Red Hat Reference Architecture Series

Deploying Highly Available Red Hat Enterprise Linux OpenStack Platform 6 with Red Hat Ceph Storage

Using Red Hat Enterprise Linux OpenStack Platform Installer

Keith Schincke

Senior Software Engineer

RHCA

Balaji Jayavelu

Principal Software Engineer

RHCE

Version 1.2

June 2015

100 East Davie Street
Raleigh NC 27601 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park NC 27709 USA

Linux is a registered trademark of Linus Torvalds. Red Hat, Red Hat Enterprise Linux and the Red Hat "Shadowman" logo are registered trademarks of Red Hat, Inc. in the United States and other countries.

Ceph is a registered trademark of Red Hat Inc.

WordPress is a registered trademark of WordPress Foundation.

UNIX is a registered trademark of The Open Group.

Intel, the Intel logo and Xeon are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

© 2015 by Red Hat, Inc. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V1.0 or later (the latest version is presently available at http://www.opencontent.org/openpub/).

The information contained herein is subject to change without notice. Red Hat, Inc. shall not be liable for technical or editorial errors or omissions contained herein.

Distribution of modified versions of this document is prohibited without the explicit permission of Red Hat Inc.

Distribution of this work or derivative of this work in any standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from Red Hat Inc.

The GPG fingerprint of the security@redhat.com key is:
CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

Send feedback to refarch-feedback@redhat.com

Comments and Feedback

In the spirit of open source, we invite anyone to provide feedback and comments on any reference architectures. Although we review our papers internally, sometimes issues or typographical errors are encountered. Feedback allows us to not only improve the quality of the papers we produce, but allows the reader to present thoughts on potential improvements and topic expansion to the papers.

Feedback on the papers can be provided by emailing refarch-feedback@redhat.com. Please refer to the title within the email.

Staying In Touch

Join us on some of the popular social media sites where we keep our audience informed on new reference architectures as well as offer related information on things we find interesting.

Like us on Facebook:

https://www.facebook.com/rhrefarch

Follow us on Twitter:

https://twitter.com/RedHatRefArch

Plus us on Google+:

https://plus.google.com/u/0/b/114152126783830728030/

Table of Contents

1 Executive Summary

The OpenStack environment enables enterprises an on-demand computing resource, by provisioning and maintaining vast amounts of self managed virtual machines. Compute resources are accessible via APIs for developers building cloud applications and via web interfaces for administrators and users. The architecture is designed to scale horizontally on standard hardware while keeping cloud computing requirements at a low cost.

The purpose of this reference architecture is to demonstrate how a Red Hat Enterprise Linux OpenStack Platform is deployed on servers running the Red Hat Enterprise Linux 7 operating system. The Red Hat Enterprise Linux OpenStack Platform, provides an Infrastructure-as-a-Service (IaaS) foundation for a cloud computing environment on top of Red Hat Enterprise Linux. It meets enterprise requirements with the ability to extensively scale and provide a fault tolerant and highly available environment.

The deployment is orchestrated by using the Red Hat Enterprise Linux OpenStack Platform Installer. The Installer is a wizard-based tool that makes enterprise-grade installations much easier. The Installer is built upon the Foreman deployment tool's capabilities to discover and provision the hosts, orchestrate the deployment, and configure the OpenStack services and components.

This reference architecture introduces Red Hat Enterprise Linux OpenStack Platform version 6 running on Red Hat Enterprise Linux version 7.0, through a detailed use case:

- Installing Red Hat's OpenStack technology using the Staypuft Installer
- Deploying a Ceph Storage environment and integrating with OpenStack
- Network configuration using Neutron networking - Virtual Extensible Local Area Network (VXLAN) network type and Modular Layer 2 (ML2) plugin.
- Highly available (HA) environment using Pacemaker and HAProxy
- Create a tenant user and network
- Deploy demonstration web application

Every step of this use case has been tested in Red Hat's engineering lab with production code on bare-metal hardware for the OpenStack deployed nodes.

Additional Information:

- OpenStack "Juno" Release Notes https://wiki.openstack.org/wiki/ReleaseNotes/Juno
- Red Hat Enterprise Linux OpenStack Platform 6 Release Notes https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/6/html/Release_Notes/ch02s03.html
- Red Hat Enterprise Linux OpenStack Platform details https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/

2 Components Overview

2.1 Red Hat Enterprise Linux OpenStack Platform

2.1.1 Overview

Red Hat Enterprise Linux OpenStack Platform (RHEL OSP), provides Infrastructure-as-a-Service (IaaS) foundation for public, private or hybrid cloud computing environment on top of Red Hat Enterprise Linux (RHEL). RHEL OSP meets enterprise requirements with ability to extensively scale and provide a fault tolerant and highly available environment.

OpenStack is made up of many different moving parts. Because of its open nature, anyone can add additional components to OpenStack to meet their requirements. 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.

2.1.2 OpenStack Services

OpenStack has a modular architecture with various services as its components.

2.1.2.1 Compute Service (Nova)

Nova is the primary computing engine behind OpenStack and provides the base for Openstack IaaS functionality. It can scale out horizontally on standard hardware in a distributed and asynchronous fashion, providing a fault tolerant and cost effective computing environment for virtual machines. The access to compute resources can be controlled by virtual hardware profiles and tenants. They are used for deploying and managing large numbers of virtual machines and other instances that handle computing tasks.

2.1.2.2 Identity Service (Keystone)

Keystone provides identity services for OpenStack. Identity Service provides a central directory of user mappings to the OpenStack services they can access. It acts as a common authentication system across the cloud environment and can integrate with existing backend directory services. The Identity Service is comprised of the Keystone service, which responds to service requests, places messages in the messaging queue, grants access tokens, and updates the state database.

The Keystone Service has two primary functions:

- User Management
- Service Catalog

2.1.2.3 Block Storage (Cinder)

OpenStack Cinder service provides compute instances with persistent block storage. Block storage is appropriate for performance sensitive scenarios such as databases, expandable file systems, or providing a server with access to raw block level storage. Persistent block storage can survive instance termination and can also be moved across instances like any external storage device. Cinder has volume snapshots capabilities for backing up the volumes.

The RHEL OSP Installer provides four driver backends for Cinder:

- NFS
- LVM
- Ceph
- EqualLogic

2.1.2.4 Network Service (Neutron)

OpenStack Networking is a scalable API-driven service for managing networks and IP addresses. OpenStack Networking gives users self-service control over their network configurations. Users can define, separate, and join networks on demand. Neutron API includes support for Layer 2 (L2) networking as well as an extension for layer 3 (L3) router construction that enables routing between L2 networks and gateways to external networks. This allows for flexible network models to fit the requirements of different applications. OpenStack Networking has a pluggable architecture that supports numerous virtual networking technologies as well as native Linux networking mechanisms including Open vSwitch and Linux Bridge.

2.1.2.5 Dashboard (Horizon)

Horizon is the dashboard behind OpenStack that provides administrators and users a graphical interface to access, provision and automate cloud- based resources. . Developers can access the components of OpenStack individually through an application programming interface (API) The dashboard provides system administrators a view of what is going on in the cloud, and to manage it as necessary. The dashboard runs via the HTTP service.

2.1.2.6 Image Service (Glance)

OpenStack Image Service (Glance) provides discovery, registration, and delivery services for disk and server images. It can also be used to store and catalog multiple backups. Glance allows these images to be used as templates when deploying new virtual machine instances. The Image Service can store disk and server images in a variety of back-ends, including OpenStack object storage. The Image Service API provides a standard REST interface for querying information about disk images and lets clients stream the images to new servers.

RHEL OSP Installer provides three Driver Backends for glance:

- Local File
- NFS
- Ceph storage

NOTE: Local File is applicable only for non HA deployment with a single controller. This reference architecture utilizes Ceph Storage for Glance and Cinder.

2.1.2.7 Telemetry Service (Ceilometer)

Ceilometer provides telemetry services for billing services to individual users of the cloud. It keeps a verifiable count of each user's usage of various components in the OpenStack cloud. The delivery of counters is traceable and auditable.

2.1.2.8 Orchestration Service (Heat)

OpenStack Heat is an orchestration service that manages the life-cycle of applications within an OpenStack environment using templates. Heat is capable of deploying multi- instance applications called stacks and managing application lifecycle.

Heat Templates

Heat templates are written in a declarative format. A template defines what resources to deploy rather than how to deploy those resources. This is similar to the approach used by popular configuration tools such as Puppet, Ansible, and Chef. Configuration tools focus on the configuration of a system, whereas Heat focuses on resource provision and relies on scripts provided by the cloud-init package to handle system configuration. A template may create and configure a large list of resources thus supporting complex application stacks.

2.1.2.9 High Availability Services

OpenStack Environment consists of stateless, shared-nothing services that serve their APIs and underlying infrastructure components OpenStack services use for inter-service communication and to save persistent data.

Building a scale-out controller requires setting the services and infrastructure components (database and message broker) in Active-Active configuration. This provides the ability to add more nodes to the cluster as the load grows while load balancing the API requests among the available nodes. While most of the services are Active-Active, there are some services still in Active-Passive mode.

Red Hat Enterprise Linux OpenStack Platform is with the Red Hat Enterprise Linux High Availability Add-On and supports highly available environments for customer deployments. This means that a cloud infrastructure can now be set up so that if one of its controller nodes or services fail, the node or service can be brought back up with minimal impact on production.

- **Memcached** is a fast in-memory key-value cache software that is used by OpenStack components for caching data and increasing performance. Memcached runs on all controller nodes, ensuring that should one go down, another instance of Memcached is available.
- **Galera** Cluster is a synchronous multi-master cluster for MariaDB with the following features:
    - Synchronous replication
    - Active-Active multi-master topology
    - Read and write to any cluster node
    - Automatic membership control, failed nodes drop from the cluster

- Automatic node joining
- True parallel replication, on row level

- **HAProxy** is a software layer-7 load balancer used to proxy all clustered OpenStack API components and perform SSL terminations. HAProxy can be added as a resource to the Pacemaker software that runs on the Controller nodes where HAProxy is situated.
- **Pacemaker** is the clustering software used to ensure the availability of services and systems running on the controller nodes. It also handles high availability by leveraging 'corosync' service. Pacemaker manages the Galera nodes and HAProxy.
- **Fencing** is an operation that completely isolates a failed node to preserve cluster functionality.. Pacemaker has a built in integration with fencing.
- **HA Modes**

  The following modes are supported:

  1. **Active-Active :** In an Active-Active configuration, all system components are kept online; if a component fails, its load is passed to the next active component. The majority of OpenStack services are configured to run in active/active configuration via the Pacemaker resource manager.
  2. **Active-Passive :** In this configuration, only one instance of the service runs in the cluster at a time and get started if pacemaker detects the service is offline. A small number of OpenStack services use an active/passive configuration for high availability.

     Please refer to 4.6.4 PCS resources availability for availability details by resource.

Figure 2.1.2 1: High Availability By Pacemaker describes the services that are Active-Active or Active-Passive in the current RHELOSP 6 version deployed through current version of Installer.

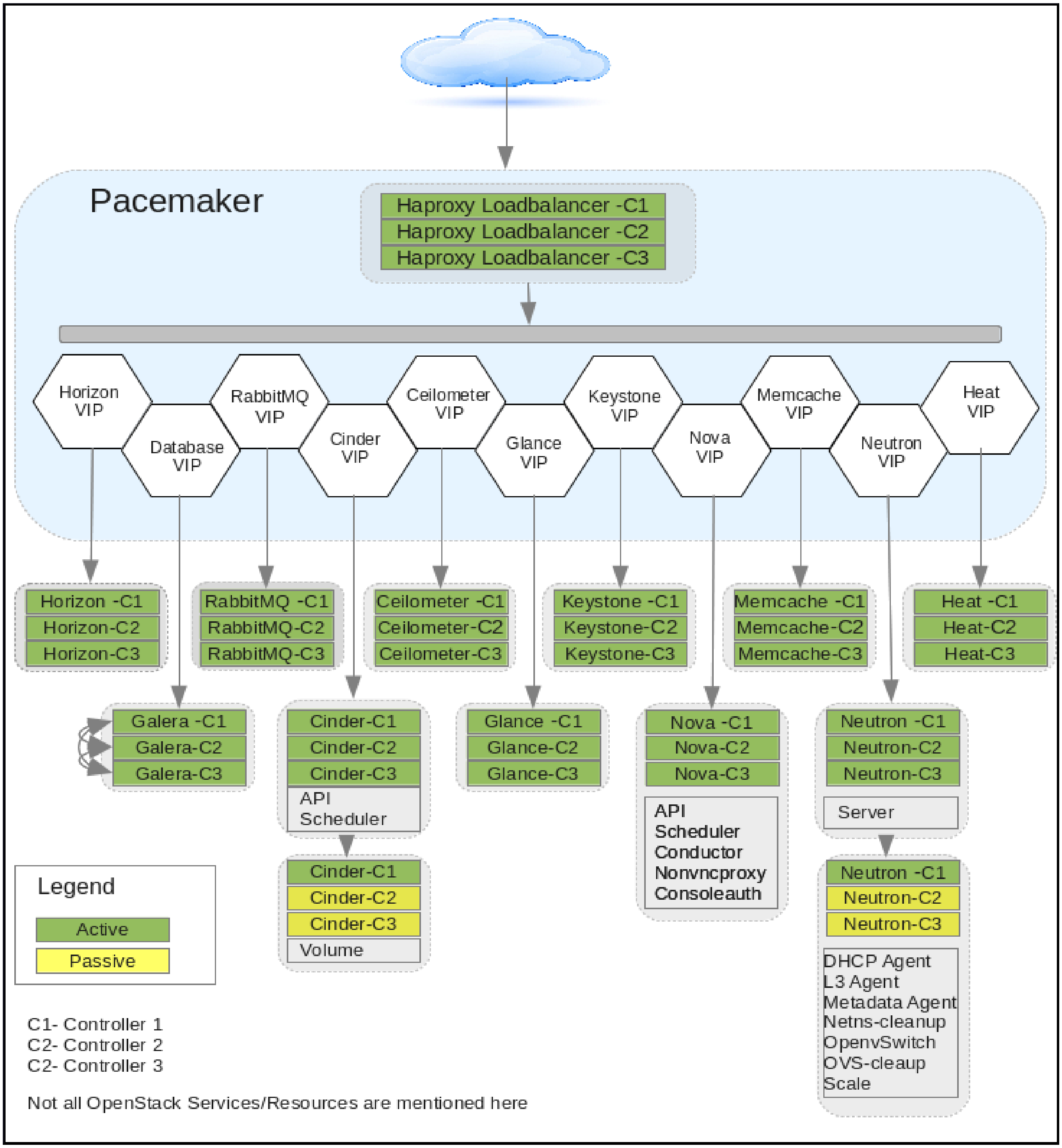


Figure 2.1.2 1: High Availability By Pacemaker

NOTE: Table 4.6.4.1: Resource Availability displays a list of services/resources by availability and location.

2.1.2.10 Other services

- **MariaDB** is the database shipped with Red Hat Enterprise Linux 7 and used by RHEL OSP 6 as the default.
- **RabbitMQ** is the default (and recommended) message broker on RHEL OSP 6.

2.2 Ceph

Ceph is an open-source, scalable, software-defined storage system which uniquely delivers object, block and file system storage in one unified system.

2.2.1 Ceph Storage Cluster

The Ceph Storage Cluster is the foundation for all Ceph deployments. Based upon RADOS (Reliable Autonomic Distributed Object Store), Ceph Storage Cluster consist of two types of daemons: a Ceph Object Storage daemon (OSD) that stores data as objects on a storage node, and a Ceph Monitor that maintains a master copy of the cluster map.

Ceph stores data objects within two logical groups: pools and placement groups (PGs).

- **Pools:** Pools are logical groups for storing objects. Pools manage the number of placement groups, the number of object replicas, and the CRUSH ruleset for the pool. Ceph can snapshot pools. Each pool has a number of placement groups. CRUSH (Controlled Replication Under Scalable Hashing) maps PGs to OSDs dynamically. When a Ceph Client stores objects, CRUSH maps each object to a placement group.
- **Placement Groups:** A Placement Group (PG) aggregates a series of objects into a group, and maps the group to a series of OSDs. Tracking object placement and object metadata on a per-object basis is computationally expensive–i.e., a system with millions of objects cannot realistically track placement on a per-object basis. Placement groups address this barrier to performance and scalability. Additionally, placement groups reduce the number of processes and the amount of per-object metadata Ceph must track when storing and retrieving data.

2.2.2 Red Hat Ceph Storage (RHCS)

Red Hat Ceph Storage combines open-source Ceph designed to present object, from a single distributed computer cluster to connected clients. Ceph's main goals are to be completely distributed without a single point of failure, scalable to the exabyte level, and freely-available.

RHCS has gone through Red Hat's extensive QA and testing for increased reliability and interoperability. By adding Ceph Storage into its bandwagon of products, Red Hat offers a single source of support to deploy and manage a complex OpenStack environment integrated with Ceph storage.

For details on the RHCS, please refer to the URL https://access.redhat.com/documentation/en-US/Red_Hat_Ceph_Storage/

2.2.3 Calamari

Calamari is a management and monitoring system for Ceph storage cluster which provides a dashboard user interface for monitoring and managing the Ceph cluster.

2.3 Red Hat Enterprise Linux OpenStack Installer (RHEL OSP Installer)

Here are descriptions of the services running on the Red Hat OpenStack Installer:

- **Web-based Interface** - From the Red Hat OpenStack Installer web-based interface (HTTP port 80), one can create a deployment configuration that provisions all the Controller and Compute nodes in the OpenStack setup. Once the set of nodes are deployed, the interface can monitor them as well as provide hardware/system information.
- **PXE Servic**e - The PXE boot service (via tftpd) allows the Installer to discover the managed nodes and deploy them.
- **DHCP Service** - The DHCP server on the Installer allows the Controller, Compute and Ceph nodes on the PXE network to automatically lease IP addresses from the Installer.
- **Gateway Service** - The PXE/Management network on the installer service as the default gateway during host provisioning.
- **DNS Service** – The Installer runs a BIND DNS server (*named* service) that dynamically adds the names and IP addresses of all hosts in the OpenStack configuration. Node names are assigned by the MAC address of the provisioning network NIC of the managed hosts.

For details on the installer, please refer to the URL https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/6/html/Installer_and_Foreman_Guide/index.html

2.4 Network Information

Some of the networking information worth mentioning have been highlighted below.

2.4.1 Modular Layer 2 (ML2) core plug-in

The Modular Layer 2 (ml2) plug-in is a framework which supports a variety of networking technologies via plugins. The networking technologies include Open vSwitch, Linux Bridge and so on.

By default, Red Hat Enterprise OpenStack Platform 6 Installer uses the open source ML2 core plug-in with the Open vSwitch (OVS) mechanism driver. This reference architecture focuses on using ML2 with OVS and VXLAN tunneling for virtual machine traffic isolation.

ML2 network types:

- **Flat** - All instances reside on the same network, which can also be shared with the hosts. No VLAN tagging or other network segregation takes place.
- **Local** - Instances reside on the local compute host and are effectively isolated from

any external networks.

- **VLAN** - Networking allows users to create multiple provider or tenant networks using VLAN IDs (802.1Q tagged) that correspond to VLANs present in the physical network. This allows instances to communicate with each other across the environment. They can also communicate with dedicated servers, firewalls, load balancers and other networking infrastructure on the same layer 2 VLAN.
- **VXLAN** - Virtual E**x**tensible Local Area Network helps create a logical network for virtual machines across different networks. This allows the creation of a layer 2 network on top of layer 3 through encapsulation. The basic use case for VXLAN is to connect two or more layer three (L3) networks and make them look like they share the same layer two (L2) domain. This allows for virtual machines to live in two disparate networks yet still operate as if they were attached to the same L2 network thus enhancing scalability. The VXLAN networks are broken down as segments and the same IP address can exist across different segments. A combination of Machine Address Control (MAC) and VXLAN Network Identifier (VNI) makes each VM connection unique.
- **GRE**- Generic Routing Encapsulation (GRE) segmentation. A network layout in which tunnels are used to segregate and carry network traffic over individual tenant networks.

  NOTE: Based on the physical network fabric and other network requirements, third-party Networking plug-ins can be deployed instead of the default ML2/Open vSwitch driver due to the pluggable architecture of OpenStack Neutron Networking.

  VXLAN and GRE are overlay technologies and could result in a small overhead on each packet.

2.4.2 Network Traffic Types

RHEL OSP Installer uses the following network traffic types for deployment. All of the below except “external” and “tenant” get assigned to the “default” traffic type. At a minimum, besides default, two additional subnets are ideal for “external” and “tenant” networks. If there is capacity to add additional network interfaces, the traffic types can be further isolated by creating additional subnets.

The following subnets and assignment details are described in 4.3 OpenStack Deployment.

- **Provisioning/PXE** - Used for PXE/provisioning of hosts. This one is set to the default subnet in Foreman which is mapped to the boot/PXE interface on the hosts and cannot be changed. Foreman manages DHCP for this network.
- **External** - Used for external connectivity/bridges for instances. The IP addresses on this network should be reachable outside the intranet.
- **Tenant -** For tenant network traffic internally among virtual Instances.
- **Management** - Private API for services. Provides internal communication between OpenStack components. IP addresses on this network should be reachable only within the data center.

- **Public API** - Sets up access to the RESTful API, and the Horizon GUI. It exposes all OpenStack APIs, including the Networking API, to tenants. IP addresses on this network should be reachable by anyone on the intranet.
- **Admin API** – For admin access for various services.
- **Cluster Management** – Used by Pacemaker and Galera for cluster communications.
- **Storage clustering** – For Ceph cluster communications among OSD nodes.
- **Storage** – For connectivity between Controller, Compute, and Ceph storage nodes for storage traffic.

3 Reference Architecture Configuration Details

This section of the paper describes the hardware, software, and procedures used to configure this reference architecture in the lab. Good practices learned in the lab are shared throughout this document.

3.1 Environment

The reference architecture environment consists of the components required to build a highly available Red Hat Enterprise Linux OpenStack Platform cloud infrastructure.

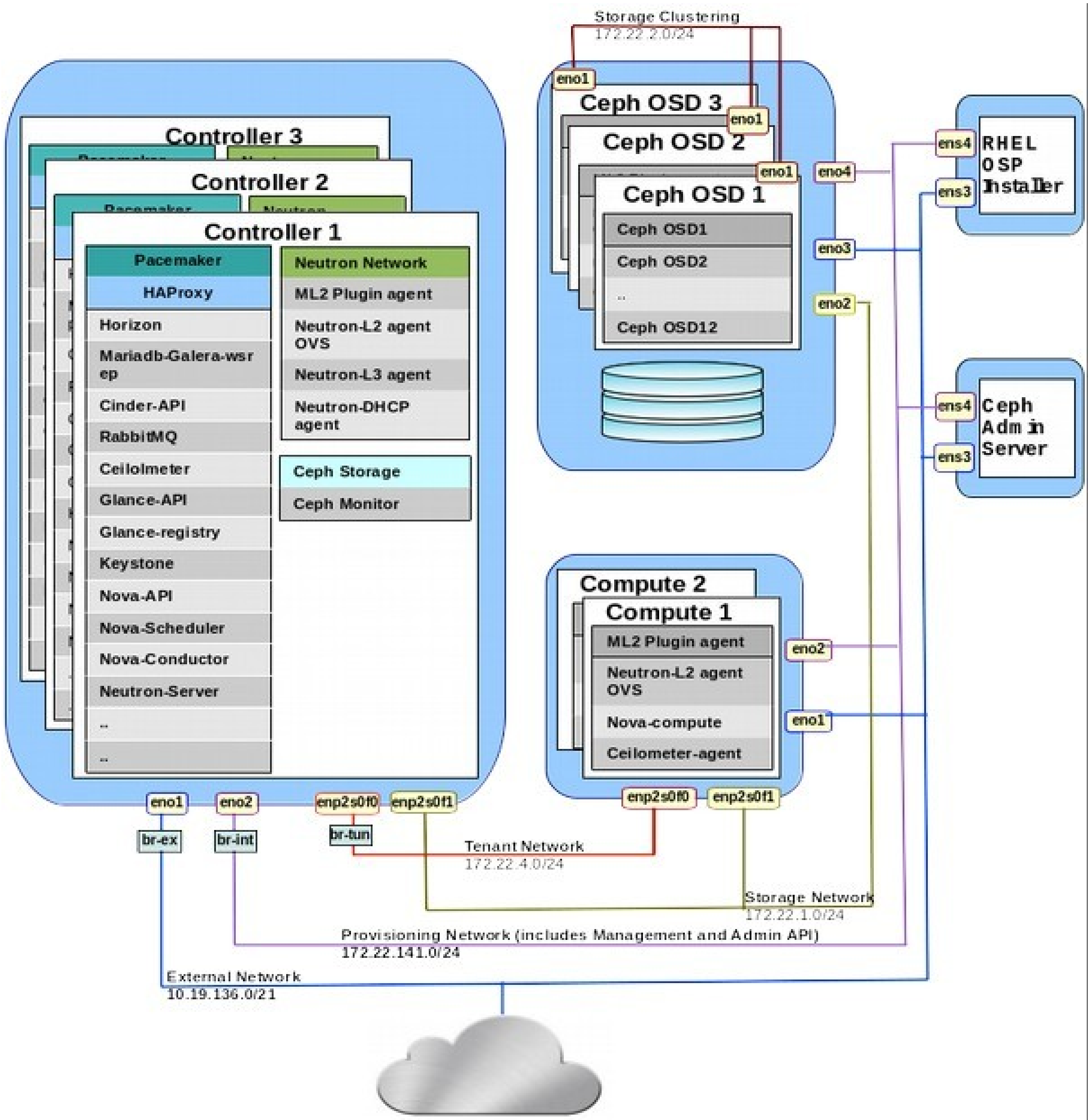


Figure 3.1 1: OpenStack-Ceph Environment

- Figure Figure 3.1 1: OpenStack-Ceph Environment describes the deployed OpenStack environment that consists of:
- Three node pacemaker cluster of controller hosts that also act as neutron network node and Ceph monitor nodes.
- Two compute hosts.
- Three node Ceph OSD cluster.

There are isolated cluster interconnects for the pacemaker and Ceph OSD clusters. The provisioning network (shared with public api, management and Admin API), tenant network and storage network are dedicated networks. External has an assigned gateway for external connectivity. RHEL OSP Installer and Ceph Admin Server use provisioning network for server provisioning and Ceph deploy commands respectively.

3.1.1 Required Channels

3.1.1.1 Channels for RHEL OSP Installer server

Channel	Repository Name
Red Hat Enterprise Linux 7 Server (RPMS)	rhel-7-server-rpms
Red Hat Enterprise Linux 7 Server - RH Common (RPMs)	rhel-7-server-rh-common-rpms
Red Hat Enterprise Linux OpenStack Platform Installer 6.0 (RPMS)	rhel-7-server-openstack-6.0-installer-rpms
Red Hat Enterprise Linux RHSCL	rhel-server-rhscl-7-rpms

Table 3.1.1.1: Required Channels -Installer

3.1.1.2 Channels for deployed servers

Channel	Repository Name
Red Hat Enterprise Linux OpenStack Platform 6.0 (RPMS)	rhel-7-server-openstack-6.0-rpms
Red Hat Enterprise Linux OpenStack Platform Installer 6.0 (RPMS)	rhel-7-server-openstack-6.0-installer-rpms
Red Hat Enterprise Linux 7 Server - RH Common (RPMs)	rhel-7-server-rh-common-rpms

Table 3.1.1.2: Required Channels -Deployed Servers

3.1.2 Software Versions

3.1.2.1 RHEL OSP Installer

Software	Version
rhel-osp-installer	0.5.7-1.el7ost
rhel-osp-installer-client	0.5.7-1.el7ost
puppet	3.6.2-2.el7
foreman	1.6.0.49-6.el7ost
dnsmasq	2.66-13.el7_1
dhcp	4.2.5-36.el7
foreman-discovery-image	7.0-20150227.0.el7ost
foreman-installer	1.6.0-0.3.RC1.el7ost
foreman-postgresql	1.6.0.49-6.el7ost
foreman-proxy	1.6.0.30-6.el7ost
foreman-selinux	1.6.0.14-1.el7sat

Table 3.1.2.1: RHEL OSP Installer Software Versions

3.1.2.2 Ceph Admin Server

Software	Version
calamari-server	1.2.3-11.el7cp
calamari-clients	1.2.3-3.el7cp
ceph-deploy	1.5.22-0.2.rc1.el7cp
httpd	2.4.6-31.el7

Table 3.1.2.2: Ceph Admin Server Software Versions

3.1.2.3 Ceph OSD Servers

Software	Version
foreman-installer	1:1.6.0-0.2.RC1.el7ost
ceph	0.80.6-0.el7
ceph-common	0.80.6-0.el7
openstack-selinux	0.6.17-1.el7ost
dnsmasq	2.66-12.el7
iproute	3.10.0-13.el7
puppet	3.6.2-2.el7

Table 3.1.2.3: Ceph Software Versions

3.1.2.4 Compute Servers

Software	Version
openstack-nova-compute	2014.2.3-9.el7ost
openstack-utils	2014.2-1.el7ost
python-cinderclient	1.1.1-1.el7ost
openstack-neutron-openvswitch	2014.2.3-2.el7ost
ceph-common	0.80.8-7.el7cp
dnsmasq.x86_64	2.66-13.el7_1
iproute	3.10.0-21.el7
openstack-neutron	2014.2.3-2.el7ost
openstack-neutron-openvswitch	2014.2.3-2.el7ost
openstack-selinux	0.6.29-1.el7ost
foreman-selinux	1.6.0.14-1.el7sat
foreman-installer	1.6.0-0.4.RC1.el7ost
puppet	3.6.2-2.el7

Table 3.1.2.4: Compute Software Versions

3.1.2.5 Controller Servers

Software	Version
fence-agents	4.0.11-11.el7_1
iproute	3.10.0-21.el7
memcached	1.4.15-9.el7
mariadb	5.5.41-2.el7_0
mariadb-galera-common	5.5.41-2.el7ost
mariadb-galera-server	5.5.41-2.el7ost
openstack-cinder	2014.2.3-3.el7ost
openstack-dashboard	2.3-2.el7ost
openstack-glance	2014.2.3-1.el7ost
openstack-keystone	2014.2.3-1.el7ost
openstack-neutron	2014.2.3-2.el7ost
openstack-neutron-openvswitch	2014.2.3-2.el7ost
openstack-nova-api	2014.2.3-9.el7ost
openstack-nova-conductor	2014.2.3-9.el7ost
openstack-nova-console	2014.2.3-9.el7ost
openstack-nova-novncproxy	2014.2.3-9.el7ost
openstack-nova-scheduler	2014.2.3-9.el7ost
openstack-selinux	0.6.29-1.el7ost
openstack-utils	2014.2-1.el7ost
pacemaker	1.1.12-22.el7_1.1
pcs	0.9.137-13
rabbitmq-server	3.3.5-3.el7ost
resource-agents	3.9.5-40.el7_1.3
ceph	0.80.8-5.el7cp
ceph-common	0.80.8-7.el7cp
foreman-installer	1.6.0-0.4.RC1.el7ost
foreman-selinux	1.6.0.14-1.el7sat
puppet	3.6.2-2

3.1.3 Security Reference

All the managed hosts being deployed and the installers have SELinux set to enforcing by default. Some of the important ports handled in each host are mentioned below:

3.1.3.1 RHEL OSP Installer

Service	Port
rpc	111
ssh	22
filenet rpc	32769
ssl	443
dns	53
pftp	662
tftp	69
http/https	80/443
puppet	8140
rquota	875

3.1.3.2 Ceph Admin Server

Service	Port
ssh	22
http/https	80/443
Calamari	2003
Ceph Salt master	4505,4506

3.1.3.3 Compute nodes

Service	Port
ssh	22
dns	53
bootp server/client	67/68
VXLAN	4789
Remote frame buffer	5900:5999

Table 3.1.3.1: Compute node ports

3.1.3.4 Controller nodes

Service	Port
rabbitmq (amqp)	5672
ceilometer-api	8777
cinder-api	8776
galera	3306
glance-api	9292
glance-registry	9191
heat-api	8004
heat-cfn	8000
heat-cloudwatch	8003
horizon	80
keystone-admin	35357
keystone-public	5000
neutron-api	9696
nova-api	8774
nova-metadata	8775
nova-novncproxy	6080
nova-xvpvncproxy	6081
haproxy stats	81
vxlan	4789

Table 3.1.3.2: HAProxy port configuration

3.1.3.5 Ceph OSD node

Three ports are used per OSD service running on each host. This range will need to be adjusted depending on the number of OSDs per host.

Service	Port
Ceph osd	6800:6850

Table 3.1.3.3: Ceph OSD node ports

3.1.4 Server Hardware Configuration

3.1.4.1 Selecting the hardware

- The deployed RHEL OSP servers do not require identical hardware. The hardware used in this reference architecture meets the minimum requirements outlined in the OpenStack documentation.
- The Compute nodes offer computing resource to the virtual instances and therefore must have adequate memory and processing power. However, a shortage of resources can be met by horizontally scaling the environment with the addition of more compute nodes.
- Ceph OSDs calculate data placement, replicate data, and maintain their own copy of the cluster map. Hence Ceph OSD nodes are CPU intensive and require significant processing power.
- Monitors simply maintain a master copy of the cluster map and may not CPU intensive and can coexist on a controller node.

3.1.4.2 Server Hardware specifications

The table lists the hardware specifications for the servers used in this reference architecture.

Hardware	Specifications
Controller nodes Dell PowerEdge M520 blade server Count – 3 nodes	Processor 2 x Intel(R) Xeon(R) CPU E5-2450 @ 2.10GHz (8 core)
	Network 4 x Broadcom Gigabit Ethernet BCM5720 2 x Broadcom NetXtreme II 10 Gb Ethernet BCM57810
	Memory 6 x DDR3 8GB @1600 MHZ DIMMs (48GB)
	Disk 2 x 136GB SAS internal disk drives
Compute nodes Dell PowerEdge M520 blade server Count – 2 nodes	Processor 2 x Intel(R) Xeon(R) CPU E5-2450 @ 2.10GHz (8 core)
	Network 4 x Broadcom Gigabit Ethernet BCM5720 2 x Broadcom NetXtreme II 10 Gb Ethernet BCM57810
	Memory 6 x DDR3 16GB @1600 MHZ DIMMs (96GB)
	Disk 2 x 136GB SAS internal disk drives
Ceph Storage nodes (OSD) Dell PowerEdge R720xd rack mount server Count – 3 nodes	Processor 2 x Intel(R) Xeon(R) CPU E5-2670 @ 2.50GHz (10 core)
	Network 4 x Broadcom Gigabit Ethernet BCM5720 2 x Broadcom NetXtreme II 10 Gb Ethernet BCM57810
	Memory 8 x DDR3 16GB @ 1866 MHZ DIMM (128GB)
	Disk 2 x 136GB SAS internal disk drives 12x 1TB SAS Drives 3 x 400GB SSD Drives
KVM server for VMs • **RHEL-OSP-Installer** • **Ceph admin server** Dell PowerEdge M520 blade server Count – 1 node	Processor 2 x Intel(R) Xeon(R) CPU E5-2450 @ 2.10GHz (8 core)
	Network 4 x Broadcom Gigabit Ethernet BCM5720 2 x Broadcast NetXtreme II 10 Gb Ethernet BCM57810
	Memory 6 x DDR3 16GB @1600 MHZ DIMMs - 96GB
	Disk 2 x 136GB SAS internal disk drives

Table 3.1.4.1: Server Hardware Specifications

3.1.5 Storage Configuration for Ceph Nodes

In this reference architecture, the shared storage requirement for the highly available Openstack environment has been met by using Ceph storage across a private 10GB network that runs between Controller, Compute, and Ceph nodes.

Each OSD server has 12 x 1TB SAS drives for data and 3 x 400GB SSD drives for journaling.

3.1.6 Network Interfaces

Table 3.1.6.1: Network Traffic Type describes the network traffic types used in each host group. Ideally network separation is a good practice where possible. This helps in handling performance, scalabilty and isolation requirements. At a minimum, three dedicated networks are required for “external”, “tenant” and the remaining networks.

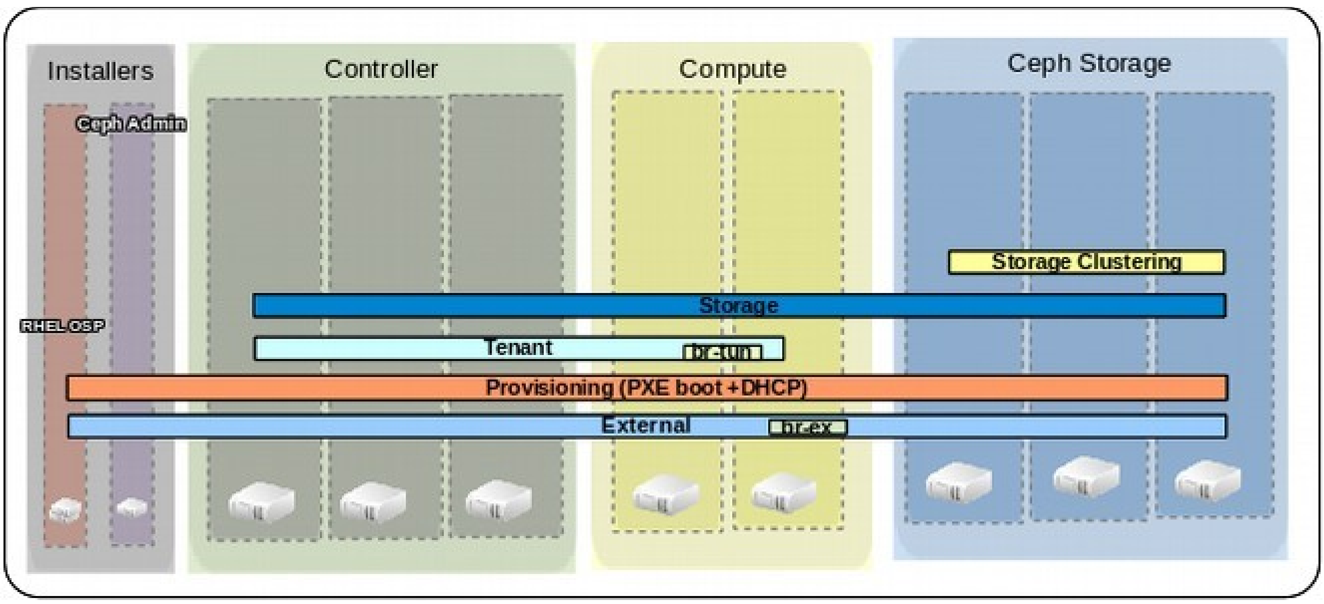


Figure 3.1.6 1: Network Traffic Type

Network Traffic Type	Controller	Compute	Ceph Storage Nodes	RHEL OSP Installer	Ceph Admin Node
Provisioning	√	√	√	√	√
External	√	√	√	√	√
Tenant	√	√			
Storage	√	√	√		
Storage Clustering			√		

Table 3.1.6.1: Network Traffic Type

This reference architecture combines a common network for Provisioning, Admin API and Management and provides dedicated network for the remaining networks.

- Controller nodes use four dedicated networks as described in Table 3.1.6.1: Network Traffic Type
- Compute nodes use four networks
- Ceph OSD nodes use four networks
- The installers use two each

NOTE: The RHEL OSP Installer can configure network bonding for the interfaces for performance and high availability. Network bonding is not in scope for this reference architecture. Please refer to the RHEL OSP Installer documentation for details at: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/6/html-single/Installer_and_Foreman_Guide/index.html#

There are limitations with converging Provisioning, Management, and Admin API networks on a single interface. Please refer to Bugzilla https://bugzilla.redhat.com/show_bug.cgi?id=1176423 for details.

The External and Public API networks cannot be on the same subnet.

3.1.7 Network bridges

The OpenStack virtual instances connect to the physical network via OVS Bridges. The following describes the bridges that get created during RHEL OSP Installer deployment.

- Integration bridge br-int - The br-int Open vSwitch bridge is the integration bridge. All guests running on the compute host connect to this bridge. Networking implements isolation across these guests by configuring the br-int ports.
- Physical connectivity bridge br-ex- The br-ex bridge provides dedicated external connectivity for VM traffic. It connects to the integration bridge by a virtual interface pair: (int-br-ex, phy-br-ex).
- Tunnel bridge br-tun – Handles traffic management.

3.1.8 Network Addresses

The following is the list of network addresses used in this reference architecture:

- Installer servers

Host	Role	Network	Interface	Network Address
ospceph-rinst2 (VM)	RHEL OSP Installer	Public	ens3	10.19.141.57
		Provisioning	ens4	172.22.141.57
osp-ice (VM)	Ceph Admin Server	External	ens3	10.19.141.150
		Provisioning	ens4	172.22.141.150

Table 3.1.8.1: Network Information- Installers

- Ceph OSD nodes

Host	Role	Network	Interface	Network Address
osp-osd-1	Ceph OSD	External	eno3	10.19.141.206
		Provisioning	eno4	172.22.141.206
		Storage Clustering	eno1	172.22.2.206
		Storage	eno2	172.22.1.206
osp-osd-2	Ceph OSD	External	eno3	10.19.141.207
		Provisioning	eno4	172.22.141.207
		Storage Clustering	eno1	172.22.2.207
		Storage	eno2	172.22.1.207
osp-osd-3	Ceph OSD	External	eno3	10.19.141.208
		Provisioning	eno4	172.22.141.208
		Storage Clustering	eno1	172.22.2.208
		Storage	eno2	172.22.1.208

Table 3.1.8.2: Network Information - Ceph OSD nodes

- Controller and Ceph Monitor nodes

Host	Role	Network	Interface	Network Address
osp-cont-1	Controller/ Ceph Monitor	External	eno1	10.19.141.201
		Provisioning	eno2	172.22.141.201
		Tenant	enp2s0f0	172.22.4.201
		Storage	enp2s0f1	172.22.1.201
osp-cont-2	Controller/ Ceph Monitor	External	eno1	10.19.141.202
		Provisioning	eno2	172.22.141.202
		Tenant	enp2s0f0	172.22.4.202
		Storage	enp2s0f1	172.22.1.202
osp-cont-3	Controller/ Ceph Monitor	External	eno1	10.19.141.203
		Provisioning	eno2	172.22.141.203
		Tenant	enp2s0f0	172.22.4.203
		Storage	enp2s0f1	172.22.1.203

Table 3.1.8.3: Network Information- Controller Nodes

- Compute nodes

Host	Role	Network	Interface	Network Address
osp-comp-1	Compute	External	eno1	10.19.141.204
		Provisioning	eno2	172.22.141.204
		Tenant	enp2s0f0	172.22.4.204
		Storage	enp2s0f1	172.22.1.204
osp-comp-2	Compute	External	eno1	10.19.141.205
		Provisioning	eno2	172.22.141.205
		Tenant	enp2s0f0	172.22.4.206
		Storage	enp2s0f1	172.22.1.206

Table 3.1.8.4: Network Information - Compute nodes

4 Installation

4.1 Installation process

The following lists the steps and their sequence followed in this reference architecture to render a functional OpenStack environment:

1. Install and configure RHEL OSP Installer system
2. Boot managed hosts into discovery mode
3. Deploy managed hosts with required OpenStack parameters
4. Install and configure Ceph Admin installer
5. Configure Ceph environment and integrate with deployed OpenStack environment
6. Create tenant projects, users, network capabilities and virtual instances
7. Install a demo application to verify the functioning of the virtual instance
8. Test high availability despite node failure

4.2 Installing and configuring RHEL OSP Installer system

The following section describes setting up a RHEL OSP Installer system.

4.2.1 Setting up the RHEL OSP Installer server

Provision either a virtual machine or a physical machine with the following features:

- Red Hat Enterprise Linux 7. For details please refer to the following

  https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/index.html
- At least 1 vCPU and at least 3GB memory in case of a virtual machine.
- An interface with public access for External connectivity.

```
[root@ospceph-rinst2 ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens3
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
NAME=ens3
ONBOOT=yes
IPADDR=10.19.141.57
PREFIX=21
GATEWAY=10.19.143.254
```

- An interface connected to a private network. This private network must be connected to all the Openstack and Ceph nodes as well. This network is used for PXE booting and provisioning. Since this network broadcasts DHCP/PXE boots, care must be taken to ensure it is isolated from other networks to avoid interference with other networks in the datacenter.

```
[root@ospceph-rinst2 ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens4
DEVICE=ens4
BOOTPROTO=none
ONBOOT=yes
TYPE=Ethernet
IPADDR=172.22.141.57
PREFIX=24
PEERDNS=yes
```

- Configure NTP
- A gateway is required for external access for the OpenStack managed nodes. The Installer private network interface IP can act as a gateway address. It is also possible to point to an external gateway. If the installer is configured as the gateway, IP forwarding must be enabled on the system.

```
[root@ospceph-rinst2 ~]# echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf
&& sysctl -p
```

- Enable firewall and configure iptables. The installer server must broadcast DHCP and allow PXE boot to the clients. Hence the relevant ports must be open. In this reference architecture, firewalld has been disabled and replaced by iptables service and configured as follows.

```
[root@ospceph-rinst2 ~]# cat /etc/sysconfig/iptables
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [17:1776]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 53 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 53 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 111 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 111 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 32803 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 32769 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2020 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 2020 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 662 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 662 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 892 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 892 -j ACCEPT
```

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 875 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 875 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2049 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 2049 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 69 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 8140 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 8140 -j ACCEPT
-A FORWARD -o ens+ -j ACCEPT
-A FORWARD -s 172.22.141.0/24 -j ACCEPT
-A FORWARD -d 172.22.141.0/24 -j ACCEPT
-A FORWARD -i ens4 -j ACCEPT
COMMIT
*nat
:PREROUTING ACCEPT [1:76]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -s 172.22.141.0/24 -j MASQUERADE
COMMIT
```

NOTE: The RHEL OSP Installer system in this case is a VM created on a KVM server that has connectivity to the private VLAN for provisioning network (as available to the deployed nodes).

4.2.2 Subscribing to the required channels using Subscription Manager

Register the installer server using the customer delivery network (CDN) customer portal user name/password, list and attach the respective pools for the channels listed in Table 3.1.1.1: Required Channels -Installer. Please refer to Subscription management documentation for details https://access.redhat.com/documentation/en-US/Red_Hat_Subscription_Management/

4.2.3 Download the RHEL OSP Installer

Download the installer script.

```
[root@osp-rinst2 ~]# yum install -y rhel-osp-installer
```

4.2.4 Install the RHEL OSP Installer

1. Run the installer and follow through the prompts to enter appropriate settings.

```
[root@osp-rinst2 ~]# rhel-osp-installer

Please select NIC on which you want provisioning enabled:
1. ens3
2. ens4
?  2                          (Choose ens4 as the provisioning interface)
```

2. Based on the network interface selected, the network setup for the deployed nodes are configured in the installer

```
Networking setup:
       Network interface: 'ens4'
              IP address: '172.22.141.57'  --> IP Address of RHEL OSP Installer
            Network mask: '255.255.255.0'
         Network address: '172.22.141.0'   --> IP Address for PXE Network
            Host Gateway: '10.19.143.254'  --> Gateway for the install host
        DHCP range start: '172.22.141.54'   Address range for deployed servers
          DHCP range end: '172.22.141.254'
            DHCP Gateway: '172.22.141.57'  --> Gateway for PXE Network
           DNS forwarder: '10.19.143.247' --> External DNS server
                  Domain: 'example.com'  --> Domain name for deployed servers
           NTP sync host: '0.rhel.pool.ntp.org'
                Timezone: 'America/New_York'
Configure networking on this machine: ✓
Configure firewall on this machine: ✓

The installer can configure the networking and firewall rules on this
machine with the above configuration. Default values are populated from the
this machine's existing networking configuration.

If you DO NOT want to configure networking please set 'Configure networking
on this machine' to No before proceeding. Do this by selecting option 'Do
not configure networking' from the list below.
```

3. The predetermined parameters can be changed. In this case, NTP server has been modified as described below.

```
How would you like to proceed?:
1. Proceed with the above values
2. Change Network interface
3. Change IP address
4. Change Network mask
5. Change Network address
6. Change Host Gateway
7. Change DHCP range start
8. Change DHCP range end
9. Change DHCP Gateway
10. Change DNS forwarder
11. Change Domain
12. Change NTP sync host
13. Change Timezone
14. Configure networking
15. Configure firewall
16. Cancel Installation
12
new value for NTP sync host
clock1.example.com,clock2.example.com
Networking setup:
       Network interface: 'ens4'
              IP address: '172.22.141.57'
            Network mask: '255.255.255.0'
         Network address: '172.22.141.0'
```

```
             Host Gateway: '10.19.143.254'
         DHCP range start: '172.22.141.58'
           DHCP range end: '172.22.141.254'
             DHCP Gateway: '172.22.141.57'
            DNS forwarder: '10.19.143.247'
                   Domain: 'osplocal.example.com'
            NTP sync host: 'clock1.example.com,clock2.example.com'
                 Timezone: 'America/New_York'
Configure networking on this machine: ✓
Configure firewall on this machine: ✓

The installer can configure the networking and firewall rules on this
machine with the above configuration. Default values are populated from the
this machine's existing networking configuration.

If you DO NOT want to configure networking please set 'Configure networking
on this machine' to No before proceeding. Do this by selecting option 'Do
not configure networking' from the list below.
```

4. Additional values typically changed include the time zone and the DNS domain for the provisioning network. In this example, the provisioning network domain name is updated to be “osplocal.example.com”. The IP addresses assigned to the OSP hosts on the provisioning network will resolve to this domain.

5. If all the values are correct, the network configuration is setup by selecting the “proceed” option.

```
How would you like to proceed?:
1. Proceed with the above values
2. Change Network interface
3. Change IP address
4. Change Network mask
5. Change Network address
6. Change Host Gateway
7. Change DHCP range start
8. Change DHCP range end
9. Change DHCP Gateway
10. Change DNS forwarder
11. Change Domain
12. Change NTP sync host
13. Change Timezone
14. Configure networking
15. Configure firewall
16. Cancel Installation
1
Configure client authentication
          SSH public key: ''
           Root password: '********************************************'

Please set a default root password for newly provisioned machines. If you
choose not to set a password, it will be generated randomly. The password
must be a minimum of 8 characters. You can also set a public ssh key which
will be deployed to newly provisioned machines.
```

6. The root password for the deployed servers can be set and verified. SSH key

authentication can be configured to allow passwordless root authentication between the OSP installation server and the OSP infrastructure. The usage of SSH key authentication and the complexity of the root password should meet local security compliance requirements.

```
How would you like to proceed?:
1. Proceed with the above values
2. Change SSH public key
3. Change Root password
4. Toggle Root password visibility
2
You may either use a path to your public key file or enter the whole key
(including type and comment)
file or key
/root/.ssh/id_rsa.pub
Configure client authentication
          SSH public key: 'ssh-rsa AAAAB ... M4dz root@ospceph-rinst2'
           Root password: '********************************************'

Please set a default root password for newly provisioned machines. If you
choose not to set a password, it will be generated randomly. The password
must be a minimum of 8 characters. You can also set a public ssh key which
will be deployed to newly provisioned machines.

How would you like to proceed?:
1. Proceed with the above values
2. Change SSH public key
3. Change Root password
4. Toggle Root password visibility
3
new value for root password
********
enter new root password again to confirm
********
Configure client authentication
          SSH public key: ''
           Root password: '********'

Please set a default root password for newly provisioned machines. If you
choose not to set a password, it will be generated randomly. The password
must be a minimum of 8 characters. You can also set a public ssh key which
will be deployed to newly provisioned machines.

How would you like to proceed?:
1. Proceed with the above values
2. Change SSH public key
3. Change Root password
4. Toggle Root password visibility
4
Configure client authentication
          SSH public key: ''
           Root password: 'redhat11'

Please set a default root password for newly provisioned machines. If you
choose not to set a password, it will be generated randomly. The password
```

```
must be a minimum of 8 characters. You can also set a public ssh key which
will be deployed to newly provisioned machines.

How would you like to proceed?:
1. Proceed with the above values
2. Change SSH public key
3. Change Root password
4. Toggle Root password visibility
4
Configure client authentication
          SSH public key: ''
           Root password: '********'

Please set a default root password for newly provisioned machines. If you
choose not to set a password, it will be generated randomly. The password
must be a minimum of 8 characters. You can also set a public ssh key which
will be deployed to newly provisioned machines.

How would you like to proceed?:
1. Proceed with the above values
2. Change SSH public key
3. Change Root password
4. Toggle Root password visibility
1
Installing             Done
[100%] [.......................................................]
Starting configuration...
Redirecting to /bin/systemctl stop  puppet.service
Redirecting to /bin/systemctl start  puppet.service

Now you should configure installation media which will be used for
provisioning.
Note that if you don't configure it properly, host provisioning won't work
until you configure installation media manually.
```

7. Specify the RHEL 7.1 URL that is used for deployed servers.

```
Enter RHEL repo path:
1. Set RHEL repo path (http or https URL): http://
2. Proceed with configuration
3. Skip this step (provisioning won't work)
1
Enter RHEL repo path:
1.  Set RHEL repo path (http or https URL):
http://download.example.com/released/RHEL-7/7.1/Server/x86_64/os/
(Use appropriate repo path)
2. Proceed with configuration
3. Skip this step (provisioning won't work)
2
```

8. Provide subscription manager inputs along with username, password and required channels

```
Enter your subscription manager credentials:
1. Subscription manager username:
2. Subscription manager password:
```

```
3. Comma separated repositories: rhel-7-server-openstack-6.0-rpms
rhel-7-server-openstack-6.0-installer-rpms rhel-7-server-rh-common-rpms
4. Subscription manager pool (recommended):
5. Subscription manager proxy hostname:
6. Subscription manager proxy port:
7. Subscription manager proxy username:
8. Subscription manager proxy password:
9. Proceed with configuration
10. Skip this step (provisioning won't subscribe your machines)
1
Username: redhat_subscription_user
```

9. Select 2 to enter password

```
2
Password: *******
```

10. Select **3** to enter the required repositories

```
rhel-7-server-openstack-6.0-rpms,rhel-7-server-openstack-6.0-installer-rpms,
rhel-7-server-rh-common-rpms
```

11. (Optional) Select 4 to enter a specific subscription manager pool id

```
4
Pool: 8a85..........12
```

12. Proceed with installation

```
9
Starting to seed provisioning data
Use 'base_RedHat_7' hostgroup for provisioning
  Success!
  * Foreman is running at https://ospceph-rinst2.example.com
      Initial credentials are admin / tYZdbSzWZMCvDTw3
  * Foreman Proxy is running at https://ospceph-rinst2.example.com:8443
  * Puppetmaster is running at port 8140
  The full log is at /var/log/rhel-osp-installer/rhel-osp-installer.log
```

13. The OSP installer installation is complete. The application URL and login credentials are displayed in the final output. The login credentails are also stored in /etc/foreman/rhel-osp-installer.answers.yaml.

    Please refer to C.1 Changing Installer GUI Password for instructions to change the password for the admin account.

4.2.5 Handling installer failures

If the "rhel-osp-installer" script gets interrupted or fails, or if changes are required, the recommended approach is to re-run the script.

Please refer to Appendix B.1 Troubleshooting failed RHEL OSP Installer script for details on how to re-execute the script.

4.2.6 Boot managed nodes into discovery mode

After the managed nodes have been set to PXE boot on the provisioning NIC, booting them initiates a Foreman Discovery process. Once the nodes have PXE booted properly, they show as "discovered" by the RHEL OpenStack Installer on the Discovered hosts page. This can take a few minutes. The network details are displayed in 3.1Figure 3.1 1: OpenStack-Ceph Environment.The assigned hostname will be based on the MAC addresses of the network interface on the provisioning network.

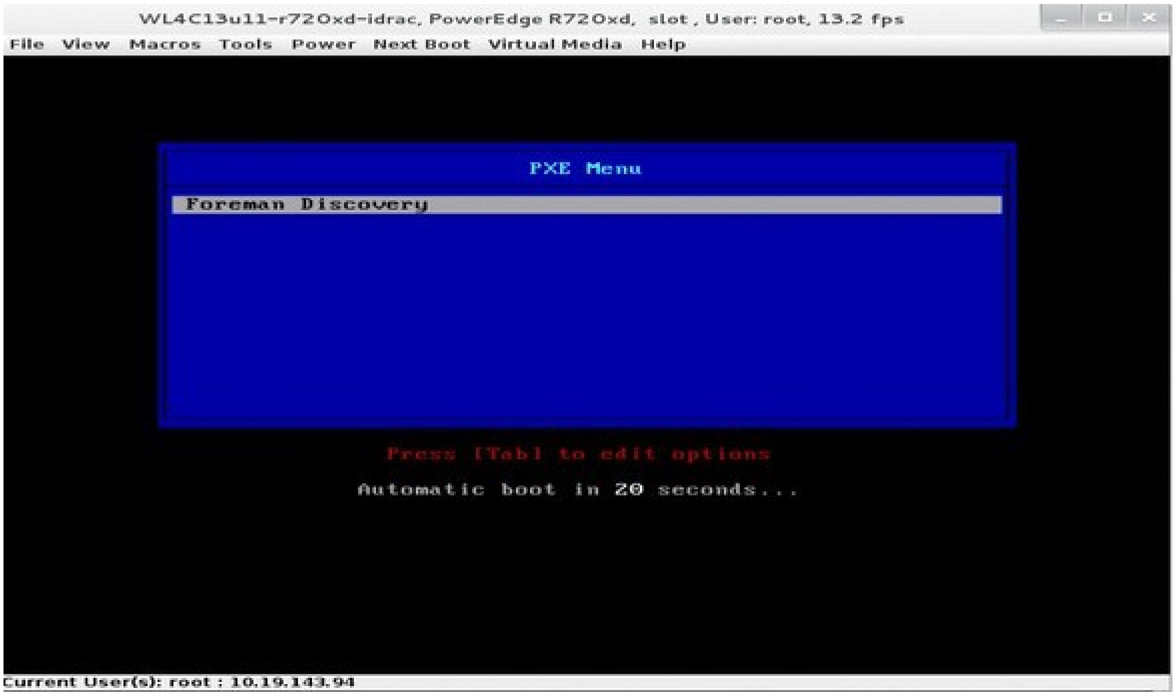


Figure 4.2.6 1: Discovery

On the Installer Gui main page ***Hosts-> Discovered Hosts***

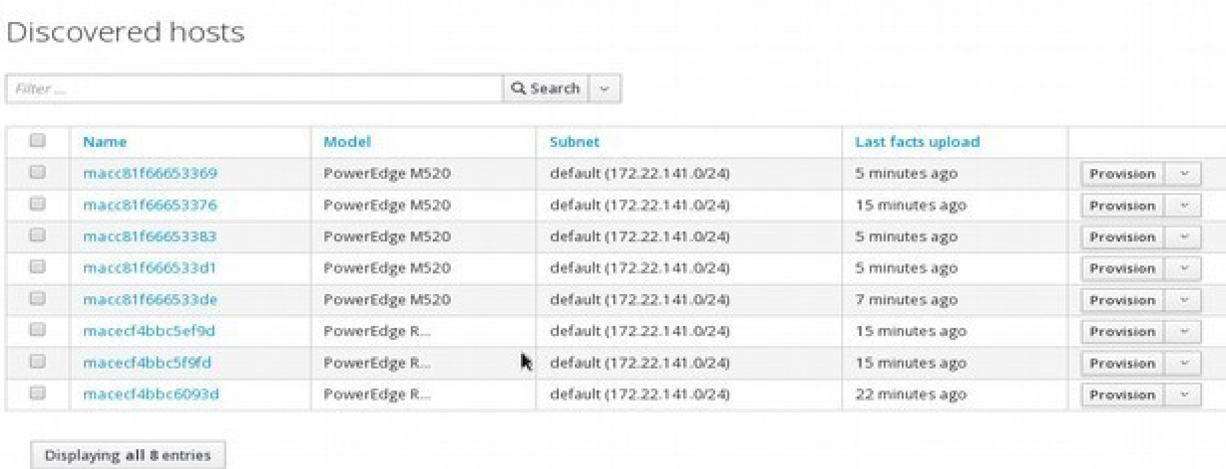


Figure 4.2.6 2: Discovered Hosts

NOTE: Please refer to Appendix B.2 Troubleshooting discovery failure for discovery failures related to multiple DHCP networks under certain conditions.

4.3 OpenStack Deployment

The following is an HA deployment consisting of three Controller nodes, two compute nodes and three Ceph OSD nodes. Each controller node hosts the functions of Controller node, Neutron Network nodes, and Ceph Monitor nodes.

1. Create a new deployment.

   Select ***OpenStack Installer -> New Deployment***

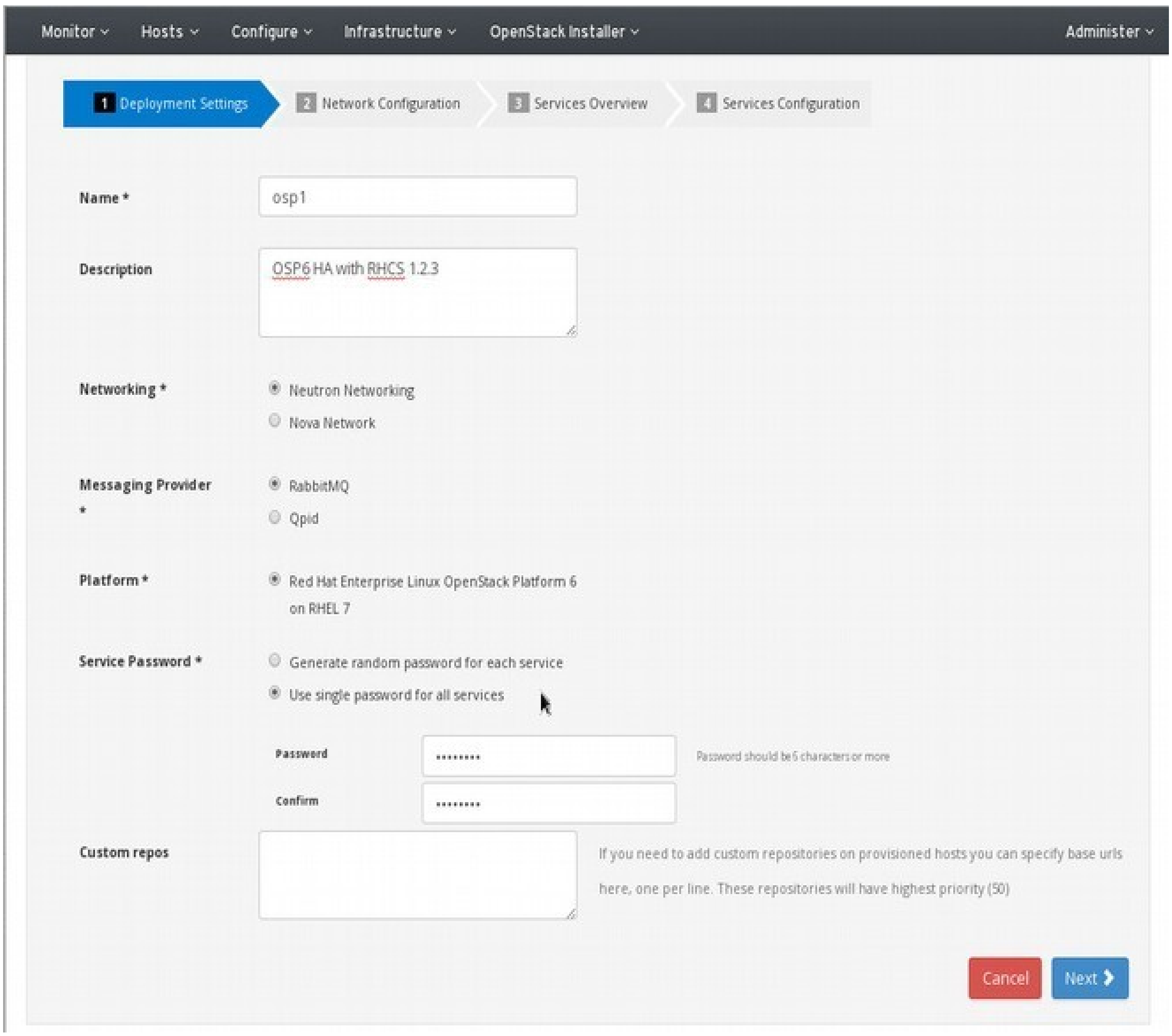


Figure 4.3 1: RHEL OSP Installer New Deployment

2. Some of the options to select are:
   - Networking (Neutron or Nova)
   - Messaging Provider (RabbitMQ or Qpid)
   - Service Password (Random password for each service or a Single manually assigned password for all services)

For this reference archeticutre, Neutron is used for networking. RabbitMQ is used as the Message Providor and a uniform password is use for all services. Uniform passwords are not recommended for a production installation.

3. Network Configuration

   This page enables assigning the right traffic type to their respective subnets. This setting determines the number of network interfaces used in the OpenStack deployment, type of IP assignment and other network related configurations set in the subnets. The subnets can be created ahead of deployment. There is also an option to create subnets from the new deployment page as well, as shown below.

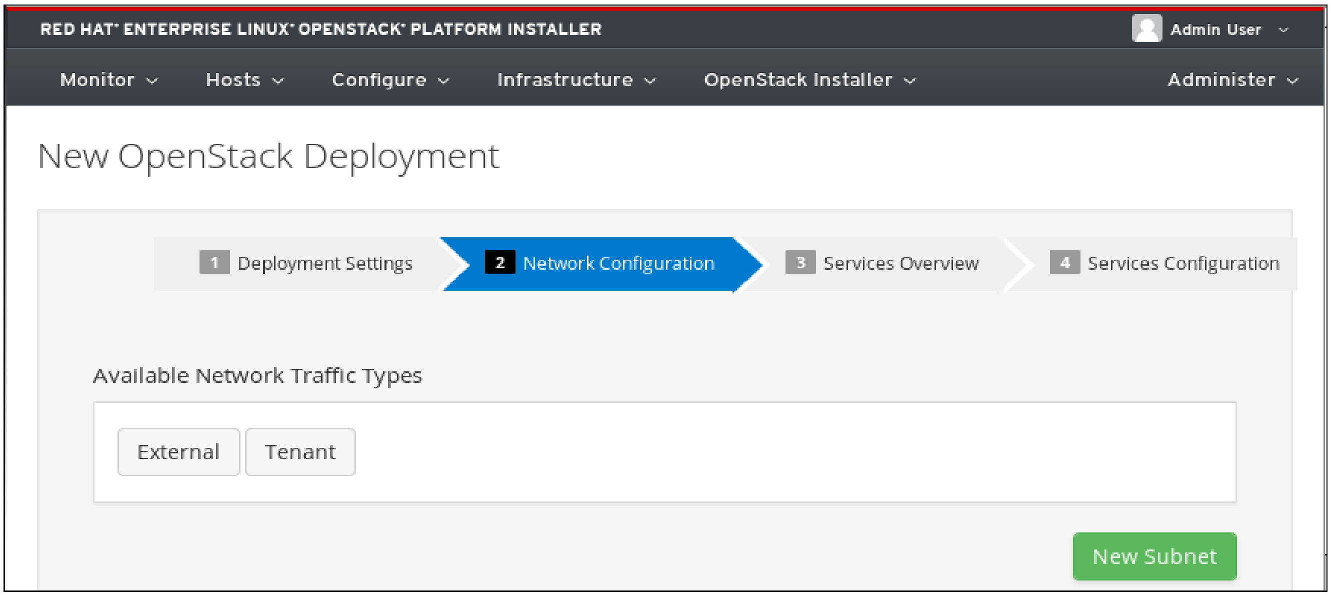


Figure 4.3 2: Create New Subnets

The following displays “tenant subnet” details.

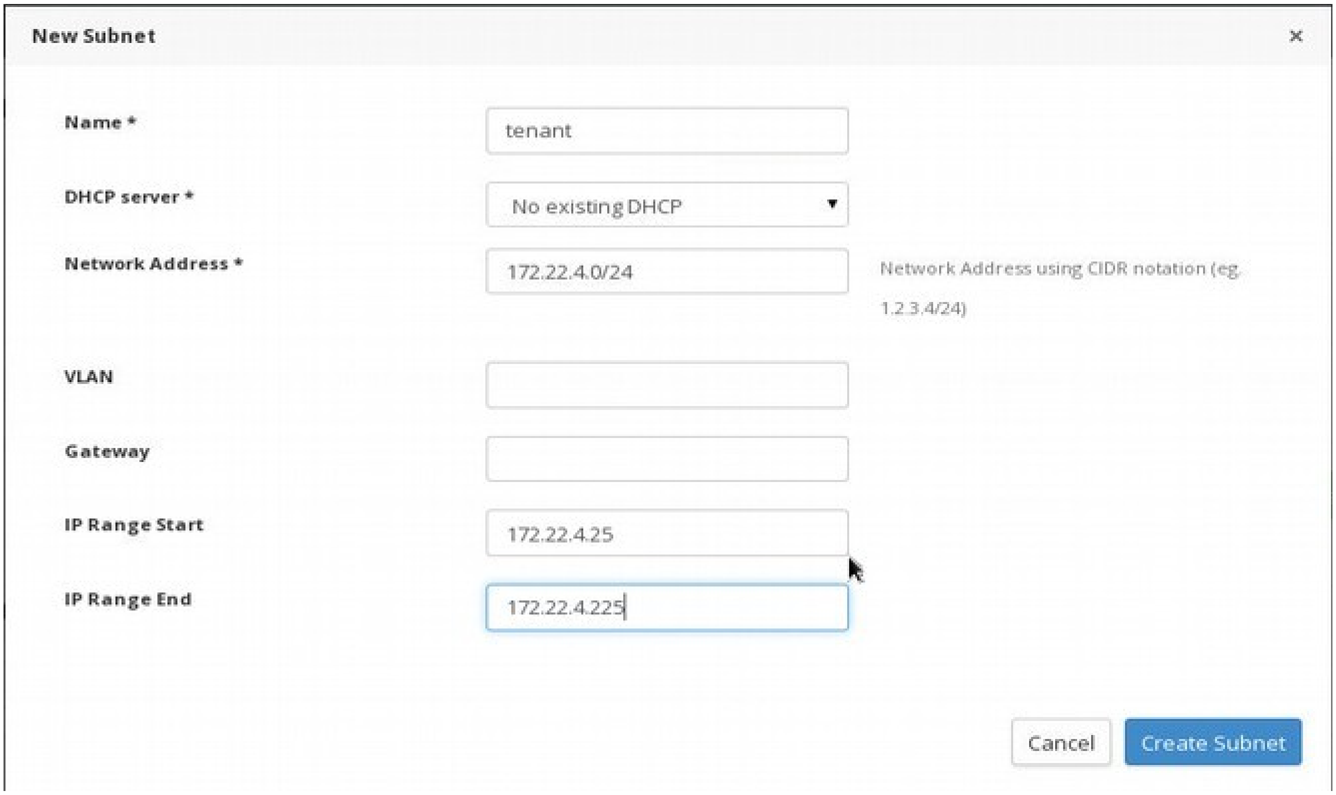


Figure 4.3 3: Create New Subnets - Assign Values

In this reference architecture, the “No Existing DHCP” option is selected for parameter “DHCP Server”. This mandates providing an IP Start and End range. For details on creating a subnet with IPAM setting as DHCP, please refer to https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/6/html/Installer_and_Foreman_Guide/chap-Deployment_Scenario_1_Basic_Environment.html#Creating_an_External_Subnet1

4. Figure 4.3 4: Subnets List lists all the subnets created and used in this deployment. This information is accessible via Infrastructure -> Subnets

Subnets

Filter ... Search

Name	Network address	Domains	VLAN ID	DHCP Ca
storage	172.22.1.0/24			
storage-clsuter	172.22.2.0/24			
tenant	172.22.4.0/24			
external	10.19.136.0/21			
default	172.22.141.0/24	osplocal.example.com		ospceph

Figure 4.3 4: Subnets List

Table 4.3.1 lists the Network range used for each traffic type.

Subnet name	Network Range	
	From	To
• PXE • Admin API • Cluster Management • Public API • Management	172.22.141.58	172.22.141.254
External	10.19.141.25	10.19.141.250
Tenant	172.22.4.25	172.22.4.225
Storage	172.22.1.25	172.22.1.225
Storage Clustering	172.22.2.25	172.22.2.225

Table 4.3.1: Subnets - Network Range

5. Assign respective traffic type to the available subnets by dragging the traffic type from default to target subnet. If a traffic type is still under the default subnet, then it shares the network along with provisioning and others in the default subnet.

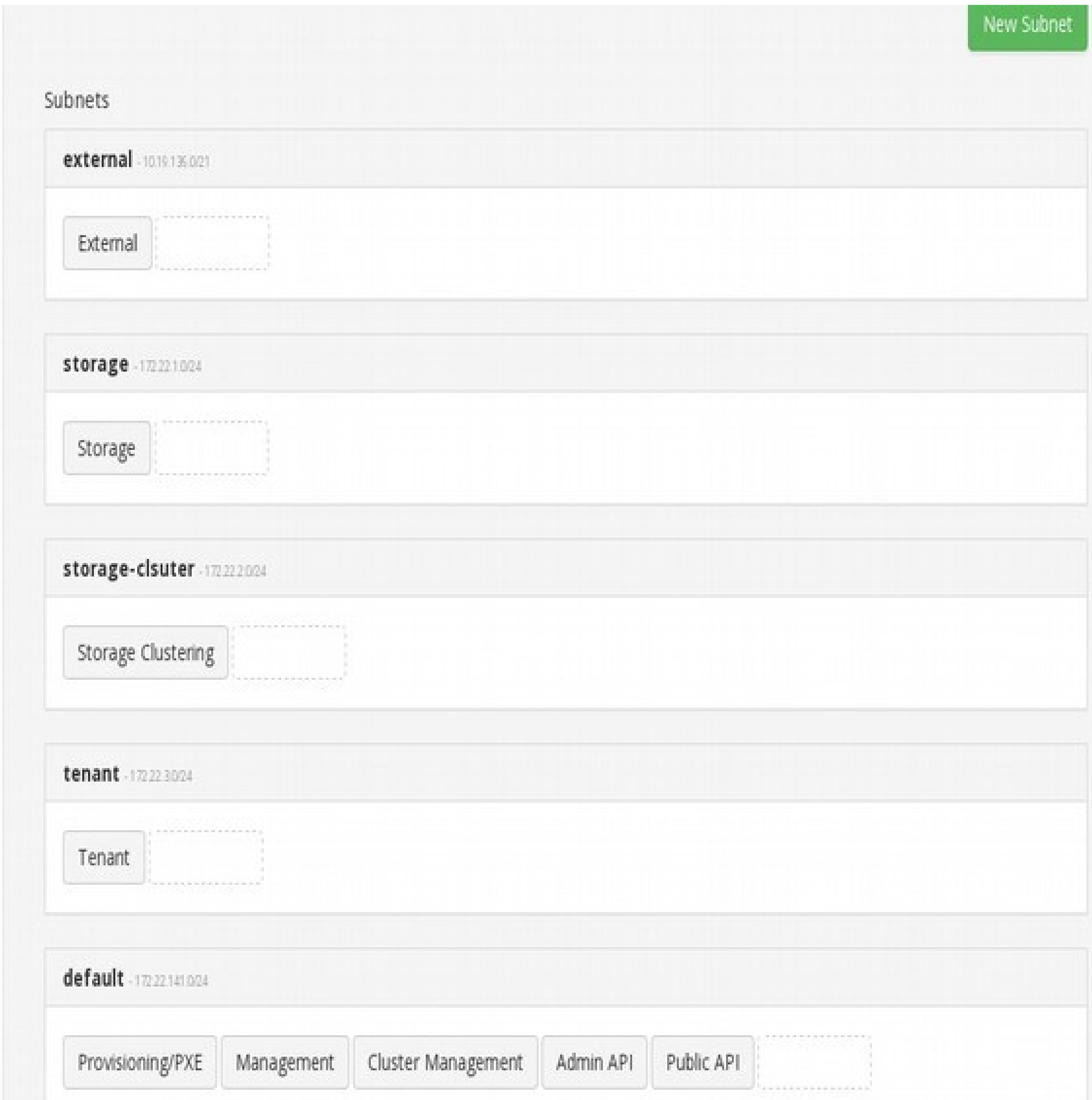


Figure 4.3 5: Assign Traffic Type to Subnets

NOTE: Provisioning is assigned to "default" and cannot be moved to a different subnet. By default, all the traffic types except "external and Tenant" are assigned to "default" subnet. If left that way, the deployment requires only three networks – External, Tenant and Provisioning. This setup is not recommended for production and can be used only for a proof of concept setup.

6. Display Services Overview for the Deployment Roles.

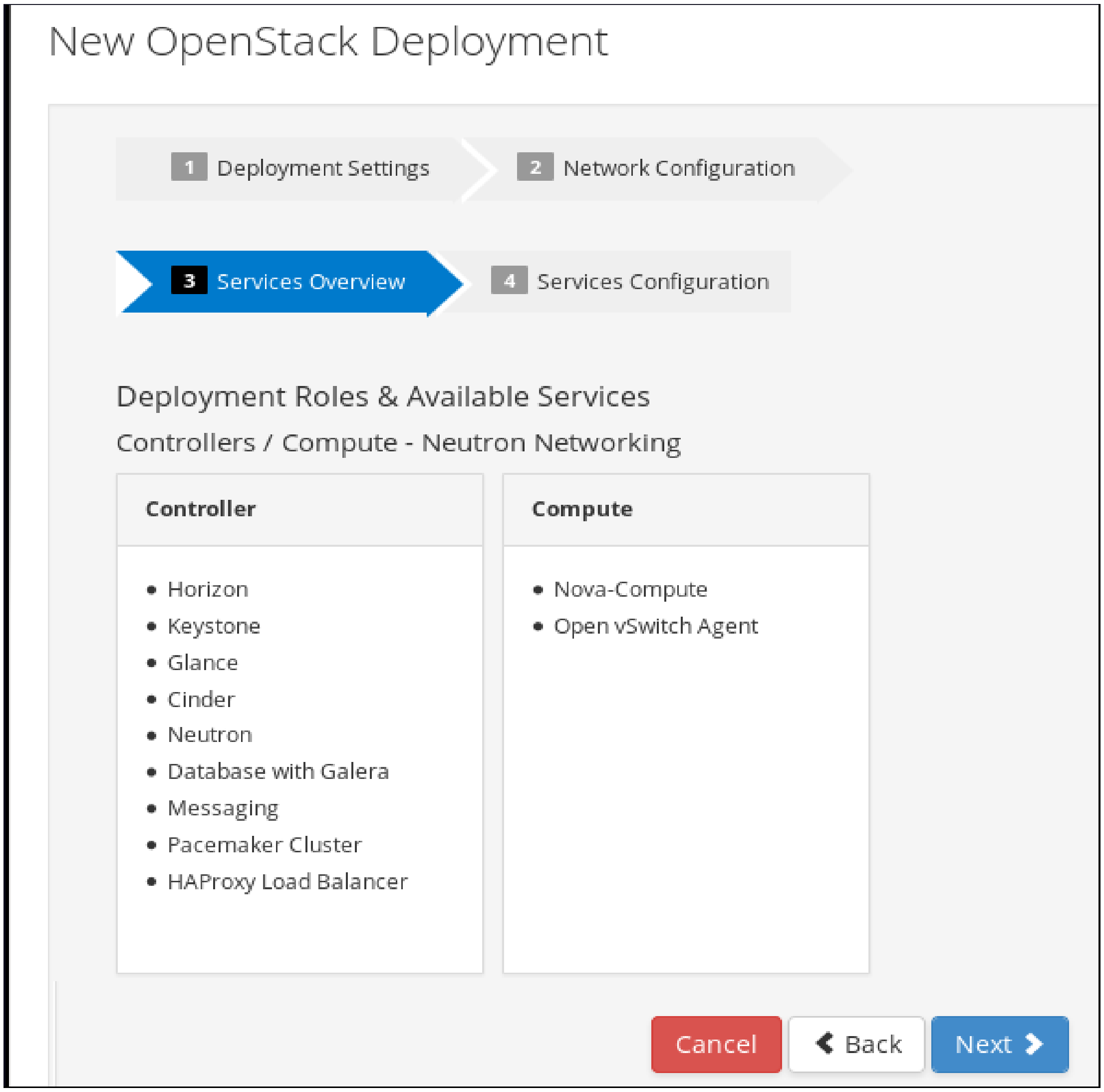


Figure 4.3 6: Services Overview

7. Select Neutron Service configuration

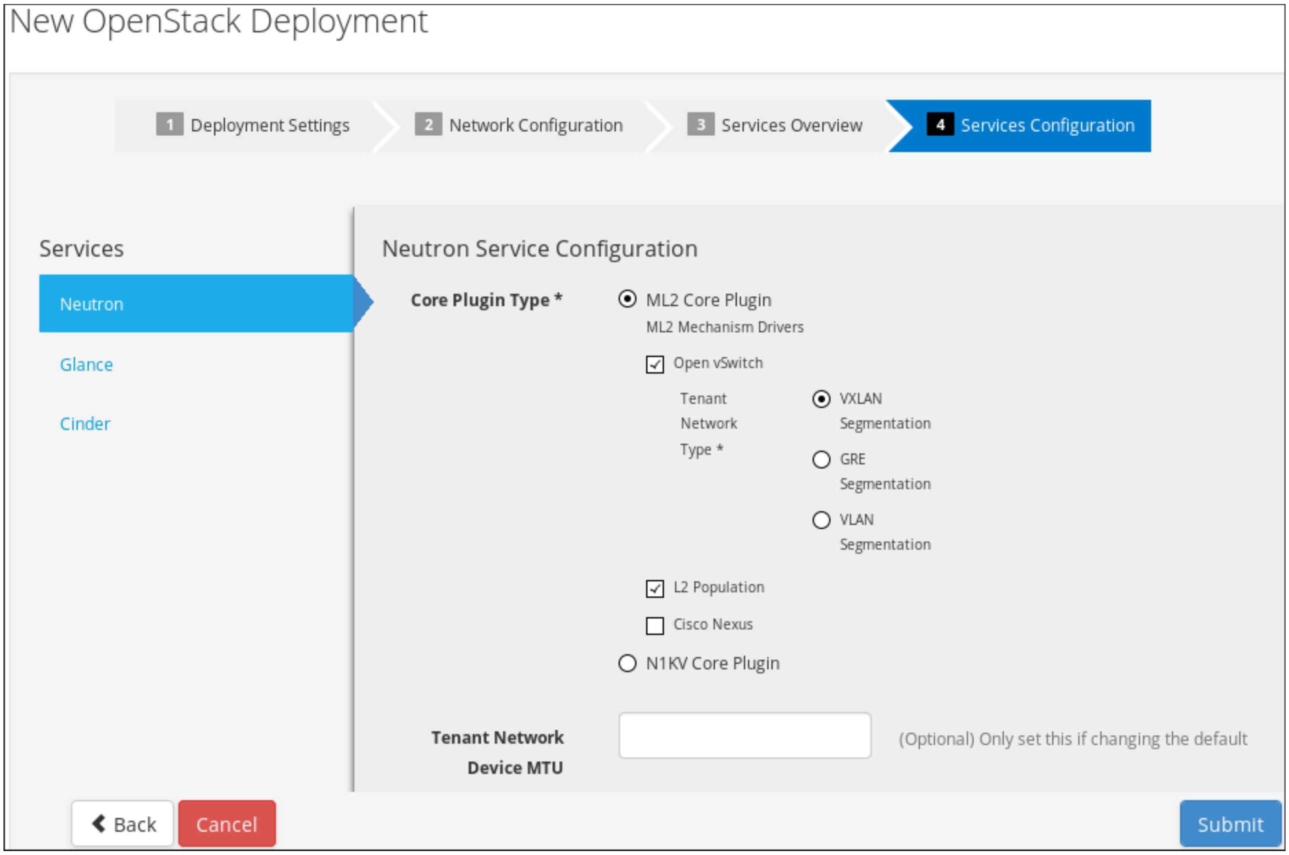


Figure 4.3 7: Neutron Service Configuration

8. Select Glance backend.

   In this reference architecture, Ceph is the backend storage for Glance with manual configuration steps are detailed in this document.

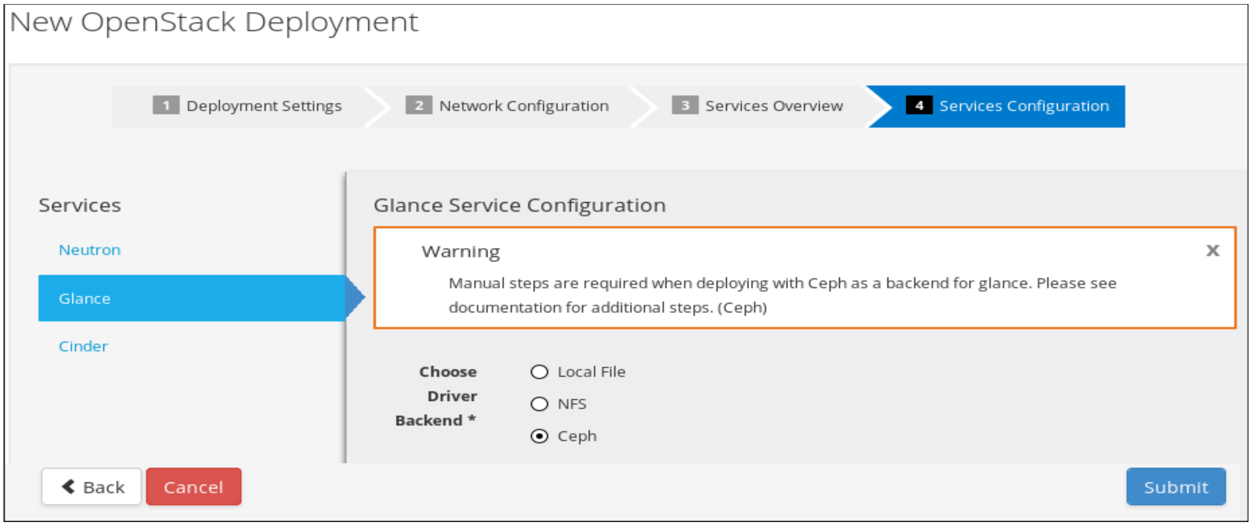


Figure 4.3 8: Glance backend - Ceph

9. Select Cinder backend.

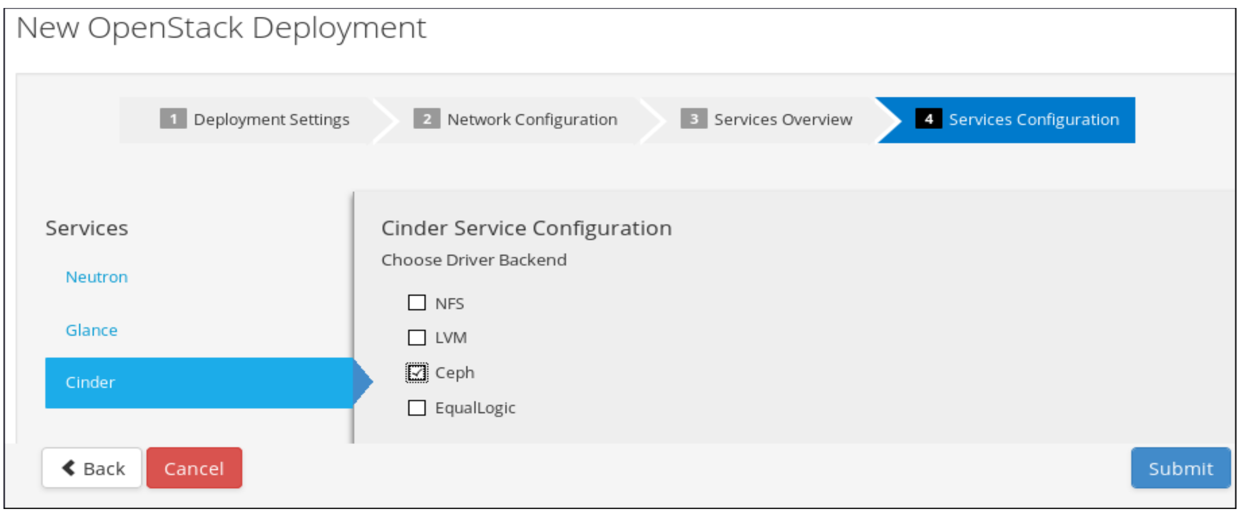


Figure 4.3 9: Cinder Backend - Ceph

In this reference architecture Ceph is the backend storage for Cinder.

10. Three TCP ports are required for each OSD. The allowed TCP port range must to be modified if a large number of OSDs are hosted on a server. Additional ports should be allowed to permit OSD restart. For example, if an OSD server provides 12 OSDs then 36 TCP ports are required for basic operation. This value can be changed via the Advanced Configuration tab of the deployment configuration screen. Additional ports are included in the range to permit service restart. The port range is updated by editing the deployment by clicking “Advanced Configuraiton” and changing the ports parameter of the Ceph Storage (OSD) (node) host group

Host Group	Key	Value
Ceph Storage (OSD) (node)	Ports	`6800:6850`

Table 4.3.2: Ceph OSD Port

11. The OSD XFS mount parameters must be updated due to BZ1212580. The following configuration parameters must be updated in the advanced configuration of OSP deployment.

Host Group	Key	Value
Controller	Ceph osd mount options xfs	`inode64,noatime,logbsize=256k`
Ceph Storage (OSD) (node)	Osd mount options xfs	`inode64,noatime,logbsize=256k`

Table 4.3.3: Ceph XFS Mount Options

12. The basic settings for the deployment have been completed.

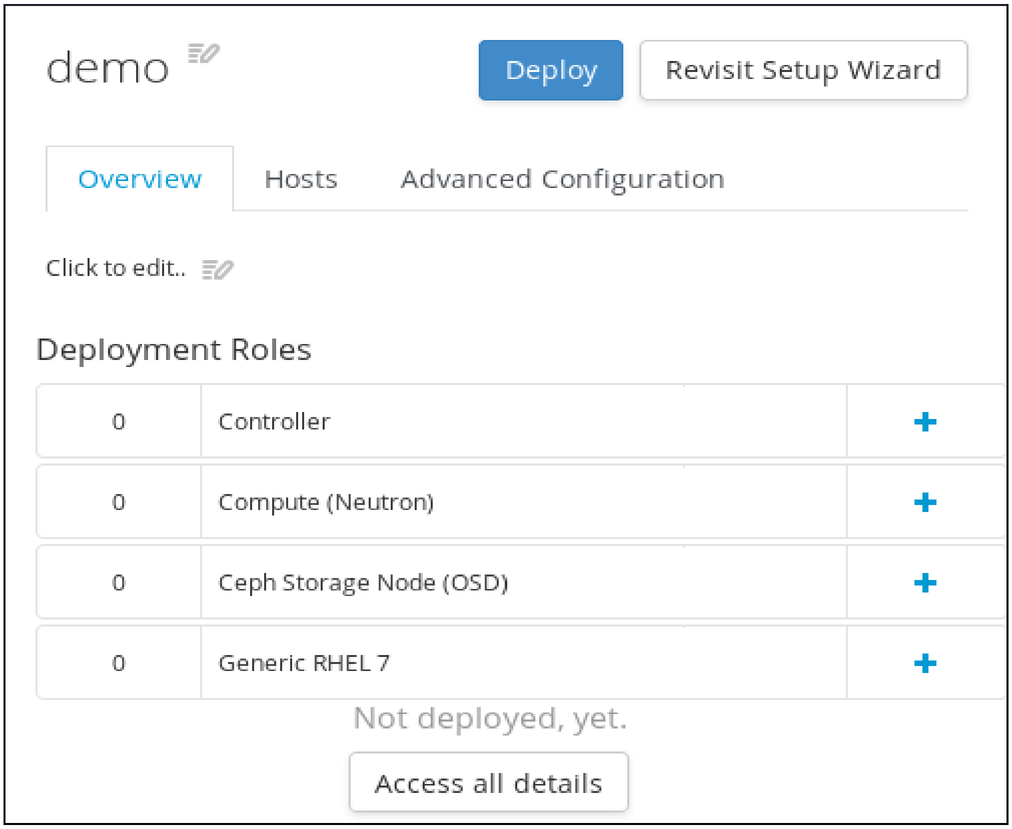


Figure 4.3 10: Deployment Basics Completed

13. The “Access all details” button can is used to view:
    - the allocated VIPs by the subnets assigned to the deployment.
    - users for the service and their (random or manually defined) passwords.
    - database for the services and their (random or manually defined) passwords.

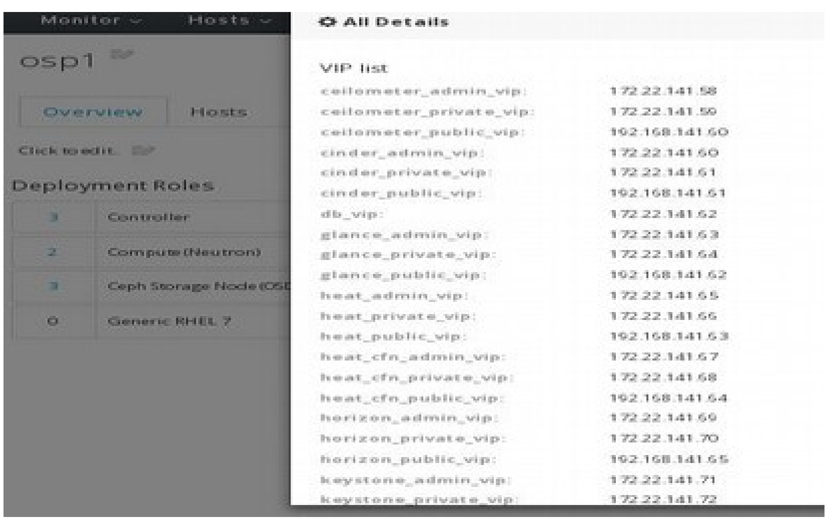


Figure 4.3 11: VIPs Users and Database Details

Refer to A.6 Deployment Assigned Values for full list of auto assigned values.

14. Assign the managed hosts to different Deployment Roles. Designated hosts are determined by the interface MAC address and available storage configuration.

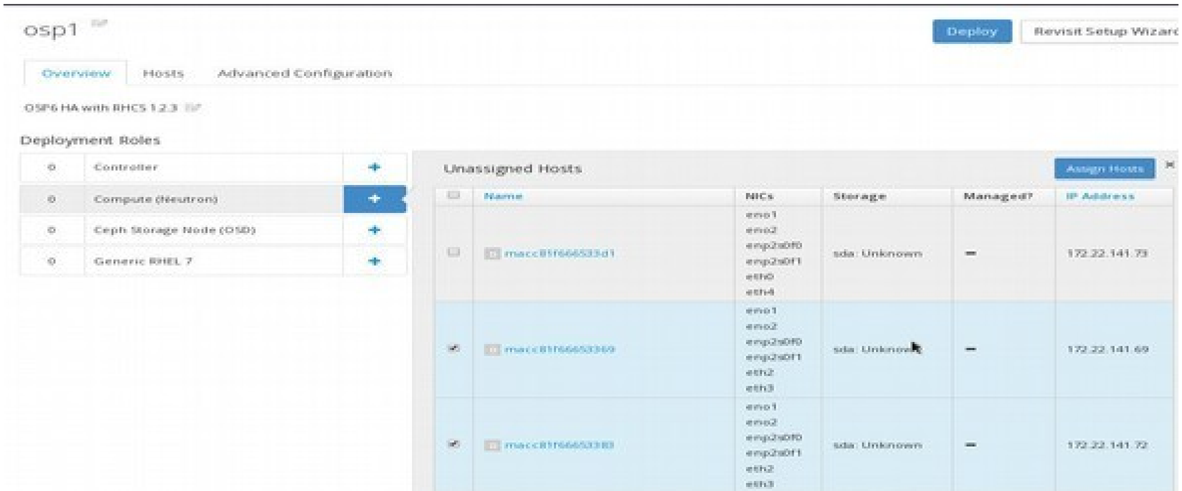


Figure 4.3 12: Assign Hosts to Deployment Roles

15. Display the hosts by the host group to review host assignment

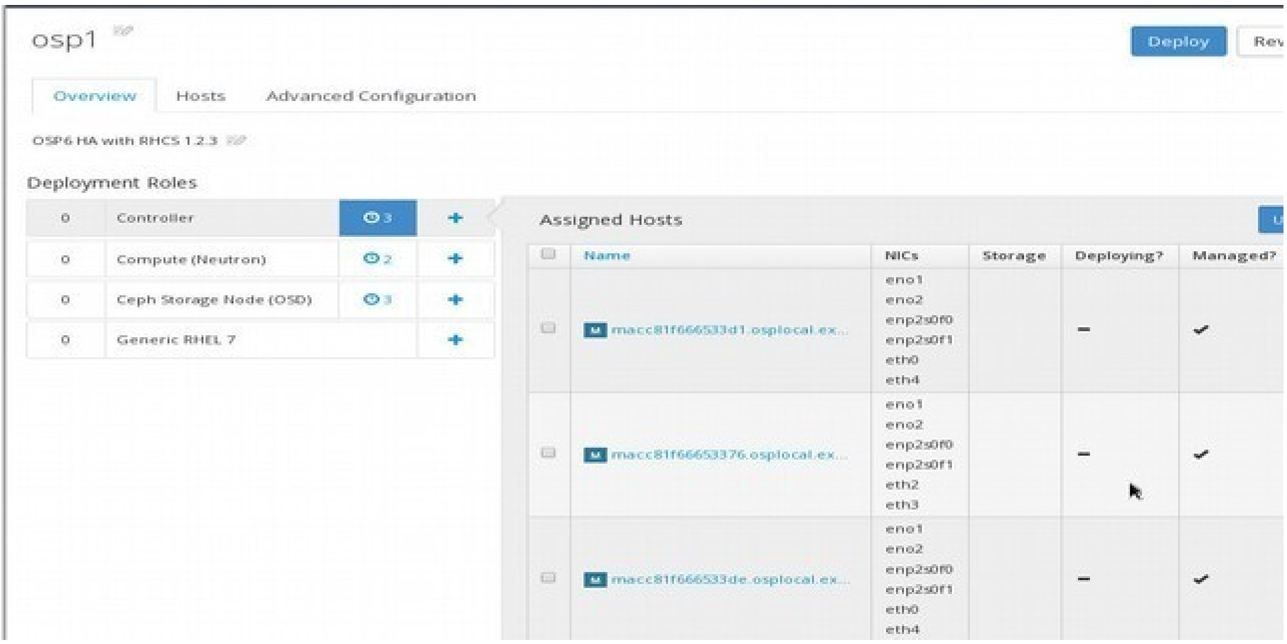


Figure 4.3 13: Assigned Hosts By Deployment Roles

16. The assigned hosts can be renamed to improve host identification by providing user friendly names. Assigned hosts can be displayed by selecting Hosts -> All Hosts

17. Select the Hosts tab and select the Assigned button to list the assigned hosts

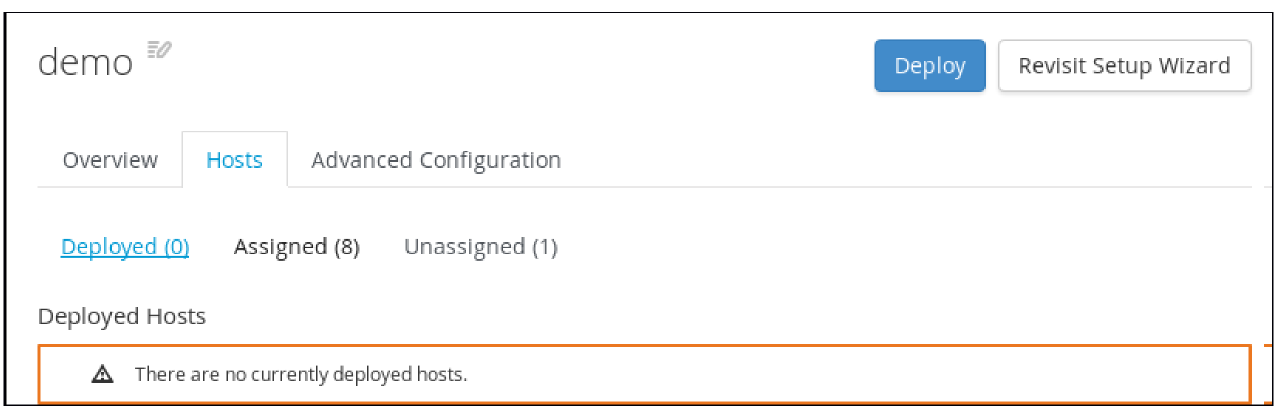


Figure 4.3 14: Total Hosts Assigned

18. Select a host or several hosts in the same Deployment Role and select “Configure Network” on the top right corner of the screen.

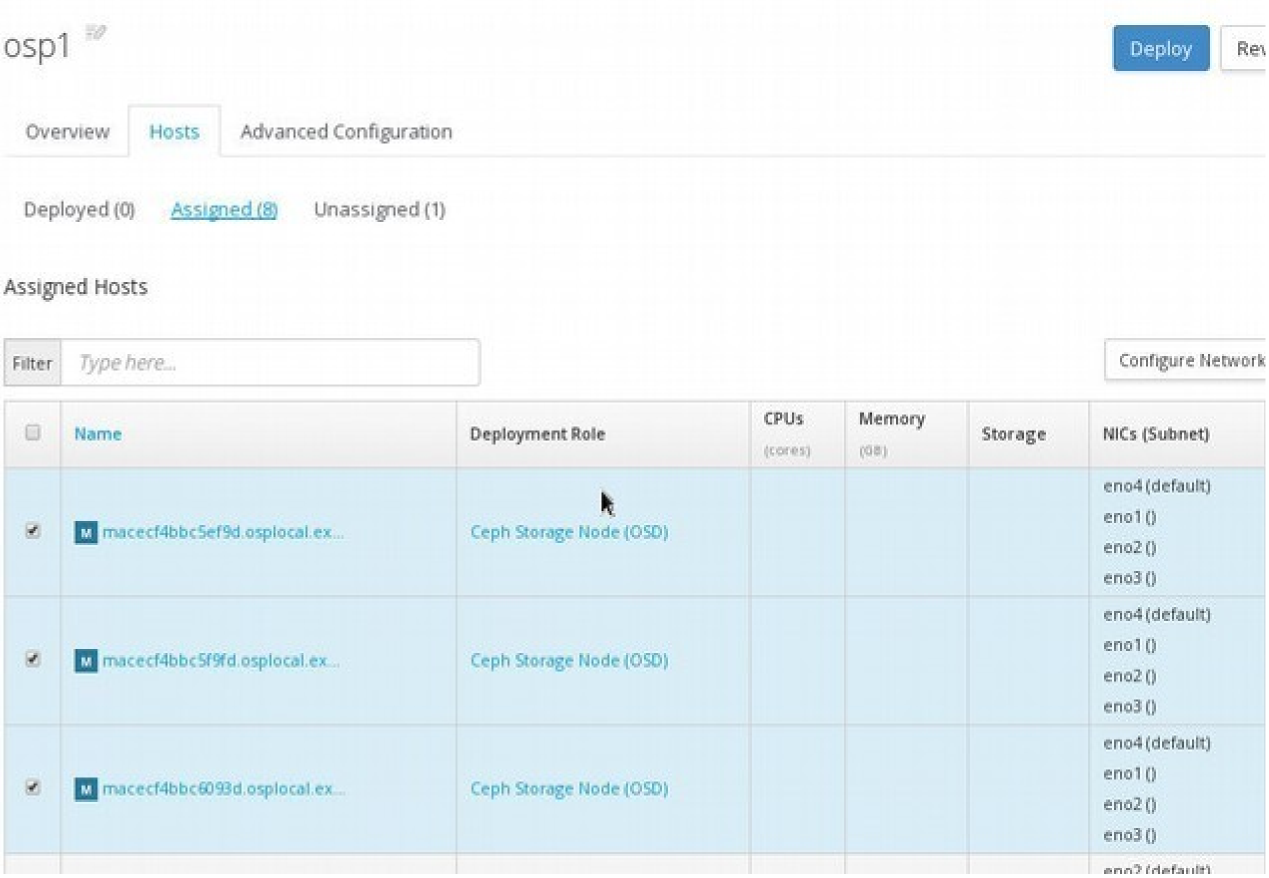


Figure 4.3 15: Configure Network For Hosts

NOTE: Multiple hosts can be selected to avoid repetitions. However, the available network device names and final subnet assignments must be identical.

19. The following displays the available network ports and the subnets.

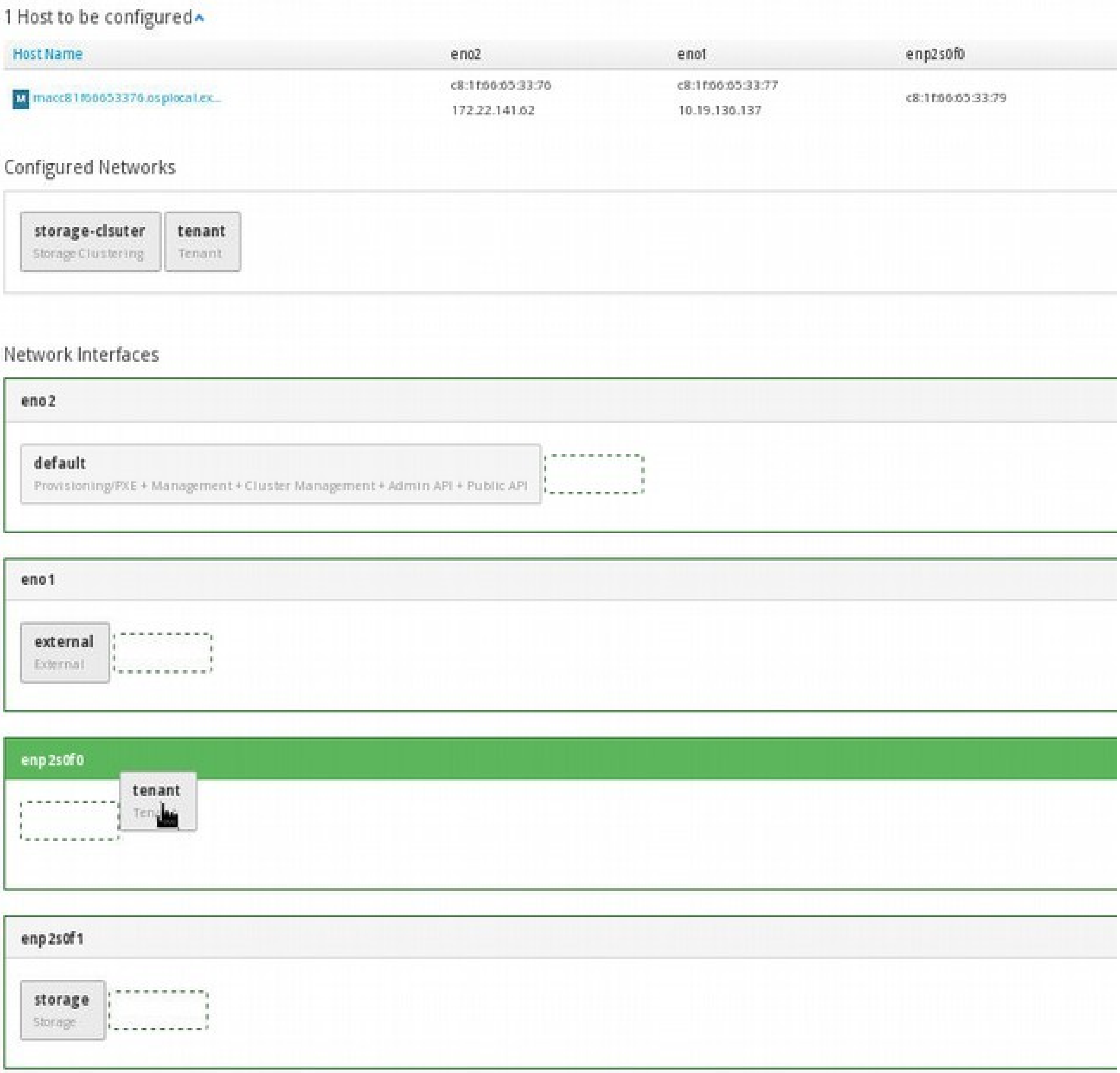


Figure 4.3 16: Available Network & Ports

NOTE: It is not necessary to assign all the subnets. In fact, they vary by deployment groups.

20. After the assignment has been completed, they appear as follows. The storage-cluster subnet is not assigned to this host

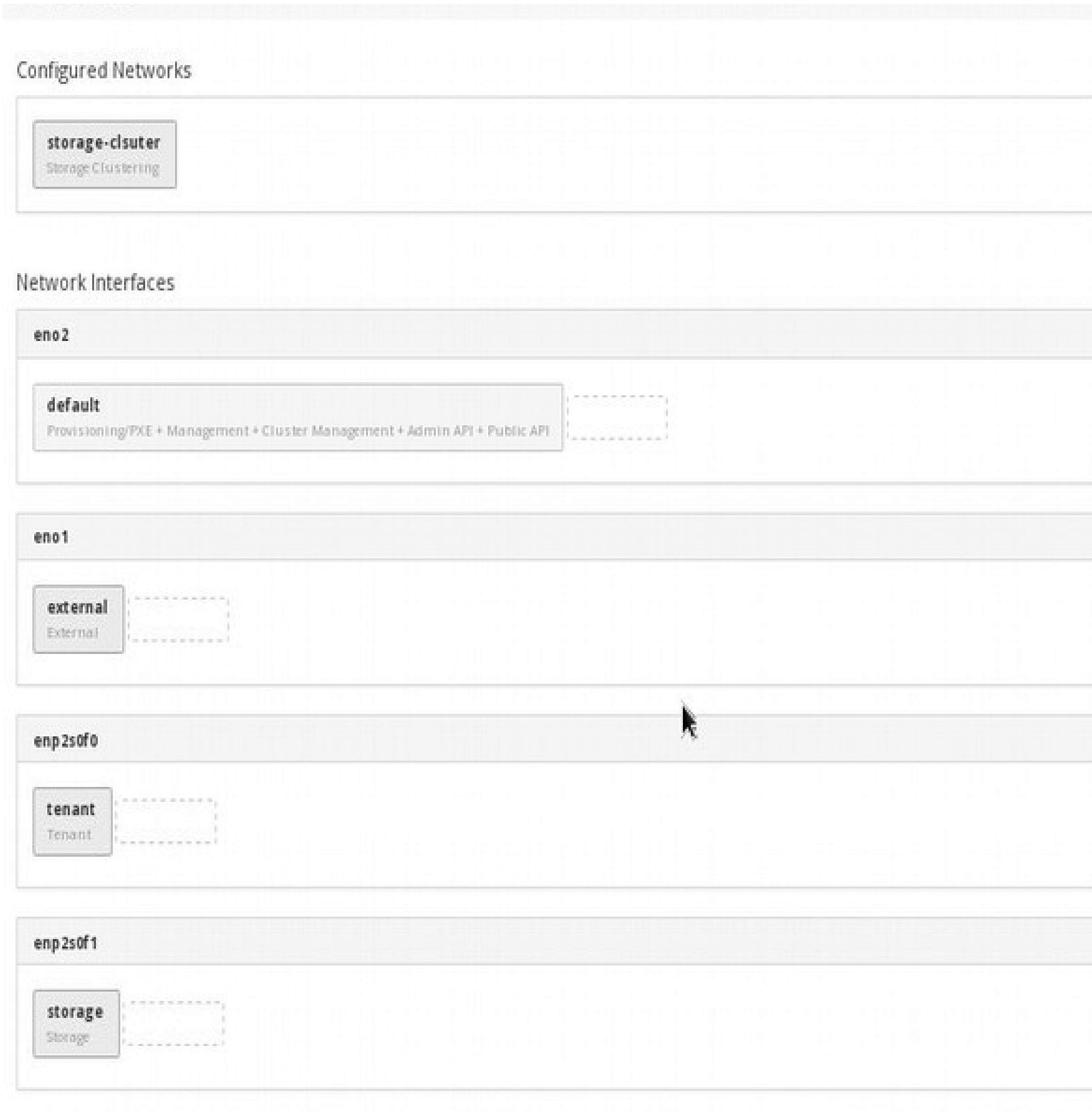


Figure 4.3 17: Assignment Completed View

21. After all the hosts have been configured for network, they can be edited/verified as follows.

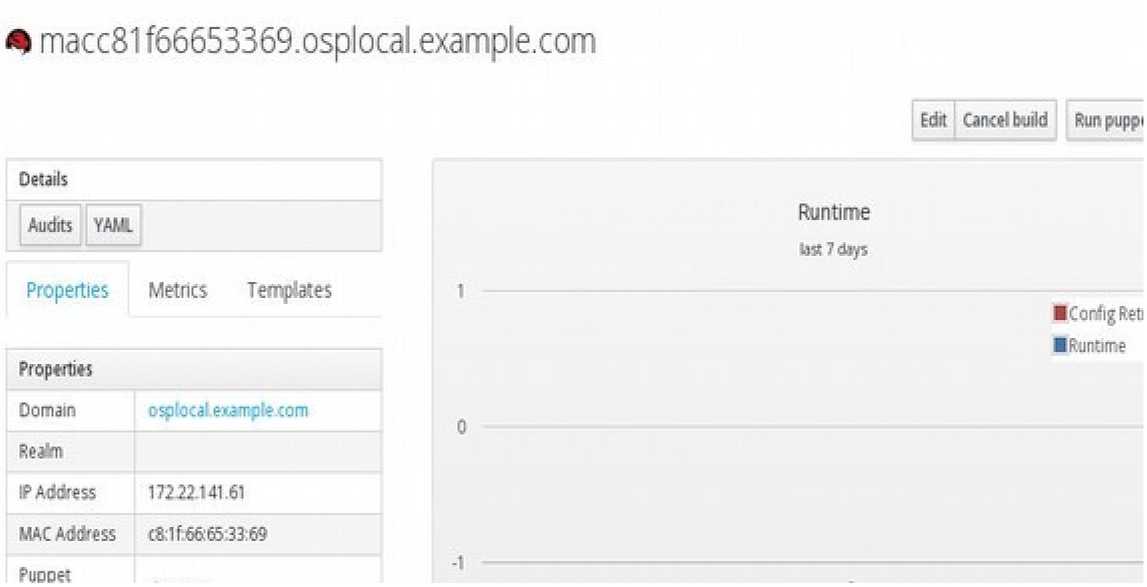
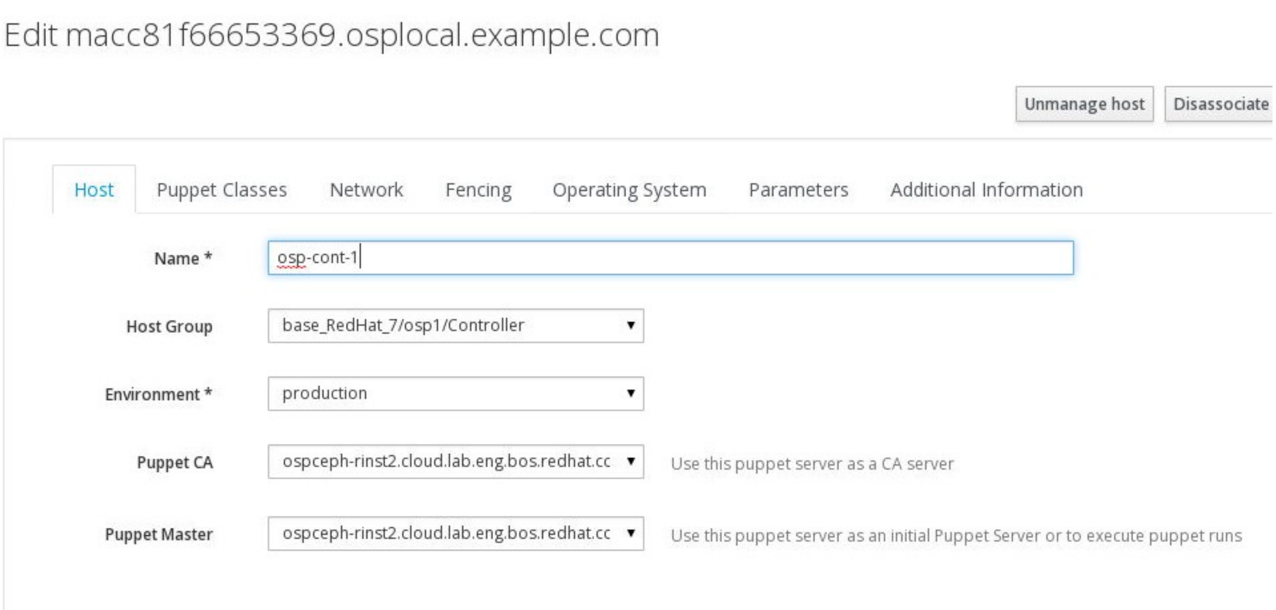


Figure 4.3 18: Edit Hosts

22. The hostname is updated to be user friendly. In this example, macc81f66653369 is changed to osp-cont-1



23. The controller nodes run pacemaker cluster and are protected by hardware fencing. The controller nodes must be configured in the following screen with fencing related information as follows. The IPMI settings, IP, User and Password are preset values on

the Dell blade chassis.

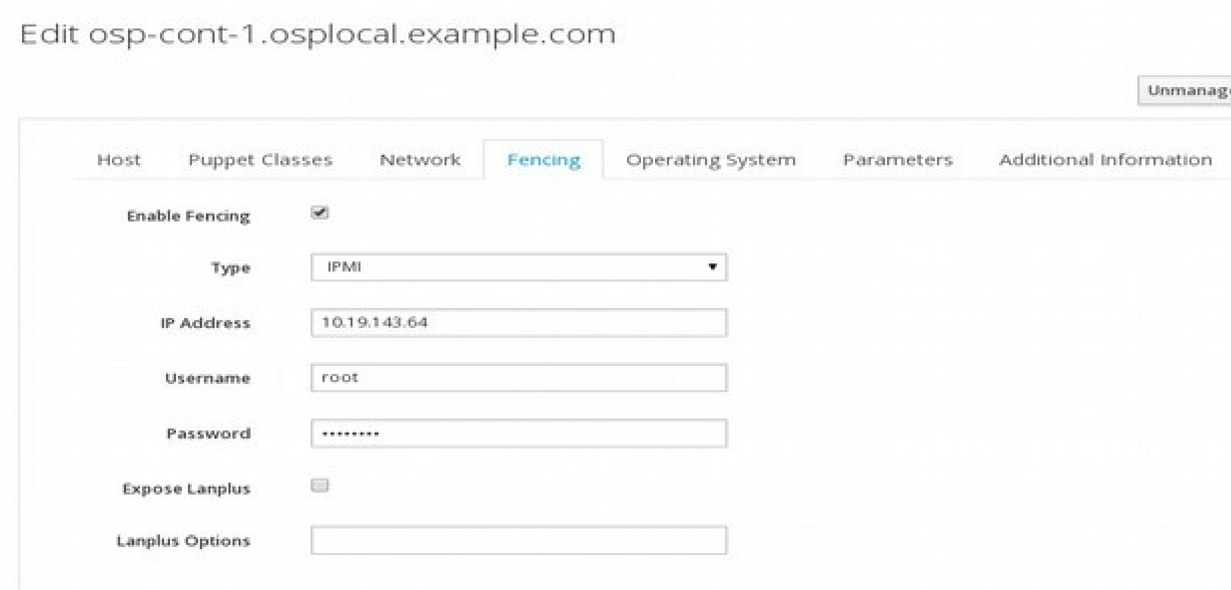


Figure 4.3 19: Configure Fencing

24. The subnet and network assignments can be verified/modified in the following screen.

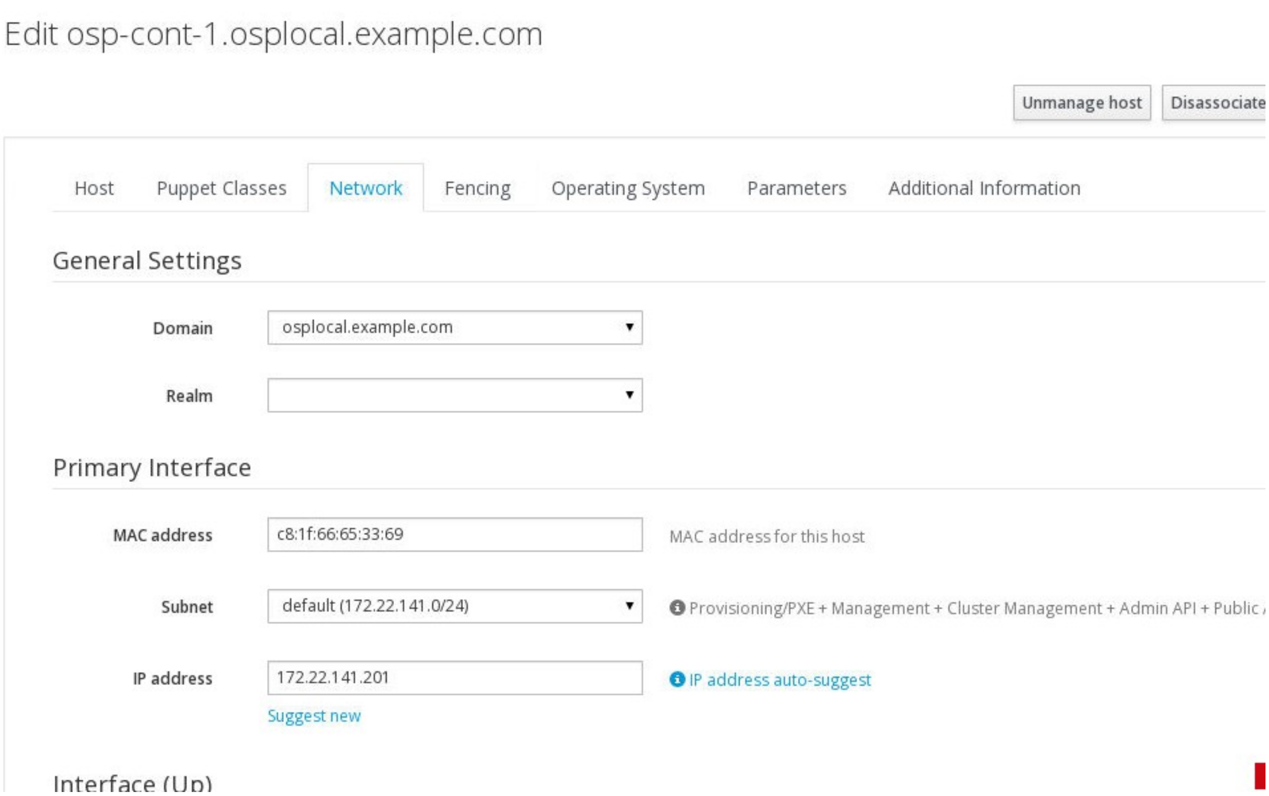


Figure 4.3 20: Network Edit

It is a good practice to verify and ensure all the networks are proper before selecting "Deploy" button. If there is misrepresentation of network, fixing networks post provisioning is a time consuming and daunting process.

Ensure IPs:

- are assigned to the correct subnet & network port
- fall within the subnet IP range
- are not duplicated elsewhere

25. After all the network assignments have been confirmed, deployment can be initiated by selecting the "deploy" button in the top right corner of the selected deployment screen. If the deployment has been successful, the status bar displays successful completion.

Please refer to B.3Troubleshooting Network assignment to ensure network assignment is in order before proceeding with deploy.

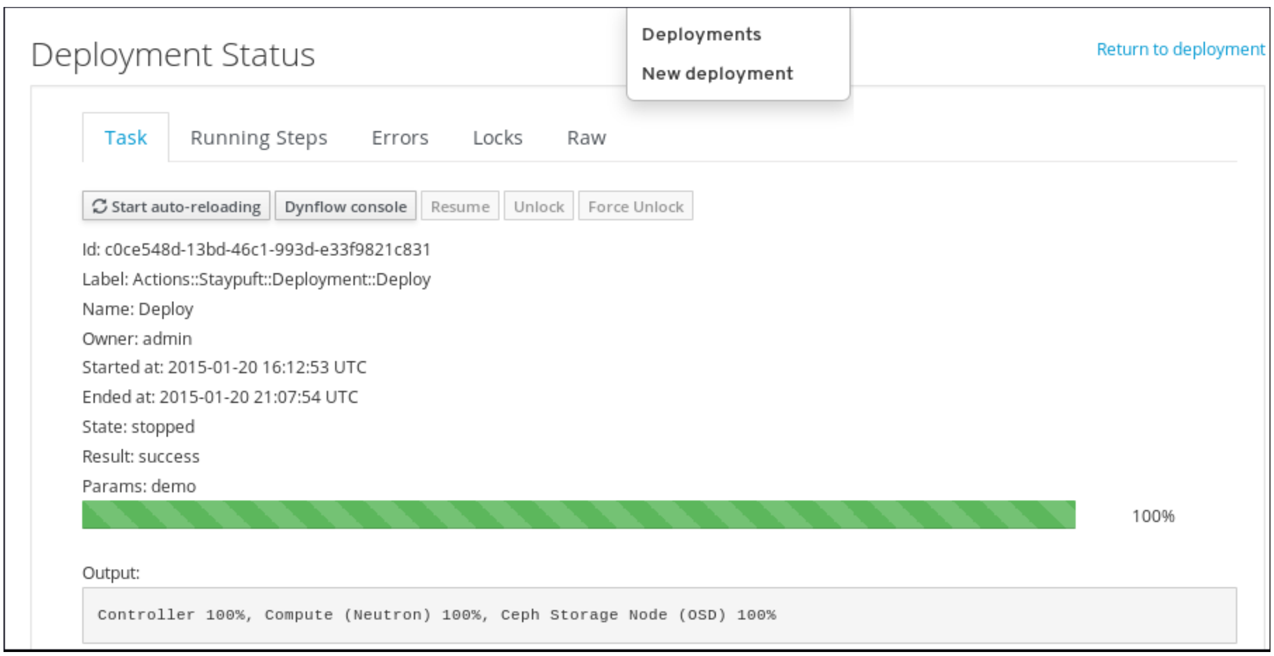


Figure 4.3 21: Deployment Status Bar

26. After successful deployment, Horizon URL is displayed on the deployment screen as

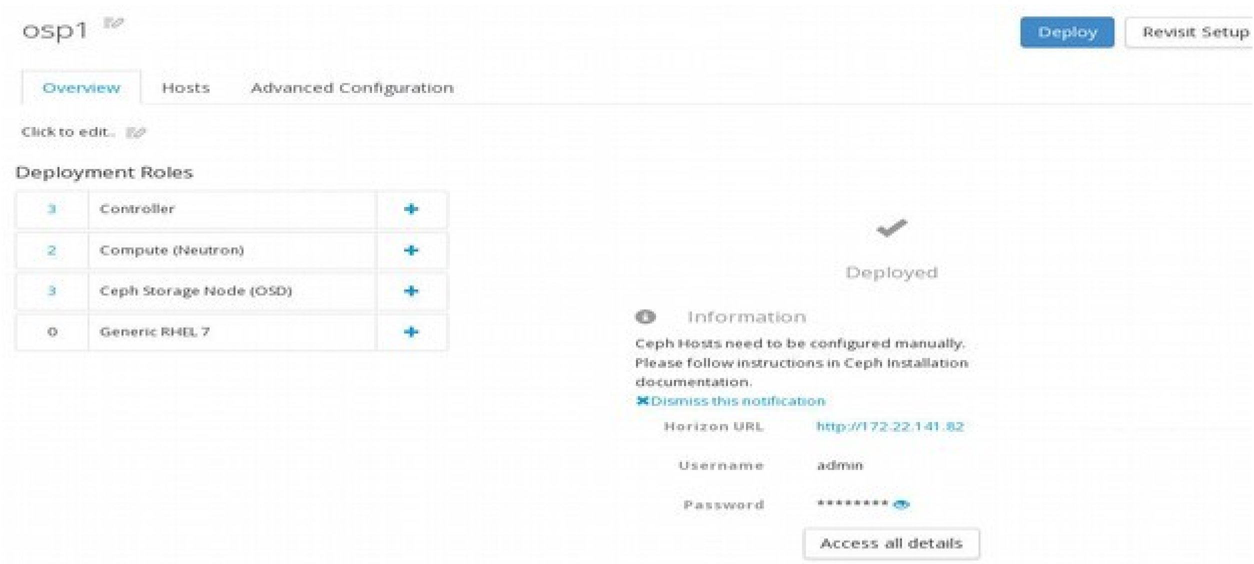


Figure 4.3 22: Successful Deployment

follows.

This completes the deployment of the OpenStack environment.

NOTE: For detailed information on RHEL OSP Installer procedure and capabilities, please refer to https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/6/html/Installer_and_Foreman_Guide/index.html

4.4 Troubleshooting a deployment failure:

- Host provisioning failure- Please refer to B.4Troubleshooting during provisioning
- Post provisioning failure- Please refer to B.5Troubleshooting post provisioning
- Rebuilding host from scratch B.6 Rebuilding host from scratch
- To restart the entire deployment all over again, please refer to B.7 Restart Deployment from scratch

4.5 Ceph storage setup and integration

4.5.1 Install and configure Ceph Admin server

The Ceph admin server, as the name suggests, administers and monitors the Ceph storage cluster. It runs Calamari, which is a management and monitoring system for the cluster. This server could be a virtual machine and is not resource intensive.

Fully details for installing, configuring and deploying Red Hat Ceph storage can be found here:

https://access.redhat.com/documentation/en-US/Red_Hat_Ceph_Storage/

1. Install RHEL 7.1 operating system
2. Configure NTP
3. Modify firewall rules

```
[root@osp-ice ~]# firewall-cmd --zone=public --add-port=80/tcp \
--permanent
[root@osp-ice ~]# firewall-cmd --zone=public --add-port=2003/tcp \
--permanent
[root@osp-ice ~]# firewall-cmd --zone=public --add-port=4505-4506/tcp \
--permanent
```

4. Update DNS resolution

   In this reference architecture, communication between the Ceph installer and OpenStack nodes happens via Provisioning network. For better manageability, DNS resolution needs to be updated to resolve the host names and IPs of the Provisioning network.

```
[root@osp-ice ~]# cat /etc/resolv.conf
domain example.com #Domain of the Ceph server
search osplocal.example.com example.com #OSP managed and local domains
nameserver 10.19.141.57 #IP address of the OSP installer
```

5. The Ceph administration command, ceph-deply, uses SSH to issue commands on managed hosts. Generate SSH Keys on the Ceph Admin server and copy it to the managed hosts

```
[root@osp-ice ~]# ssh-keygen
[root@osp-ice ~]# for HOST in osp-cont-{1,2,3} osp-osd-{1,2,3}; do
ssh-copy-id $HOST; done
```

4.5.2 Install Red Hat Ceph Storage 1.2.3

1. Download the installation ISO from the downloads section of http://access.redhat.com
2. Mount the ISO image

```
[root@osp-ice ~]# mkdir /mnt/iso
[root@ osp-ice ~]# mount -o loop /tmp/rhceph-1.2.3-rhel-7-x86_64.iso \
/mnt/iso
```

3. Copy Ceph package PEM files

```
[root@ osp-ice ~]# cp \
 /mnt/iso/RHCeph-Calamari-1.2-x86_64-c1e8ca3b6c57-285.pem \
 /etc/pki/product/285.pem
[root@osp-ice ~]# cp \
/mnt/iso/RHCeph-Installer-1.2-x86_64-8ad6befe003d-281.pem \
/etc/pki/product/281.pem
[root@osp-ice ~]# cp \
/mnt/iso/RHCeph-MON-1.2-x86_64-d8afd76a547b-286.pem \
/etc/pki/product/286.pem
```

```
[root@osp-ice ~]# cp \
/mnt/iso/RHCeph-OSD-1.2-x86_64-25019bf09fe9-288.pem \
/etc/pki/product/288.pem
```

4. Install the Ceph install package

```
[root@osp-ice ~]# yum -y install \
 /mnt/iso/ice_setup-0.2.2-1.el7cp.noarch.rpm
```

5. The ICE 1.2.2 must be downloaded or copied to the ice-1.2.2 directory on the Ceph Admin server (ICE Admin host).

```
[root@osp-ice ~]# mkdir cluster
[root@osp-ice ~]# cd cluster
[root@osp-ice cluster]# ice_setup.py -d /mnt/iso
-> press enter to install
-> press enter to accept the location of the install packages
-> press enter to accept hostname
-> press enter to accept http
```

4.5.2.1 Initialize Calamari

The calamari application must to be configured and initialized on the ICE Admin host

```
[root@osp-ice cluster]# calamari-ctl initialize
-> press enter to accept root as username
-> enter an email address press enter to enter a null email address
-> type in password
-> retype in password
```

Alternatively, the calamari application can be initialized from the command line.

```
[root@osp-ice cluster]# calamari-ctl initialize --admin-username root
--admin-password redhat11 --admin-email xxx@example.com
```

4.5.2.2 Configure Ceph cluster and deploy monitors

RHEL OSP Installer deploys the basic configuration files necessary for a Ceph cluster. However, additional commands are required to complete the Ceph cluster deployment.

The following commands pull the RHEL OSP Installer deployed ceph.conf file from one of the Ceph OSDs and uses it to complete the installation process on the Ceph monitors. At the end of this process, the monitors end up in a quorum state and the necessary keyfiles get passed onto Ceph Admin server. The ceph.conf file must be pulled from one of the OSD hosts due to puppet not deploying the same file to all of the nodes. BZ1184256 has been opened to track this issue.

```
[root@osp-ice cluster]# ceph-deploy config pull osp-osd-1
[root@osp-ice cluster]# for HOST in osp-cont-{1,2,3} ; do ceph-deploy
install $HOST ; done
[root@cephadm cluster]# ceph-deploy --overwrite-conf mon create-initial
```

The cluster status can be monitored from a Ceph monitor with the command

```
[root@osp-cont-1 ~]# ceph mon_status
```

4.5.2.3 Configure Ceph OSD

1. Install Ceph onto the OSDs.

```
[root@osp-ice cluster]#for HOST in osp-osd-{1,2,3} ; do ceph-deploy install
$HOST ; done
```

2. Gather the disk list for each OSD server. (This step must be repeated for each OSD server)

```
[root@osp-ice cluster]# ceph-deploy disk list osp-osd-1
```

The table lists out the storage and corresponding journal disks used in this reference architecture.

Storage Disk	Journal Disk
/dev/sdb, /dev/sdc, /dev/sdd, /dev/sde	/dev/sdn
/dev/sdf, /dev/sdg, /dev/sdh, /dev/sdi	/dev/sdo
/dev/sdj, /dev/sdk, /dev/sd', /dev/sdm	/dev/sdp

Table 4.5.2.1: Ceph OSD Disks

3. Initialize the disks. This step will delete any existing partitions on the disks.

```
[root@osp-ice cluster]# for HOST in osp-osd-{1,2,3}
                       > do
                       > for DISK in sd{b,c,d,e,f,g,h,i,j,k,l,m,n,o,p}
                             > do
                             > ceph-deploy disk zap $HOST:/dev/$DISK
                             > done
                       > done
```

4. Add the OSDs to the Ceph cluster.

Once the disks have been initialized, they must be prepared for use as an OSD , and activated. The "activate" command causes the OSD to come up and be placed in the cluster. It is possible to prepare OSDs, deploy them to the OSD nodes and activate them in one step with the "create" command.

This command must be repeated for each OSD by associating each storage disk with its respective journal disk. In this reference architecture four storage disks are assigned to one SSD journal disk as described in table Table 4.5.2.1: Ceph OSD Disks. This must be repeated on each OSD node.

```
[root@osp-ice cluster]# ceph-deploy --overwrite-conf osd create
osp-osd-1:/dev/sdb:/dev/sdn
[root@osp-ice cluster]# ceph-deploy --overwrite-conf osd create
osp-osd-1:/dev/sdc:/dev/sdn
.
.
.
.
[root@osp-ice cluster]# ceph-deploy –-overwrite-conf osd create
osp-osd-1:/dev/sdm:/dev/sdp
```

NOTE: The "--overwrite-conf" command line argument must to be used for each execution due to puppet periodically updating the deployed configuration file. Updating the configuration file will result in the remaining commands erroring out.

5. Monitoring the OSDs.

   The OSD creation process can be monitored on the controller nodes

```
[root@osp-cont-1 ~]# watch -d "ceph osd tree| tail -20 "
```

6. Verify the OSDs.

```
[root@osp-ice cluster]# ceph-deploy disk list osd1
                          .
                          .
                          .
[osp-osd-1][DEBUG ] /dev/sdn :
[osp-osd-1][DEBUG ]  /dev/sdn1 ceph journal, for /dev/sdb1
[osp-osd-1][DEBUG ]  /dev/sdn2 ceph journal, for /dev/sdc1
[osp-osd-1][DEBUG ]  /dev/sdn3 ceph journal, for /dev/sdd1
[osp-osd-1][DEBUG ]  /dev/sdn4 ceph journal, for /dev/sde1
[osp-osd-1][DEBUG ] /dev/sdo :
[osp-osd-1][DEBUG ]  /dev/sdo1 ceph journal, for /dev/sdf1
[osp-osd-1][DEBUG ]  /dev/sdo2 ceph journal, for /dev/sdg1
[osp-osd-1][DEBUG ]  /dev/sdo3 ceph journal, for /dev/sdh1
[osp-osd-1][DEBUG ]  /dev/sdo4 ceph journal, for /dev/sdi1
[osp-osd-1][DEBUG ] /dev/sdp :
[osp-osd-1][DEBUG ]  /dev/sdp1 ceph journal, for /dev/sdj1
[osp-osd-1][DEBUG ]  /dev/sdp2 ceph journal, for /dev/sdk1
[osp-osd-1][DEBUG ]  /dev/sdp3 ceph journal, for /dev/sdl1
[osp-osd-1][DEBUG ]  /dev/sdp4 ceph journal, for /dev/sdm1
```

4.5.2.4 Add Hosts to Calamari

Add controller and OSD nodes to calamari. Repeat this command for each host.

```
[root@osp-ice cluster]# for HOST in osp-osd-{1,2,3} osp-cont-{1,2,3}; do
ceph-deploy calamari connect $HOST; done
```

Log into calamari, accept all new hosts and verify status of cluster.

4.5.2.5 Storage Pool Creation

On a controller, create new pools.

```
[root@osp-cont-1 ~]# ceph osd pool create volumes 2048
[root@osp-cont-1 ~]# ceph osd pool create images 2048
```

4.5.2.6 Ceph Controller Integration

1. In a controller, import the RHEL OSP Installer images and volumes key rings.

```
[root@osp-cont-1 ~]# ceph auth import -i \
/etc/ceph/ceph.client.images.keyring
[root@osp-cont-1 ~]# ceph auth import -i \
/etc/ceph/ceph.client.volumes.keyring
```

2. On each controller, restart cinder.

```
[root@osp-cont-1 ~]# systemctl restart openstack-cinder-volume
```

4.5.2.7 Glance Configuration

After the Ceph cluster is built, glance support is installed into the OSP cluster. This is achieved by updating the parameter listed in Table 4.5.2.2: Glance Parameter. Once the configuration is saved, puppet will install glance and configure it to use Ceph as the storage backend.

Service	Key	Value
Controller	Include glance	true

Table 4.5.2.2: Glance Parameter

4.6 Post Installation verification

The status of the OpenStack cluster and managed services needs to be verified once deployment is complete. Pacemaker and OpenStack managed services need to be reviewed and checked for any failures.

4.6.1 Galera Status

Refer to C.3 List of Galera Cluster Status Variables for information on how to display the galera configuration settings.

> **NOTE**: It is important that Galera nodes are brought back up in the right order if all the nodes are down. The node that went down last in the cluster must be the first to be brought back up. Please refer to C.2 Bootstrapping Galera cluster. RHELOSP 6 runs a PCS agent that handles this automatically.

4.6.2 Pacemaker Cluster (PCS) Status

The condition of the cluster can be verified by the status command option. This displays the condition of each PCS resource and the cluster node it is running on. This is a good indicator to confirm if a resource or service is active-active or active-passive. Truncated output is provided below. See A.4 PCS Full Status for the complete output.

```
[root@osp-cont-1 ~]# pcs status
Cluster name: openstack
Last updated: Sun May 10 23:25:04 2015
Last change: Thu May  7 15:41:40 2015
Stack: corosync
Current DC: pcmk-osp-cont-3 (3) - partition with quorum
Version: 1.1.12-a14efad
3 Nodes configured
132 Resources configured


Online: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]

Full list of resources:

 ip-ceilometer-pub-172.22.141.60     (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-2
 ip-neutron-pub-172.22.141.91 (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
.
.
.
 ip-glance-pub-172.22.141.67  (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-glance-prv-172.22.141.66  (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-glance-adm-172.22.141.65  (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 Clone Set: glance-registry-clone [glance-registry]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: glance-api-clone [glance-api]
```

```
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]

Failed actions:
    openstack-heat-engine_monitor_60000 on pcmk-osp-cont-3 'not running' (7): call=610, status=complete, exit-reason='none', last-rc-change='Sun May 10 00:23:53 2015', queued=0ms, exec=0ms
    neutron-server_monitor_60000 on pcmk-os`p-cont-3 'not running' (7): call=563, status=complete, exit-reason='none', last-rc-change='Sun May 10 00:23:42 2015', queued=0ms, exec=0ms


PCSD Status:
  pcmk-osp-cont-1: Online
  pcmk-osp-cont-2: Online
  pcmk-osp-cont-3: Online

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

4.6.3 Cleaning up PCS resources

The failed PCS resources can be started or restarted with the cleanup command. The above output shows the failed resources:

```
Failed actions:
    openstack-heat-engine_monitor_60000 on pcmk-osp-cont-3 'not running' (7): call=610, status=complete, exit-reason='none', last-rc-change='Sun May 10 00:23:53 2015', queued=0ms, exec=0ms
    neutron-server_monitor_60000 on pcmk-osp-cont-3 'not running' (7): call=563, status=complete, exit-reason='none', last-rc-change='Sun May 10 00:23:42 2015', queued=0ms, exec=0ms
```

These can be restarted as follows:

```
[root@osp-cont-1 ~]# pcs resource cleanup openstack-heat-engine
Resource: openstack-heat-engine successfully cleaned up
[root@osp-cont-1 ~]# pcs resource cleanup neutron-server
Resource: neutron-server successfully cleaned up
```

This removes the failed actions.

4.6.4 PCS resources availability

The following table describes the resources that are active-active (running on all three pacemaker) controller nodes. Please refer to Figure 2.1.2 1: High Availability By Pacemaker that provides a high level view of service and resource availability among controller nodes.

Resource	Osp-cont-1	Osp-cont-2	Osp-cont-3	Active-Active	Active-Passive	Notes
ceilometer-delay	*	*	*	√		
cinder-api	*	*	*	√		
cinder-scheduler	*	*	*	√		
cinder-volume	*				√	
stonith-ipmilan	*	*	*		√	Note 1
galera	*	*	*	√		
haproxy	*	*	*	√		
heat-api	*	*	*	√		
heat-api-cfn	*	*	*	√		
heat-api-cloudwatch	*	*	*	√		
horizon	*	*	*	√		
ip-amqp-pub			*		√	
ip-ceilometer-adm		*			√	
ip-ceilometer-prv		*			√	
ip-ceilometer-pub		*			√	
ip-cinder-adm			*		√	
ip-cinder-prv			*		√	
ip-cinder-pub			*		√	
ip-galera-pub			*		√	Note 2
ip-glance-adm			*		√	
ip-glance-prv			*		√	
ip-glance-pub			*		√	
ip-heat_cfn-adm		*			√	
ip-heat_cfn-prv		*			√	
ip-heat_cfn-pub		*			√	
ip-heat-adm		*			√	
ip-heat-prv		*			√	
ip-heat-pub		*			√	
ip-horizon-adm			*		√	
ip-horizon-prv			*		√	
ip-horizon-pub			*		√	
ip-keystone-adm			*		√	
ip-keystone-prv			*		√	
ip-keystone-pub			*		√	
ip-loadbalancer-pub		*	*	*	√	
ip-neutron-adm			*		√	

ip-neutron-pub			*		√	
ip-nova-adm		*			√	
ip-nova-prv		*			√	
ip-nova-pub		*			√	
keystone	*	*	*	√		
memcached	*	*	*	√		
mongod	*	*	*	√		
neutron-dhcp-agent		*			√	
neutron-l3-agent	*	*	*	√		
neutron-metadata-agent		*			√	
neutron-netns-cleanup		*			√	
neutron-openvswitch-agent		*			√	
neutron-ovs-cleanup		*			√	
ip-neutron-prv			*		√	
neutron-scale		*			√	Note 3
neutron-server	*	*	*	√		
openstack-ceilometer-alarm-evaluator	*	*	*	√		
openstack-ceilometer-alarm-notifier	*	*	*	√		
openstack-ceilometer-api	*	*	*	√		
openstack-ceilometer-central	*				√	
openstack-ceilometer-collector	*	*	*	√		
openstack-ceilometer-notification	*	*	*	√		
openstack-heat-engine			*		√	
openstack-nova-api	*	*	*	√		
openstack-nova-conductor	*	*	*	√		
openstack-nova-consoleauth	*	*	*	√		
openstack-nova-novncproxy	*	*	*	√		
openstack-nova-scheduler	*	*	*	√		
rabbitmq-server	*	*	*	√		

Table 4.6.4.1: Resource Availability

Note 1 - Stonith resource is unique to each system. Hence three resources run simultaneously in Active-Passive mode, one per each system.

Note 2 - Galera is a multi master cluster. The cluster nodes are supposed to be active-active with synchronous replication using wsrep provider.

Note 3 - Expected to be changed to A/A in future releases.

4.7 Pacemaker constraints

The services must be brought up in the right order due to dependencies. The following illustration describes which services/resources can come in parallel or in sequence.

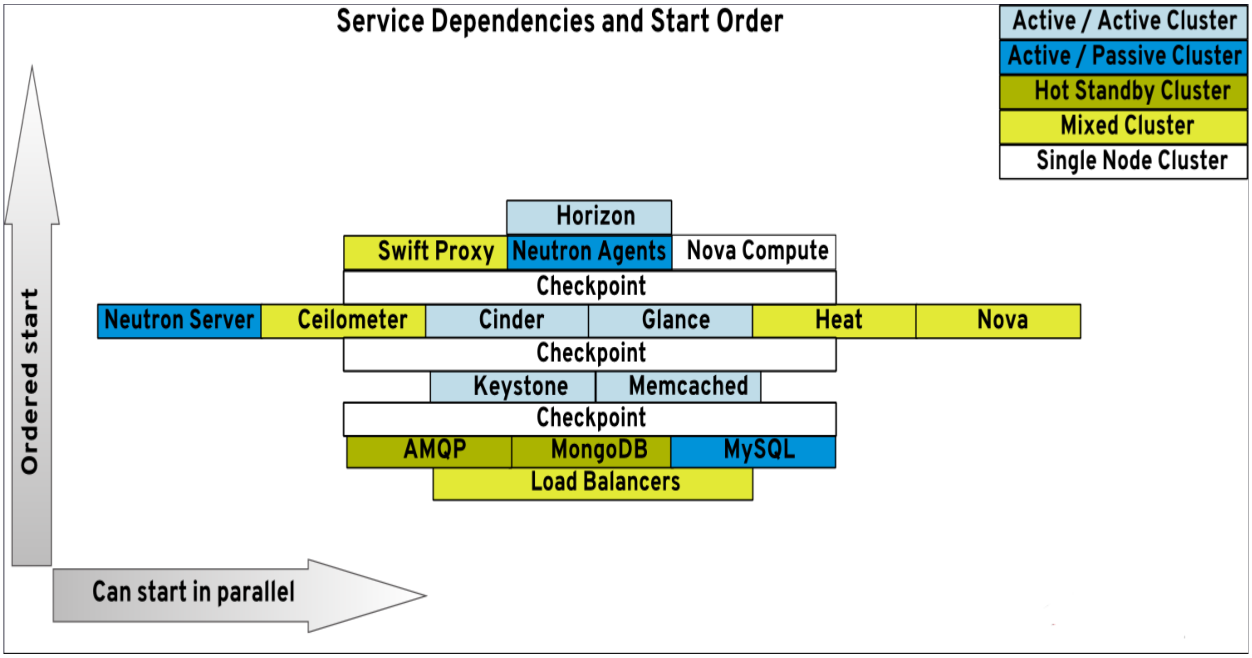


Figure 4.7 1: Service Dependencies and Start Order

This order can displayed by using Pacemaker constraints command option. The following A.5 PCS Constraint View describes the dependency/order to start services.

4.8 Fencing

Fencing protects the data from being corrupted by rogue nodes or concurrent access. Fencing is provided by '*stonithd*' daemon. **STONITH** is an acronym for Shoot-The-Other-Node-In-The-Head. In this reference architecture, IPMI interface of the hardware is used to take control of the node. Hence, they must be run for all the controller nodes.

It is important that a *stonithd* service for a node for ex: controller1, should run on controller 2 or controller 3. This is to ensure that the controller in question can be disabled by the other controllers where the service is running. If the service for controller 1 runs on itself, then it cannot effectively shut itself down when there is a problem. At times, one controller node runs stonith service for other two nodes. In this configuration, server osp-cont-1 runs stonith for other two nodes but service for itself, is running on node osp-cont-2.

```
[root@osp-cont-1 ~]# pcs status | grep stonith
 stonith-ipmilan-10.19.143.62     (stonith:fence_ipmilan):
 Started pcmk-osp-cont-3
```

```
stonith-ipmilan-10.19.143.63     (stonith:fence_ipmilan):
Started pcmk-osp-cont-1

stonith-ipmilan-10.19.143.61     (stonith:fence_ipmilan):
Started pcmk-osp-cont-3
```

4.8.1.1 Testing Fencing

Fencing can be tested by simulating a system failure on one of the nodes. The following command introduces an error where the system fails to respond to pacemaker:

```
[root@osp-cont-1 ~]# systemctl stop network.service
```

The node that runs stonith service for this node is osp-cont-3, which initiates a shutdown of node osp-cont-1 to disable any possible access by it.

```
Feb 12 00:07:34 [23844] osp-cont-3.osplocal.com        cib:     info:
crm_update_peer_proc:  pcmk_cpg_membership: Node pcmk-osp-cont-1[1] -
corosync-cpg is now offline

Feb 12 00:07:34 [23844] osp-cont-3.osplocal.com        cib:     info:
pcmk_cpg_membership:   Member[9.0] cib.2
Feb 12 00:07:34 [23849] osp-cont-3.osplocal.com       crmd:     info:
crm_update_peer_proc:  pcmk_cpg_membership: Node pcmk-osp-cont-1[1] -
corosync-cpg is now offline
Feb 12 00:07:34 [23844] osp-cont-3.osplocal.com        cib:     info:
pcmk_cpg_membership:   Member[9.1] cib.3
Feb 12 00:07:34 [23843] osp-cont-3.osplocal.com pacemakerd:     info:
crm_cs_flush:    Sent 0 CPG messages  (1 remaining, last=15): Try again (6)
Feb 12 00:07:34 [23849] osp-cont-3.osplocal.com       crmd:     info:
peer_update_callback:  Client pcmk-macc81f66653342/peer now has status
[offline]

                      ----abbreviated----


Feb 12 00:07:34 [23849] osp-cont-3.osplocal.com       crmd:   notice:
crm_update_peer_state:      pcmk_quorum_notification: Node pcmk-osp-cont-1[1]
- state is now lost (was member)

Feb 12 00:07:35 [23845] osp-cont-3.osplocal.com stonith-ng:   notice:
can_fence_host_with_device: stonith-ipmilan-10.19.143.61 can fence
pcmk-osp-cont-1: static-list

Feb 12 00:07:48 [23845] osp-cont-3.osplocal.com stonith-ng:     info:
stonith_action_create:      Initiating action monitor for agent fence_ipmilan
(target=(null))
Feb 12 00:07:48 [23845] osp-cont-3.osplocal.com stonith-ng:     info:
log_operation:   stonith-ipmilan-10.19.143.63:15835 [ Getting status of
IPMI:10.19.143.63...Done ]
 pcmk_dbus_get_property:    Calling: GetAll on org.freedesktop.systemd1
Feb 12 00:07:59 [23845] osp-cont-3.osplocal.com stonith-ng:   notice:
remote_op_done:  Operation reboot of pcmk-osp-cont-1 by pcmk-osp-cont-3 for
crmd.32191@pcmk-osp-cont-1.bb2d088e: OK
Feb 12 00:07:59 [23845] osp-cont-3.osplocal.com stonith-ng:     info:
```

```
stonith_command:      Processed st_notify reply from pcmk-osp-cont-1: OK (0)
Feb 12 00:07:59 [23849] osp-cont-1.osplocal.com       crmd:   notice:
tengine_stonith_notify:     Peer pcmk-osp-cont-1 was terminated (reboot) by
pcmk-osp-cont-3 for pcmk-osp-cont-3: OK
(ref=bb2d088e-88bb-49f8-8a23-7855da78c0e5) by client crmd.32191
```

This initiates a reboot of the node, thereby restoring services by systemd and OCF resource agents.

```
[root@osp-cont-1 ~]# pcs status | grep Online
Online: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 pcmk-osp-cont-1: Online
 pcmk-osp-cont-2: Online
 pcmk-osp-cont-3: Online
```

4.9 OpenStack

4.9.1 Controller Status

The following sections help in understanding the health of the OpenStack environment.

4.9.1.1 Source Administration Credentials

Many of the OpenStack commands require the passing of administrative credentials. Puppet deploys the /root/keystonerc_admin configuration file with the needed authentication parameters. This file needs to be sourced prior to executing any OpenStack commands.

```
[root@osp-cont-1 ~]# . ./keystonerc_admin
```

4.9.1.2 OpenStack Status

The openstack-status command is an all-in-one command able to display most information needed to monitor the status of the openstack

```
[root@osp-cont-1 ~(openstack_admin)]# openstack-status
== Nova services ==
openstack-nova-api:                     active    (disabled on boot)
openstack-nova-cert:                    active    (disabled on boot)
openstack-nova-compute:                 inactive  (disabled on boot)
openstack-nova-network:                 inactive  (disabled on boot)
openstack-nova-scheduler:               active    (disabled on boot)
openstack-nova-conductor:               active    (disabled on boot)
== Glance services ==
openstack-glance-api:                   active    (disabled on boot)
openstack-glance-registry:              active    (disabled on boot)
== Keystone service ==
openstack-keystone:                     active
== Horizon service ==
openstack-dashboard:                    uncontactable
== neutron services ==
neutron-server:                         active    (disabled on boot)
neutron-dhcp-agent:                     active    (disabled on boot)
neutron-l3-agent:                       active    (disabled on boot)
neutron-metadata-agent:                 active    (disabled on boot)
neutron-lbaas-agent:                    inactive  (disabled on boot)
neutron-openvswitch-agent:              active    (disabled on boot)
== Cinder services ==
openstack-cinder-api:                   active    (disabled on boot)
openstack-cinder-scheduler:             active    (disabled on boot)
openstack-cinder-volume:                inactive  (disabled on boot)
```

```
openstack-cinder-backup:                inactive  (disabled on boot)
== Ceilometer services ==
openstack-ceilometer-api:               active    (disabled on boot)
openstack-ceilometer-central:           inactive  (disabled on boot)
openstack-ceilometer-compute:           inactive  (disabled on boot)
openstack-ceilometer-collector:         active    (disabled on boot)
openstack-ceilometer-alarm-notifier:    active    (disabled on boot)
openstack-ceilometer-alarm-evaluator:   active    (disabled on boot)
openstack-ceilometer-notification:      active    (disabled on boot)
== Heat services ==
openstack-heat-api:                     active    (disabled on boot)
openstack-heat-api-cfn:                 active    (disabled on boot)
openstack-heat-api-cloudwatch:          active    (disabled on boot)
openstack-heat-engine:                  inactive  (disabled on boot)
== Support services ==
openvswitch:                            active
dbus:                                   active
rabbitmq-server:                        inactive  (disabled on boot)
memcached:                              active
== Keystone users ==
+----------------------------------+------------+---------+-------------------------------------+
|                id                |    name    | enabled |                email                |
+----------------------------------+------------+---------+-------------------------------------+
| ab09e9c0d2c74f21b0cf69d7efb0e8ac |   admin    |   True  | admin@example.com |
| 2b7396f6a12643f18ee9f0b330b8a264 | ceilometer |   True  |         ceilometer@localhost        |
| 86e03cfebe8948d483a99524a98fb2d3 |   cinder   |   True  |           cinder@localhost          |
| 2ca39534eb984f5a87dcedc845e35b00 |   glance   |   True  |           glance@localhost          |
| c29c4690d8ec404dafcbc2424c73bd45 |    heat    |   True  |            heat@localhost           |
| 1e0179ff1ffd479abc27bc6c4dd95cb5 |  neutron   |   True  |          neutron@localhost          |
| 249dde53757f4703ae1038892a1cf09d |    nova    |   True  |            nova@localhost           |
+----------------------------------+------------+---------+-------------------------------------+
== Glance images ==
+--------------------------------------+-------------+-------------+------------------+-----------+--
------+
| ID                                   | Name        | Disk Format | Container Format | Size      |
Status |
+--------------------------------------+-------------+-------------+------------------+-----------+--
------+
+--------------------------------------+-------------+-------------+------------------+-----------+--
------+
== Nova managed services ==
+----+------------------+------------------------------+----------+---------+-------+--------------
-------------+-----------------+
| Id | Binary           | Host                         | Zone     | Status  | State | Updated_at
| Disabled Reason |
+----+------------------+------------------------------+----------+---------+-------+--------------
-------------+-----------------+
| 2  | nova-cert        | osp-cont-1.osplocal.example.com | internal | enabled | up    |
2015-05-11T11:46:32.000000 | -               |
| 5  | nova-consoleauth | osp-cont-1.osplocal.example.com | internal | enabled | up    |
2015-05-11T11:46:29.000000 | -               |
| 8  | nova-scheduler   | osp-cont-1.osplocal.example.com | internal | enabled | up    |
2015-05-11T11:46:34.000000 | -               |
| 11 | nova-conductor   | osp-cont-1.osplocal.example.com | internal | enabled | up    |
2015-05-11T11:46:34.000000 | -               |
| 23 | nova-cert        | osp-cont-2.osplocal.example.com | internal | enabled | up    |
2015-05-11T11:46:33.000000 | -               |
| 26 | nova-cert        | osp-cont-3.osplocal.example.com | internal | enabled | up    |
2015-05-11T11:46:35.000000 | -               |
| 29 | nova-consoleauth | osp-cont-2.osplocal.example.com | internal | enabled | up    |
2015-05-11T11:46:28.000000 | -               |
| 32 | nova-consoleauth | osp-cont-3.osplocal.example.com | internal | enabled | up    |
2015-05-11T11:46:29.000000 | -               |
| 35 | nova-scheduler   | osp-cont-2.osplocal.example.com | internal | enabled | up    |
2015-05-11T11:46:32.000000 | -               |
| 38 | nova-scheduler   | osp-cont-3.osplocal.example.com | internal | enabled | up    |
2015-05-11T11:46:31.000000 | -               |
| 41 | nova-conductor   | osp-cont-2.osplocal.example.com | internal | enabled | up    |
2015-05-11T11:46:32.000000 | -               |
| 53 | nova-conductor   | osp-cont-3.osplocal.example.com | internal | enabled | up    |
2015-05-11T11:46:32.000000 | -               |
| 65 | nova-compute     | osp-comp-2.osplocal.example.com | nova     | enabled | up    |
```

```
2015-05-11T11:46:35.000000 | -                |
| 68 | nova-compute     | osp-comp-1.osplocal.example.com | nova     | enabled | up    |
2015-05-11T11:46:32.000000 | -                |
+----+-----------------+---------------------------------+----------+---------+-------+-------------
--------------+-----------------+
== Nova networks ==
+--------------------------------------+-----------------------------------------------------+------+
| ID                                   | Label                                               | Cidr |
+--------------------------------------+-----------------------------------------------------+------+
+--------------------------------------+-----------------------------------------------------+------+
== Nova instance flavors ==
+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+
| ID | Name      | Memory_MB | Disk | Ephemeral | Swap | VCPUs | RXTX_Factor | Is_Public |
+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+
| 1  | m1.tiny   | 512       | 1    | 0         |      | 1     | 1.0         | True      |
| 2  | m1.small  | 2048      | 20   | 0         |      | 1     | 1.0         | True      |
| 3  | m1.medium | 4096      | 40   | 0         |      | 2     | 1.0         | True      |
| 4  | m1.large  | 8192      | 80   | 0         |      | 4     | 1.0         | True      |
| 5  | m1.xlarge | 16384     | 160  | 0         |      | 8     | 1.0         | True      |
+----+-----------+-----------+------+-----------+------+-------+-------------+-----------+
== Nova instances ==
+----+------+--------+------------+-------------+----------+
| ID | Name | Status | Task State | Power State | Networks |
+----+------+--------+------------+-------------+----------+
+----+------+--------+------------+-------------+----------+
```

5 Test Solution - Deploy a Multi-tier Web Application

This section describes the steps taken to validate basic OpenStack functionality of this reference architecture. The test case includes:

- Commands to verify the initial configuration
- Adding an image to Glance
- Creating a Keystone tenant and user
- Defining a network that includes public and private switches and a router
- Adding a key pair
- Deploying 2 virtual machines
- Verifying the virtual machines can communicate over the private network
- Creating and attaching a Cinder volume
- Adding a floating IP address to a virtual machine and accessing it from the public network
- Log in to the Virtual instances, creating filesystems using the attached cinder volumes, deploy a web application on the web tier and a database on the database tier
- Demonstrate application functionality

5.1 Overview

This section of the reference architecture describes how to install a multi-tier web application consisting of a web server and a database server. The example uses a simple WordPress deployment. The web server is accessible via a public floating IP address on the client network. It connects to a remote MySQL database server via the tenant network. The database server stores its database on a persistent volume provided by the Ceph backend Storage service.

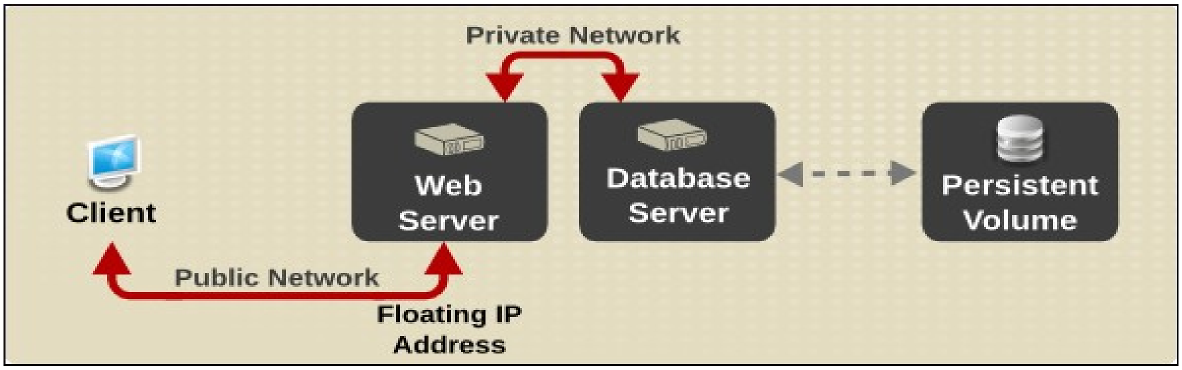


Figure 5.1 1: Multi-tier Application

The process for deploying a multi-tier application demonstrates OpenStack's complete functionality as described in this reference architecture. It is a realistic use case that touches every service discussed in the preceding pages. Multi-tier web applications are often deployed in IaaS clouds so they can be scaled as required. Keeping the database server off a publicly accessible network provides an extra layer of security. Figure 5.1 1: Multi-tier Application depicts the end state of the multi-tier application deployment.

5.1.1 Prepare the Test Environment

This section describes steps to create a realistic test environment. Execute the commands in this section as the OpenStack administrator. Execute the commands in subsequent sections as a Member user. This section will create the public network, create a tenant user, create a private network for the tenant and connect the tenant network to the public network via a router.

1. Verify the contents of */root/keystone_admin* created during deployment.

```
[root@osp-cont-1 ~]# cat /root/keystonerc_admin
export OS_USERNAME=admin
export OS_TENANT_NAME=admin
export OS_PASSWORD=redhat11
export OS_AUTH_URL=http://172.22.141.85:35357/v2.0/
export PS1='[\u@\h \W(openstack_admin)]\$ '
Source /etc/keystonerc_admin to execute commands as the OpenStack
administrator.
[root@osp-cont-1 ~(openstack-admin)]# source /root/keystonerc_admin
```

2. Create a provider network. The external network "public" is mapped to the *br-ex* bridge.

```
 [root@osp-cont-1 ~]# neutron net-create public \
--provider:network_type flat –provider:physical_network \
 physnet-external --router:external=True --shared
```

```
Created a new network:
+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | True                                 |
| id                        | 22be3c03-33b8-4015-a62e-c532d1d3df3d |
| name                      | public                               |
| provider:network_type     | flat                                 |
| provider:physical_network | physnet-external                     |
| provider:segmentation_id  |                                      |
| router:external           | True                                 |
| shared                    | True                                 |
| status                    | ACTIVE                               |
| subnets                   |                                      |
| tenant_id                 | 28235d54c43a489b881055efd8ff2f7b     |
+---------------------------+--------------------------------------+
```

3. Create a subnet named *public*.

```
[root@osp-cont-1 ~(openstack_admin)]# neutron subnet-create \
--name public --gateway 10.19.143.254 --enable_dhcp=False \
--allocation-pool start=10.19.139.209,end=10.19.139.214 \
public 10.19.139.0/21
Created a new subnet:
+-------------------+----------------------------------------------------+
| Field             | Value                                              |
+-------------------+----------------------------------------------------+
| allocation_pools  | {"start": "10.19.139.209", "end": "10.19.139.214"} |
| cidr              | 10.19.136.0/21                                     |
| dns_nameservers   |                                                    |
| enable_dhcp       | False                                              |
| gateway_ip        | 10.19.143.254                                      |
| host_routes       |                                                    |
| id                | f11d1b49-74c8-441d-9505-f6857b9bc610               |
| ip_version        | 4                                                  |
| ipv6_address_mode |                                                    |
| ipv6_ra_mode      |                                                    |
| name              | public                                             |
| network_id        | 22be3c03-33b8-4015-a62e-c532d1d3df3d               |
| tenant_id         | 28235d54c43a489b881055efd8ff2f7b                   |
+-------------------+----------------------------------------------------+
```

4. Import the virtual machine disk image.

```
[root@osp-cont-1 ~(openstack_admin)]# glance image-create \
--name rhel70unlock --is-public true --disk-format qcow2 \
--container-format bare --file /tmp/rhel70_unlock.qcow2
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | 310d10275e6325b136685d536dc2f7d6     |
| container_format | bare                                 |
| created_at       | 2015-02-04T16:58:08                  |
| deleted          | False                                |
| deleted_at       | None                                 |
| disk_format      | qcow2                                |
| id               | 13b9561e-d86c-49e8-a59c-0507feeaac9c |
```

```
| is_public        | True                                 |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | rhel70unlock                         |
| owner            | 651f07161db649b5923cf7b64cf2ebe7     |
| protected        | False                                |
| size             | 437977088                            |
| status           | active                               |
| updated_at       | 2015-02-04T16:58:19                  |
| virtual_size     | None                                 |
+------------------+--------------------------------------+
```

NOTE: For qcow2 image details, please refer to C.4 Downloading the RHEL Guest (qcow2) Image and C.5 Editing RHEL Guest (qcow2) Image .

5. Verify available glance images.

```
[root@osp-cont-1 ~(openstack_admin)]# glance image-list
+--------------------------------------+--------------+-------------+-------
-----------+-----------+--------+
| ID                                   | Name         | Disk Format |
Container Format | Size      | Status |
+--------------------------------------+--------------+-------------+-------
-----------+-----------+--------+
| d5ed3662-4421-40a8-9ee2-66be7cc31d7f | rhel65unlock | qcow2       | bare
| 308609024 | active |
| 13b9561e-d86c-49e8-a59c-0507feeaac9c | rhel70unlock | qcow2       | bare
| 437977088 | active |
+--------------------------------------+--------------+-------------+-------
-----------+-----------+--------+
+----------+----------------------------------+
|  email   |                                  |
```

6. Create a tenant and a tenant user. In this example the user is named *finance01*. The tenant is named *finance-tenant*.

```
[root@osp-cont-1 ~(openstack_admin)]# keystone user-create --name finance01
--pass finance01
+----------+----------------------------------+
| Property |              Value               |
| enabled  |               True               |
|    id    | 3d7235dd2ff24ed39506599446125a9d |
|   name   |            finance01             |
| username |            finance01             |
+----------+----------------------------------+
[root@osp-cont-1 ~(openstack_admin)]# keystone tenant-create --name
finance-tenant
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |                                  |
|   enabled   |               True               |
|      id     | 9ce5392d728a4e4086b6ed55a8a456a0 |
|     name    |          finance-tenant          |
+-------------+----------------------------------+
```

7. Add the Member role to the tenant user.

```
[root@osp-cont-1 ~(openstack_admin)]# keystone user-role-add –user-id \
 finance01 --tenant-id finance-tenant --role-id _member_
[root@osp-cont-1 ~(openstack_admin)]# keystone user-role-list –user-id \
finance01 --tenant-id finance-tenant
+----------------------------------+----------+-----------------------------------
----+----------------------------------+
|                id                |   name   |              user_id
|            tenant_id             |
+----------------------------------+----------+-----------------------------------
-----+----------------------------------+
| 9fe2ff9ee4384b1894a90878d3e92bab | _member_ |
3d7235dd2ff24ed39506599446125a9d | 9ce5392d728a4e4086b6ed55a8a456a0 |
+----------------------------------+----------+-----------------------------------
-----+----------------------------------+
-+----------------------------------+----------------------------------+
```

8. Create a file named */root/keystonerc_finance01* that contains environment variables for the tenant user.

```
[root@osp-cont-1 ~(openstack_admin)]# cat keystonerc_finance
export OS_USERNAME=finance01
export OS_TENANT_NAME=finance-tenant
export OS_PASSWORD=finance01
export OS_AUTH_URL=http://172.22.141.85:35357/v2.0/
export PS1='[\u@\h \W(openstack_finance)]\$ '
```

5.1.2 Create the Tenant Network

Execute the commands in this section as the tenant user. These commands create the tenant network, subnet, and router. They also add rules to the tenant security group.

1. Source */root/keystonerc_finance01* to execute commands as the tenant user.

```
[root@osp-cont-1 ~(openstack_admin)]# source /root/keystonerc_finance
```

2. List the *default* security group for finance01 tenant. The id of the tenant network is used for subsequent commands.

```
[root@osp-cont-1 ~(finance)]# neutron security-group-list
+--------------------------------------+---------+-------------+
| id                                   | name    | description |
+--------------------------------------+---------+-------------+
| b64c8041-a8a4-4c54-9773-9e3665762d3d | default | default     |
+--------------------------------------+---------+-------------+
```

3. Create a new security group rule for ICMP.

```
[root@osp-cont-1 ~(finance)]# neutron \
security-group-rule-create --direction ingress --protocol icmp \
b64c8041-a8a4-4c54-9773-9e3665762d3d
Created a new security_group_rule:
+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| direction         | ingress                              |
| ethertype         | IPv4                                 |
| id                | 544ca3cd-91a1-4798-bb2a-a96b25829709 |
| port_range_max    |                                      |
| port_range_min    |                                      |
```

```
| protocol          | icmp                                 |
| remote_group_id   |                                      |
| remote_ip_prefix  |                                      |
| security_group_id | b64c8041-a8a4-4c54-9773-9e3665762d3d |
| tenant_id         | 9ce5392d728a4e4086b6ed55a8a456a0     |
+-------------------+--------------------------------------+
```

4. Create a new security group rule for SSH.

```
[root@osp-cont-1 ~(finance)]# neutron \
security-group-rule-create --direction ingress --protocol tcp \
--port_range_min 22 --port_range_max 22 \
b64c8041-a8a4-4c54-9773-9e3665762d3d
Created a new security_group_rule:
+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| direction         | ingress                              |
| ethertype         | IPv4                                 |
| id                | d7f1a343-0bf1-499b-b22d-06fce2638405 |
| port_range_max    | 22                                   |
| port_range_min    | 22                                   |
| protocol          | tcp                                  |
| remote_group_id   |                                      |
| remote_ip_prefix  |                                      |
| security_group_id | b64c8041-a8a4-4c54-9773-9e3665762d3d |
| tenant_id         | 9ce5392d728a4e4086b6ed55a8a456a0     |
+-------------------+--------------------------------------+
```

5. Create a new security group rule for HTTP.

```
[root@osp-cont-1 ~(finance)]# neutron \
security-group-rule-create --direction ingress --protocol tcp \
--port_range_min 80 --port_range_max 80 \
b64c8041-a8a4-4c54-9773-9e3665762d3d
Created a new security_group_rule:
+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| direction         | ingress                              |
| ethertype         | IPv4                                 |
| id                | 7d2d2c6c-a45c-490b-9c15-ba473a3f1a9f |
| port_range_max    | 80                                   |
| port_range_min    | 80                                   |
| protocol          | tcp                                  |
| remote_group_id   |                                      |
| remote_ip_prefix  |                                      |
| security_group_id | b64c8041-a8a4-4c54-9773-9e3665762d3d |
| tenant_id         | 9ce5392d728a4e4086b6ed55a8a456a0     |
```

6. Display security group rules.

   - The truncated output of the security group settings are displayed

```
[root@osp-cont-1 ~(finance)]# neutron security-group-show
b64c8041-a8a4-4c54-9773-9e3665762d3d
+----------------------+-----------------------------------------------------
------------------
```

```
| Field                 | Value
+-----------------------+-----------------------------------------------------
------------------
| description           | default
| id                    | b64c8041-a8a4-4c54-9773-9e3665762d3d
| name                  | default
|                       | {"remote_group_id": null, "direction": "ingress",
"remote_ip_prefix": null, "protocol": "icmp", "tenant_id":
"9ce5392d728a4e4086b6ed55a8a456a0", "port_range_max":  null, ...
|                       | {"remote_group_id": null, "direction": "ingress",
"remote_ip_prefix": null, "protocol": "tcp", "tenant_id":
"9ce5392d728a4e4086b6ed55a8a456a0", "port_range_max": .. 80,
|                       | {"remote_group_id": null, "direction": "ingress",
"remote_ip_prefix": null, "protocol": "tcp", "tenant_id":
"9ce5392d728a4e4086b6ed55a8a456a0", "port_range_max": 22, ....
| tenant_id             | 28235d54c43a489b881055efd8ff2f7b
+-----------------------+-----------------------------------------------------
------------------
```

NOTE: Security groups and security group rules allow administrators and tenants the ability to specify the type of traffic and direction (ingress/egress) that is allowed to pass through a virtual interface port. A security group is a container for security group rules. When a virtual interface port is created it is associated with a security group. If a security group is not specified, the port gets associated with a 'default' security group. By default this group drops all ingress traffic and allow all egress traffic. This behavior can be modified.

7. Create a tenant network.

```
[root@osp-cont-1 ~(finance)]# neutron net-create priv
Created a new network:
+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | True                                 |
| id                        | 54d1143c-e5ef-4d62-a0b2-3aa7cd3eaabf |
| name                      | priv                                 |
| provider:network_type     | vxlan                                |
| provider:physical_network |                                      |
| provider:segmentation_id  | 10                                   |
| router:external           | False                                |
| shared                    | False                                |
| status                    | ACTIVE                               |
| subnets                   |                                      |
| tenant_id                 | 9ce5392d728a4e4086b6ed55a8a456a0     |
+---------------------------+--------------------------------------+
```

8. Create a subnet called privsub.

```
[root@osp-cont-1 ~(finance)]# neutron subnet-create --name privsub priv
10.0.100.0/21 --dns_nameservers list=true 10.19.143.247
Created a new subnet:
+-------------------+------------------------------------------------+
| Field             | Value                                          |
+-------------------+------------------------------------------------+
| allocation_pools  | {"start": "10.0.96.2", "end": "10.0.103.254"} |
| cidr              | 10.0.96.0/21                                   |
| dns_nameservers   | 10.19.143.247                                  |
```

```
| enable_dhcp        | True                                         |
| gateway_ip         | 10.0.96.1                                    |
| host_routes        |                                              |
| id                 | da815e8d-ad2d-42ae-8a20-8f7f2350e391         |
| ip_version         | 4                                            |
| ipv6_address_mode  |                                              |
| ipv6_ra_mode       |                                              |
| name               | privsub                                      |
| network_id         | 54d1143c-e5ef-4d62-a0b2-3aa7cd3eaabf         |
| tenant_id          | 9ce5392d728a4e4086b6ed55a8a456a0             |
+--------------------+----------------------------------------------+
```

9. List the networks.

```
[root@osp-cont-1 ~(finance)]# neutron net-list
+--------------------------------------+--------+--------------------------
--------------------------+
| id                                   | name   | subnets
|
+--------------------------------------+--------+--------------------------
--------------------------+
| 54d1143c-e5ef-4d62-a0b2-3aa7cd3eaabf | priv   |
da815e8d-ad2d-42ae-8a20-8f7f2350e391 10.0.96.0/21   |
+--------------------------------------+--------+--------------------------
--------------------------+
[root@osp-cont-1 ~(finance)]# neutron subnet-list
+--------------------------------------+-----------------------------------
---------------+------------------+----------------------------------------
-------------+
| id                                   | name
| cidr               | allocation_pools                                   |
+--------------------------------------+-----------------------------------
---------------+------------------+----------------------------------------
-------------+
| da815e8d-ad2d-42ae-8a20-8f7f2350e391 | privsub
| 10.0.96.0/21     | {"start": "10.0.96.2", "end": "10.0.103.254"}        |
+--------------------------------------+-----------------------------------
---------------+------------------+----------------------------------------
-------------+
```

10. Create a router.

```
[root@osp-cont-1 ~(finance)]# neutron router-create route1
Created a new router:
+-----------------------+--------------------------------------+
| Field                 | Value                                |
+-----------------------+--------------------------------------+
| admin_state_up        | True                                 |
| distributed           | False                                |
| external_gateway_info |                                      |
| ha                    | True                                 |
| id                    | 86d142ce-9ad3-4825-8ec3-e61d64944649 |
| name                  | route1                               |
| routes                |                                      |
| status                | ACTIVE                               |
| tenant_id             | 9ce5392d728a4e4086b6ed55a8a456a0     |
+-----------------------+--------------------------------------+
```

11. Display the router details.

```
[root@osp-cont-1 ~(finance)]# neutron router-show route1
+-----------------------+--------------------------------------+
| Field                 | Value
|
+-----------------------+--------------------------------------+
| admin_state_up        | True
| distributed           | False
| external_gateway_info |
| ha                    | True
| id                    | 86d142ce-9ad3-4825-8ec3-e61d64944649
| name                  | route1
| routes                |
| status                | ACTIVE
| tenant_id             | 9ce5392d728a4e4086b6ed55a8a456a0
+-----------------------+--------------------------------------+
```

12. Connect the tenant subnet to the router.

```
[root@osp-cont-1 ~(finance)]# subnet_id=$(neutron subnet-list |\
 awk ' /10.0./ { print $2 } ')
[root@macc81f6665334f ~(finance_member)]# neutron \
router-interface-add route1 $subnet_id
Added interface 8a8d40a1-9214-4e2d-8e12-61b9e23a8717 to router route1.
Add external network
```

Once the tenant network is configure, it needs to be connected to the externaldefined in 5.1.1 Prepare the Test Environment via the tenant router. This allows communication between the public and private subnets.

1. Switch to ‘admin’ environment.

```
[root@osp-cont-1 ~(finance)]# source /root/keystonerc_admin
```

2. Find the router Keystone ID.

```
[root@osp-cont-1 ~(openstack_admin)]# route_id=$(neutron router-list |\
awk ' /route1/ { print $2 } ')
```

3. Find the external network Keystone ID.

```
[root@osp-cont-1 ~(openstack_admin)]# ext_net_id=$(neutron net-list |\
awk ' /public/ { print $2 } ')
```

4. Set the router gateway to the external network ID.

```
[root@osp-cont-1 ~(openstack_admin)]# neutron router-gateway-set \
$route_id $ext_net_id
```

5. Set gateway for router 86d142ce-9ad3-4825-8ec3-e61d64944649

```
[root@osp-cont-1 ~(openstack_admin)]# source /root/keystonerc_finance01
```

6. Verify the router gateway.

```
[root@osp-cont-1 ~(finance)]# neutron router-show $route_id
+-----------------------+------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------+
| Field                 | Value
|
+-----------------------+------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------+
| admin_state_up        | True
|
| distributed           | False
```

```
|
| external_gateway_info | {"network_id": "22be3c03-33b8-4015-a62e-c532d1d3df3d", "enable_snat": true,
"external_fixed_ips": [{"subnet_id": "f11d1b49-74c8-441d-9505-f6857b9bc610", "ip_address": "10.19.139.209"}]} |
| ha                    | True
|
| id                    | 86d142ce-9ad3-4825-8ec3-e61d64944649
|
| name                  | route1
|
| routes                |
|
| status                | ACTIVE
|
| tenant_id             | 9ce5392d728a4e4086b6ed55a8a456a0
|
+----------------------+--------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------+
```

7. View the gateway port on the router to find the gateway IP address.

```
[root@osp-cont-1 ~(finance)]# neutron port-list --device-id \
398eb713-c0e0-4ce3-9f34-dd91c5e0ae8b
+--------------------------------------+------+------------- +------------------------------+
| id                                   | name | mac_address       | fixed_ips
|
+--------------------------------------+------+------------- +------------------------------+
| d8b2d6af-f43d-4ede-99c6-59d34720f18a |      | fa:16:3e:73:a1:15 | {"subnet_id": "0300da73-f9d5-
|                                                                   41df-bbf8-5c0275ac664c",
|                                                                  "ip_address": "10.0.96.1"} |
+--------------------------------------+------+------------- +------------------------------+
```

5.1.3 Boot the Virtual Machine Instances

Create a key pair, boot two virtual machine instances as the tenant user and connect them to the private tenant network.

1. Create a key pair and set its permissions.

```
[root@osp-cont-1 ~(finance)]# nova keypair-add finkp > \
/root/finkp.pem
[root@osp-cont-1 ~(finance)]# chmod 600 /root/finkp.pem
```

2. Identify network id of the private network.

```
[root@osp-cont-1 ~(finance)]# priv_net_id=$(neutron net-list | \

awk ' /priv/ { print $2 } ')
```

3. Boot two instances.

   - Booting webserver finweb1 instance:

```
[root@osp-cont-1 ~(finance)]# nova boot --flavor 3 –image \
 rhel70unlock --key-name finkp finweb1 --nic net-id=$priv_net_id
+--------------------------------------+-----------------------------------------------+
| Property                             | Value                                         |
+--------------------------------------+-----------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                        |
| OS-EXT-AZ:availability_zone          |                                               |
| OS-EXT-SRV-ATTR:host                 | -                                             |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | -                                             |
| OS-EXT-SRV-ATTR:instance_name        | instance-00000002                             |
| OS-EXT-STS:power_state               | 0                                             |
| OS-EXT-STS:task_state                | scheduling                                    |
| OS-EXT-STS:vm_state                  | building                                      |
| OS-SRV-USG:launched_at               | -                                             |
| OS-SRV-USG:terminated_at             | -                                             |
| accessIPv4                           |                                               |
```

```
| accessIPv6                           |                                                       |
| adminPass                            | 2u7U453xjmGd                                          |
| config_drive                         |                                                       |
| created                              | 2015-05-13T15:45:07Z                                  |
| flavor                               | m1.medium (3)                                         |
| hostId                               |                                                       |
| id                                   | 80291de3-a64a-41e4-ba15-4c2994768672                  |
| image                                | rhel70unlock (13b9561e-d86c-49e8-a59c-0507feeaac9c) |
| key_name                             | finkp                                                 |
| metadata                             | {}                                                    |
| name                                 | finweb1                                               |
| os-extended-volumes:volumes_attached | []                                                    |
| progress                             | 0                                                     |
| security_groups                      | default                                               |
| status                               | BUILD                                                 |
| tenant_id                            | 9ce5392d728a4e4086b6ed55a8a456a0                      |
| updated                              | 2015-05-13T15:45:07Z                                  |
| user_id                              | 3d7235dd2ff24ed39506599446125a9d                      |
+--------------------------------------+-------------------------------------------------------+
```

- Booting database server findb1 instance:

```
[root@osp-cont-1 ~(finance_member)]# nova boot --flavor 3 --image \
rhel70unlock --key-name finkp findb1 --nic net-id=$priv_net_id
+--------------------------------------+-------------------------------------------------------+
| Property                             | Value                                                 |
+--------------------------------------+-------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                |
| OS-EXT-AZ:availability_zone          |                                                       |
| OS-EXT-SRV-ATTR:host                 | -                                                     |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | -                                                     |
| OS-EXT-SRV-ATTR:instance_name        | instance-00000005                                     |
| OS-EXT-STS:power_state               | 0                                                     |
| OS-EXT-STS:task_state                | scheduling                                            |
| OS-EXT-STS:vm_state                  | building                                              |
| OS-SRV-USG:launched_at               | -                                                     |
| OS-SRV-USG:terminated_at             | -                                                     |
| accessIPv4                           |                                                       |
| accessIPv6                           |                                                       |
| adminPass                            | 5WoYgnMRuyQw                                          |
| config_drive                         |                                                       |
| created                              | 2015-05-13T16:10:04Z                                  |
| flavor                               | m1.medium (3)                                         |
| hostId                               |                                                       |
| id                                   | 917cedbe-7a39-4776-a03b-54fcaed0eee8                  |
| image                                | rhel65unlock (d5ed3662-4421-40a8-9ee2-66be7cc31d7f) |
| key_name                             | finkp                                                 |
| metadata                             | {}                                                    |
| name                                 | findb1                                                |
| os-extended-volumes:volumes_attached | []                                                    |
| progress                             | 0                                                     |
| security_groups                      | default                                               |
| status                               | BUILD                                                 |
| tenant_id                            | 9ce5392d728a4e4086b6ed55a8a456a0                      |
| updated                              | 2015-05-13T16:10:04Z                                  |
| user_id                              | 3d7235dd2ff24ed39506599446125a9d                      |
+--------------------------------------+-------------------------------------------------------+
```

- View the instances.

```
[root@osp-cont-1 ~(finance)]# nova list
+--------------------------------------+--------+--------+-------+-------+-----------------+
| ID                                   | Name   | Status | Task  | Power | Networks        |
|                                                          State | State |                 |
+--------------------------------------+--------+--------+-------+-------+-----------------+
| 917cedbe-7a39-4776-a03b-54fcaed0eee8 | findb1  | ACTIVE | -          | Running     |
priv=10.0.96.6 |
| 80291de3-a64a-41e4-ba15-4c2994768672 | finweb1 | ACTIVE | -          | Running     |
priv=10.0.96.2 |
+--------------------------------------+--------+--------+-------+-------+-----------------+
```

5.1.4 Associate a Floating IP Address with an Instance

Create a floating IP address and associate it with an instance. Ping the instance via the floating IP address.

1. Create a floating IP address from the public subnet pool. Save its Keystone ID to an environment variable.

```
[root@osp-cont-1 ~(finance)]# floatip_id=$(neutron floatingip-create public
| awk ' / id/ { print $2 } ')

[root@osp-cont-1 ~(finance)]# echo $floatip_id

93961db0-1f42-48f3-a391-23dd3f38f319
```

2. Find the port ID of an instance.

```
[root@osp-cont-1 ~(finance)]# finvm1_id=$(nova list | \
awk ' /finweb1/ { print $2 } ')

[root@osp-cont-1 ~(finance)]# finvm1_port=$(neutron port-list \

--device_id $finvm1_id | awk ' /ip_address/ { print $2 } ')
```

3. Associate the floating IP address ID with the port ID.

```
[root@osp-cont-1 ~(finance)]# neutron floatingip-associate \
$floatip_id $finvm1_port
Associated floating IP 54195e0f-a47a-4dc8-b4ca-423ae9de4b2b
```

4. Verify the associated IP. This will display which private tenat IP is associated to a public IP.

```
[root@osp-cont-1 ~(financer)]# neutron floatingip-list
+--------------------------------------+------------------+--------------------------------+
| id                                   | fixed_ip  | floating_ip    | port_id              |
+--------------------------------------+------------------+--------------------------------+
| 54195e0f-a47a-4dc8-b4ca-423ae9de4b2b | 10.0.96.2        | 10.19.139.211       |
31c3544f-21ee-4995-b6d4-a7bf3566b769 |
+--------------------------------------+------------------+--------------------------------+
```

5. Ping the first instance via floating IP address 10.19.139.233.

```
[root@osp-cont-1 ~(finance)]# ping -c 3 10.19.139.211
PING 10.19.139.233 (10.19.139.233) 56(84) bytes of data.
64 bytes from 10.19.139.211: icmp_seq=1 ttl=63 time=3.92 ms
64 bytes from 10.19.139.211: icmp_seq=2 ttl=63 time=0.871 ms
64 bytes from 10.19.139.211: icmp_seq=3 ttl=63 time=0.769 ms
```

The horizon GUI displays the network topology for the instances as described in

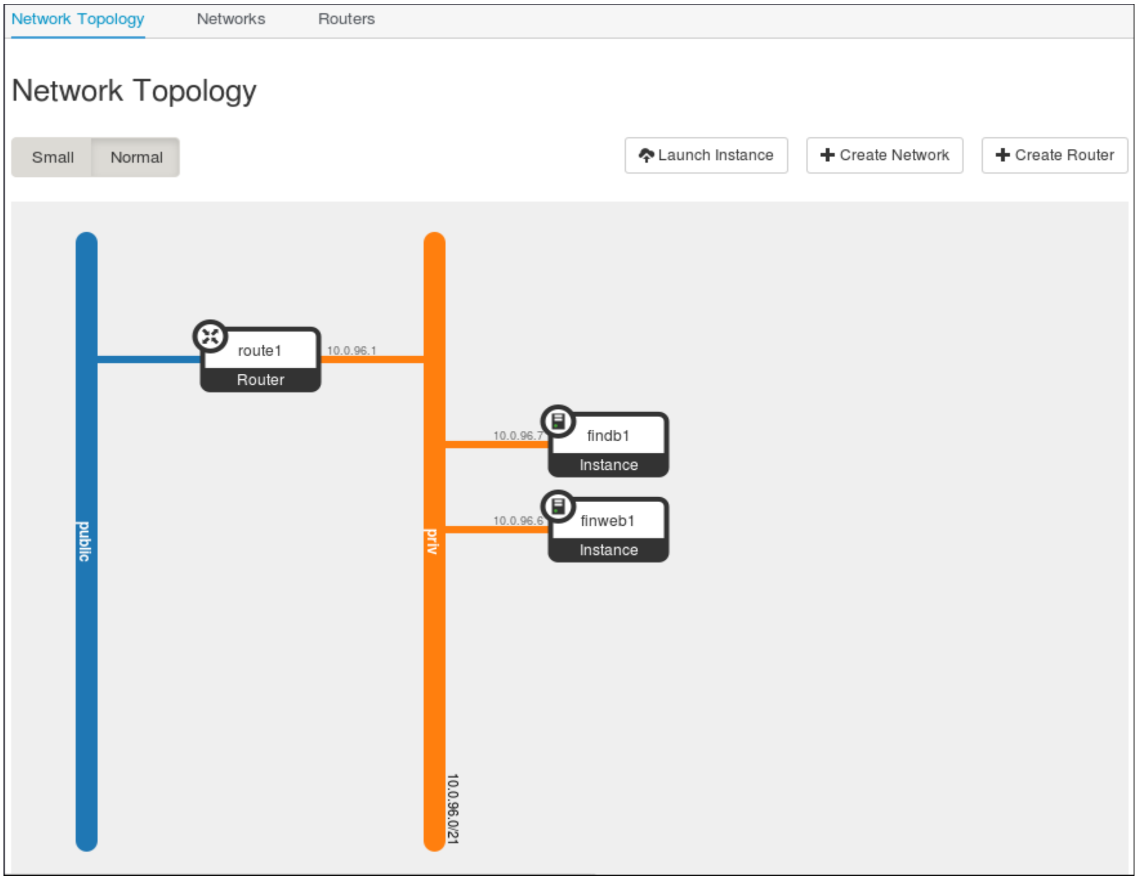


Figure 5.1.4 1: Network Topology for instances

5.1.5 Attach a Persistent Storage Volume to an Instance

1. Create cinder volume for the web server.

```
[root@osp-cont-1 ~(finance)]# cinder create --display-name \
webdisk 5
+---------------------+--------------------------------------+
|       Property      |                Value                 |
+---------------------+--------------------------------------+
|     attachments     |                  []                  |
|  availability_zone  |                 nova                 |
|       bootable      |                false                 |
|      created_at     |      2015-05-14T01:11:36.910555      |
| display_description |                 None                 |
|     display_name    |               webdisk                |
|      encrypted      |                False                 |
|          id         | 8c90bebe-2e74-4dc5-9917-b00da5cd3d4d |
|       metadata      |                  {}                  |
|         size        |                  5                   |
|     snapshot_id     |                 None                 |
|     source_volid    |                 None                 |
|        status       |               creating               |
```

```
|     volume_type     |                 None                 |
+---------------------+--------------------------------------+
```

2. Create cinder volume for database server.

```
[root@osp-cont-1 ~(finance)]# cinder create --display-name \
dbdisk 5
+---------------------+--------------------------------------+
|       Property      |                Value                 |
+---------------------+--------------------------------------+
|     attachments     |                  []                  |
|  availability_zone  |                 nova                 |
|       bootable      |                false                 |
|      created_at     |      2015-05-14T01:11:47.197583      |
| display_description |                 None                 |
|     display_name    |                dbdisk                |
|      encrypted      |                False                 |
|          id         | c3b8f310-8020-4bb2-970e-2f4672028cf8 |
|       metadata      |                  {}                  |
|         size        |                  5                   |
|     snapshot_id     |                 None                 |
|     source_volid    |                 None                 |
|        status       |               creating               |
|     volume_type     |                 None                 |
+---------------------+--------------------------------------+
```

3. Find the volume ID of the volume made for the web server.

```
[root@osp-cont-1 ~(finance)]# volid=$(nova volume-list | \
awk ' /webdisk/ { print $2 } ')
[root@osp-cont-1 ~(financer)]# echo $volid
8c90bebe-2e74-4dc5-9917-b00da5cd3d4d
```

4. Attach the web volume to the running instance of the web server.

```
[root@osp-cont-1 ~(finance)]# nova volume-attach finweb1 \
$volid
+----------+--------------------------------------+
| Property | Value                                |
+----------+--------------------------------------+
| device   | /dev/vdb                             |
| id       | 41e39fee-8da0-4573-a1ed-e004f8888dd5 |
| serverId | 80291de3-a64a-41e4-ba15-4c2994768672 |
| volumeId | 8c90bebe-2e74-4dc5-9917-b00da5cd3d4d |
+----------+--------------------------------------+
```

5. Repeat this step for database disk. After the volumes have been attached, they are verified via the nova volume-list command.

```
[root@osp-cont-1 ~(finance)]# nova volume-list
+--------------------------------------+--------+------+--------------------------------------+
| ID                                   | Display| Size |              Attached
|                                      | Name   |      |             to
+--------------------------------------+--------+------+--------------------------------------+
| c3b8f310-8020-4bb2-970e-2f4672028cf8 | in-use | dbdisk      | 5    | None         |
917cedbe-7a39-4776-a03b-54fcaed0eee8 |
| 8c90bebe-2e74-4dc5-9917-b00da5cd3d4d | in-use | webdisk     | 5    | None         |
80291de3-a64a-41e4-ba15-4c2994768672
|+--------------------------------------+--------+------+--------------------------------------+
```

6. Log into the instance as follows.

```
[root@osp-cont-1 ~(finance)]# ssh -i finkp.pem \
```

```
cloud-user@10.19.139.211
Last login: Wed Feb  4 14:29:46 2015 from 10.19.141.57

[cloud-user@finweb1 ~]$ su -
Password:
Last login: Thu May 14 08:10:44 EDT 2015 on pts/0
Last failed login: Thu May 14 17:43:21 EDT 2015 on pts/1
There was 1 failed login attempt since the last successful login.
[root@finweb1 ~]#
```

NOTE: The root password had been set earlier in the qcow2 image before creating the glance image. Please refer to C.5 Editing RHEL Guest (qcow2) Image for details.

5.1.6 Deploying the Multi-tier Web Application

5.1.6.1 Install basic web server

1. Install the web server and needed support packages for the test application.

```
[root@finweb1 /]# yum install -y httpd php php-mysql mariadb
```

2. Creating a filesystem based on the attached cinder volume (/dev/vdb):

```
[root@finweb1 ~]# parted /dev/vdb mklabel gpt
Information: You may need to update /etc/fstab.
[root@finweb1 ~]# mkfs.ext4 /dev/vdb
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
327680 inodes, 1310720 blocks
65536 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1342177280
40 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

3. Mount the filesystem

```
[root@finweb1 ~]# blkid /dev/vdb
/dev/vdb: UUID="8ad347af-fbe1-4006-a4c5-050961b5da3f" TYPE="ext4"

[root@finweb1 ~]# echo “UUID=8ad347af-fbe1-4006-a4c5-050961b5da3f
/var/www/html ext4 defaults 0 0 ” >> /etc/fstab

[root@finweb1 ~]# mount -a

[root@finweb1 ~]# df -h | grep html
```

```
/dev/vdb          4.8G   20M  4.6G   1% /var/www/html
```

NOTE: The Virtual Instance must be registered to Channel Distribution Network or Satellite to be able to download the packages.

4. Enable Apache service and start it

```
[root@finweb1 ~]# systemctl enable httpd.service
ln -s '/usr/lib/systemd/system/httpd.service'
'/etc/systemd/system/multi-user.target.wants/httpd.service'
[root@finweb1 ~]# systemctl start httpd.service
```

5. Verify if web service is running

Figure 5.1.6 1: Web Server Verification

5.1.6.2 Deploy database server

1. Install the following packages

```
[root@findb1 /]# yum install -y mariadb php php-mysql
```

NOTE: The Virtual Instance must be registered to Channel Distribution Network or Satellite to be able to download the packages.

2. Create filesystem: On the database server findb1, repeat the filesystem creation process as described for finweb1 5.1.6.1 Install basic web server to create */usr/share/mariadb-galera* filesystem.

```
[root@findb1 ~]# df -h | grep maria
/dev/vdb        4.8G   23M  4.6G   1% /usr/share/mariadb-galera
```

1. Enable and start MariaDB

```
[root@findb1 ~]# systemctl enable mariadb.service
ln -s '/usr/lib/systemd/system/mariadb.service'
'/etc/systemd/system/multi-user.target.wants/mariadb.service'

[root@findb1 ~]# systemctl start mariadb.service
```

3. Create a database called wordpress , user called finweb1 and set it up with password 'finpass'. The tenant network for the database server is "10.0.96.2".

```
[root@findb1 /]# mysql -u root

MariaDB [(none)]> create database wordpress ;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant all on wordpress.* to finweb1@'%';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> set password for finweb1@'%' =  password('finpass');
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>  grant all on wordpress.* to finweb1@'localhost';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> set password for finweb1@'localhost' =
password('finpass');
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant all on wordpress.* to finweb1@'10.0.96.2';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> set password for finweb1@'10.0.96.2' =
password('finpass');
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> show databases ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
| wordpress          |
+--------------------+
```

4. Display user and host details

```
[root@findb1 /]# mysql -u root -e "select user,host from mysql.user";
+---------+------------------+
| user    | host             |
+---------+------------------+
| finweb1 | %                |
```

```
| finweb1 | 10.0.96.2    |
| root    | 127.0.0.1        |
| root    | ::1              |
|         | findb1.novalocal |
| root    | findb1.novalocal |
|         | localhost        |
| finweb1 | localhost        |
| root    | localhost        |
```

5. Verify connectivity via IP

```
[cloud-user@finweb1 ~]$ mysql -u finweb1 -p -h 10.0.96.6 wordpress
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 28
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2014, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

MySQL [wordpress]>   (This confirms that the user with the correct password
is able to connect via the private tenant IP)
```

5.1.6.3 Deploying web application

The web application used is WordPress.

1. On the web server download WordPress.

```
[root@finweb1 ~]# cd /var/www/html/

[root@finweb1 html]# curl -O https://wordpress.org/latest.tar.gz

[root@finweb1 html]# tar -xvzf latest.tar.gz

[root@finweb1 wordpress]# cd wordpress/
```

2. Copy the sample config file.

```
[root@finweb1 wordpress]# cp -pr wp-config-sample.php wp-config.php
```

3. Edit the content as mentioned in bold.

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'finweb1');

/** MySQL database password */
define('DB_PASSWORD', 'finpass');

/** MySQL hostname */
```

```
define('DB_HOST', '10.0.96.6'); (Tenant network IP of findb1 server)
```

4. From a browser invoke the URL http://10.19.139.211/wordpress/wp-admin/install.php to complete WordPress installation.

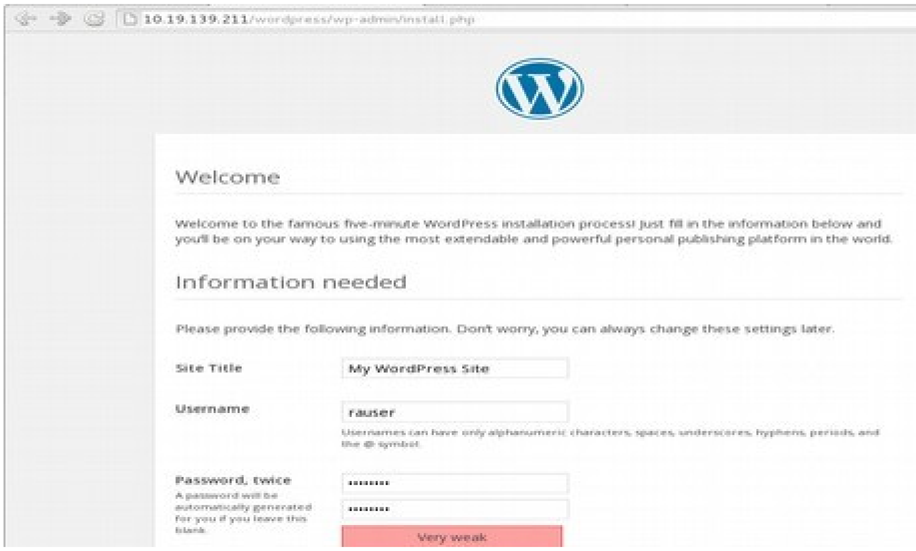


Figure 5.1.6 2: WordPress Install

5. Once the installation has been completed, test the Web Server from a Client by invoking a browser with the following URL http://10.19.139.211/wordpress/wp-login.php

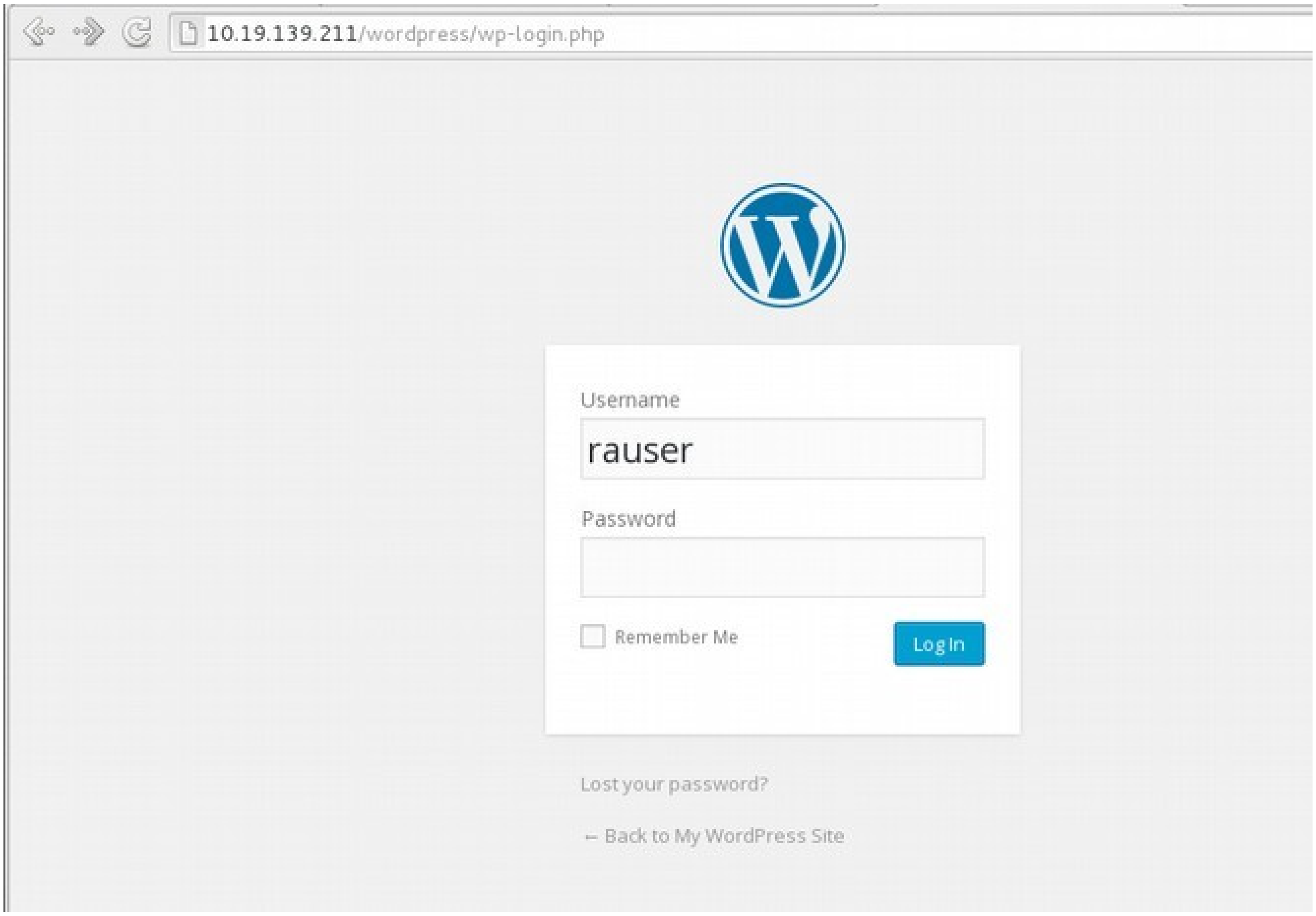


Figure 5.1.6 3: Web App Login

6 Conclusion

This reference architecture describes Red Hat's approach to high availability and scaling with RHEL OSP 6 utilizing an integrated Installer. The approach leverages the enterprise-tested RHEL HA Add-on along with the stable code base afforded by RHEL OSP 6 so that customers can deploy a scale out solution with confidence. The steps described in this reference architecture were performed on realistic hardware in the Systems Engineering lab using only released code.

Although OpenStack supports many messaging, database, and load balancer technologies, this document focuses on a solution comprised entirely from shipping Red Hat software components. Red Hat tests and supports all of these components. These components are bundled into a single solution to ensure that Red Hat supports and stands behind the end-to-end stack.

This document covers each of these topics in sufficient detail to allow Red Hat customers to reproduce them in their own environments. The Red Hat Enterprise Linux OpenStack Platform configuration described in this document can be customized and expanded to meet specific requirements. Future reference architectures build on the features exposed in this document add additional OpenStack services as well as strategies for integrating Red Hat Enterprise Linux OpenStack Platform with other Red Hat products.

Appendix A: Deployed Node Outputs

A.1 Iptables Output

A.1.1 RHEL OSP Installer iptables output

```
[root@ospceph-rinst2 ~]# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state
RELATED,ESTABLISHED
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            state NEW tcp
dpt:22
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            state NEW tcp
dpt:80
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            state NEW tcp
dpt:443
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            state NEW tcp
```

```
dpt:53
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0            state NEW udp
dpt:53
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            state NEW tcp
dpt:111
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0            state NEW udp
dpt:111
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            state NEW tcp
dpt:32803
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0            state NEW udp
dpt:32769
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            state NEW tcp
dpt:2020
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0            state NEW udp
dpt:2020
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            state NEW tcp
dpt:662
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0            state NEW udp
dpt:662
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            state NEW tcp
dpt:892
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0            state NEW udp
dpt:892
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            state NEW tcp
dpt:875
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0            state NEW udp
dpt:875
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            state NEW tcp
dpt:2049
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0            state NEW udp
dpt:2049
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0            state NEW udp
dpt:69
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0            state NEW udp
dpt:8140
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            state NEW tcp
dpt:8140

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  172.22.141.0/24      0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            172.22.141.0/24
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  192.168.141.0/24     0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            192.168.141.0/24
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@ospceph-rinst2 ~]#
```

A.1.2 Ceph admin server iptables output

```
[root@osp-ice ~]# iptables -nL
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
```

```
target     prot opt in     out     source               destination
ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
tcp dpt:80
ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
tcp dpt:2003
ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
tcp dpt:4506
ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
tcp dpt:4505
ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
tcp dpt:22
ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
state RELATED,ESTABLISHED
ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
state NEW tcp dpt:22
REJECT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
target     prot opt in     out     source               destination
REJECT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT 16M packets, 1769M bytes)
target     prot opt in     out     source               destination
```

A.2 Open vSwitch Configuration

```
[root@osp-cont-1 ~(openstack_admin)]# ovs-vsctl show
52734b0c-3787-43e1-a86d-4a692a2ddb86
    Bridge br-tun
        fail_mode: secure
        Port "vxlan-0a138886"
            Interface "vxlan-0a138886"
                type: vxlan
                options: {df_default="true", in_key=flow,
local_ip="172.22.4.25", out_key=flow, remote_ip="10.19.136.134"}
        Port patch-int
            Interface patch-int
                type: patch
                options: {peer=patch-tun}
        Port "vxlan-c000026f"
            Interface "vxlan-c000026f"
                type: vxlan
                options: {df_default="true", in_key=flow,
local_ip="172.22.4.25", out_key=flow, remote_ip="192.0.2.111"}
        Port br-tun
            Interface br-tun
                type: internal
        Port "vxlan-0a138877"
            Interface "vxlan-0a138877"
                type: vxlan
                options: {df_default="true", in_key=flow,
```

```
local_ip="172.22.4.25", out_key=flow, remote_ip="10.19.136.119"}
        Port "vxlan-0a138880"
            Interface "vxlan-0a138880"
                type: vxlan
                options: {df_default="true", in_key=flow,
local_ip="172.22.4.25", out_key=flow, remote_ip="10.19.136.128"}
    Bridge br-ex
        Port phy-br-ex
            Interface phy-br-ex
                type: patch
                options: {peer=int-br-ex}
        Port "eno1"
            Interface "eno1"
        Port br-ex
            Interface br-ex
                type: internal
    Bridge br-int
        fail_mode: secure
        Port "ha-58389435-ac"
            tag: 1
            Interface "ha-58389435-ac"
                type: internal
        Port "qg-60c4a27a-8c"
            tag: 3
            Interface "qg-60c4a27a-8c"
                type: internal
        Port patch-tun
            Interface patch-tun
                type: patch
                options: {peer=patch-int}
        Port int-br-ex
            Interface int-br-ex
                type: patch
                options: {peer=phy-br-ex}
        Port br-int
            Interface br-int
                type: internal
        Port "tapcf33d9d0-f8"
            tag: 2
            Interface "tapcf33d9d0-f8"
                type: internal
        Port "qr-8a8d40a1-92"
            tag: 2
            Interface "qr-8a8d40a1-92"
                type: internal
    ovs_version: "2.1.3"
```

A.3 HAProxy settings

The */etc/haproxy/haproxy.cfg* file details the HAProxy settings for each service and their ports:

```
[root@osp-cont-1 ~]# cat /etc/haproxy/haproxy.cfg
# This file managed by Puppet
global
```

```
  daemon
  group  haproxy
  maxconn  10000
  pidfile  /var/run/haproxy.pid
  user  haproxy

defaults
  log  127.0.0.1 local2 warning
  mode  tcp
  option  tcplog
  option  redispatch
  retries  3
  timeout  connect 5s
  timeout  client 30s
  timeout  server 30s

listen amqp
  bind 172.22.141.95:5672
  mode  tcp
  option  tcpka
  option  tcplog
  timeout  client 900m
  timeout  server 900m
  server lb-backend-osp-cont-1 172.22.141.201:5672  check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:5672  check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:5672  check inter 1s

listen ceilometer-api
  bind 172.22.141.60:8777
  bind 172.22.141.59:8777
  bind 172.22.141.58:8777
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:8777  check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:8777  check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:8777  check inter 1s

listen cinder-api
  bind 172.22.141.63:8776
  bind 172.22.141.62:8776
  bind 172.22.141.61:8776
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:8776  check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:8776  check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:8776  check inter 1s

listen galera
  bind 172.22.141.64:3306
  mode  tcp
  option  tcplog
  option  httpchk
  option  tcpka
  stick  on dst
  stick-table  type ip size 2
```

```
  timeout  client 90m
  timeout  server 90m
  server pcmk-osp-cont-1 172.22.141.201:3306  check inter 1s port 9200
on-marked-down shutdown-sessions
  server pcmk-osp-cont-2 172.22.141.202:3306  check inter 1s port 9200
on-marked-down shutdown-sessions
  server pcmk-osp-cont-3 172.22.141.204:3306  check inter 1s port 9200
on-marked-down shutdown-sessions

listen glance-api
  bind 172.22.141.67:9292
  bind 172.22.141.66:9292
  bind 172.22.141.65:9292
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:9292  check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:9292  check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:9292  check inter 1s

listen glance-registry
  bind 172.22.141.67:9191
  bind 172.22.141.66:9191
  bind 172.22.141.65:9191
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:9191  check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:9191  check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:9191  check inter 1s

listen heat-api
  bind 172.22.141.70:8004
  bind 172.22.141.69:8004
  bind 172.22.141.68:8004
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:8004  check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:8004  check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:8004  check inter 1s

listen heat-cfn
  bind 172.22.141.75:8000
  bind 172.22.141.72:8000
  bind 172.22.141.71:8000
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:8000  check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:8000  check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:8000  check inter 1s

listen heat-cloudwatch
  bind 172.22.141.69:8003
  bind 172.22.141.68:8003
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:8003  check inter 1s
```

```
  server lb-backend-osp-cont-2 172.22.141.202:8003  check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:8003  check inter 1s

listen horizon
  bind 172.22.141.82:80
  bind 172.22.141.79:80
  bind 172.22.141.76:80
  mode  http
  cookie  SERVERID insert indirect nocache
  option  httplog
  server lb-backend-osp-cont-1 172.22.141.201:80 cookie
lb-backend-osp-cont-1 check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:80 cookie
lb-backend-osp-cont-2 check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:80 cookie
lb-backend-osp-cont-3 check inter 1s

listen keystone-admin
  bind 172.22.141.87:35357
  bind 172.22.141.86:35357
  bind 172.22.141.83:35357
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:35357  check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:35357  check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:35357  check inter 1s

listen keystone-public
  bind 172.22.141.87:5000
  bind 172.22.141.86:5000
  bind 172.22.141.83:5000
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:5000  check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:5000  check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:5000  check inter 1s

listen neutron-api
  bind 172.22.141.91:9696
  bind 172.22.141.90:9696
  bind 172.22.141.89:9696
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:9696  check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:9696  check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:9696  check inter 1s

listen nova-api
  bind 172.22.141.94:8774
  bind 172.22.141.93:8774
  bind 172.22.141.92:8774
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:8774  check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:8774  check inter 1s
```

```
  server lb-backend-osp-cont-3 172.22.141.204:8774  check inter 1s

listen nova-metadata
  bind 172.22.141.94:8775
  bind 172.22.141.93:8775
  bind 172.22.141.92:8775
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:8775  check
  server lb-backend-osp-cont-2 172.22.141.202:8775  check
  server lb-backend-osp-cont-3 172.22.141.204:8775  check

listen nova-novncproxy
  bind 172.22.141.94:6080
  bind 172.22.141.93:6080
  bind 172.22.141.92:6080
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:6080  check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:6080  check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:6080  check inter 1s

listen nova-xvpvncproxy
  bind 172.22.141.94:6081
  bind 172.22.141.93:6081
  bind 172.22.141.92:6081
  mode  tcp
  option  tcplog
  server lb-backend-osp-cont-1 172.22.141.201:6081  check inter 1s
  server lb-backend-osp-cont-2 172.22.141.202:6081  check inter 1s
  server lb-backend-osp-cont-3 172.22.141.204:6081  check inter 1s

listen stats
  bind *:81
  mode  http
  stats  enable
[root@osp-cont-1 ~]#
```

A.4 PCS Full Status

```
[root@osp-cont-1 ~]# pcs status
Cluster name: openstack
Last updated: Sun May 10 23:25:04 2015
Last change: Thu May  7 15:41:40 2015
Stack: corosync
Current DC: pcmk-osp-cont-3 (3) - partition with quorum
Version: 1.1.12-a14efad
3 Nodes configured
132 Resources configured


Online: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]

Full list of resources:

 ip-ceilometer-pub-172.22.141.60     (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-2
 ip-neutron-pub-172.22.141.91 (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-ceilometer-prv-172.22.141.59     (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-2
 ip-ceilometer-adm-172.22.141.58     (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-2
 ip-horizon-pub-172.22.141.82 (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-horizon-adm-172.22.141.76 (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
```

```
 ip-amqp-pub-172.22.141.95    (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-2
 ip-loadbalancer-pub-172.22.141.88   (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-neutron-prv-172.22.141.90 (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-neutron-adm-172.22.141.89 (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-horizon-prv-172.22.141.79 (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 Clone Set: memcached-clone [memcached]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: haproxy-clone [haproxy]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 ip-galera-pub-172.22.141.64  (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-2
 Master/Slave Set: galera-master [galera]
     Masters: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: rabbitmq-server-clone [rabbitmq-server]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 ip-keystone-pub-172.22.141.87       (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-2
 ip-keystone-adm-172.22.141.83       (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-2
 ip-keystone-prv-172.22.141.86       (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-2
 Clone Set: keystone-clone [keystone]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 ip-nova-pub-172.22.141.94    (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-nova-prv-172.22.141.93    (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-nova-adm-172.22.141.92    (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 Clone Set: openstack-nova-novncproxy-clone [openstack-nova-novncproxy]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: openstack-nova-consoleauth-clone [openstack-nova-consoleauth]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: openstack-nova-conductor-clone [openstack-nova-conductor]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: openstack-nova-api-clone [openstack-nova-api]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: openstack-nova-scheduler-clone [openstack-nova-scheduler]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 ip-cinder-pub-172.22.141.63  (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-cinder-adm-172.22.141.61  (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-cinder-prv-172.22.141.62  (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 Clone Set: cinder-api-clone [cinder-api]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: cinder-scheduler-clone [cinder-scheduler]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 cinder-volume (systemd:openstack-cinder-volume):   Started pcmk-osp-cont-2
 ip-heat-pub-172.22.141.70    (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-heat-prv-172.22.141.69    (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-heat-adm-172.22.141.68    (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-heat_cfn-pub-172.22.141.75       (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-2
 ip-heat_cfn-prv-172.22.141.72       (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-2
 Clone Set: neutron-server-clone [neutron-server]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: neutron-scale-clone [neutron-scale] (unique)
     neutron-scale:0  (ocf::neutron:NeutronScale):  Started pcmk-osp-cont-3
     neutron-scale:1  (ocf::neutron:NeutronScale):  Started pcmk-osp-cont-2
     neutron-scale:2  (ocf::neutron:NeutronScale):  Started pcmk-osp-cont-1
 Clone Set: neutron-ovs-cleanup-clone [neutron-ovs-cleanup]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 ip-heat_cfn-adm-172.22.141.71       (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-2
 Clone Set: neutron-netns-cleanup-clone [neutron-netns-cleanup]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: neutron-openvswitch-agent-clone [neutron-openvswitch-agent]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: neutron-dhcp-agent-clone [neutron-dhcp-agent]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: neutron-l3-agent-clone [neutron-l3-agent]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: neutron-metadata-agent-clone [neutron-metadata-agent]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: heat-api-clone [heat-api]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Resource Group: heat
     openstack-heat-engine    (systemd:openstack-heat-engine):     Started pcmk-osp-cont-3
 Clone Set: heat-api-cfn-clone [heat-api-cfn]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: heat-api-cloudwatch-clone [heat-api-cloudwatch]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
```

```
 Clone Set: horizon-clone [horizon]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: mongod-clone [mongod]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 openstack-ceilometer-central (systemd:openstack-ceilometer-central):     Started pcmk-osp-cont-2
 Clone Set: openstack-ceilometer-api-clone [openstack-ceilometer-api]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: openstack-ceilometer-alarm-evaluator-clone [openstack-ceilometer-alarm-evaluator]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: openstack-ceilometer-collector-clone [openstack-ceilometer-collector]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: openstack-ceilometer-notification-clone [openstack-ceilometer-notification]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: openstack-ceilometer-alarm-notifier-clone [openstack-ceilometer-alarm-notifier]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: ceilometer-delay-clone [ceilometer-delay]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 ip-glance-pub-172.22.141.67  (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-glance-prv-172.22.141.66  (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 ip-glance-adm-172.22.141.65  (ocf::heartbeat:IPaddr2):     Started pcmk-osp-cont-3
 Clone Set: glance-registry-clone [glance-registry]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]
 Clone Set: glance-api-clone [glance-api]
     Started: [ pcmk-osp-cont-1 pcmk-osp-cont-2 pcmk-osp-cont-3 ]

Failed actions:
    openstack-heat-engine_monitor_60000 on pcmk-osp-cont-3 'not running' (7): call=610,
status=complete, exit-reason='none', last-rc-change='Sun May 10 00:23:53 2015', queued=0ms, exec=0ms
    neutron-server_monitor_60000 on pcmk-osp-cont-3 'not running' (7): call=563, status=complete,
exit-reason='none', last-rc-change='Sun May 10 00:23:42 2015', queued=0ms, exec=0ms


PCSD Status:
  pcmk-osp-cont-1: Online
  pcmk-osp-cont-2: Online
  pcmk-osp-cont-3: Online

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

A.5 PCS Constraint View

```
[root@osp-cont-1 ~(openstack_admin)]# pcs constraint show
Location Constraints:
  Resource: stonith-ipmilan-10.19.143.61
    Disabled on: pcmk-macc81f66653342
  Resource: stonith-ipmilan-10.19.143.62
    Disabled on: pcmk-macc81f6665334f
  Resource: stonith-ipmilan-10.19.143.63
    Disabled on: pcmk-macc81f6665335c
Ordering Constraints:
  start openstack-nova-consoleauth-clone then start
openstack-nova-novncproxy-clone (kind:Mandatory)
  start openstack-nova-novncproxy-clone then start openstack-nova-api-clone
(kind:Mandatory)
  start openstack-nova-api-clone then start openstack-nova-scheduler-clone
(kind:Mandatory)
  start openstack-nova-scheduler-clone then start
openstack-nova-conductor-clone (kind:Mandatory)
  start cinder-scheduler-clone then start cinder-volume (kind:Mandatory)
  start cinder-api-clone then start cinder-scheduler-clone (kind:Mandatory)
  start neutron-scale-clone then start neutron-ovs-cleanup-clone
(kind:Mandatory)
```

```
  start heat-api-clone then start heat-api-cfn-clone (kind:Mandatory)
  start heat-api-cfn-clone then start heat-api-cloudwatch-clone
(kind:Mandatory)
  start heat-api-cloudwatch-clone then start openstack-heat-engine
(kind:Mandatory)
  start keystone-clone then start neutron-server-clone (kind:Mandatory)
  start keystone-clone then start openstack-nova-consoleauth-clone
(kind:Mandatory)
  start keystone-clone then start cinder-api-clone (kind:Mandatory)
  start keystone-clone then start openstack-ceilometer-central
(kind:Mandatory)
  start mongod-clone then start openstack-ceilometer-central
(kind:Mandatory)
  start openstack-ceilometer-central then start
openstack-ceilometer-collector-clone (kind:Mandatory)
  start openstack-ceilometer-collector-clone then start
openstack-ceilometer-api-clone (kind:Mandatory)
  start openstack-ceilometer-api-clone then start ceilometer-delay-clone
(kind:Mandatory)
  start ceilometer-delay-clone then start
openstack-ceilometer-alarm-evaluator-clone (kind:Mandatory)
  start openstack-ceilometer-alarm-evaluator-clone then start
openstack-ceilometer-alarm-notifier-clone (kind:Mandatory)
  start openstack-ceilometer-alarm-notifier-clone then start
openstack-ceilometer-notification-clone (kind:Mandatory)
  start neutron-ovs-cleanup-clone then start neutron-netns-cleanup-clone
(kind:Mandatory)
  start neutron-netns-cleanup-clone then start
neutron-openvswitch-agent-clone (kind:Mandatory)
  start neutron-openvswitch-agent-clone then start neutron-dhcp-agent-clone
(kind:Mandatory)
  start neutron-dhcp-agent-clone then start neutron-l3-agent-clone
(kind:Mandatory)
  start neutron-l3-agent-clone then start neutron-metadata-agent-clone
(kind:Mandatory)
  start ip-nova-pub-172.22.141.94 then start haproxy-clone (kind:Optional)
  start ip-neutron-pub-172.22.141.91 then start haproxy-clone
(kind:Optional)
  start ip-horizon-prv-172.22.141.83 then start haproxy-clone
(kind:Optional)
  start galera-master then start keystone-clone (kind:Mandatory)
  start rabbitmq-server-clone then start keystone-clone (kind:Mandatory)
  start memcached-clone then start keystone-clone (kind:Mandatory)
  start ip-ceilometer-adm-172.22.141.58 then start haproxy-clone
(kind:Optional)
  start ip-heat-adm-172.22.141.76 then start haproxy-clone (kind:Optional)
  start ip-neutron-adm-172.22.141.89 then start haproxy-clone
(kind:Optional)
  start ip-keystone-adm-172.22.141.85 then start haproxy-clone
(kind:Optional)
  start ip-ceilometer-pub-172.22.141.60 then start haproxy-clone
(kind:Optional)
  start ip-ceilometer-prv-172.22.141.59 then start haproxy-clone
(kind:Optional)
  start ip-horizon-adm-172.22.141.82 then start haproxy-clone
```

```
(kind:Optional)
  start ip-cinder-prv-172.22.141.66 then start haproxy-clone (kind:Optional)
  start ip-cinder-pub-172.22.141.68 then start haproxy-clone (kind:Optional)
  start ip-amqp-pub-172.22.141.95 then start haproxy-clone (kind:Optional)
  start ip-keystone-pub-172.22.141.87 then start haproxy-clone
(kind:Optional)
  start ip-keystone-prv-172.22.141.86 then start haproxy-clone
(kind:Optional)
  start haproxy-clone then start keystone-clone (kind:Mandatory)
  start ip-nova-adm-172.22.141.92 then start haproxy-clone (kind:Optional)
  start ip-galera-pub-172.22.141.69 then start haproxy-clone (kind:Optional)
  start ip-horizon-pub-172.22.141.84 then start haproxy-clone
(kind:Optional)
  start ip-heat-pub-172.22.141.78 then start haproxy-clone (kind:Optional)
  start ip-cinder-adm-172.22.141.63 then start haproxy-clone (kind:Optional)
  start ip-heat-prv-172.22.141.77 then start haproxy-clone (kind:Optional)
  start ip-neutron-prv-172.22.141.90 then start haproxy-clone
(kind:Optional)
  start ip-nova-prv-172.22.141.93 then start haproxy-clone (kind:Optional)
  start glance-registry-clone then start glance-api-clone (kind:Mandatory)
  start keystone-clone then start glance-registry-clone (kind:Mandatory)
  start ip-glance-adm-172.22.141.71 then start haproxy-clone (kind:Optional)
  start ip-glance-prv-172.22.141.73 then start haproxy-clone (kind:Optional)
  start ip-glance-pub-172.22.141.74 then start haproxy-clone (kind:Optional)
Colocation Constraints:
  ip-ceilometer-prv-172.22.141.59 with ip-ceilometer-pub-172.22.141.60
(score:INFINITY)
  ip-horizon-adm-172.22.141.82 with ip-horizon-pub-172.22.141.84
(score:INFINITY)
  ip-neutron-adm-172.22.141.89 with ip-neutron-pub-172.22.141.91
(score:INFINITY)
  ip-horizon-prv-172.22.141.83 with ip-horizon-pub-172.22.141.84
(score:INFINITY)
  ip-ceilometer-adm-172.22.141.58 with ip-ceilometer-pub-172.22.141.60
(score:INFINITY)
  ip-neutron-prv-172.22.141.90 with ip-neutron-pub-172.22.141.91
(score:INFINITY)
  ip-keystone-adm-172.22.141.85 with ip-keystone-pub-172.22.141.87
(score:INFINITY)
  ip-keystone-prv-172.22.141.86 with ip-keystone-pub-172.22.141.87
(score:INFINITY)
  ip-nova-adm-172.22.141.92 with ip-nova-pub-172.22.141.94 (score:INFINITY)
  ip-nova-prv-172.22.141.93 with ip-nova-pub-172.22.141.94 (score:INFINITY)
  openstack-nova-novncproxy-clone with openstack-nova-consoleauth-clone
(score:INFINITY)
  openstack-nova-api-clone with openstack-nova-novncproxy-clone
(score:INFINITY)
  openstack-nova-scheduler-clone with openstack-nova-api-clone
(score:INFINITY)
  openstack-nova-scheduler-clone with openstack-nova-conductor-clone
(score:INFINITY)
  ip-cinder-prv-172.22.141.66 with ip-cinder-pub-172.22.141.68
(score:INFINITY)
  ip-cinder-adm-172.22.141.63 with ip-cinder-pub-172.22.141.68
(score:INFINITY)
```

```
  cinder-volume with cinder-scheduler-clone (score:INFINITY)
  cinder-scheduler-clone with cinder-api-clone (score:INFINITY)
  ip-heat-prv-172.22.141.77 with ip-heat-pub-172.22.141.78 (score:INFINITY)
  ip-heat-adm-172.22.141.76 with ip-heat-pub-172.22.141.78 (score:INFINITY)
  ip-heat_cfn-adm-172.22.141.79 with ip-heat_cfn-pub-172.22.141.81
(score:INFINITY)
  ip-heat_cfn-prv-172.22.141.80 with ip-heat_cfn-pub-172.22.141.81
(score:INFINITY)
  heat-api-cloudwatch-clone with heat-api-cfn-clone (score:INFINITY)
  openstack-heat-engine with heat-api-cloudwatch-clone (score:INFINITY)
  heat-api-cfn-clone with heat-api-clone (score:INFINITY)
  neutron-ovs-cleanup-clone with neutron-scale-clone (score:INFINITY)
  neutron-netns-cleanup-clone with neutron-ovs-cleanup-clone
(score:INFINITY)
  neutron-openvswitch-agent-clone with neutron-netns-cleanup-clone
(score:INFINITY)
  neutron-dhcp-agent-clone with neutron-openvswitch-agent-clone
(score:INFINITY)
  neutron-l3-agent-clone with neutron-dhcp-agent-clone (score:INFINITY)
  neutron-metadata-agent-clone with neutron-l3-agent-clone (score:INFINITY)
  ip-horizon-prv-172.22.141.83 with haproxy-clone (score:INFINITY)
  ip-neutron-adm-172.22.141.89 with haproxy-clone (score:INFINITY)
  ip-heat-adm-172.22.141.76 with haproxy-clone (score:INFINITY)
  ip-keystone-adm-172.22.141.85 with haproxy-clone (score:INFINITY)
  ip-horizon-adm-172.22.141.82 with haproxy-clone (score:INFINITY)
  ip-neutron-pub-172.22.141.91 with haproxy-clone (score:INFINITY)
  ip-ceilometer-adm-172.22.141.58 with haproxy-clone (score:INFINITY)
  ip-ceilometer-pub-172.22.141.60 with haproxy-clone (score:INFINITY)
  ip-cinder-pub-172.22.141.68 with haproxy-clone (score:INFINITY)
  ip-cinder-prv-172.22.141.66 with haproxy-clone (score:INFINITY)
  ip-amqp-pub-172.22.141.95 with haproxy-clone (score:INFINITY)
  ip-keystone-prv-172.22.141.86 with haproxy-clone (score:INFINITY)
  ip-nova-adm-172.22.141.92 with haproxy-clone (score:INFINITY)
  ip-galera-pub-172.22.141.69 with haproxy-clone (score:INFINITY)
  ip-horizon-pub-172.22.141.84 with haproxy-clone (score:INFINITY)
  ip-heat-pub-172.22.141.78 with haproxy-clone (score:INFINITY)
  ip-cinder-adm-172.22.141.63 with haproxy-clone (score:INFINITY)
  ip-ceilometer-prv-172.22.141.59 with haproxy-clone (score:INFINITY)
  ip-nova-pub-172.22.141.94 with haproxy-clone (score:INFINITY)
  ip-keystone-pub-172.22.141.87 with haproxy-clone (score:INFINITY)
  ip-heat-prv-172.22.141.77 with haproxy-clone (score:INFINITY)
  ip-neutron-prv-172.22.141.90 with haproxy-clone (score:INFINITY)
  ip-nova-prv-172.22.141.93 with haproxy-clone (score:INFINITY)
  ip-glance-prv-172.22.141.73 with ip-glance-pub-172.22.141.74
(score:INFINITY)
  ip-glance-adm-172.22.141.71 with ip-glance-pub-172.22.141.74
(score:INFINITY)
  glance-api-clone with glance-registry-clone (score:INFINITY)
  ip-glance-prv-172.22.141.73 with haproxy-clone (score:INFINITY)
  ip-glance-adm-172.22.141.71 with haproxy-clone (score:INFINITY)
  ip-glance-pub-172.22.141.74 with haproxy-clone (score:INFINITY)
[root@osp-cont-1 ~(openstack_admin)]#
```

A.6 Deployment Assigned Values

Virtual Interface	Virtual IP	Traffic Type
ceilometer_admin_vip	172.22.141.58	Admin API
ceilometer_private_vip	172.22.141.59	Public API
ceilometer_public_vip	172.22.141.60	Public API
cinder_admin_vip	172.22.141.63	Admin API
cinder_private_vip	172.22.141.66	Management
cinder_public_vip	172.22.141.68	Public API
db_vip	172.22.141.69	Management
glance_admin_vip	172.22.141.71	Admin API
glance_private_vip	172.22.141.73	Management
glance_public_vip	172.22.141.74	Public API
heat_admin_vip	172.22.141.76	Admin API
heat_private_vip	172.22.141.77	Management
heat_public_vip	172.22.141.78	Public API
heat_cfn_admin_vip	172.22.141.79	Admin API
heat_cfn_private_vip	172.22.141.80	Management
heat_cfn_public_vip	172.22.141.81	Public API
horizon_admin_vip	172.22.141.82	Admin API
horizon_private_vip	172.22.141.83	Management
horizon_public_vip	172.22.141.84	Public API
keystone_admin_vip	172.22.141.85	Admin API
keystone_private_vip	172.22.141.86	Management
keystone_public_vip	172.22.141.87	Public API
loadbalancer_vip	172.22.141.88	Public API
neutron_admin_vip	172.22.141.89	Admin API
neutron_private_vip	172.22.141.90	Management
neutron_public_vip	172.22.141.91	Public API
nova_admin_vip	172.22.141.92	Admin API
nova_private_vip	172.22.141.93	Management
nova_public_vip	172.22.141.94	Public API
amqp_vip	172.22.141.95	Management
swift_public_vip	172.22.141.96	Public API

User	Password
Admin	redhat11 *
Ceilometer	redhat11
Cinder	redhat11
Glance	redhat11
Heat	redhat11
Heat cfn	redhat11
Keystone	redhat11
Neutron	redhat11
Nova	redhat11
Swift	redhat11
Amqp	redhat11

Database	Password
Cinder	Redhat11 *
Glance	redhat11
Heat	redhat11
Mysql root	redhat11
Keystone	redhat11
Neutron	redhat11
Nova	redhat11

* Set in the deployment first page

Table 6.1: Deployed VIPs, Users and Database Details

Appendix B: Troubleshooting Deployment

B.1 Troubleshooting failed RHEL OSP Installer script

If the "rhel-osp-installer" script gets interrupted/fails or if there is requirement to make changes, the recommended approach is to re-run the script. The inputs provided during execution of the installer script updates configuration files under */etc/foreman* directory. However they do not take effect until the installer script is re-executed.

Here is the procedure to be followed when the rhel-osp-installer script must be re-executed.

1. Stop foreman, named and DHCP processes

```
 [root@ospceph-rinst2 ~]# for i in foreman foreman-proxy foreman-tasks dhcpd
named; do   systemctl stop $i.service; done
```

2. Clean up the foreman entries

```
[root@ospceph-rinst2 ~]# foreman-rake db:drop
```

3. Restart the services

```
[root@ospceph-rinst2 ~]# for i in foreman foreman-proxy foreman-tasks dhcpd
named; do   systemctl start $i.service;
```

4. Run the installer script again

```
[root@ospceph-rinst2 ~]# rhel-osp-installer
```

5. Follow through the same installation procedure to complete the installation.

B.2 Troubleshooting discovery failure

There has been frequent discovery failures into dracut. Multiple reboots sometimes fixed the issue. This has been prevalent due to combination of two factors:

- More than one network had DHCP for the managed hosts (for example both external and provisioning networks were broadcasting DHCP)
- Both the networks were part of the same internal Dell blade switch.

There is a bugzilla open for this issue https://bugzilla.redhat.com/show_bug.cgi?id=1171329

B.3 Troubleshooting Network assignment

After the configure network step has been completed, it is better to ensure the IP assignments are proper. Double click on each host-> Edit-> Network and scroll down to verify if the network assignments are correct for the corresponding network port.

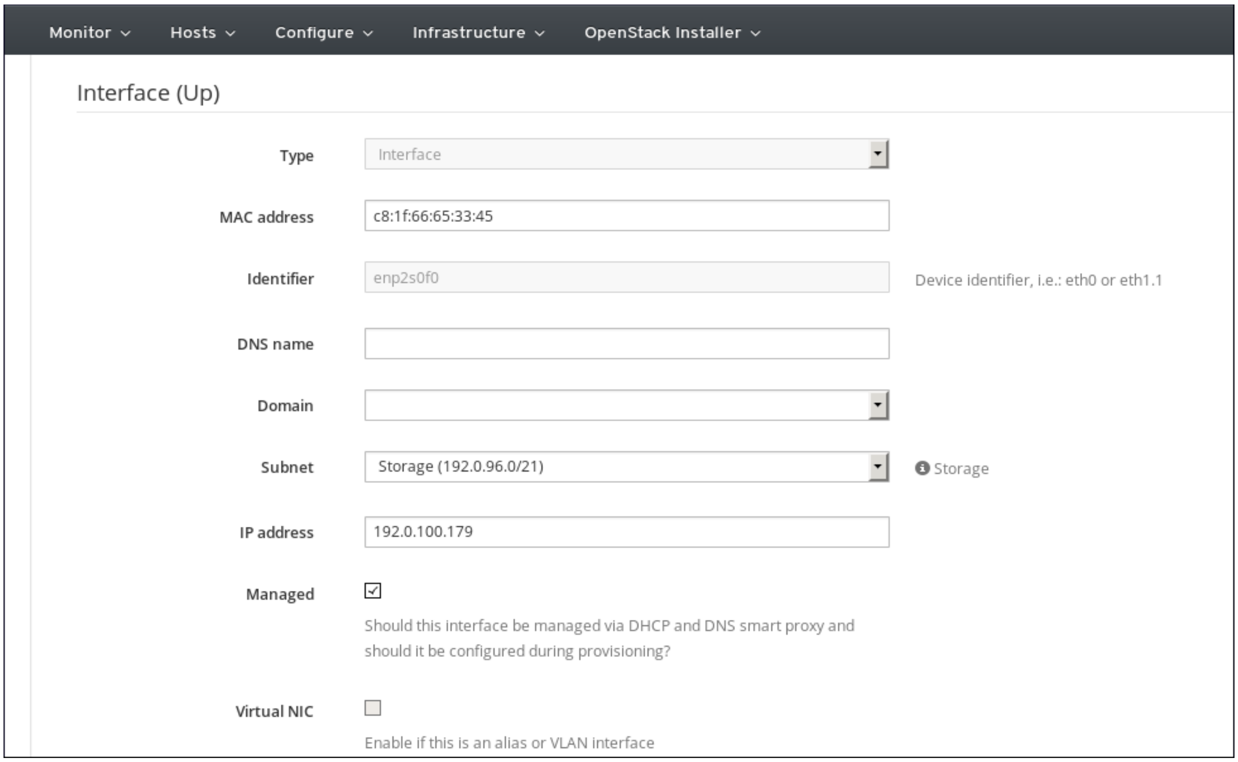


Figure 6 1: Verify Subnet and Network assigned

Below is an example for ensuring port “enp2s0f0” is assigned to subnet “Tenant” (as per network configuration step) and has an unique IP within the range for “Tenant subnet”. Please note the subnet assigned to the port is “Storage subnet” instead of “Tenant subnet”

This can be fixed by selecting the correct subnet in the “subnet” drop down menu.

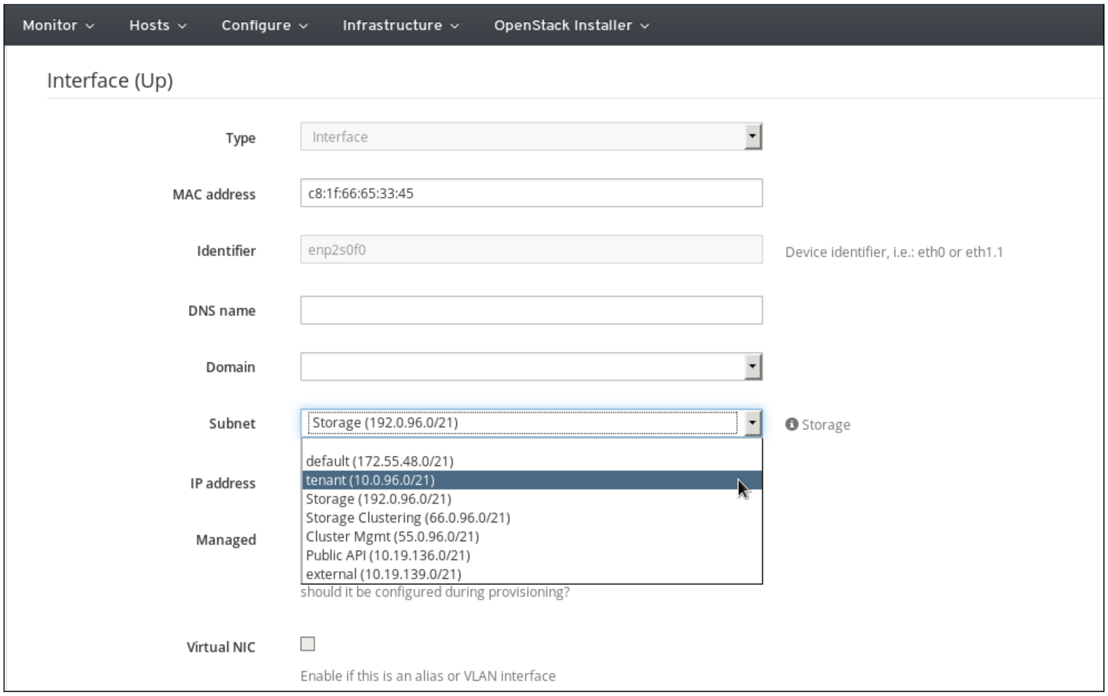


Figure 6 2: Change to Correct Subnet

With the correct subnet, a Tenant IP gets auto generated as displayed below. Ensure there is no duplication of this IP elsewhere and the IP is in the “Tenant” subnet range. In case of duplicate IP assignment, it is possible to manually enter an IP.

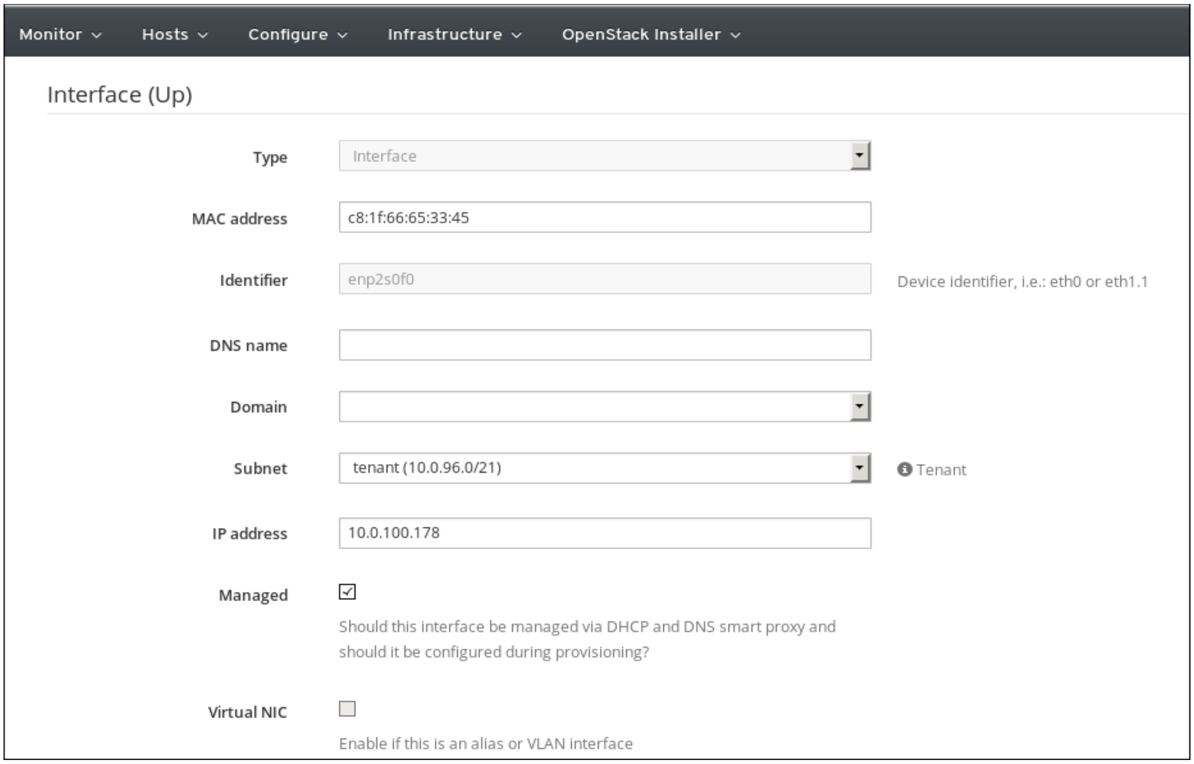


Figure 6 3: Corrected IP

Repeat these steps as required on all ports on each host being deployed and save the change by selecting “Submit” button. This ensures that all the deployed systems have correct network settings.

B.4 Troubleshooting during provisioning

Sometimes, the provisioning fails leaving failed node in “dracut” mode. If all the settings are correct (assignments, external network connectivity etc.) a reboot of the server (sometimes, repeated reboots) makes the node proceed further with provisioning.

B.5 Troubleshooting post provisioning

After the nodes have been provisioned, ensure network configuration is correct.

```
#ip address show
#route
```

If there is a problem with external connectivity, the deployment stalls since puppet cannot be executed. Please troubleshoot in the following order:

1. Verify “route” output is correct and default router points to the correct interface.
2. Verify output of “ip address show” command to see if all the required interfaces are up. If some of them are not, identify the network device name associated with a failed network, match it with network configuration files under */etc/sysconfig/network-scripts.* With RHEL 7 “systemd” setting up the network device names, there has been occasions where there is a difference between the device name in the discovery phase to device names after provisioning for the same mac address. For example, device

name shows up as “eno1” during discovery but changes to eth1 after provisioning for the same mac address. Puppet configures network config files based on discovered device names, hence should fail to bring up the network. Manually moving the network config file (in this case */etc/sysconfig/network-scripts/eno1* to */etc/sysconfig/network-scripts/eth1*) and changing the device name (“NAME=eno1” to NAME=eth1) should fix the network.

NOTE: This device name variation issue has surfaced only on blade servers possibly due to shared network architecture. This has never happened on three standalone Dell 720XDs used for Ceph in this setup. There is an open BZ https://bugzilla.redhat.com/show_bug.cgi?id=1180763 for this issue.

Also network interfaces fail to activate when the IPs assigned belong to wrong subnet or out of range. Refer to step 21 under 4.3 OpenStack Deployment for correcting this issue.

3. Once the network has been sorted out, puppet can be initiated to be rerun in either of the following two ways:

```
[root@osp-cont-1 ~]# script -c "puppet run agent --test --verbose" \
puppet-1.txt
```

Alternatively on the installer GUI, select the ***host -> Run puppet***

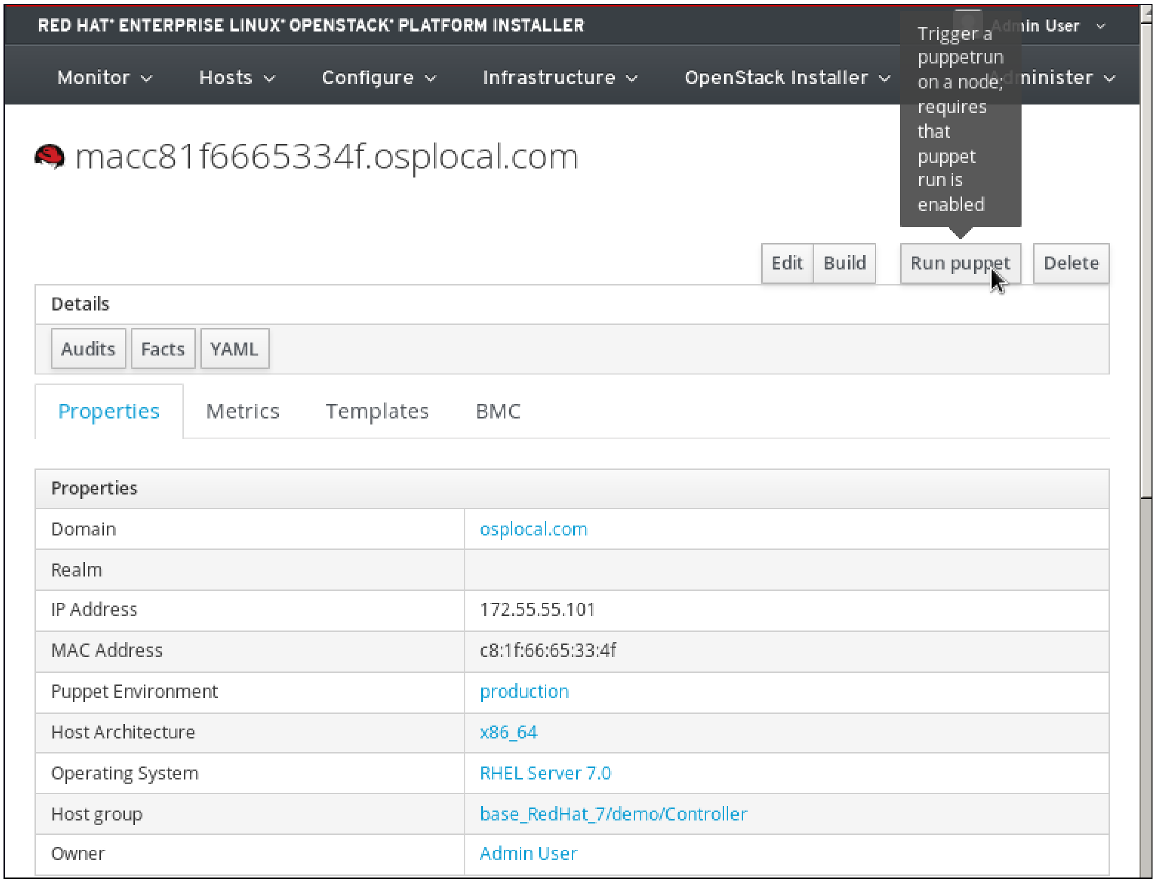


Figure 6 4: Re-run puppet

This initiates puppet to run on the node.

4. Watch the puppet run status in the log using a different login session

```
[root@osp-cont-1 ~]# tail -f /var/log/messages
```

NOTE: For better troubleshooting and reference, please make a note of all the relevant network port mac addresses, IPs etc. for the nodes being deployed.

5. Further troubleshooting can be performed with outputs displayed under "Facts", "Reports" and "YAML" tabs when you click on a deployed host as described in Figure 6 5: Deployed Server Details

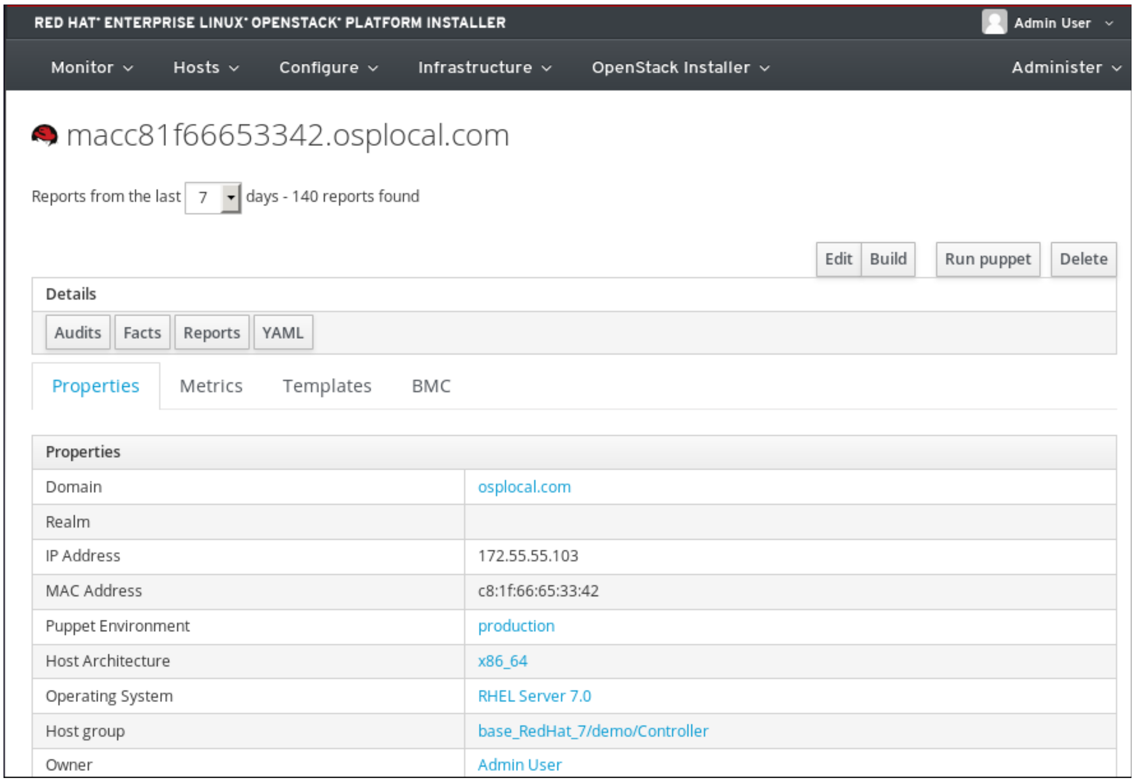


Figure 6 5: Deployed Server Details

B.6 Rebuilding host from scratch

If the integrity of the provisioned node is in question, it can rebuilt from scratch just for the node. This avoids the requirement to redeploy the entire set of nodes all over again.
On the installer GUI, select the ***host -> Build***

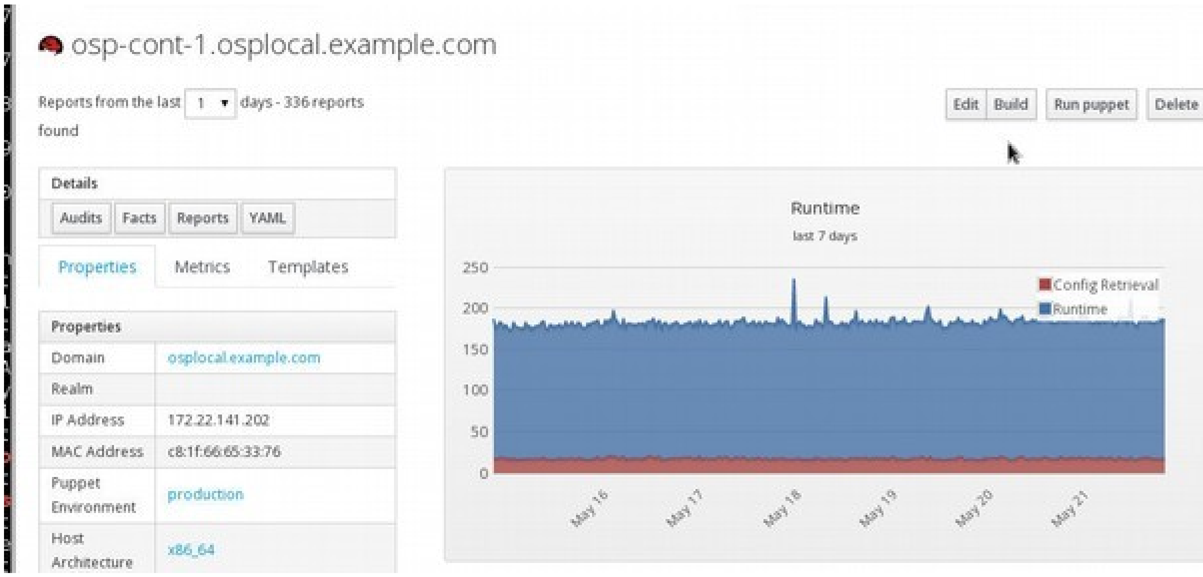


Figure 6 6: Rebuilding Host

The installer has configured the host so that it may be provisioned upon next PXE boot.

B.7 Restart Deployment from scratch

If the deployment is not successful and cannot be fixed easily, the failed deployment can be discarded and restarted all over again. However, these nodes must have been already discovered earlier with their mac addresses, hostnames, leased IPs already present in the RHEL OSP Installer config/lease files. They must be flushed out before restarting afresh.

Steps for redeployment:

1. Delete the failed deployment (OpenStack *Installer-> Deployment->* Select the right one-> *delete*).
2. Delete the hosts (*Hosts-> All hosts->Select the hosts-> Select Action-> Delete hosts)*

   This should delete the files associated for each host (the file is named after the provisioning mac address) from */var/lib/tftpboot/pxelinux.cfg/*

   If the file for any of these host does exist still, please delete them manually on the installer.
3. Log in to the RHEL OSP Installer system:
   - Stop the relevant services

```
[root@ospceph-rinst2 ~]# for i in dhcpd named foreman-proxy; do systemctl
stop $i.service ; done
```

- Remove the following files

```
[root@ospceph-rinst2 ~]# rm -rf \   /var/named/dynamic/db.141.22.172.arpa.jnl
\ /var/named/dynamic/db.osplocal.example.com.jnl
```

(*Where db.141.22.172.arpa.jnl and db.osplocal.example.com.jnl are relevant to this deployment.*)

- Zero out the entry from the dhcp leases file

```
[root@ospceph-rinst2 ~]# cat /dev/null > /var/lib/dhcpd/dhcpd.leases
```

- Restart the services

```
[root@ospceph-rinst2 ~]# for i in dhcpd named foreman-proxy; do systemctl
start $i.service ; done
```

4. Reboot all the deployment nodes and they get PXE booted and discovered
5. Verify the discovered node in the RHEL OSP Installer GUI *Hosts-> Discovered Hosts*
6. Create a new deployment as before.

Appendix C: Miscellaneous Information

C.1 Changing Installer GUI Password

1. Login to the RHEL OSP Installer GUI using the URL and credentials from the installation complete output under section 4.2.4 Install the RHEL OSP Installer. It is preferable to change the password by using Administrator menu on the right.

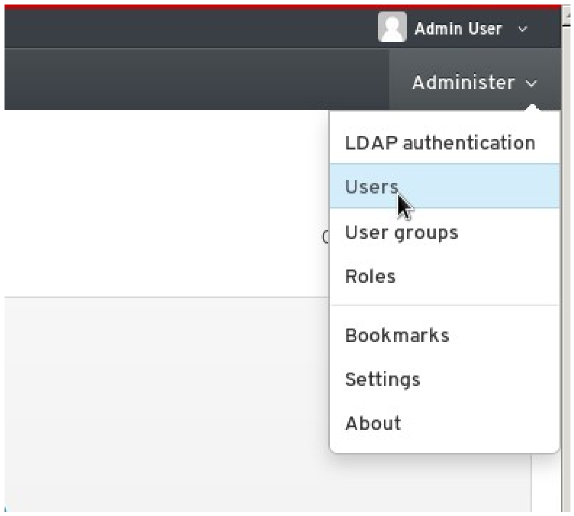


Figure 6 7: Change Password 1

2. Enter a new password

C.2 Bootstrapping Galera cluster

The following describes how bootstrapping Galera is handled in this deployment to ensure the Galera cluster nodes are brought up in the right sequence in the event where all the nodes go down.

1. Pacemaker performs resource discovery to determine if any active Galera instances are already up.
2. If no Galera instances are up. pacemaker starts the Galera resources on each node in the cluster in "slave" mode. The "slave" mode does not mean what it typically means for other resources. In slave mode the Galera instances are brought up in read only non-clustered mode. This is performed on each node to determine which node has the highest write sequence number.
3. After all nodes in the "gcom://" node list have reported their sequence number, the node that has the highest sequence number is picked to bootstrap the Galera cluster. Pacemaker promotes this bootstrap node to Master first.
4. After the bootstrap node is up, pacemaker promotes all the other Galera instances to "Master" which means they are allowed join the Galera cluster.

C.3 List of Galera Cluster Status Variables

For details on Galera cluster variables please refer to URL
https://mariadb.com/kb/en/mariadb/galera-cluster-status-variables/

C.4 Downloading the RHEL Guest (qcow2) Image

The qcow2 RHEL guest image is distributed via "rhel-7-server-rh-common-rpms" channel.

```
[root@ospceph-rinst2 ~]# yum install rhel-guest-image-7
                 ...
Installed:
  rhel-guest-image-7.noarch 0:7.0-20150127.0.el7

[root@ospceph-rinst2 ~]# ls /usr/share/rhel-guest-image-7 | grep qcow

rhel-guest-image-7.0-20150127.0.x86_64.qcow2
```

C.5 Editing RHEL Guest (qcow2) Image

This section directs editing the image to enable root access and set a password. The qcow2 image downloaded has root login disabled. By default "cloud-user" is the user available to login when the instance is booted up. In order to add a user and set its password or add a password to root user, guestfish tool can be Installed on a local system. Edit the qcow2 image, prior to creating a glance image in an OpenStack environment.

```
[root@osceph-risnt2 qcow_unlock]# yum install guestfish -y
```

Edit the downloaded qcow2 image by running the following command

```
[root@ospceph-rinst2 qcow_unlock]# guestfish --rw -a
rhel-guest-image-7.0-20150127.0.x86_64.qcow2
Welcome to guestfish, the libguestfs filesystem interactive shell for
editing virtual machine filesystems.

Type: 'help' for help on commands
      'man' to read the manual
      'quit' to quit the shell

><fs> run
 100%
 00:00
><fs> list-filesystems
/dev/sda1: xfs
><fs> mount /dev/sda1 /
><fs> vi /etc/shadow
><fs> quit
```

In the */etc/shadow* file, replace "!!" in the second field corresponding to root with the encrypted password. This sets the password for the image permanently. An encrypted password can be created using "openssl".

```
[root@ospceph-rinst2 qcow_unlock]# openssl passwd -1 new_password
```

Save this edited qcow2 image as **/tmp/rhel70_unlock.qcow2** and use it for Glance image creation in the OpenStack environment.

Appendix D: Contributors

Contributor	Title	Contribution
Balaji Jayavelu	Principal Software Engineer	Original Author
John Herr	Senior Software Engineer	Content Review
James Shubin	Configuration Management Architect	Content Review

Appendix E: Revision History

Revision 1.0 Thursday March 05, 2015 Balaji Jayavelu

- Initial release

Revision 1.1 Friday March 20, 2015 Balaji Jayavelu

- Minor updates and corrections

Revision 1.1 Friday May XX, 2015 Keith Schincke

- Updated Ceph and ICE references to Red Hat Ceph Storage
- Updated OSP network subnets and domain names
- Content rearrangement and update