG_HEAT_KS_PW	**--os-heat-ks-passwd**		The password to use for the Heat service to authenticate with Identity.
CONFIG_HEAT_CLOUDWATCH_INSTALL	**--os-heat-cloudwatch-install**	*n*	Set to *y* if you would like Packstack to install Heat CloudWatch API.
CONFIG_HEAT_CFN_INSTALL	**--os-heat-cfn-install**	*n*	Set to *y* if you would like Packstack to install Heat CloudFormation API.
CONFIG_HEAT_DOMAIN	**--os-heat-domain**	*heat*	The name of the Identity domain for Orchestration.
CONFIG_HEAT_DOMAIN_ADMIN	**--os-heat-domain-admin**	*heat-admin*	The name of the Identity domain admin user for Orchestration.
CONFIG_HEAT_DOMAIN_PASSWORD	**--os-heat-domain-password**		The password for the Identity domain admin user for Orchestration.
CONFIG_CEILOMETER_SECRET	**--ceilometer-secret**		Secret key for signing Telemetry messages.

Configuration Key	Command Line Argument	Default Value	Description
CONFIG_CEILOMETER_KS_PW	**--ceilometer-ks-passwd**		The password to use for Ceilometer to authenticate with Identity.
CONFIG_MONGODB_HOST	**--mongodb-host**	*192.0.43.10*	The IP address of the server on which to install MongoDB.
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.

5.3.3. 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 5.5. 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 5.6. 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 6. PackStack and Passwords

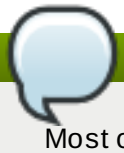
When PackStack deploys OpenStack, it generates passwords for each of the services. You will be using a subset of these passwords for authentication. This chapter describes the location of the passwords and also the steps to be followed in order to change them.

6.1. Password Locations

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

Table 6.1. PackStack Passwords locations

Service	Location of the Passwords
Identity	**~/keystonerc_admin**
Compute	**/etc/nova/nova.conf**
OpenStack Networking	**/etc/neutron/neutron.conf**
Image	**/etc/glance/glance-api.conf**
Block Storage	**/etc/cinder/cinder.conf**
Object Storage	**/etc/swift/proxy-server.conf**
MySQL Database	**~/.my.cnf**
Telemetry	**/etc/ceilometer/ceilometer.conf**
Orchestration	**/etc/heat/heat.conf**
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 the Image service, **sql_connection = mysql://glance:12345678abcdefgh@192.0.43.10/glance** where,

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

6.2. 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

```
# mysqladmin -u root -pOLDPASS password NEWPASS
```

Replace *OLDPASS* with the existing password, leaving no space between **-p** and the password, and *NEWPASS* with the new password.

- Nagios

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

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

To change Identity's authentication passwords on the Identity server, use the following command:

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

Replace *USERNAME* with the name of the service-user you want to change the password for. You will have to enter the new password when prompted.

The locations of Identity's authentication passwords for OpenStack services are shown in the table in Section 6.1, “Password Locations” .

Important

Using the **user-password-update** command changes the passwords only on the Identity server. Ensure that you also manually 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 the Dashboard.

With the goal of launching an instance in mind, this chapter discusses the procedures to create a keypair, upload an image, create a network with a subnet, launching an instance and creating a volume.

For more procedures on using the Dashboard and CLI clients, refer *Red Hat Enterprise Linux OpenStack Platform 5 End User Guide* and *Red Hat Enterprise Linux OpenStack Platform 5 Admininistration User Guide* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

Chapter 7. A First Start: Launching an Instance

7.1. Accessing the Dashboard

Dashboard is the web-based UI for managing an OpenStack installation. To access the Dashboard you must have first:

- Installed the Dashboard service.
- Obtain the hostname and login password.
  - **HOSTNAME** is the host name or IP address of the server on which you installed the Dashboard.
  - For logging in as **admin**, the password stored as **export OS_PASSWORD=** in the **~/keystonerc_admin** file.
  - If you have enabled the **demo** Identity (Keystone) tenant, for example by running **packstack --allinone**, you should log into the Dashboard using the **demo** account instead of the **admin**. The **demo** password is stored as **export OS_PASSWORD=** in the **~/keystonerc_demo** file.

Procedure 7.1. 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.

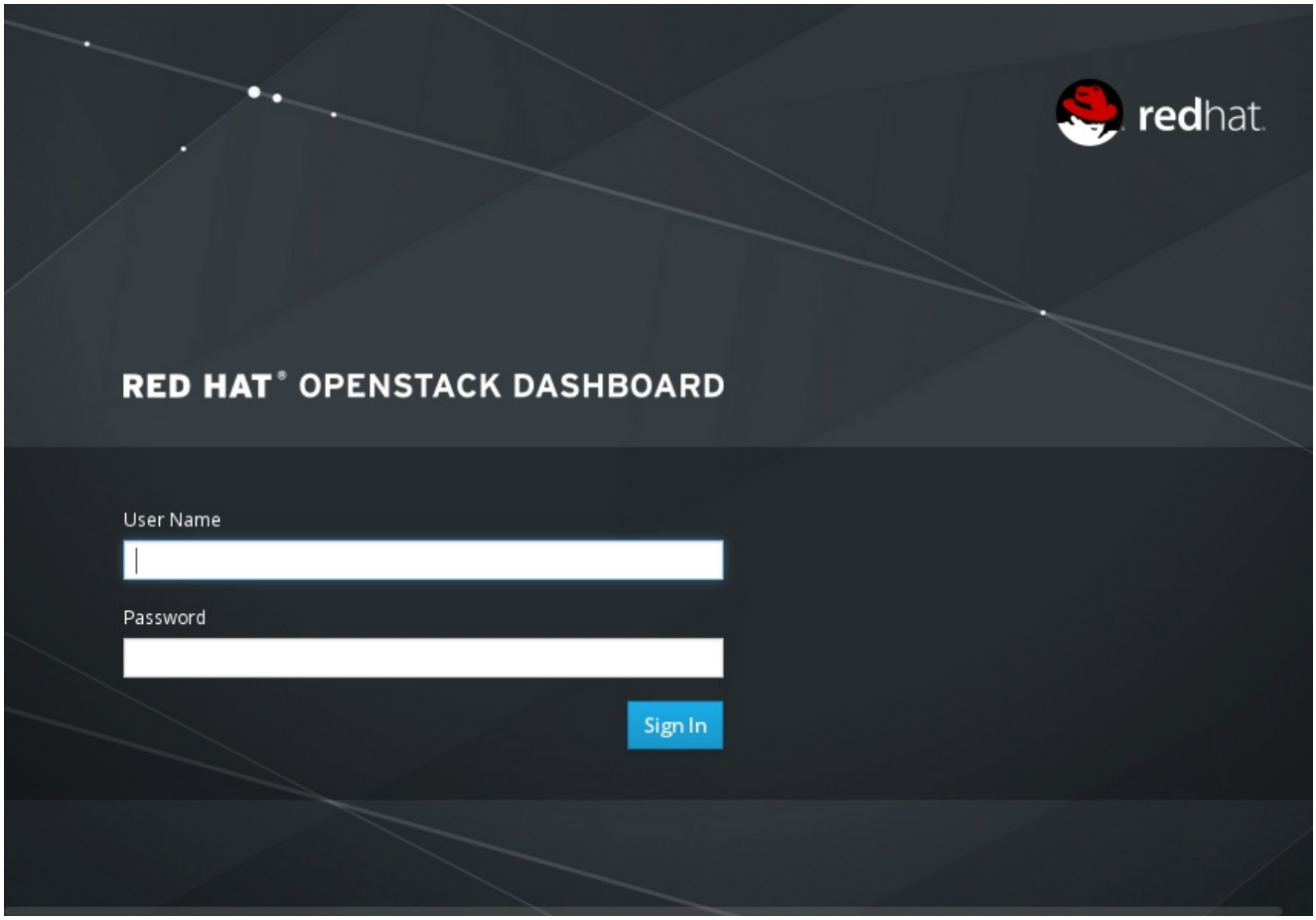


Figure 7.1. Log In Screen

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

The Overview page of the Dashboard displays. This page displays the **`Limit Summary`**, **`Usage Summary`** and **`Usage`**.

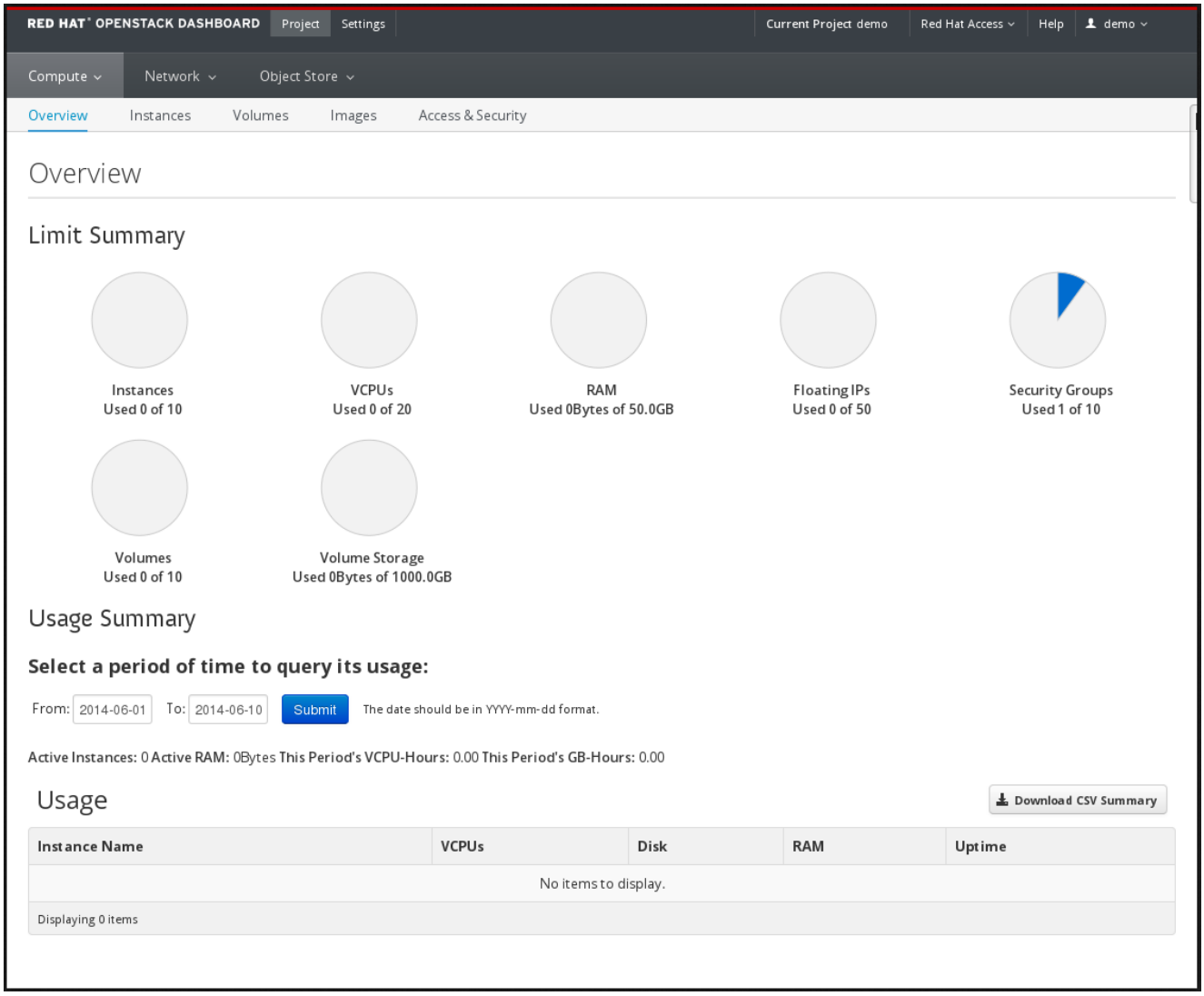


Figure 7.2. Dashboard - Project Tab

7.2. Uploading a Disk Image

To launch instances based on images stored in the OpenStack Image service, you must first have either 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'.

Note

Deploying OpenStack using **`packstack --allinone`** creates a default **`cirros`** image. You can use this to launch instances.

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

- Installed the Dashboard service.

Procedure 7.2. Uploading an Image using the Dashboard

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

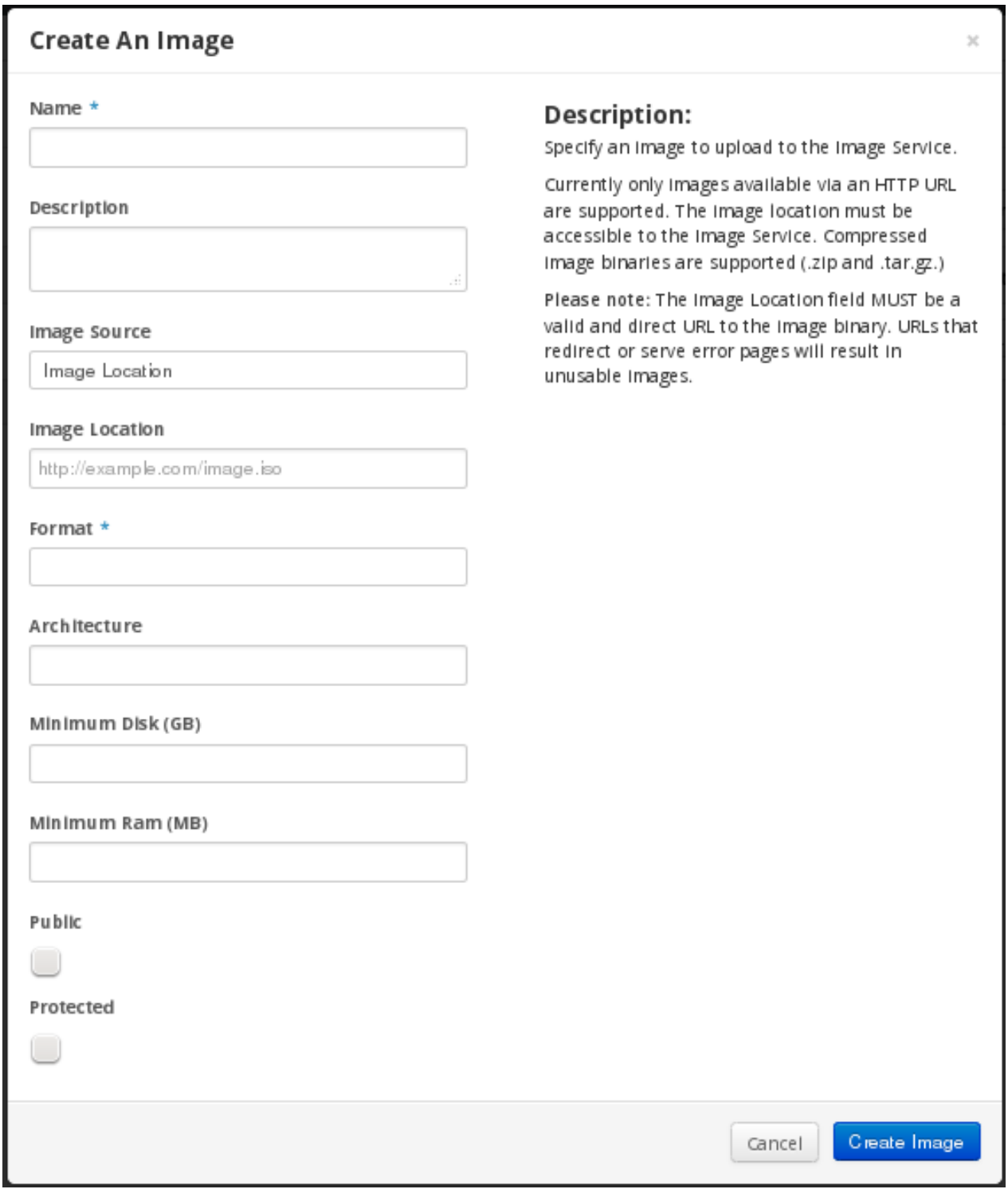


Figure 7.3. Create An Image Dialog

4. Configure the settings that define your image.
    a. Enter a name for the image.
    b. Enter a brief description about the image.

c. Choose the image source from the dropdown list. Your choices are **Image Location** and **Image File**.

d. Based on your selection, there is an **Image Location** or **Image File** field. You can include the location URL or browse to the image file on your file system and include it.

e. Select the correct type from the dropdown menu in the **Format** field (for example, **QCOW2**).

f. Leave the **Minimum Disk (GB)** and **Minimum RAM (MB)** fields empty.

These values provide guidance to OpenStack as to the appropriate default *flavor* (for example, tiny or medium) for new instances using this image. These fields are optional.

g. Select the **Public** checkbox, to let other users access the image.

h. Select the **Protected** checkbox, to keep the image protected from being accidentally deleted.

i. Click the **Create Image** button.

You have successfully uploaded an image. 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**.

7.3. 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 7.3. 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.

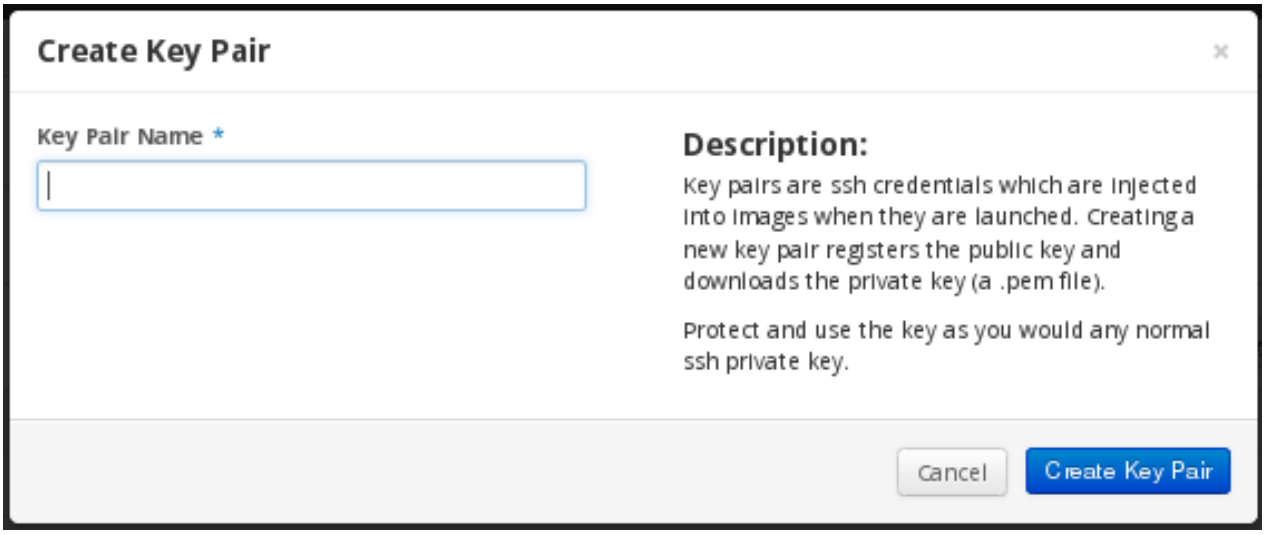


Figure 7.4. Create Keypair

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.

7.4. Creating an OpenStack Network

This procedure lets you use the dashboard to create a virtual network for use by your instances. After deploying OpenStack with **packstack --allinone**, a default **private** network is created. You can use that network for creating an instance or create a new one.

By default, **packstack --allinone** installs OpenStack Networking (neutron) service. For information on Nova Networking, see *Deploying OpenStack: Learning Environments (Manual Setup)* at https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/.

Note

You do not have to initially specify a subnet. But to launch an instance, a network with subnet should be created. So this procedure discusses the steps to create subnet.

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

- Installed the Dashboard service.
- Installed OpenStack Networking services.

Procedure 7.4. Creating a Network Using the Dashboard

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

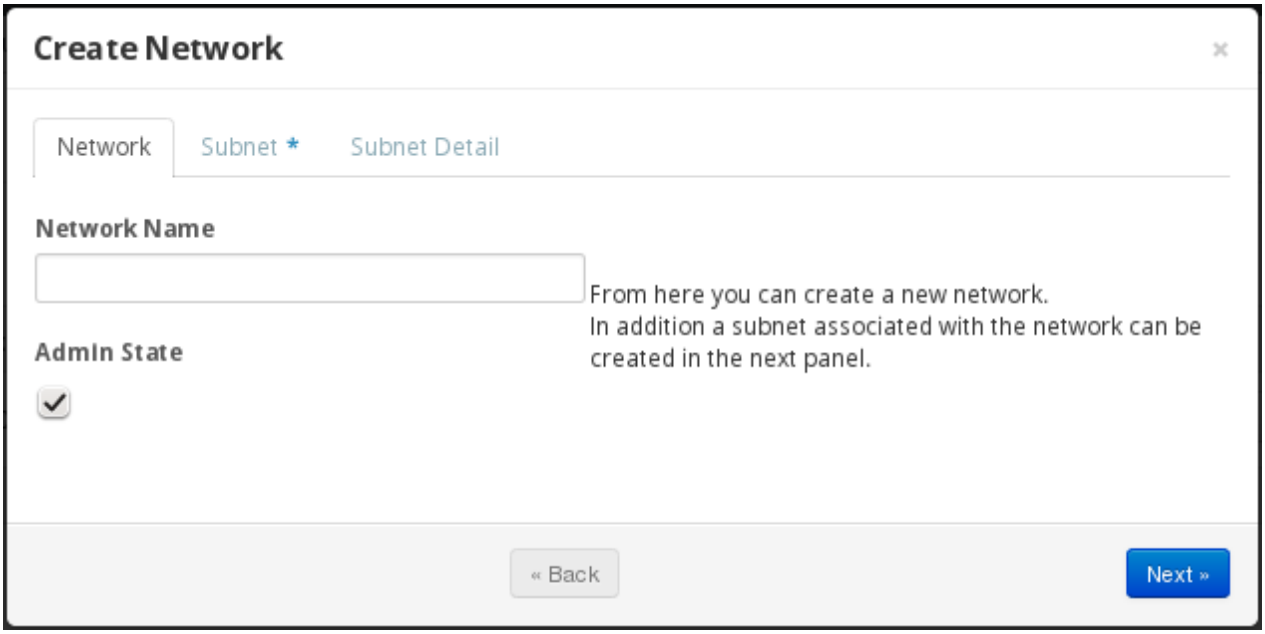


Figure 7.5. Create Network: Network Tab

4. By default, the dialog opens to the **Network** tab. You have the option of specifying a network name.

Note

If you do not define a specific subnet, clear the **Create Subnet** check box.

5. Click on the **Subnet** tab and enter the following values:
    a. Enter the **Subnet** name.
    b. Enter the IP address for the network.
    c. Select the **IP Version** from the drop down list.
    d. Enter the **Gateway IP** address.
6. To specify additional attributes for the subnet, click on the **Subnet Detail** tab and enter allocation pools, DNS name servers and host routes.
7. Click the **Create** button.

You have successfully created a new network. Use the **Network Topology** tab to verify the newly created network and subnets.

7.5. Launching an Instance

This procedure discusses how to use the dashboard to create a new virtual machine instance. To launch an instance from the Dashboard you must have first:

- Installed the Dashboard service.
- Have an image available. For more information, see Section 7.2, “Uploading a Disk Image”.

- Have a network available. For more information, see Section 7.4, "Creating an OpenStack Network".

Procedure 7.5. Launching an Instance using the Dashboard

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

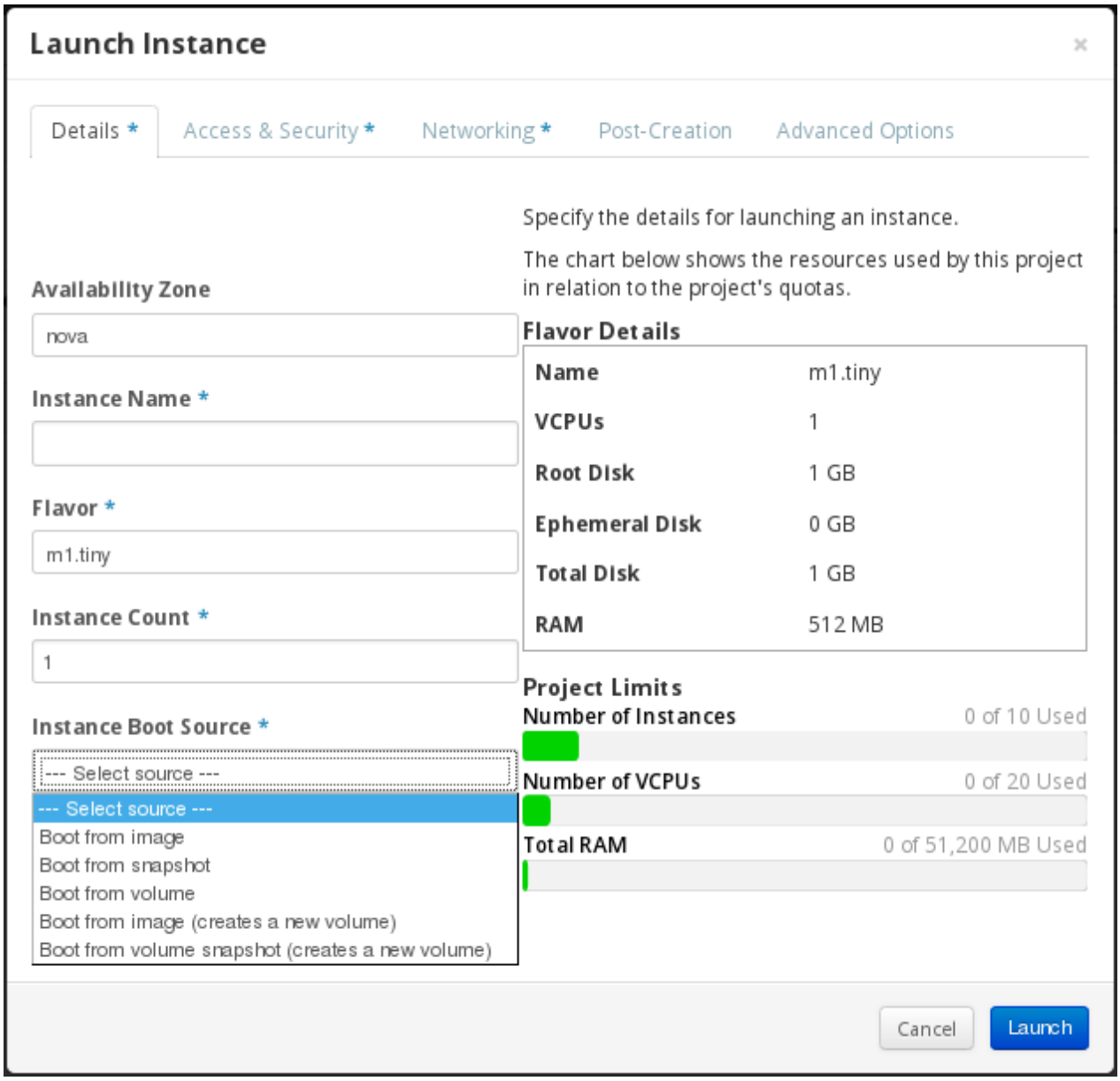
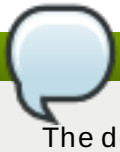


Figure 7.6. Launch Instance: Details Tab

4. By default, the dialog opens to the **Details** tab.
   a. Select an **Availability Zone** for your instance from the dropdown list. The default option is **nova**.
   b. Enter an **Instance Name** to identify your instance.

c. Select a **Flavor** for your instance. The flavor determines the compute resources available to your instance. After a flavor is selected, their resources are displayed in the **Flavor Details** pane for preview.



Note

The default flavor is selected based on the minimum RAM and CPU settings on the selected image.

d. Enter an **Instance Count**. This determines how many instances to launch using the selected options.

e. Select a boot option from the **Instance Boot Source** dropdown list.

You have the following boot options:

- **Boot from image** - If you choose this option, a new field for **Image Name** is displayed. You can select the image from the dropdown list.
- **Boot from snapshot** - If you choose this option, a new field for **Instance Snapshot** is displayed. You can select the snapshot from the dropdown list.
- **Boot from volume** - If you choose this option, a new field for **Volume** is displayed. You can select the volume from the dropdown list.
- **Boot from image (creates a new volume)** - With this option, you can boot from an image and create a volume by choosing **Device Size** and **Device Name** for your volume. Click the **Delete on Terminate** option to delete the volume on terminating the instance.
- **Boot from volume snapshot (creates a new volume)** - Using this option, you can boot from a volume snapshot and create a new volume by choosing **Volume Snapshot** from a dropdown list and adding a **Device Name** for your volume. Click the **Delete on Terminate** option to delete the volume on terminating the instance.

5. Click the **Access & Security** tab and configure the security settings for your instance.

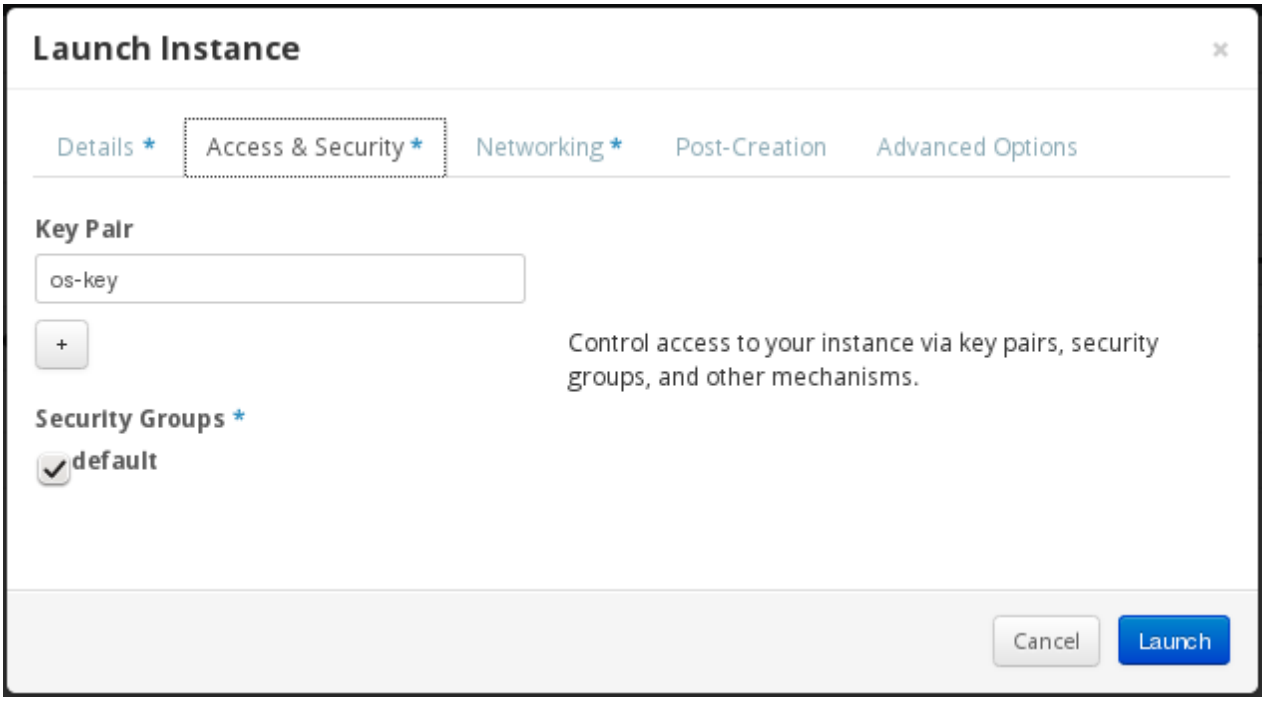


Figure 7.7. Launch Instance: Access & Security Tab

a. Select an existing keypair from the **Keypair** dropdown box or click the **+** button to upload a new keypair. For more information, see Section 7.3, "Creating a 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** Identity (Keystone) tenant, choose the **private** network.

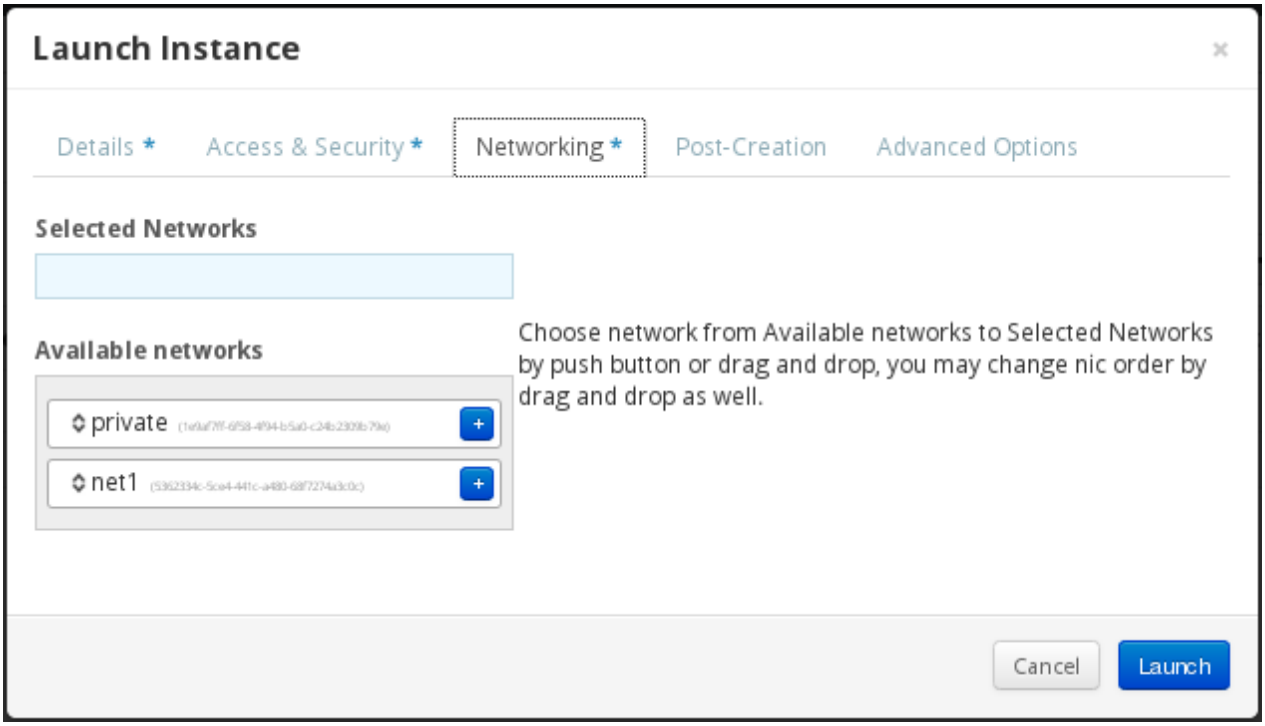


Figure 7.8. Launch Instance: Networking Tab

7. Click the **Launch** button.

You have just created a Compute instance.

To view 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.

A VNC session to the instance's console is run within the browser.

7.6. 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 service.
- Installed the Block Storage service.

Procedure 7.6. 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.

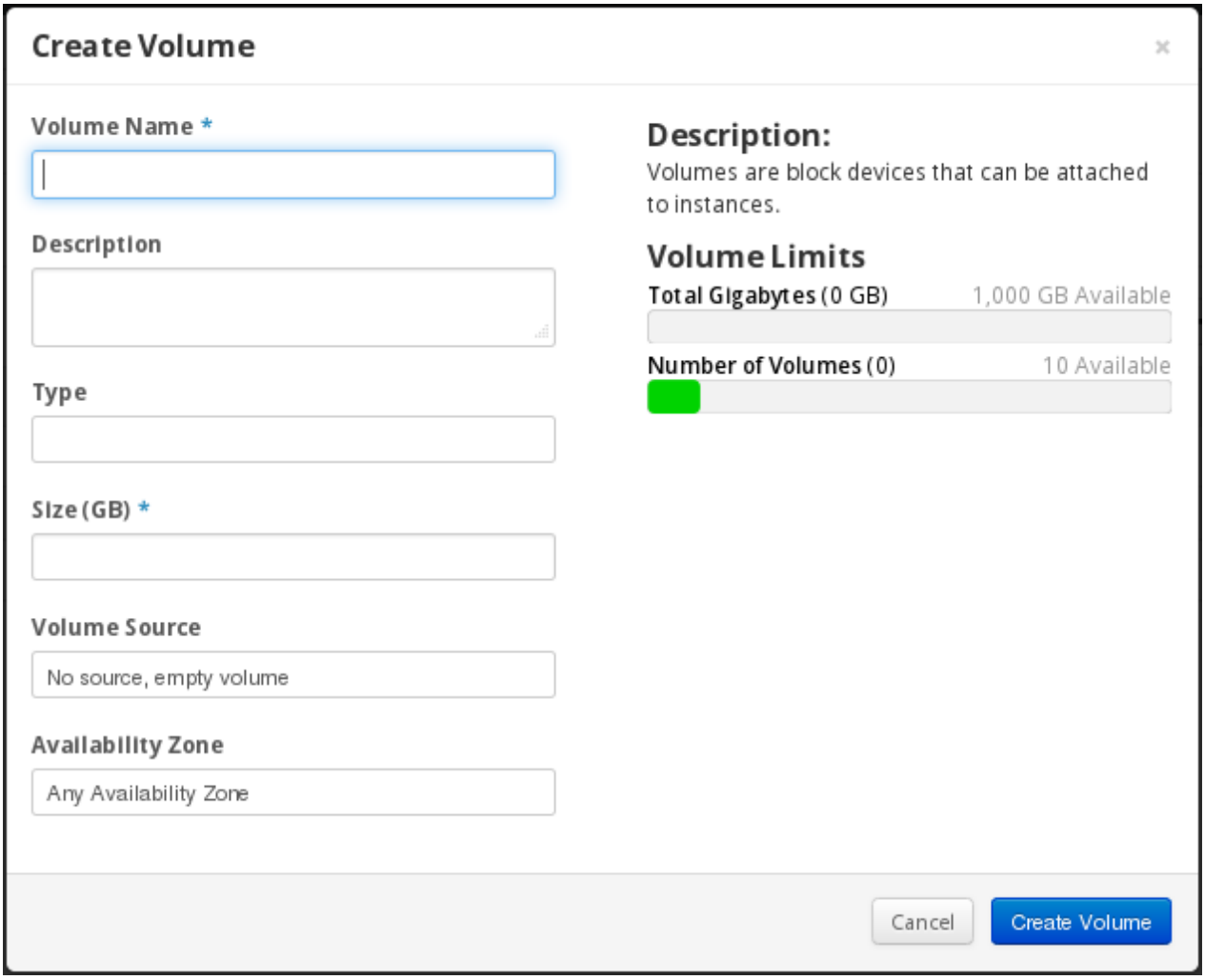


Figure 7.9. Create Volume Dialog

4. Configure the values that will define your new volume.
    a. Enter a **Volume Name** to identify your new volume.
    b. Enter a **Description** to further describe your new volume.
    c. Enter the **Size** of your new volume in gigabytes (GB).
    d. Choose a source for your volume from the **Volume Source** dropdown list. You can choose to load an empty volume or from an image.

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 volume using the Dashboard.

Removing PackStack Deployments

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 use either of the following scripts:

- The first procedure 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.

Important

These procedures must be carried out on all OpenStack hosts as a root user.

Note

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.

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.

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

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

```
#!/usr/bin/bash
# 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, including many
```

```
# which may be unrelated to RDO.
yum remove -y nrpe "*nagios*" puppet ntp ntp-perl ntpdate
"*openstack*" \
"*nova*" "*keystone*" "*glance*" "*cinder*" "*swift*" \
mysql mysql-server httpd "*memcache*" scsi-target-utils \
iscsi-initiator-utils perl-DBI perl-DBD-MySQL ;

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

# 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 ;

umount /srv/node/device* ;
killall -9 dnsmasq tgtd httpd ;
setenforce 1 ;
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.

Important

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

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

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

```
#!/usr/bin/bash
# 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*" "*rdo-release*";

# Optional - makes database cleanup cleaner.
```

```
# If you do this bit, the database cleanup stuff below is superfluous.
# yum remove -y "*mysql*"

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

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 /tmp/keystone-signing-nova ;

# Ensure there is a root user and that we know the password
service mysql stop
cat > /tmp/set_mysql_root_pwd << EOF
UPDATE mysql.user SET Password=PASSWORD('MyNewPass') WHERE User='root';
FLUSH PRIVILEGES;
EOF

# mysql cleanup
/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 5.0.0-31 **Thu Apr 16 2015** **Summer Long**

Updated base OS from 7.0 to 7.1.

Revision 5.0.0-30 **Wed Feb 11 2015** **Andrew Dahms**

Corrected the description of configuring channels.

Revision 5.0.0-29 **Wed Feb 11 2015** **Andrew Dahms**

Corrected a description of the product version.

Revision 5.0.0-28 **Wed Feb 11 2015** **Andrew Dahms**

Updated the link to the article outlining supported guest operating systems.

Revision 5.0.0-27 **Tue Jan 20 2015** **Andrew Dahms**

Corrected a reference to the Red Hat Enterprise Linux OpenStack Platform installer.

Revision 5.0.0-26 **Fri Jan 16 2015** **Don Domingo**

BZ#1122258 - Updated the syntax of subscription manager commands.

Revision 5.0.0-25 **Tue Oct 14 2014** **Andrew Dahms**

BZ#1150796 - Removed an unnecessary note stating a restriction on changing database user passwords.

Revision 5.0.0-24 **Tue Oct 14 2014** **Martin Lopes**

BZ#1124984 - Added procedure for updating Qpid packages.

Revision 5.0.0-23 **Fri Oct 03 2014** **Andrew Dahms**

BZ#1145397 - Updated the command for changing a database user password.

Revision 5.0.0-22 **Fri Aug 29 2014** **Bruce Reeler**

BZ#1134234 - Corrected description of CONFIG_VMWARE_BACKEND in table PackStack Answer Fi Configuration Keys.

Revision 5.0.0-21 **Fri Aug 22 2014** **Bruce Reeler**

BZ#1131744 - Corrected Notes in Orchestration and Telemetry steps in Running PackStack Interactively section.
BZ#1131798 - Changed section on PackStack and Passwords, updated refs to table of config files, updated notes and clarified some steps.
BZ#1127042 - Replaced references to installation guides with new titles and changed this document's title from GSG to Deploying OS: PoC Environments (PackStack).

Revision 5.0.0-20 **Thu Jul 24 2014** **Andrew Dahms**

BZ#1120446 - Added a reference to the Red Hat Enterprise Linux OpenStack Platform Installer and Foreman Guide.

Revision 5.0.0-19 **Fri Jul 18 2014** **Bruce Reeler**

BZ#1118112 and BZ#1118110 - Updated table ""PackStack Answer File Configuration Keys" in section "Running PackStack Non-interactively". Added note that RabbitMQ is recommended AMQP service.

Revision 5.0.0-18 **Mon Jul 7 2014** **Don Domingo**

Red Hat Enterprise Linux OpenStack 5.0 (for Red Hat Enterprise Linux 7.0) GA release build.

Revision 5.0.0-17 Fri Jul 4 2014 Bruce Reeler

BZ#1105430 - Edited multiple sections based on doc QE. Amongst others: Removed OpenStack service details section. Moved 'Supported Virtual Machine Operating Systems' section to Product Requirements. Changed Configure Software Repositories. Replaced chkconfig with systemctl in 'Disable Network Manager' section.
BZ#1040519 - Standardized use of subscription-manager and yum-config-manager.
BZ#1093198 - Changed from Qpid to rabbitMQ as default message broker.
BZ#1105440 - Multiple edits and fixes to 'Using PackStack' and appendices based on doc QE.

Revision 5.0.0-13 Tue Jun 24 2014 Don Domingo

Applied new recommended publication settings based on new documentation portal features.

Revision 5.0.0-12 Mon Jun 23 2014 Don Domingo

Applied new recommended publication settings based on new supported build features.

Revision 5.0.0-11 Wed Jun 18 2014 Don Domingo

Applied recommended build settings as per updates to https://access.redhat.com/site/documentation/.

Revision 5.0.0-10 Tue Jun 17 2014 Bruce Reeler

BZ#1065160 - Updated Abstract to summarize that GSG is for PackStack and what the outcome is after running PackStack.
BZ#1040519 - Updated the section on configuring software repositories to be more consistent using configuration-manager and changed procedure names and steps to be more consistent.
BZ#1064085 - Corrected various typos and ambiguities found during translation.

Revision 5.0.0-9 Mon May 26 2014 Bruce Reeler

BZ#964184 - Added examples of options when running PackStack interactively.
BZ#1022873 - Added option of configuring a custom hash for Object Storage instead of the hard coded one.
BZ#1038865 - Updated storage location of answer file.
BZ#1064044 - Fixed table formatting problems.
BZ#1066192 - Added deployment utilities to Product Introduction section.
BZ#1071107 - Updated Dashboard output when creating a network.
BZ#1072204 - Included a note in the 'Launching an Instance' section to indicate that the flavors are selected based on the image source.
BZ#1084834 - Added Trove as tech preview in introduction.
BZ#1095246 - Updated Dashboard screen capture.
BZ#1092694 - Added information about using the Answer File to manually configure storage.

Revision 5.0.0-8 Mon May 12 2014 Deepti Navale

BZ#1072204 - Included a note regarding the selection of flavors based on resources allocated in th image.
BZ#1095246 - Updated the dashboard image in section 'Dashboard Service' in the intro chapter. Updates to the 'Using OpenStack with the Dashboard' chapter based on RHEL OpenStack Platform 5 changes.
BZ#1071107 - Updated 'Note' in the Creating a Network section. When subnet is not created, the status of the network still changes to Active.

Revision 5.0.0-5 Fri May 9 2014 Bruce Reeler

BZ#1022873 - Added CONFIG_SWIFT_HASH key and config option --os-swift-hash, for creating a custom hash, to table of PackStack Configuration Keys in answerfile.
BZ#964184 - Added examples of interfaces and bridges to Running PackStack Interactively section
BZ#1066192 - Added section on deployment utilities.
BZ#1092687 - Added missing instruction in Procedure 2.1 Disabling NetworkManager to add NM_CONTROLLED key, corrected typos.

Revision 5.0.0-3 Tue May 6 2014 Summer Long

BZ#1092694 - Restructured and updated Configuring Storage section.

Revision 5.0.0-2 Thu Apr 24 2014 Summer Long

Red Hat Enterprise Linux OpenStack Platform 5.0 - Beta preview.