Reference Architectures 2020

Deploying Red Hat OpenShift Container Platform 4.4 on Red Hat OpenStack Platform 13 and 16.0

Last Updated: 2020-05-13

Reference Architectures 2020 Deploying Red Hat OpenShift Container Platform 4.4 on Red Hat OpenStack Platform 13 and 16.0

August Simonelli
asimonel@redhat.com

Legal Notice

Abstract

The purpose of this document is to provide guidelines and considerations for deploying Red Hat OpenShift Container Platform 4.4 on Red Hat OpenStack Platform 13 and 16.0.

Table of Contents

PART I. EXECUTIVE SUMMARY

Organizations across the globe continue to rapidly develop innovative software applications in hybrid and multi-cloud environments in order to achieve competitive advantages and ensure customer satisfaction. Many of these applications are deployed in a private cloud due to security and compliance, data affinity, and performance requirements. The IT organizations responsible for operating the private cloud value simplicity, agility, flexibility, security, and cost efficiency, as these features reduce their own barriers to innovation as part of their overall hybrid and multi-cloud strategy.

This reference architecture showcases a prescriptive, pre-validated, private cloud solution from Red Hat that provides IT as a Service (ITaaS), and the rapid provisioning and lifecycle management of containerized apps, virtual machines (VMs), and associated application and infrastructure services for cloud users such as software developers, data scientists, and solution architects. Red Hat OpenShift Container Platform, Red Hat OpenStack Platform, and Red Hat Ceph Storage are the key architecture components of this solution.

This reference architecture updates Deploying Red Hat OpenShift Container Platform 3.11 on Red Hat OpenStack Platform 13 for the Red Hat OpenShift Container Platform 4.x stream. If you want to implement Red Hat OpenShift Container Platform 3.11, or are working with existing Red Hat OpenShift Container Platform 3.11 installs, then use the reference architecture for Red Hat OpenShift Container Platform 3.11, as there are significant differences in the 4.x release stream.

CHAPTER 1. ABOUT THIS DOCUMENT

This document provides an overview of the options available for implementing Red Hat OpenShift Container Platform (RHOCP) on Red Hat OpenStack Platform (RHOSP), and reviews how we implement those solutions to deploy the reference architecture within a lab-based environment.

This document is not an implementation guide. Complete, supported, and fully tested documentation exists for all the components included within this reference architecture, including:

- Red Hat OpenStack Platform 13
- Red Hat OpenStack Platform 16.0
- Red Hat OpenShift Platform 4.4
- Red Hat Ceph Storage 3
- Red Hat Ceph Storage 4

CHAPTER 2. SOLUTION OVERVIEW

2.1. TARGET USE CASES

This reference architecture is valuable for enterprise, telecommunications, government, and IT service provider organizations that want to deploy a private cloud solution with programmable Infrastructure as a Service (IaaS), Containers as a Service (CaaS), and Platform as a Service (PaaS) capabilities.

2.2. SOLUTION BENEFITS FOR IT AND BUSINESS

The key benefits and value of this solution for IT organizations and the business are as follows:

- **Faster innovation and time-to-value** The reference architecture saves the organization time as it supplements the design, deploy, and test of a full solution stack composed of several Red Hat products.
- **Simpler and safer deployments** All the desired best practices for the full solution stack are already pre-validated by Red Hat, resulting in a highly prescriptive solution to ensure success.
- **Cost efficient**: The reference architecture is based on open source technologies that are fully supported by Red Hat.

CHAPTER 3. ARCHITECTURE OVERVIEW

This is a reference architecture for running Red Hat OpenShift Container Platform 4.4 on Red Hat OpenStack Platform 13 or Red Hat OpenStack Platform 16.0.

3.1. INSTALLING RED HAT OPENSTACK PLATFORM

Red Hat OpenStack Platform (RHOSP) is deployed to physical servers using the RHOSP director. Director is a toolset for installing and managing a complete RHOSP environment from installation to Day 2 operations.

3.2. INSTALLING RED HAT OPENSHIFT CONTAINER PLATFORM 4.X

Red Hat OpenShift Container Platform (RHOCP) has a new installation program for the 4.x stream. It features a streamlined interface and simplified installation process allowing a faster, easier, and more precise installation. For more information, see the Red Hat OpenShift Container Platform 4.4 Installing guide.

The RHOCP 4 installation program offers the following types of deployment:

- **Installer-provisioned infrastructure clusters**: The RHOCP 4 installation program manages all aspects of the installation, including infrastructure provisioning, with a RHOCP best practice deployment.

- **User-provisioned infrastructure clusters**: Administrators are responsible for preparing, creating and managing their own underlying infrastructure for clusters. This approach allows greater customization prior to installing RHOCP.

Both types of clusters have the following characteristics:

- Highly available infrastructure with no single points of failure by default.

- A deep integration between RHOCP and the underlying operating system, Red Hat Enterprise Linux CoreOS (RHCOS), that provides “appliance-like” integration.

- Administrators maintain control over what updates are applied, and when.

This reference architecture features the installer-provisioned infrastructure method for installing RHOCP onto RHOSP. Following this method, the installation program creates all the networking, machines, and operating systems required when using the OpenStack APIs. This results in an architecture that is highly available, fully tested, and entirely supported, suitable for production today.

NOTE

- The new installer-provisioned infrastructure method for RHOCP is highly prescriptive, as it installs a “best practice” deployment. The infrastructure of an installer-provisioned infrastructure deployment should not be customised after deployment. Any infrastructure changes must be implemented by the installation program, which interacts directly with the underlying infrastructure and APIs. Only Day 2 infrastructure operations, such as machine scale outs, are recommended.

- For enterprises that need additional infrastructure customisations and requirements, the simplicity of the installer-provisioned infrastructure method may be limiting. In this case, the user-provisioned infrastructure method may be more appropriate.

This document describes a reference architecture suitable for the majority of RHOCP 4.4 on RHOSP use cases. The reference architecture represents the best practice for getting RHOCP on RHOSP up and running quickly and in the most reliable and supported way. It also shares important design considerations and key integrations between the products.

This reference architecture is fully supported by Red Hat.

3.3. RELATIONSHIP BETWEEN RED HAT OPENSHIFT CONTAINER PLATFORM AND RED HAT OPENSTACK PLATFORM

The relationship between Red Hat OpenShift Container Platform (RHOCP) and Red Hat OpenStack Platform (RHOSP) is complementary. RHOSP exposes resources through its Application Programming Interface (API) and RHOCP requests those resources.

RHOSP provides RHOCP with compute, storage, and networking infrastructure, plus additional resources such as self-service load balancers and encryption.

RHOCP runs its containerized applications on the infrastructure provisioned by RHOSP. The products are tightly integrated, allowing RHOCP to consume RHOSP resources on demand and without user intervention.

3.3.1. Introducing Red Hat Enterprise Linux (RHEL) CoreOS

RHOCP 4 nodes run on Red Hat Enterprise Linux CoreOS (RHCOS) . RHCOS provides over-the-air updates on a Red Hat Enterprise Linux (RHEL) kernel to deliver a secure, easily managed container host. In an installer-provisioned infrastructure deployment, and for this reference architecture, RHCOS is the supported operating system for all the RHOCP nodes, and is used by default for worker and master nodes. It is also a RHOCP requirement that the master nodes run RHCOS.

NOTE

RHCOS is currently only used with RHOCP. It is not for use as an independent operating system. For more information, see CoreOS has joined the Red Hat family .

3.3.1.1. Ignition

Ignition is a RHCOS utility that is used to manipulate disks during initial configuration. It completes common disk tasks, including partitioning disks, formatting partitions, writing files, and configuring users.

On first boot, Ignition reads the bootstrap configuration files, generated by the RHOCP installation program, from the OpenStack Image service.

For more information, see About Ignition.

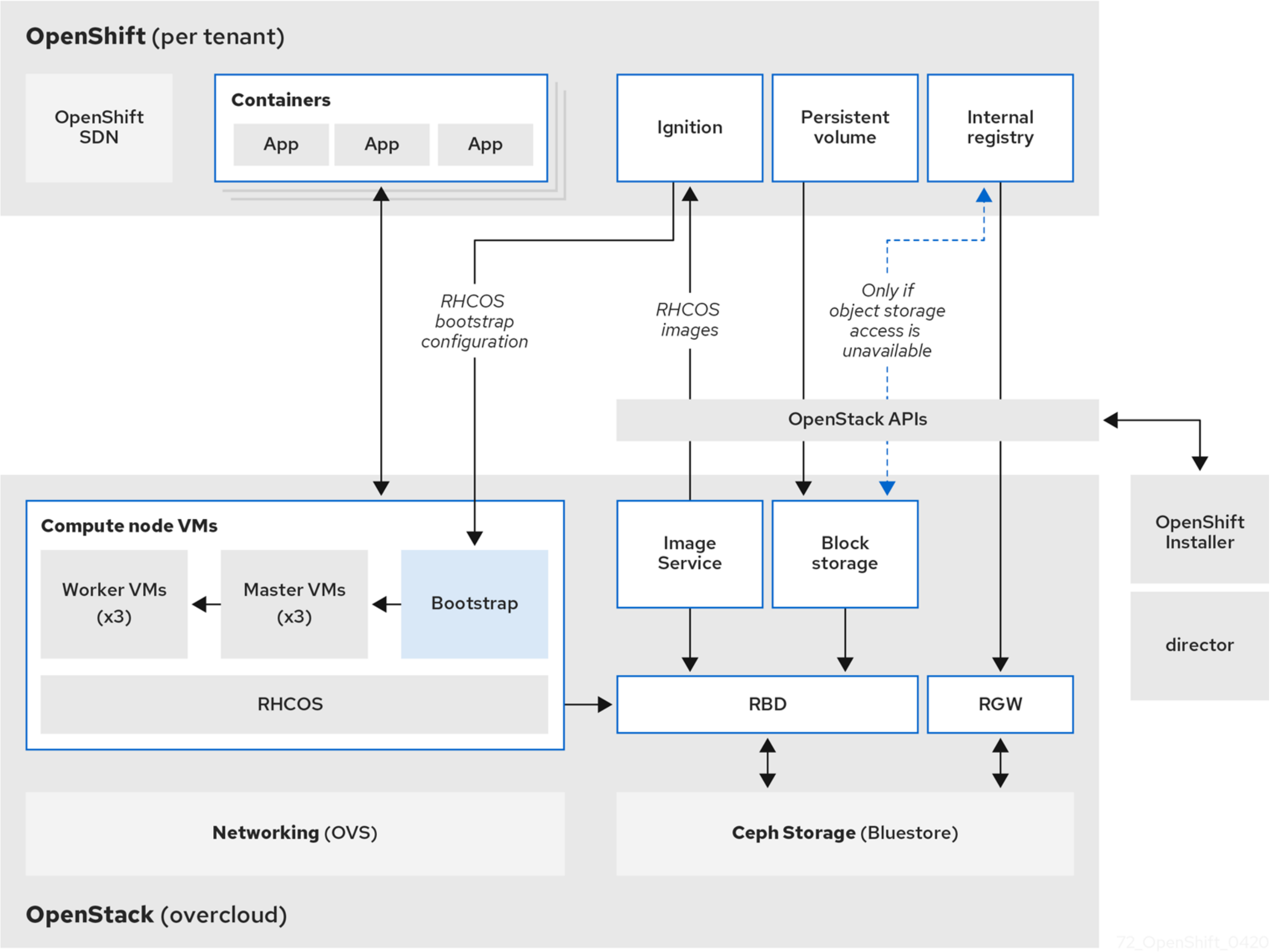
3.4. REFERENCE ARCHITECTURE HIGH LEVEL DESIGN

This reference architecture is for the following products:

Table 3.1. Reference architecture products

Product	Version used to test the deployment
Red Hat OpenStack Platform (RHOSP)	13, version used 13.0.11
	16, version used 16.0.0
Red Hat OpenShift Container Platform (RHOCP)	4.4
Red Hat Ceph Storage	3.3, as deployed by director with RHOSP 13; BlueStore enabled as the OSD back end.
	4, as deployed by director with RHOSP 16.0; BlueStore is the default OSD back end.

The following diagram provides a high level overview of the reference architecture.



The following table describes the components of this reference architecture.

Table 3.2. Reference architecture components

Component/Service	Product
Object storage	Red Hat Ceph Object Gateway (RGW) (replaces the default OpenStack Object Storage (swift))
Block storage	Red Hat OpenStack Block Storage (cinder) backed by Red Hat Ceph Block Devices (RBD)
Image storage	Red Hat OpenStack Image Service (glance) backed by Red Hat Ceph Block Devices (RBD)
Compute service	Red Hat OpenStack Compute (nova) backed by Red Hat Ceph Block Devices (RBD)
Networking service - OpenStack	Red Hat OpenStack Networking with Open vSwitch (OVS)
Networking service - OpenShift	Red Hat OpenShift software-defined networking (SDN)

Component/Service	Product
Ignition	Red Hat Enterprise Linux CoreOS (RHCOS) tool, used in the Red Hat OpenShift installation process

Key reference architecture features

- The reference architecture uses TLS for external endpoint encryption.
- A public, external network with routable IPs in a floating IP pool is available to all tenants.
- The default network policy mode is implemented for OpenShift SDN.
- You have administrative access to a DNS zone to host the required domains.

Red Hat OpenShift Container Platform deployment

RHOCP does not require administrative access to the RHOSP cloud. The RHOSP administrator prepares a suitable tenant for RHOCP. RHOCP is deployed by that tenant on a RHOSP instance that is running Red Hat Enterprise Linux CoreOS (RHCOS).

This reference architecture deploys RHOCP on instances that run the RHCOS operating system installed from a RAW image.

The Ignition installation files, generated by the installation program for creating the bootstrap node, are stored in the Red Hat OpenStack Image Service (glance). The RHOCP bootstrap node requires external DNS resolution and external Internet access to set up the RHOCP cluster.

CHAPTER 4. DESIGN CONSIDERATIONS

This section describes how this reference architecture addresses key design considerations. The recommendations in this reference architecture were developed in conjunction with Red Hat Field Consultants, Engineers, and Product teams.

4.1. CONNECTIVITY IMPLEMENTATION

You can deploy Red Hat OpenShift Container Platform (RHOCP) to different on-premises architectures.

NOTE

This reference architecture assumes a direct connection to the Internet, including full DNS resolution for director and all RHOCP nodes.

4.1.1. Deploying RHOCP with a direct Internet connection

Both the Red Hat OpenStack Platform (RHOSP) and RHOCP installation programs can take advantage of direct Internet access and an active subscription to Red Hat products. The installation processes are not self-contained and require access to external sources to retrieve core assets such as:

- RHOSP containers
- RHOCP containers
- RHOCP images

Therefore, an active, working DNS resolution is required.

4.1.2. Deploying RHOCP using a corporate proxy service

A RHOCP installer-provisioned infrastructure deployment supports the implementation of a HTTP or HTTPS cluster-wide proxy at both install time, by using the configuration file, and after an install, by using the cluster-wide egress proxy. For more information on setting up a cluster-wide proxy, see Configuring the cluster-wide proxy in the *OpenShift Container Platform Networking* guide.

4.1.3. Deploying RHOCP in a restricted network environment

While RHOCP 4 on RHOSP 13 fully supports a restricted network deployment, it requires additional steps and components not tested in this reference architecture. Before implementing this reference architecture in a restricted network environment, read Creating a mirror registry for installation in a restricted network in the *OpenShift Container Platform Installing* guide, and consult Red Hat support.

4.2. INSTALLATION METHODS AND TOOLING

It is always best to install Red Hat products and product integrations with the tools recommended and supported by Red Hat.

For this reference architecture we install Red Hat OpenStack Platform (RHOSP) by using director. The installation includes the minimum number of Controller nodes, Compute nodes, and Storage nodes required for a high-availability (HA) control plane and storage replication.

We install Red Hat OpenShift Container Platform (RHOCP) using the installer-provisioned infrastructure method. We use the guided install to generate an installation configuration file that we can modify with some basic customisations.

We run the RHOCP installation from the director host by using a non-privileged RHOSP tenant.

4.2.1. Red Hat OpenStack Platform director

RHOSP director is a management and installation tool based on the OpenStack TripleO project. TripleO stands for “OpenStack on OpenStack.” This reference architecture uses director to install RHOSP.

The fundamental concept behind director is that there are two clouds. The first cloud, called the undercloud, is a standalone RHOSP deployment. The undercloud can be deployed on a single physical server or virtual machine. The administrator uses the undercloud’s RHOSP services to define and deploy the production RHOSP cloud. Director is also used for day two management operations, such as applying software updates and upgrading between RHOSP versions.

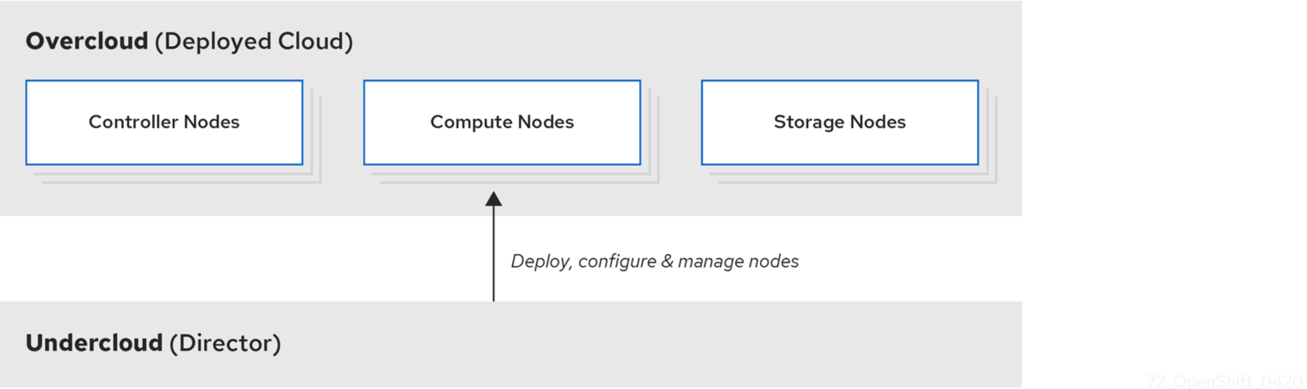
The second cloud, called the overcloud, is the full-featured production environment deployed by the undercloud. The overcloud is comprised of physical servers with various roles:

- Controller nodes run the OpenStack API endpoints. They also store RHOSP’s stateful configuration database and messaging queues.
- Compute nodes run virtual machine hypervisors. They host the computing resources allocated for user workloads.
- Storage nodes provide block, object, or software defined storage for the user workloads.

RHOCP runs within a project, or tenant, on the overcloud. Each tenant’s RHOCP cluster is isolated from other tenant clusters.

Director is required for all production RHOSP deployments. The configuration settings tested and validated by Red Hat Engineering are embedded into the deployment tooling provided by the director.

An administrator should only provision and manage their RHOSP cloud using director. Customizations should only be performed using director and never added manually.


Overcloud (Deployed Cloud)
Controller Nodes
Compute Nodes
Storage Nodes
Deploy, configure & manage nodes
Undercloud (Director)
72_OpenShift_0420

4.2.2. Red Hat OpenShift Container Platform installation program

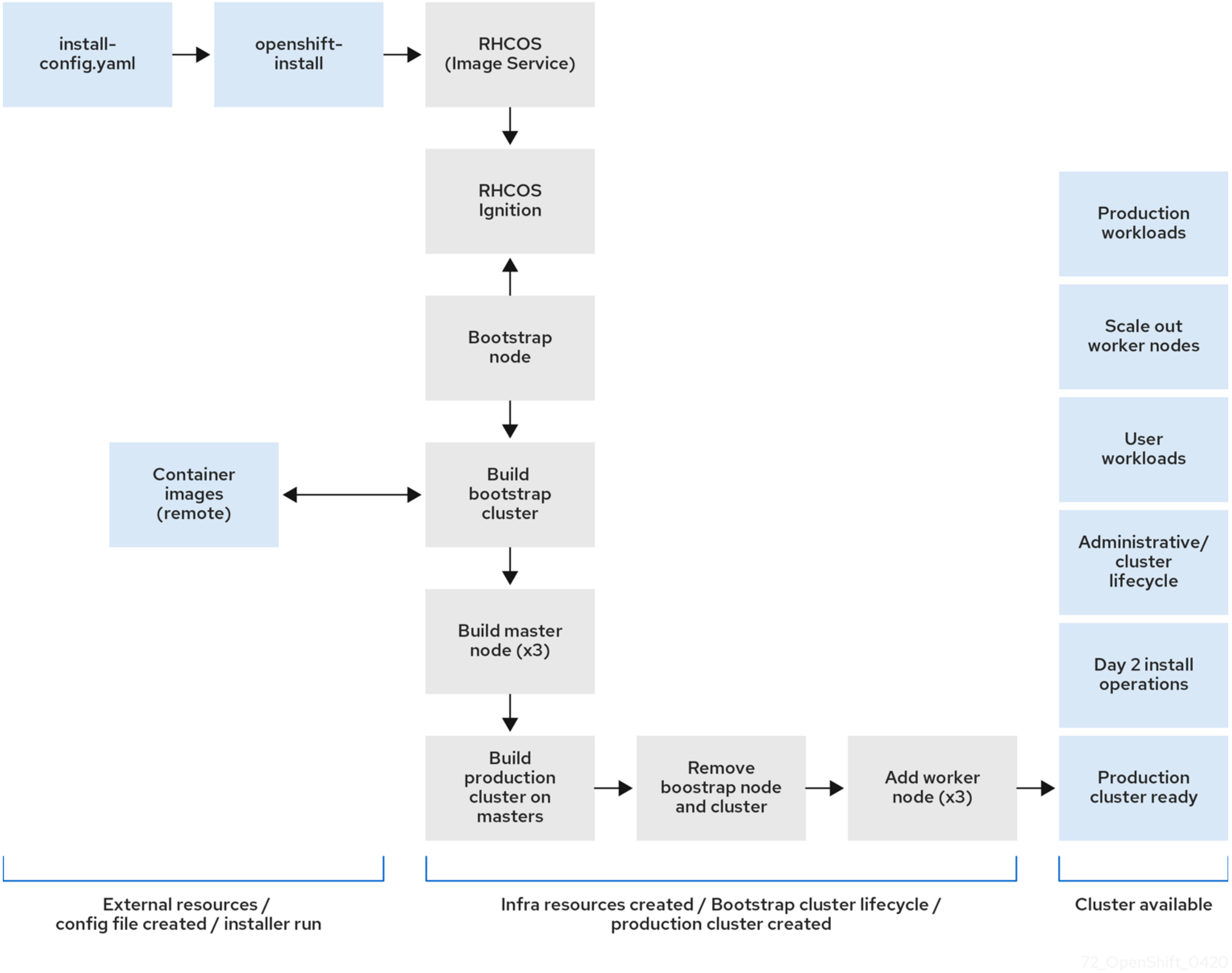
The RHOCP 4 installation program supports two methods of deployment: installer-provisioned infrastructure and user-provisioned infrastructure.

The RHOCP 4 installation program is primarily concerned with ensuring a simple path to configuring the infrastructure. Day 2 operations handle the bulk of the RHOCP configuration. To configure the infrastructure, the installation program creates a minimal RHOCP cluster.

4.2.2.1. Installer-provisioned infrastructure

With the introduction of the new RHOCP 4 installer-provisioned infrastructure deployment method, users no longer need to pre-provision RHOSP resources or describe them in Ansible playbooks.

Installer-provisioned infrastructure deployments are prescriptive and limit the amount of variance for the installation profile. This ensures an easier to manage and better supported deployment for the majority of use cases.



4.2.2.2. Generating the installation configuration file

The RHOCP installation program prompts you for your requirements, and uses your responses to create the install configuration file, **install-config.yaml**. This configuration file describes the end state of the installation and allows an operator to add additional customizations to their installation, while still remaining within the installer-provisioned infrastructure footprint.

4.3. IMPLEMENTING HIGH AVAILABILITY

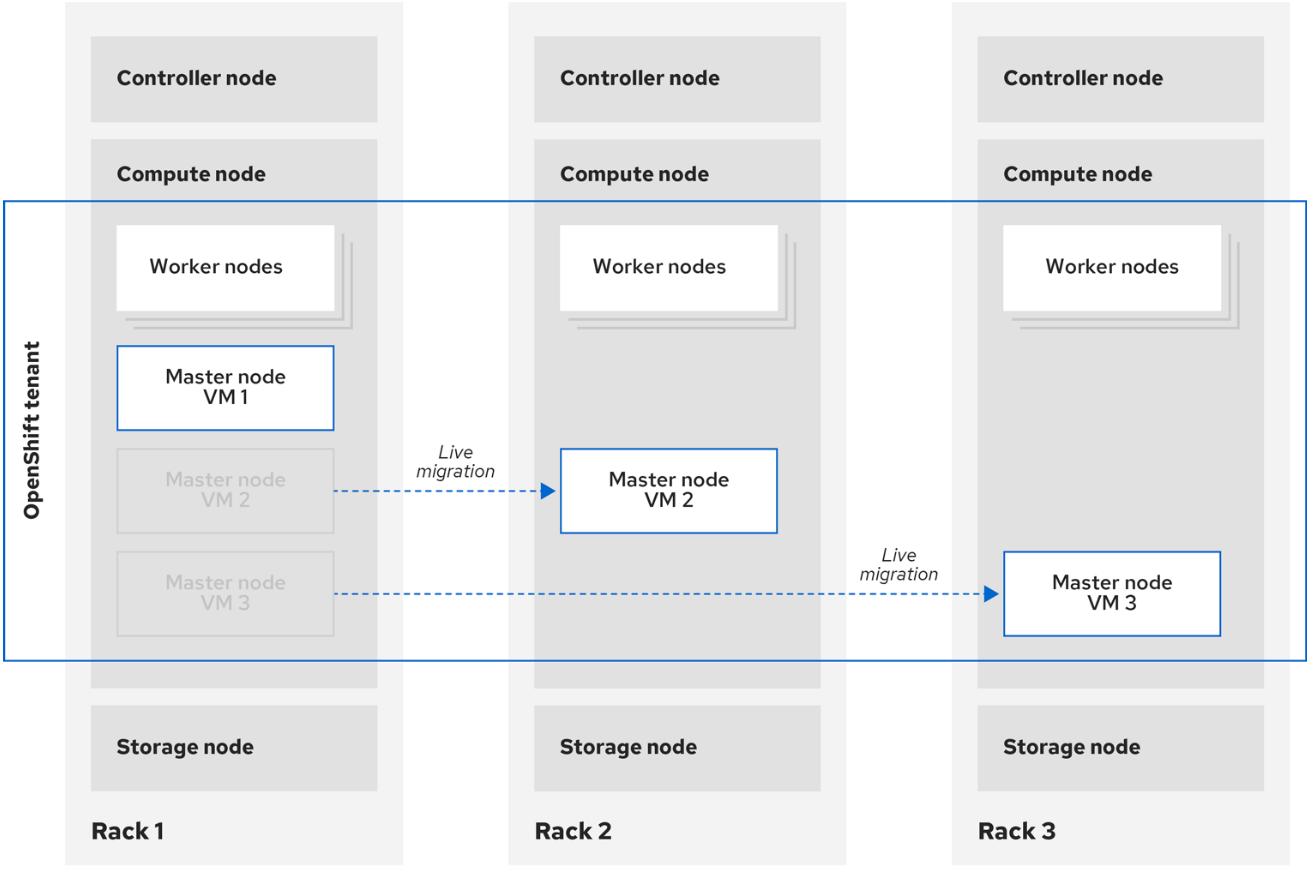
High availability (HA) is a requirement for any production deployment. A crucial consideration for HA is removing single points of failure. This reference architecture is highly available at the Red Hat

OpenStack Platform (RHOSP), Red Hat OpenShift Container Platform (RHOCP), and Red Hat Ceph Storage layers.

This reference architecture uses RHOSP director to deploy:

- three RHOSP Controller nodes
- three Red Hat Ceph Storage nodes that use BlueStore as the OSD back end
- three RHOSP Compute nodes

The following diagram shows an example of a highly available RHOCP on RHOSP deployment. Each bare-metal server contains three RHOSP nodes: Controller, Storage, and Compute. RHOCP masters and workers are distributed across the Compute nodes using live migration. NIC's are bonded and servers use RAID as required.



72_OpenShift_0520

As shown in the above diagram, you cannot place the master nodes that constitute the control plane on the same RHOSP Compute nodes. To ensure this, you must provide at least three bare-metal servers to host three distinct Compute nodes. You must use live migration to ensure masters are never hosted on the same Compute node. For more information, see the following references:

- RHOSP 13: Migrating virtual machines between Compute nodes.
- RHOSP 16.0: Migrating virtual machines between Compute nodes.

NOTE

- Live migration is an administrative function, therefore you need your cloud administrator to perform it.
- You cannot use ephemeral backing storage for master instances, as you cannot live migrate ephemeral backing storage.

This reference architecture represents a minimal starting point. It does not account for all HA possibilities. Therefore, when creating your own HA implementation consider all layers and plan accordingly. For instance, plan to have enough Compute nodes to ensure losing one does not compromise the RHOCP HA best practice of hosting master nodes on different Compute nodes.

4.3.1. RHOSP HA

The default level of HA enforced by the RHOSP director deploys three Controller nodes. Multiple RHOSP services and APIs run simultaneously on all three Controller nodes.

HAproxy load balances connections across the Controller API endpoints to ensure service availability. The Controller nodes also run the RHOSP state database and message bus. A Galera cluster protects the state database. RabbitMQ queues are duplicated across all nodes to protect the message bus.

4.3.2. RHOCP HA

RHOCP is also deployed for HA. The control plane manages the RHOCP cluster. This reference architecture colocates the **etcd** state database across the master nodes. As **etcd** requires a minimum of three nodes for HA, you must deploy three master nodes.

RHOCP hosts critical control plane components on three master nodes for HA. In RHOCP 4.x, the various infrastructure services, such as router pods, container image registry, metrics and monitoring services, and cluster aggregated logging, no longer run on separate infrastructure nodes by default. Instead, infrastructure services now run on the worker nodes. You can deploy the infrastructure services to separate infrastructure nodes on Day 2. For more information, see Creating infrastructure MachineSets.

Worker nodes are where the actual application workloads run. The masters manage the worker nodes. You should ensure that you have enough worker nodes to handle failures due to excessive load and lost infrastructure.

4.3.3. Storage HA

RHOCP and RHOSP services that use Red Hat Ceph Storage can make use of the Red Hat Ceph Storage built-in HA capabilities. By default, deploying RHOSP with Red Hat Ceph Storage results in integrated storage for Compute services, the Image Service, and Block Storage services.

You can add Ceph OSDs to increase performance and availability. Sizing and customizing your Red Hat Ceph Storage cluster requires special attention and is beyond the scope of this reference architecture. For information on customizing Red Hat Ceph Storage, see the following references:

- RHOSP 13: Customizing the Ceph Storage Cluster
- RHOSP 16.0: Customizing the Ceph Storage cluster
- Red Hat Ceph Storage 3: Red Hat Ceph Storage Hardware Selection Guide .

- Red Hat Ceph Storage 4: Hardware Guide
- Red Hat Ceph Storage: Supported configurations

Consult Red Hat support for guidance on implementing any Red Hat Ceph Storage customizations.

4.3.4. Hardware HA

You need to eliminate single points of failure at the hardware layer. Hardware fault tolerance may include the following:

- RAID on server internal hard disks.
- Redundant server power supplies connected to different power sources.
- Bonded network interfaces connected to redundant network switches.
- If the RHOSP deployment spans multiple racks, you should connect the racks to different PDUs.

A complete description of how to configure hardware fault tolerance is beyond the scope of this document.

4.4. STORAGE

Both Red Hat OpenStack Platform (RHOSP) and Red Hat OpenShift Container Platform (RHOCP) have independent and mature storage solutions. Combining these solutions can increase complexity and introduce performance overhead. You need to plan the storage carefully to avoid deploying duplicate storage components. Select storage backends that meet the needs of the RHOCP applications, while minimizing complexity.

4.4.1. Red Hat Ceph Storage

This reference architecture uses Red Hat Ceph Storage. Red Hat Ceph Storage provides scalable block, object, and file storage, and is tightly integrated with the Red Hat cloud software stack.

Red Hat Ceph Storage is a distributed data object store designed to provide excellent performance, reliability, and scalability. At the heart of every Red Hat Ceph Storage deployment is the Ceph Storage Cluster, which consists of two types of daemons:

Ceph OSD (Object Storage Daemon)

Ceph OSDs store data on behalf of Red Hat Ceph Storage clients. Ceph OSDs use the CPU and memory of Red Hat Ceph Storage nodes to perform data replication, rebalancing, recovery, monitoring and reporting functions.

Ceph Monitor

A Ceph monitor maintains a master copy of the Red Hat Ceph Storage cluster map with the current state of the storage cluster.

The RHOSP director deploys Red Hat Ceph Storage using the **ceph-ansible.yaml** environment file. This environment file automatically configures Red Hat Ceph Storage as the backing store for OpenStack Compute (nova) ephemeral disks, OpenStack Block Storage (cinder) volumes, OpenStack Image Service (glance), and OpenStack Telemetry (ceilometer).

4.4.2. Red Hat Ceph Storage backends

The version of Red Hat Ceph Storage you are using determines the storage backend to use:

- For Red Hat Ceph Storage 3.1 or earlier, use FileStore as the backend.
- For Red Hat Ceph Storage 3.2 or later, use BlueStore as the backend.

You should use BlueStore for all new deployments, as it performs best. RHOSP 13 is integrated with Red Hat Ceph Storage 3.3, and director supports the creation of OSDs with BlueStore. RHOSP 16.0 implements Red Hat Ceph Storage 4 and uses BlueStore as the backend by default. RHOSP 16.0 discontinues support for the FileStore backend. All director deployed OSD nodes must run on dedicated physical servers.

For more information, see the following references:

- RHOSP 13: Using BlueStore in Ceph 3.2 and later
- RHOSP 16.0: Using BlueStore
- Red Hat Ceph Storage 3: The ObjectStore Interface
- Red Hat Ceph Storage 4: Ceph ObjectStore

4.4.3. Persistent volumes for RHOCP

The Kubernetes persistent volume (PV) framework allows RHOCP users to request block storage volumes without any knowledge of the underlying storage.

RHOCP supports the OpenStack Block Storage service (cinder) as the provider for this functionality. A RHOCP user can use a PersistentVolume object definition to access an OpenStack Block Storage volume in RHOSP.

The RHOCP installation program automates the creation of an OpenStack Block Storage storage class for dynamic persistent volume creation using the **kubernetes.io/cinder** driver:

```
$ oc get storageclass
NAME               PROVISIONER          RECLAIMPOLICY   VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION   AGE
standard (default)   kubernetes.io/cinder   Delete          WaitForFirstConsumer   true                 3d20h
```

NOTE

The OpenStack Block Storage storage class access mode is ReadWriteOnce (**RWO**). This is sufficient for most use cases. The ReadWriteMany (**RWX**) access mode will be supported in a future reference architecture.

For more information, see Persistent storage using Cinder .

4.4.4. Object storage

Director does not consider object storage when installing RHOSP. By default, director installs OpenStack Object Storage (swift) on the Controller nodes to provide an easy-to-use, highly available object storage backend for the OpenStack Image Service (glance). This is not the preferred object storage solution for tenants to use, and access must be explicitly granted by the cloud administrator.

Instead, this reference architecture replaces the default OpenStack Object Storage with a Red Hat Ceph Object Gateway (RGW or **radosgw**), which can provide a highly available object storage backend that is compatible with both Amazon S3 and OpenStack Object Storage RESTful APIs.

4.4.4.1. RHOSP registry and object storage

The RHOCP installation program follows the recommended practice for a scaled registry and attempts to use the RHOSP object storage service as the preferred location for the internal RHOSP image registry backend. To do this, the RHOCP installation program checks for an accessible object storage location and uses it. If the RHOCP installer cannot find an accessible object storage location, it follows the recommended best practice for a non-scaled registry and creates a ReadWriteOnce (RWO) OpenStack Block Storage (cinder) volume to use instead. For more information, see Optimizing storage.

Therefore, to ensure the best possible deployment pattern for the registry, Red Hat recommends deploying RHOSP with Red Hat Ceph Object Gateway (RGW).

4.4.5. Image storage

When deploying Red Hat Ceph Storage with director, the default installation of OpenStack Image Service (glance) is backed by Red Hat Ceph Block Devices (RBD). The RHOCP installation program uses the OpenStack Image Service for two purposes:

- To store the RHOCP Ignition files used to bring up the bootstrap node and cluster.
- To store the Red Hat Enterprise Linux CoreOS (RHCOS) image.

4.5. NETWORKING

You need to plan the networking requirements of your deployment. Select networking backends that meet the needs of the applications running in the containers, while minimizing complexity. Both Red Hat OpenStack Platform (RHOSP) and Red Hat OpenShift Container Platform (RHOCP) have independent and mature networking solutions. However, naively combining the native solutions for each platform can increase complexity and unwanted performance overhead. Consider your workloads as you may want to avoid duplicate networking components. For example, consider the implications of running OpenShift SDN on top of an OpenStack SDN.

4.5.1. OpenStack Networking (neutron)

The OpenStack Networking (neutron) service provides a common abstraction layer for various backend network plugins. RHOSP 13 and 16.0 support the following backend network plugins.

OVS

Both RHOSP 13 and 16.0 deploy an Open vSwitch (OVS) backend for the OpenStack Networking service. OVS is an open-source, multi-layer software switch designed as a virtual switch for use in virtualized server environments. It is used across multiple Red Hat products. OVS is fully tested and supported.

ML2/OVS

RHOSP 13 uses the ML2 plugin by default. When using the ML2/OVS plugin, VXLAN encapsulation carries layer 2 traffic between nodes. All L3 traffic is routed through the VXLAN tunnel, to the centralized OpenStack Networking agents running on the Controller nodes.

OVS-OVN

RHOSP 16.0 uses the Open Virtual Network (OVN) component of Open vSwitch (OVS) by default.

OVN is a network virtualization solution that is built into the OVS project. OVN supports the Neutron API, and offers a clean and distributed implementation of the most common networking capabilities, such as bridging, routing, security groups, NAT, and floating IPs.
When using OVN as a backend, GENEVE (Generic Network Virtualization Encapsulation) is used for network encapsulation. The details of GENEVE are beyond the scope of this document, but key advantages of using OVN and GENEVE include increased flexibility through an extendable header format and distribution of L3 traffic by default.

OVS-OVN has been tested and is supported for RHOCP on RHOSP 16.0 deployments, including for this reference architecture when using RHOSP 16.0.

Third-party SDN solution

Multiple networking vendors support their own plugins for OpenStack Networking. For future iterations of RHOCP on RHOSP there will be increased partnerships and testing to support as many third party solutions as possible.

NOTE

RHOSP 13 and 16.0 have different default backend network plugins. This reference architecture uses the default backend for the RHOSP version implemented:

- RHOSP 13: ML2/OVS
- RHOSP 16.0: OVS-OVN

Moving from OVS to OVN on a running RHOSP 13 cloud is only supported when upgrading to RHOSP 16.0.

A provider network bridges a range of externally accessible floating IP addresses to the tenant instances. Remote clients use the floating IP addresses to access exposed RHOCP services, the OpenShift API endpoint, and the web console.

A detailed explanation of the OpenStack Networking service is beyond the scope of this document. For more information, see the following references:

- RHOSP 13: Networking Guide
- RHOSP 16.0: Networking Guide

4.5.2. Networking in Red Hat OpenShift Container Platform

Networking in Red Hat OpenShift Container Platform (RHOCP) is a complex topic and a detailed review is beyond the scope of this document. For more information, see Networking.

4.5.2.1. OpenShift SDN

The OpenShift SDN is a fully functioning, mature, and supported SDN solution for RHOCP on RHOSP. For more information, see About OpenShift SDN.

When choosing OpenShift SDN for a RHOCP on RHOSP deployment, consider the general network requirements for the workloads. When running OpenShift SDN on a standard RHOSP deployment, the network will be subject to "double encapsulation". This happens when running an encapsulated OpenShift SDN over an already encapsulated RHOSP network. This is true for both OVS and OVN implementations.

In some cases, running under double encapsulation can cause performance and complexity issues. You should therefore perform a detailed review of the requirements of your workloads to best understand the true impact of double encapsulation as it relates to your own workloads.

This reference architecture uses OpenShift SDN for RHOCP networking.

4.5.2.2. Kuryr

The Kubernetes Container Networking Interface (CNI) specification provides a mechanism for generic plugin-based networking solutions to be implemented for RHOCP application containers. The default RHOCP networking plugin can be replaced with Kuryr, a kubernetes plugin, which enables the OpenStack Networking service solution to configure networking for OpenShift's Kubernetes containers, rather than layer the two SDNs on top of each other. The Kuryr plugin makes it possible to run both RHOSP virtual machines and Kubernetes containers on the same RHOSP network. Kuryr achieves this by working with the OpenStack Networking component, along with Load Balancing as a Service (LBaaS) with Octavia. This provides combined networking for pods and services. It is primarily designed for RHOCP clusters running on RHOSP virtual machines. Kuryr aims to improve network performance and has shown positive performance results in testing. For more information, see Accelerate your OpenShift Network Performance on OpenStack with Kuryr.

Kuryr is a complex topic and beyond the scope of this document. For more information, see About Kuryr SDN.

While Kuryr is fully supported for RHOCP 4.4 on RHOSP installations, this reference architecture does not use it. When using RHOSP 13, Kuryr uses many Octavia load balancers, which adds extra overhead to the Compute nodes. You need to consider the following to determine if your installation should use Kuryr:

- Do your workloads require the benefits of removing double encapsulation?
- You need Octavia deployed to your RHOSP cloud.
- Your RHOSP infrastructure has the necessary scale to support Kuryr. For more information, see Resource guidelines for installing OpenShift Container Platform on OpenStack with Kuryr .

NOTE

RHOSP 16.0 provides a plugin for Octavia, **octavia-ovn**, which integrates Octavia with OVN allowing for reduced resource requirements when using Kuryr with RHOCP on RHOSP. A future planned supplement to this reference architecture will cover Kuryr in more detail.

4.6. DNS

The RHOCP 4 installation program greatly simplifies the DNS requirements seen in previous RHOCP on RHOSP installations. All internal DNS resolution for the cluster, certificate validation, and bootstrapping is provided through a self-hosted, installer-controlled solution that uses the mDNS plugin for CoreDNS. This solution was developed to perform DNS lookups based on discoverable information from mDNS. This plugin will resolve both the etcd-NNN records, as well as the **_etcd-server-ssl._tcp.SRV** record. It is also able to resolve the name of the nodes.

With this solution, you do not need to add the IP addresses of the master nodes, worker nodes, and the bootstrap node, either manually or dynamically, to any form of public DNS. The RHOCP on RHOSP installation is entirely self-contained in this respect.

This reference architecture uses the RHOCP installation program parameter **externalDNS** to allow the installer-built subnets to offer external DNS resolution to their instances.

NOTE

The **externalDNS** parameter did not exist for RHOCP 4.3 on RHOSP implementations. If using those versions, you need to provide cloud-wide default DNS resolution, or manually update the subnet after creating it.

For more information, see OpenStack IPI Networking Infrastructure.

With this new solution there is no need to run a self-hosted name server, which was required with the RHOCP 3.11 on RHOSP reference architecture. There is also no requirement for external solutions such as the Designate DNSaaS project.

4.6.1. DNS setup

The RHOCP on RHOSP deployment has the following DNS requirements, both before and after installation:

- The installation host must be able to resolve the OpenShift API address, for example, api.ocpra.example.com, to the RHOSP floating IP defined by using the **lbFloatingIP** parameter in **install-config.yaml**. The value you enter for **APIFloatingIPAddress** during the RHOCP 4 guided installation populates the **lbFloatingIP** parameter in **install-config.yaml**.
- The bootstrap node must be able to resolve external, public domains.
- A wild card domain must resolve an additional floating IP. That floating IP must be manually allocated to the ingress port of the installation to allow for access to deployed applications.

4.6.1.1. OpenShift API DNS

The OpenShift API floating IP address needs to be in place for installing the cluster, and to ensure the **api.<cluster name>.<base domain>.** address space resolves to it.

Assign this floating IP by using the **lbFloatingIP** parameter in the RHOSP section of **install-config.yaml**. The value you enter for **APIFloatingIPAddress** during the RHOCP 4 installation populates the **lbFloatingIP** parameter.

For example:

```
platform:
  openstack:
    cloud: openstack
    computeFlavor: m1.large
    externalNetwork: public
    lbFloatingIP: 192.168.122.152
    octaviaSupport: "1"
    region: ""
    trunkSupport: "1"
```

4.6.1.2. Application DNS

The RHOCP 4 installation program does not configure the floating IP address for the applications running on RHOCP containers. You must manually assign the floating IP address for applications to the correct port after installation.

You need a wildcard entry in your DNS for this IP to resolve the following naming structure:

***.apps.<cluster name>.<base domain>.**

4.6.1.3. Bootstrap node

The bootstrap node must be able to resolve external domain names directly. The RHOCP 4 installation program uses this name resolution to connect externally and retrieve the containers required to stand up the bootstrap cluster used to instantiate the production cluster. No other RHOCP nodes require this external resolution.

4.7. SECURITY AND AUTHENTICATION

Red Hat OpenStack Platform (RHOSP) and Red Hat OpenShift Container Platform (RHOCP) are open source enterprise software. Both support Role Based Access Control and flexible options for integrating with existing user authentication systems. Additionally, both inherit the Red Hat Enterprise Linux security features, such as SELinux.

Several RHOSP security features are used to secure RHOCP in this reference architecture.

The installer-provisioned infrastructure method creates and manages all RHOSP security groups necessary to secure a RHOSP tenant’s installation from other tenants. These groups isolate the tenant’s RHOCP install from others on the cloud.

A TLS SSL certificate is used to encrypt the OpenStack public API endpoints. This reference architecture uses a self-signed certificate and a local certificate authority on the installation host. This means that when RHOSP TLS SSL public endpoint encryption is enabled, the certificates must be imported to any hosts issuing commands against the endpoints. This reference architecture uses the director host to run the installation program, meaning all interaction with encrypted endpoints is from the director host. This follows the officially supported and documented procedures for RHOSP 13 and 16.0:

- RHOSP 13: Creating an SSL/TLS Certificate Signing Request
- RHOSP 16.0: Creating an SSL/TLS Certificate Signing Request

NOTE

At this time Internal TLS (“TLS Everywhere”) is not tested for RHOCP on RHOSP installations.

4.7.1. Authentication

By default, the RHOSP identity service stores user credentials in its state database, or it can use an LDAP-compliant directory server. Similarly, authentication and authorization in RHOSP can be delegated to another service.

RHOCP master nodes issue tokens to authenticate user requests with the API. RHOCP supports various identity providers, including HTPassword and LDAP.

Beginning with RHOCP 4.x, the RHOCP and RHOSP identity providers are integrated. Administrators can configure the RHOCP keystone identity provider with the OpenStack Identity (keystone) service. This configuration allows users to log in to RHOCP with their OpenStack Identity (keystone) credentials. For more information, see Configuring a Keystone identity provider .

4.7.2. Security

4.7.2.1. RHOSP security

The RHOSP Security and Hardening Guide recommends best practices for securing RHOSP. For more information, see the document for the version of RHOSP you are using:

- RHOSP 13: Security and Hardening Guide
- RHOSP 16.0: Security and Hardening Guide

Many of the security best practices are embedded into a default RHOSP deployment. In the field, we have secured RHOSP to various levels of standard security compliance, such as ANSSI and FedRamp. This reference architecture is not a comprehensive resource for securing RHOSP.

4.7.2.2. RHOCP security

The *OpenShift Security Guide* provides a comprehensive and detailed look into the many challenges related to security in the cloud. This guide provides assistance in the triaging of security trade-offs and risk, policy enforcement, reporting, and the validation of system configuration. The guide is available to download from the Red Hat Customer Portal .

CHAPTER 5. RESOURCE CONSIDERATIONS AND LIMITATIONS

5.1. DISK

Review the resource requirements in Resource guidelines for installing OpenShift Container Platform on OpenStack before implementation. This section highlights additional disk considerations relevant to this reference architecture.

It is essential that you ensure fast, redundant disk is available to your Red Hat OpenShift Container Platform (RHOCP) installation. To make this possible on RHOCP on RHOSP, you need to consider the minimum disk requirements for etcd, and how you are going to provide the disk to the RHOCP nodes.

5.1.1. Minimum disk requirements for etcd

As the control plane virtual machines run the etcd key-value store, they need to meet the known resource requirements. Fast disks are the most critical for etcd stability and performance. etcd is sensitive to disk write latency.

Minimum disk requirement

50 sequential IOPS, for example, a 7200 RPM disk

Minimum disk requirement for heavily loaded clusters

500 sequential IOPS, for example, a typical local SSD or a high performance virtualized block device.

For more information on the etcd disk requirements, see Hardware recommendations - Disks .

For this reference architecture, all masters need access to fast disk (SSD or better) to ensure stability and guarantee supportability. Contact your Red Hat account team for assistance on reviewing your disk performance needs.

5.1.2. Providing disk to RHOCP nodes

The full range of options for providing storage to your cluster is outside the scope of this document. The following sections provide a simple analysis of the most popular options.

5.1.2.1. Ephemeral on Compute nodes

This is the easiest way to provide disk to the masters as there is no additional configuration of RHOSP or RHOCP. All the instance volumes are stored directly on the disk local to the Compute service. As long as that disk is fast, SSD or better, the requirement for fast disk is fulfilled.

However, while etcd does have resiliency built into the software, choosing to use local Compute disks means any loss of the underlying physical node destroys the instance and etcd member, which removes a layer of protection.

5.1.2.2. Ceph-backed ephemeral

This is the default configuration for a director deployed Red Hat Ceph Storage environment, using the configuration file **/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible.yaml**. This sets **NovaEnableRBDBackend** to **True**, ensuring that the Compute service uses Red Hat Ceph Block Devices (RBD) for instances. The volumes are still ephemeral, but are sourced from the Red Hat Ceph Storage cluster. The cluster itself can then provide resilience. You can also build the Red Hat Ceph Storage cluster on fast disks to meet this requirement. The Red Hat Ceph Storage cluster may also include tiering options, as described in the following documents:

- RHOSP 13: Deploying second-tier Ceph storage on OpenStack .
- RHOSP 16.0: Deploying second-tier Ceph storage on OpenStack .

Using this method allows for added resilience for volumes. Additionally, instances backed with Red Hat Ceph Storage can be live migrated, which provides resilience and ensures correct RHOCP master node placement across the Compute nodes.

This reference architecture uses the Ceph-Backed Ephemeral default provided. This allows the Red Hat Ceph Storage cluster to be properly configured to allow for the fast disk requirements of etcd.

5.1.2.3. Volumes provided by Red Hat OpenStack Block Storage

Red Hat OpenStack Block Storage (cinder) provides an API front end that uses multiple backend storage providers, including Red Hat Ceph Storage. Compute node instances can request a volume from OpenStack Block Storage, which in turn requests the storage from the backend plugin.

RHOSP deployments with Red Hat Ceph Storage create a default setup that can be used with the RHOCP installation program to provide RHOCP nodes with OpenStack Block Storage (cinder) backed by Red Hat Ceph Block Devices (RBD). This is achieved by using machine pools. You can use machine pools to customize some aspects of each node type in an installer-provisioned infrastructure.

For example, a default RHOSP 13 installation provides the following preconfigured OpenStack Block Storage volume type:

```
$ openstack volume type list
+--------------------------------------+---------+-------------------+
| ID                                   | Name    | Is Public         |
+--------------------------------------+---------+-------------------+
| 37c2ed76-c9f7-4b0d-9a35-ab02ca6bcbb2 | tripleo | True              |
+--------------------------------------+---------+-------------------+
```

This pool can then be manually added to a machine pool subsection of the **install-config.yaml** file for the node type. In the following example, we request that the RHOCP installation program create a 25GB root volume on all masters.

```
controlPlane:
  hyperthreading: Enabled
  name: master
  platform:
    openstack:
      type: m1.large
      rootVolume:
        size: 25
        type: tripleo
  replicas: 1
```

The RHOCP installation program creates the volume for the RHOSP instance on deployment:

```
$ openstack volume list
---------------------------------------------------------------------------------------------------------------------
```

```
| ID                                   | Name                  | Status | Size | Attached to                                  |
------------------------------------------------------------------------------------------------------------------------------
| b555cc99-3317-4334-aa19-1184a78ee3f8 | ostest-8vt7s-master-0 | in-use |   25 | Attached to ostest-8vt7s-master-0 on /dev/vda  |
-----------------------------------------------------------------------------+-----------------------------------------------
---
```

The instance is built and the volume is used by the instance:

```
[core@ostest-8vt7s-master-0 ~]$ df -h
Filesystem                            Size  Used Avail Use% Mounted on
devtmpfs                              7.8G     0  7.8G   0% /dev
tmpfs                                 7.9G   84K  7.9G   1% /dev/shm
tmpfs                                 7.9G   18M  7.9G   1% /run
tmpfs                                 7.9G     0  7.9G   0% /sys/fs/cgroup
/dev/mapper/coreos-luks-root-nocrypt   25G   19G  6.2G  75% /sysroot
none                                  2.0G  435M  1.6G  22% /var/lib/etcd
/dev/vda1                             364M   84M  257M  25% /boot
/dev/vda2                             127M  3.0M  124M   3% /boot/efi
tmpfs                                 1.6G  4.0K  1.6G   1% /run/user/1000
```

Using OpenStack Block Storage for volume backends, combined with the installer-provisioned infrastructure method, allows the easy use of flexible storage backends for all installer-provisioned infrastructures. Using OpenStack Block Storage, administrators can create complex and refined tiers (classes) of storage to present to the different nodes. For example, you can set different values for the machine pools for workers and masters:

```
compute:
- hyperthreading: Enabled
  name: worker
  platform:
    openstack:
      type: m1.large
      rootVolume:
        size: 25
        type: tripleo-ssd
  replicas: 3
controlPlane:
  hyperthreading: Enabled
  name: master
  platform:
    openstack:
      type: m1.large
      rootVolume:
        size: 25
        type: tripleo-nvme
  replicas: 3
```

5.2. LIMITATIONS

This reference architecture is intentionally opinionated and prescriptive. You should only customise your deployment by using the **install-config.yaml** file, and not add additional infrastructure customization manually after deployment.

This reference architecture does not attempt to handle all uses for RHOCP on RHOSP. You should consider your individual requirements and undertake a detailed analysis of the capabilities and restrictions of an installer-provisioned infrastructure-based installation.

The following sections detail the FAQs our field teams have received around limitations when using an installer-provisioned infrastructure-based installation. This list is not conclusive and may change over time. Contact Red Hat Support or your Red Hat Specialized Solutions Architect before proceeding with the implementation of this reference architecture if you have questions around specific functionalities.

5.2.1. Internal TLS (TLS Everywhere) with Red Hat Identity Management

This reference architecture has not been tested with the steps documented in the following procedures:

- RHOSP 13: Enabling SSL/TLS on Internal and Public Endpoints with Identity Management .
- RHOSP 16.0: Enabling SSL/TLS on Overcloud Public Endpoints

Contact Red Hat Support if you want to use this functionality with RHOCP on RHOSP.

5.2.2. RHOCP installer-provisioned infrastructure subnets

Sizing

With the installer-provisioned infrastructure method, the network ranges presented within the networking section should not be edited from their default subnet values. Future changes to the networking structure for IPI will allow more flexibility for this.

MTU

The installer-provisioned infrastructure method creates all networks required by the installation. This means that modifying those networks with special settings is not possible unless done manually after the installation.
Settings for values such as MTU must be set from the RHOSP deployment as described in the RHOSP Networking Guide:

- RHOSP 13: Configure maximum transmission unit (MTU) settings.
- RHOSP 16.0: Configure maximum transmission unit (MTU) settings

5.2.3. ReadWriteMany (RWX) PersistentVolumes (PVs)

When deploying RHOCP on RHOSP there is no support for ReadWriteMany (RWX) PersistentVolumes. An RWX volume is a volume that can be mounted as read-write by many nodes. For more information, see Understanding persistent storage .

While there are many PersistentVolume plugins for RHOCP, the only one currently tested for RHOCP on RHOSP is Red Hat OpenStack Block Storage (cinder). OpenStack Block Storage provides a convenient front end for many different storage providers. Using OpenStack Block Storage ensures that the underlying storage requests are managed using the OpenStack APIs, like all other cloud resources. This provides consistent management for infrastructure administrators across all elements of the on-demand infrastructure, and allows easier management and tracking of resource allocations.

OpenStack Block Storage does not support RWX, therefore RHOCP 4.4 on RHOSP deployments are not be able to offer RWX PersistentVolumes. Support for RWX Persistent Volumes is due in a future release of RHOCP on RHOSP through the OpenStack Shared-Filesystems-as-a-Service (manila) CSI plugin.

5.2.4. Red Hat OpenShift Container Storage 4

Red Hat OpenShift Container Storage is persistent software-defined storage, integrated with and optimized for RHOCP. OpenShift Container Storage offers storage to a RHOCP installation through a containerized solution that is run within RHOCP directly.

For RHOCP, OpenShift Container Storage provides a fully self-contained containerized Red Hat Ceph Storage deployment by using underlying storage to create a unique Red Hat Ceph Storage cluster for the RHOCP installation it runs within. OpenShift Container Storage uses the upstream Rook-Ceph operator to do this.

At the time of this writing, using OpenShift Container Storage in a RHOCP on RHOSP environment is not supported. Support for this scenario is planned for a future release of OpenShift Container Storage.

CHAPTER 6. REFERENCE ARCHITECTURE IMPLEMENTATION

This section describes the deployed reference architecture.

NOTE

The reference architecture does not include step-by-step instructions for deploying Red Hat Openshift Container Platform (RHOCP) 4.4 on Red Hat Openstack Platform (RHOSP) 13 or 16.0.

For detailed installation steps, see Installing on OpenStack.

For simplicity, this reference architecture uses director stack user to host the RHOCP installation. All files and actions described in the following section were performed on this host. Unlike the RHOCP 3.11 reference architecture, a dedicated installation/bastion host is not required.

6.1. RED HAT OPENSTACK PLATFORM INSTALLATION

This reference architecture uses the 13.0.11 and 16.0 releases of Red Hat OpenStack (RHOSP). For RHOSP 13 deployments, the specific 13.0.11 maintenance release is required as it includes the required enhancements to Red Hat Ceph Object Gateway (RGW), the OpenStack Networking (neutron) service, and the OpenStack Image Service (glance), for use with installer-provisioned infrastructure deployments. RHOSP 16.0 includes all these enhancements as default.

You can check your release version by viewing the release file on director and overcloud hosts:

```
(overcloud) [stack@undercloud ]$ cat /etc/rhosp-release
Red Hat OpenStack Platform release 13.0.11 (Queens)
```

6.1.1. RHOSP deployment

The overcloud deployment consists of:

- three monolithic Controller nodes without custom roles
- three Compute nodes
- three SSD-backed storage nodes running Ceph.

Endpoints

The Public API endpoint is created using Predictable VIPs (PublicVirtualFixedIPs). For more information, see the following documents:

- RHOSP 13: Assigning Predictable Virtual IPs .
- RHOSP 16.0: Assigning Predictable Virtual IPs .

The endpoint does not use a DNS hostname to access the overcloud through SSL/TLS, as described in the following documents:

- RHOSP 13: Configuring DNS Endpoints.
- RHOSP 16.0: Configuring DNS Endpoints.

SSL/TLS

To implement external TLS encryption, use a self-signed certificate and a certificate authority on the director host. This reference architecture follows the steps in the following procedures to create a self-signed certificate and a certificate authority file called **ca.crt.pem**:

- RHOSP 13: Enabling SSL/TLS on Overcloud Public Endpoints.
- RHOSP 16.0: Enabling SSL/TLS on Overcloud Public Endpoints

The RHOCP installation program requires these certificates in front of IP-based endpoints to be part of the certificate's Subject Alternative Name (SAN).

The reference architecture also adds the **ca.crt.pem** file to the local CA trust on the director host, as described in the following reference:

- RHOSP 13: Adding the Certificate Authority to Clients
- RHOSP 16.0: Adding the Certificate Authority to Clients

This allows the undercloud to communicate with the overcloud endpoints' self-signed certificate, and to allow the RHOCP installation program to share the private CA with the necessary components of RHOCP during installation.

Storage

Director deploys Red Hat Ceph Storage using the configuration file **/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible.yaml**. This default configuration sets Red Hat Ceph Storage as the backend for the following services:

- OpenStack Image Service (glance)
- OpenStack Compute (nova)
- OpenStack Block Storage (cinder)

This reference architecture deploys Red Hat Ceph Storage across three dedicated, entirely SSD-backed Storage nodes. This is done to ensure that any storage provided to RHOCP nodes in this reference architecture is guaranteed to be fast without any extra configuration requirements. As mentioned earlier, your storage set up may differ and should be considered based on individual requirements and your unique hardware.

This reference architecture deploys Red Hat Ceph Storage with BlueStore as the OSD backend. Red Hat recommends using Red Hat Ceph Storage with BlueStore as the OSD backend for all new deployments of RHOSP 13 using Red Hat Ceph Storage 3.3 and later. Red Hat Ceph Storage with BlueStore as the OSD backend is the default for RHOSP 16.0 using Ceph 4.x, therefore no specific actions are required.

Object storage

This reference architecture uses the Red Hat Ceph Object Gateway (RGW) for object storage, which is backed by Red Hat Ceph Storage. This reference architecture deploys RGW with the following default template provided by director, with no customizations:

/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-rgw.yaml

Using this template automatically disables the default OpenStack Object Storage (swift) installation on the Controller nodes.

A RHOSP cloud administrator must explicitly allow access to Object Storage. To allow access to RGW, the administrator grants the tenant the “Member” role.

Roles

This reference architecture does not use custom roles.

Network

This reference architecture uses the standard networking isolation provided by using **/usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml** configuration file. This configuration creates the following five default networks: Storage, StorageMgmt, InternalApi, Tenant, and External. This reference architecture does not create any additional networks.

Director deploys the default Open vSwitch/OVS plugin backend.

The external network is a provider network offering a range of routable addresses that can be added to DNS and are accessible from a client web browser.

This reference architecture uses a HostnameMap and pre-assigns IPs for all node types by using the configuration file **ips-from-pool-all.yaml**.

Image storage

This reference architecture uses Red Hat Ceph Storage as the backend for the OpenStack Image Service (glance). This allows images to be stored on redundant, fast storage, that provides copy-on-write cloning (CoW) for faster boot and optimal storage.

However, this also means that using QCOW2 formatted images is not advised. For more information, see Converting an image to RAW format . Instead, to boot VMs from an ephemeral back end or from a volume, the image format must be RAW.

The RHOCP installation program automatically downloads and uses a publicly available QCOW2-formatted image. You can change this by setting the **clusterOSImage** installation variable to the URL of an external RAW-formatted image, or to the name of an existing, pre-deployed RAW-formatted image already stored in the OpenStack Image Service.

The **clusterOSImage** variable is not available from the guided installation. You must manually add it to the **install-config.yaml** file.

6.1.2. Preparing the environment

We performed the following actions to prepare the environment after overcloud deployment.

6.1.2.1. RHOSP administration

The tasks described in this section are for illustrative purposes only, to allow both RHOSP administrators and RHOCP deployers and administrators to understand all elements of the reference architecture. The steps were performed on a RHOSP 13 install. There is no requirement for administrative access to a RHOSP cloud to deploy and run RHOCP on it.

Create the public network

This reference architecture created an external flat network to provide external access and routable floating IPs:

```
$ openstack network create public --external --provider-network-type flat --provider-physical-network
```

```
datacentre
$ openstack subnet create --dhcp --gateway 192.168.122.1 --network public --subnet-range
192.168.122.0/24 --allocation-pool start=192.168.122.151,end=192.168.122.200 public
```

This network is the source for IPs to use for the API, the applications, and the bootstrap virtual IPs (VIPs), and must be accessible by cloud tenants.

Create the flavor

This reference architecture created a suitable flavor to align with the minimum requirements detailed in the Resource guidelines for installing OpenShift Container Platform on OpenStack.

```
$ openstack flavor create --ram 16384 --disk 25 --vcpu 4 --public m1.large
```

TIP

Each node type can be configured by using a custom machine pool. You can use machine pools to set different flavors per node type by setting the **type: value** in the **openstack:** section of the node machine pool in **install-config.yaml**. For more information, see Custom machine pools.

Create the user and project

This reference architecture created a simple project (tenant) called "shiftstack", and a user named "shiftstack_user".

```
$ openstack project create shiftstack
$ openstack user create --password 'redhat' shiftstack_user
```

Add the tenant to a basic role to allow them to use the cloud:

```
$ openstack role add --user shiftstack_user --project shiftstack _member_
```

NOTE

RHOCP projects and users do not need the admin role to access your cloud.

Grant access to tenants to use object storage

The cloud administrator for this reference architecture grants access to tenants to use Red Hat Ceph Object Gateway (RGW) by granting the "Member" role. The "Member" role is a special role created by the Red Hat Ceph Storage installation specifically for granting access to RGW.

NOTE

“Member” for Ceph and **_member_** for the shiftstack_user are different, distinct roles and both are required for our purposes.

```
$ openstack role add --user shiftstack_user --project shiftstack Member
```

Quotas

This reference architecture changed the default quotas to meet the resource requirements of the RHOCP nodes. Each RHOCP node requires 4 VCPUs and 16 GB RAM.

```
$ openstack quota set --cores 28 --ram 120000 shiftstack
```

The reference architecture now has a RHOSP user and project with the ability to save Ignition files in the RGW object store, and use a public, external provider network with routable floating IPs and plenty of resources available.

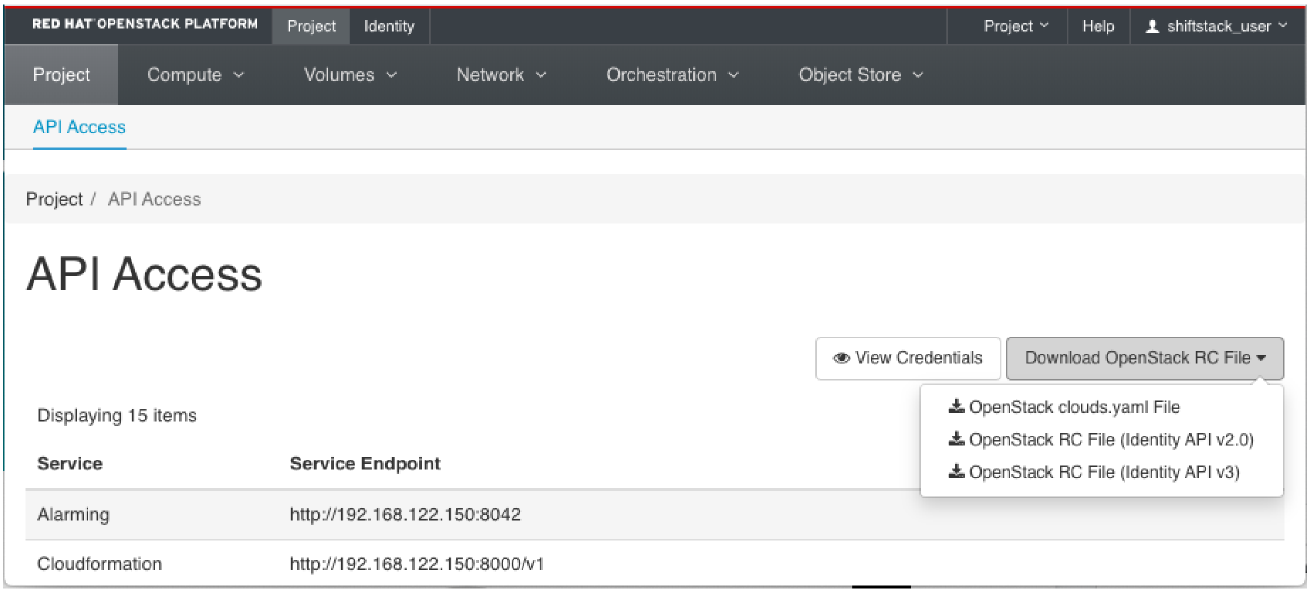
6.2. RHOCP TENANT OPERATIONS

Before starting the RHOCP installation program, the shiftstack_user performs the following tasks from the director host, logged in as the stack user.

Download the OpenStack RC file

An OpenStack RC file is an environment file that sets the environment variables required to use the RHOSP command line clients.

The reference architecture uses the shiftstack_user's RC file downloaded from the OpenStack Dashboard (horizon) GUI and placed on the deployment host. The file was not modified in any way.


RED HAT OPENSTACK PLATFORM
Project
Identity
Project
Help
shiftstack_user
Project
Compute
Volumes
Network
Orchestration
Object Store
API Access
Project / API Access
API Access
View Credentials
Download OpenStack RC File
OpenStack clouds.yaml File
OpenStack RC File (Identity API v2.0)
OpenStack RC File (Identity API v3)

Displaying 15 items

Service	Service Endpoint
Alarming	http://192.168.122.150:8042
Cloudformation	http://192.168.122.150:8000/v1

Download and modify clouds.yaml

The **clouds.yaml** file contains the required configuration for connecting to one or more clouds. Download it from the same location as the OpenStack RC file.

The default location for **clouds.yaml** is **~/.config/openstack/**, and later versions of RHOSP create the file there during installation. This reference architecture places **clouds.yaml** in the home directory of the stack user on the director host, in the same location as the OpenStack RC file. You can place the file in alternative locations, for more information, see OpenStack Credentials.

For more information, see clouds.yaml.

This reference architecture makes the following modifications to the **clouds.yaml** file:

- Adds a password, as the installation program requires a password and one is not present in the downloaded file:

  ```
  password: "redhat"
  ```

- Adds support for the self-signed external TLS, by adding the Certificate Authority file, **ca.crt.pem**:

  ```
  cacert: /home/stack/ssl/ca.crt.pem
  ```

  This change is necessary before running the RHOCP installation program, as the RHOCP installation program uses this configuration to:

  - Interact with the RHOSP TLS-enabled endpoints directly from the host it is running from.
  - Share the self-signed Certificate Authority with some of the cluster operators during the installation.

> **NOTE**
>
> RHOCP tenants must run the RHOCP installation program from a host that has had the certificate authority file placed, as described in the following references:
>
> - RHOSP 13: Adding the Certificate Authority to Clients
> - RHOSP 16.0: Adding the Certificate Authority to Clients

The following example shows the edited **clouds.yaml** file:

```
# This is a clouds.yaml file, which can be used by OpenStack tools as a source
# of configuration on how to connect to a cloud. If this is your only cloud,
# just put this file in ~/.config/openstack/clouds.yaml and tools like
# python-openstackclient will just work with no further config. (You will need
# to add your password to the auth section)
# If you have more than one cloud account, add the cloud entry to the clouds
# section of your existing file and you can refer to them by name with
# OS_CLOUD=openstack or --os-cloud=openstack
clouds:
  openstack:
    auth:
      auth_url: http://192.168.122.150:5000/v3
      username: "shiftstack_user"
      password: "redhat"
      project_id: 1bfe23368dc141068a79675b9dea3960
      project_name: "shiftstack"
      user_domain_name: "Default"
    cacert: /home/stack/ssl/ca.crt.pem
    region_name: "regionOne"
    interface: "public"
    identity_api_version: 3
```

Reserve floating IPs

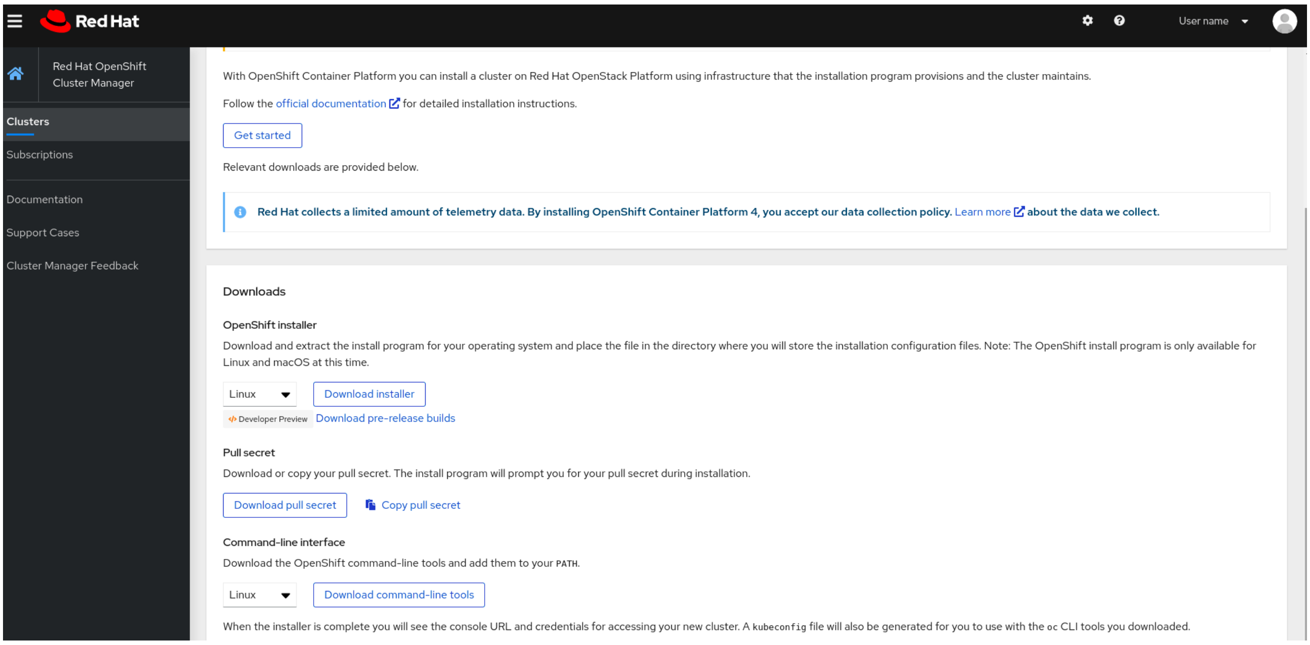
This reference architecture reserved two floating IPs from the global pool:

- 192.168.122.167 for ingress (Apps)
- 192.168.122.152 for API access

The RHOCP installation program also needs a third floating IP for the bootstrap machine. However, since the bootstrap host is temporary, this floating IP does not need to be pre-allocated or in the DNS. Instead, the installation program chooses an IP from the floating pool and allocates that to the host to allow access for troubleshooting during the installation. Once the bootstrap is destroyed, the IP is returned to the pool.

Obtain the installation program and pull secret

This reference architecture uses the downloaded RHOCP installation program, client, and pull secret from the Red Hat OpenShift Cluster Manager . You can also download the latest images, installation program, and client from the Red Hat Customer Portal .



With OpenShift Container Platform you can install a cluster on Red Hat OpenStack Platform using infrastructure that the installation program provisions and the cluster maintains.

Follow the official documentation for detailed installation instructions.

Relevant downloads are provided below.

Red Hat collects a limited amount of telemetry data. By installing OpenShift Container Platform 4, you accept our data collection policy. Learn more **about the data we collect.**

Downloads

OpenShift installer

Download and extract the install program for your operating system and place the file in the directory where you will store the installation configuration files. Note: The OpenShift install program is only available for Linux and macOS at this time.

Pull secret

Download or copy your pull secret. The install program will prompt you for your pull secret during installation.

Command-line interface

Download the OpenShift command-line tools and add them to your `PATH`.

When the installer is complete you will see the console URL and credentials for accessing your new cluster. A `kubeconfig` file will also be generated for you to use with the `oc` CLI tools you downloaded.

NOTE

New clusters are automatically registered with a 60-day evaluation subscription. Evaluation subscriptions do not include support from Red Hat. For non-evaluation use, you should attach a subscription that includes support. For more information, see OpenShift Container Platform 3 to 4 oversubscriptions during cluster migration explained.

6.3. RED HAT OPENSHIFT CONTAINER PLATFORM INSTALLATION

Before running the installation program, create a directory to store the **install-config.yaml**, and a directory to store the cluster assets. For example, the following two directories are for the "ocpra" cluster:

- "ocpra-config": Stores the “master copy” of **install-config.yaml**.
- "ocpra": The asset directory of the cluster.

The installation program can manage multiple clusters, therefore use unique directories for each cluster to keep their asset files separate.

This reference architecture uses the guided installation to generate an **install-config.yaml** file directly into the **ocpra-config** directory:

```
$ openshift-install --dir=ocpra-config create install-config
? SSH Public Key /home/stack/.ssh/id_rsa.pub
? Platform openstack
? Cloud openstack
? ExternalNetwork public
? APIFloatingIPAddress 192.168.122.152
? FlavorName m1.large
? Base Domain example.com
? Cluster Name ocpra
? Pull Secret [? for help] *****************************************
```

The guided installation populates the following variables:

- **SSH Public Key**: The specified key for all RHOCP nodes. It is saved under the “core” user (connect as core@IP). This example uses the public key of the stack user. For production deployments you need to provide separate keys in line with good security practices and your company’s security policies.

- **Platform**: The cloud provider platform you are installing RHOCP on to. This reference architecture installs on “openstack”.

- **Cloud**: This is the cloud you want to use, as defined in **clouds.yaml**.

- **ExternalNetwork**: The network defined with the “--external” flag. The installation program presents available options to choose from.

- **APIFloatingIPAddress**: The floating IP designated for the API. The installation program presents all allocated floating IPs in the project to choose from. This reference architecture allocates 192.168.122.152. This Floating IP is attached to the load balancer (haproxy) in front of the cluster. It is not a physical node.

- **FlavorName**: The flavor to use for all instances. This reference architecture uses the **m1.large** flavor created earlier with the required resource limits.

- **Base Domain**: The base domain name for the cluster. This reference architecture uses **example.com**.

- **Cluster Name**: The name of the cluster. This name is appended as a prefix to the Base Domain name as **<clustername>.<basedomain>**. This reference architecture uses “ocpra”.

- **Pull Secret** Your unique value, copied from the "Pull Secret" section on https://cloud.redhat.com/openshift/install/openstack/installer-provisioned.

TIP

You can regenerate the **install-config.yaml** for a running cluster by running the following command:

```
$ openshift-install --dir=<asset directory of the cluster> create install-config
```

For example:

```
$ openshift-install --dir=ocpra create install-config
```

Customize install-config.yaml

Using the guided installation creates a fully supported production cluster. However, you can still manually add some specific additional values to **install-config.yaml** and create an opinionated, supported, production-ready deployment. This reference architecture adds the following values to the **platform: openstack:** section:

- **externalDNS**: The RHOSP cloud this reference architecture creates does not provide a DNS by default to tenant-created subnets. Instead, this reference architecture manually sets the **externalDNS** value to allow the installer to automatically add a specific DNS to the subnet it creates.

- **clusterOSImage**: This reference architecture sets this value to a URL pointing to a RAW-formatted version of the RHCOS QCOW2 image. This overrides the use of the default QCOW2 downloaded by the installation program, which is not suitable for the Ceph backend. This RAW image is hosted on an internal webserver. For more information, see Converting an image to RAW format.

NOTE

Due to an open issue the image has a QCOW label when uploaded by the RHOCP installation program, and it returns the “disk_format” field as “qcow2”. This is incorrect. To be sure the image uploaded is RAW you can review the actual image uploaded by using the cached file that the installation program uploaded from:

```
$ qemu-img info ~/.cache/openshift-
installer/image_cache/7efb520ee8eb9ccb339fa223329f8b69
image: /home/stack/.cache/openshift-
installer/image_cache/7efb520ee8eb9ccb339fa223329f8b69
file format: raw
virtual size: 16G (17179869184 bytes)
disk size: 16G
```

The resulting **install-config.yaml** file for this reference architecture is as follows:

```
$ cat ocpra-config/install-config.yaml
apiVersion: v1
baseDomain: example.com
compute:
- hyperthreading: Enabled
  name: worker
  platform: {}
  replicas: 3
controlPlane:
  hyperthreading: Enabled
  name: master
  platform: {}
  replicas: 3
metadata:
  creationTimestamp: null
  name: ocpra
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineCIDR: 10.0.0.0/16

  # The networkType is set by default to OpenShiftSDN, even if Octavia
  # is detected and you plan to use Kuryr. Set to "Kuryr" to use Kuryr.
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  openstack:
    cloud: openstack
    computeFlavor: m1.large
    externalDNS: ['8.8.8.8']
    externalNetwork: public

    # lbFloatingIP is populated with the APIFloatingIPAddress value
    # specified in the guided installation.
    lbFloatingIP: 192.168.122.152

    # octaviaSupport is set to '0' by the installation program
    # as Octavia was not detected.
    octaviaSupport: "0"
    region: ""
    trunkSupport: "1"
    clusterOSImage: http://10.11.173.1/pub/rhcos-4.4.0-rc.1-x86_64-openstack.x86_64.raw
publish: External
pullSecret: ‘xxx'
sshKey: |
  ssh-rsa AAAAB3NzaC1yc2EsdfsdfsdsfewX0lTpqvkL/rIk5dGYVZGCHYse65W8tKT...
stack@undercloud.redhat.local
```

NOTE

The installation program defaults both the workers and masters **replica** count to “3”. Production installations do not support using more or less than three masters, as HA and over-the-air updates require three masters.

Prepare the DNS settings

This reference architecture prepared the DNS entries as follows:

- **api.ocpra.example.com** resolves to 192.168.122.152
- ***.apps.ocpra.example.com** resolves to 192.168.122.167

Specific DNS server administration is beyond the scope of this document.

Install RHOCP

Prior to installation, we copied the **install-confg.yaml** file to the asset directory for installation.

```
$ cp ocpra-config/install-config.yaml ocpra/
```

We are now ready to run the RHOCP installation program, specifying the asset directory containing the copy of the **install-config.yaml** file:

```
$ openshift-install --dir=ocpra create cluster --log-level=debug
```

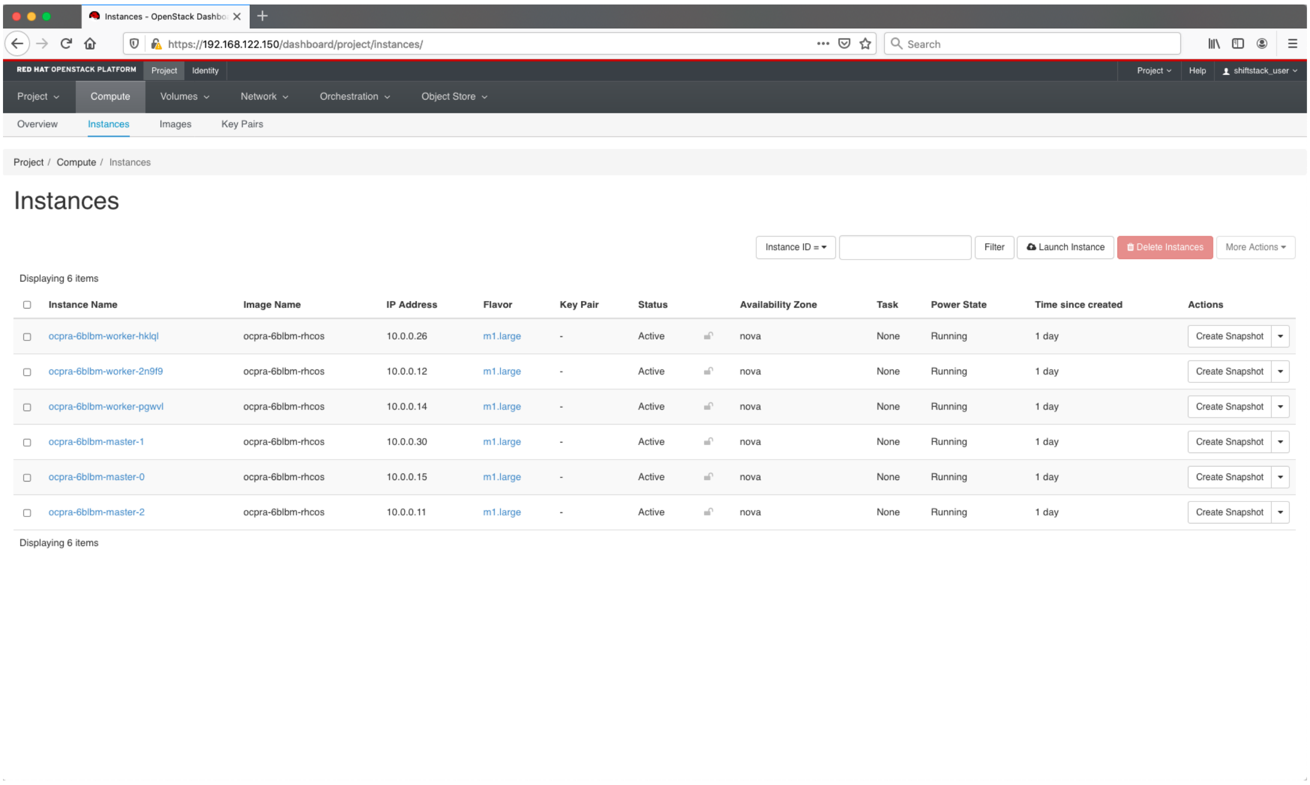
NOTE

You do not need to use **--log-level=debug** for the installation. This reference architecture uses it to clearly see the most verbose output for the installation process.

6.4. RED HAT OPENSHIFT CONTAINER PLATFORM ON RED HAT OPENSTACK PLATFORM DEPLOYMENT

The following images show the running RHOCP on RHOSP deployment using the OpenStack Dashboard.

Master and worker node instances



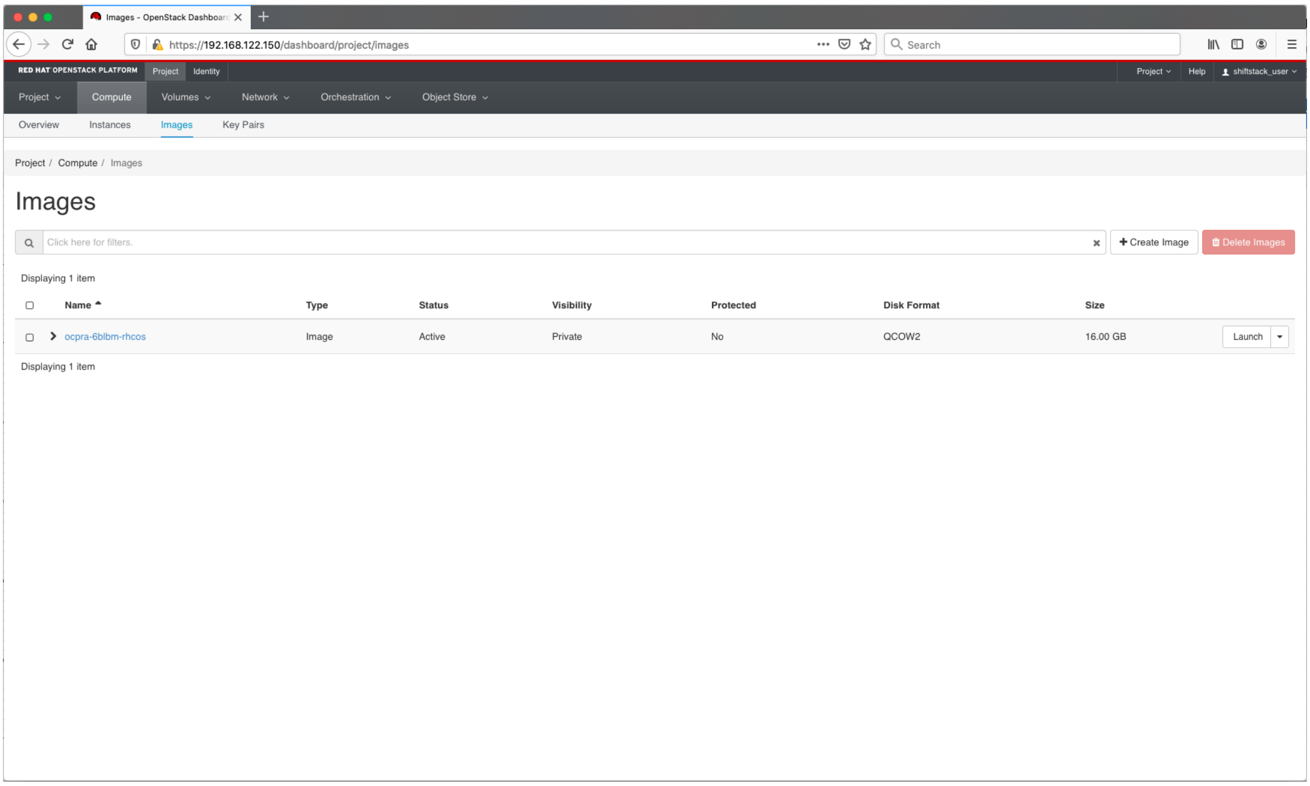
Instances

Displaying 6 items

□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status		Availability Zone	Task	Power State	Time since created	Actions
□	ocpra-6blbm-worker-hklql	ocpra-6blbm-rhcos	10.0.0.26	m1.large	-	Active		nova	None	Running	1 day	Create Snapshot
□	ocpra-6blbm-worker-2n9f9	ocpra-6blbm-rhcos	10.0.0.12	m1.large	-	Active		nova	None	Running	1 day	Create Snapshot
□	ocpra-6blbm-worker-pgwvl	ocpra-6blbm-rhcos	10.0.0.14	m1.large	-	Active		nova	None	Running	1 day	Create Snapshot
□	ocpra-6blbm-master-1	ocpra-6blbm-rhcos	10.0.0.30	m1.large	-	Active		nova	None	Running	1 day	Create Snapshot
□	ocpra-6blbm-master-0	ocpra-6blbm-rhcos	10.0.0.15	m1.large	-	Active		nova	None	Running	1 day	Create Snapshot
□	ocpra-6blbm-master-2	ocpra-6blbm-rhcos	10.0.0.11	m1.large	-	Active		nova	None	Running	1 day	Create Snapshot

Displaying 6 items

RHCOS image



Project / Compute / Images

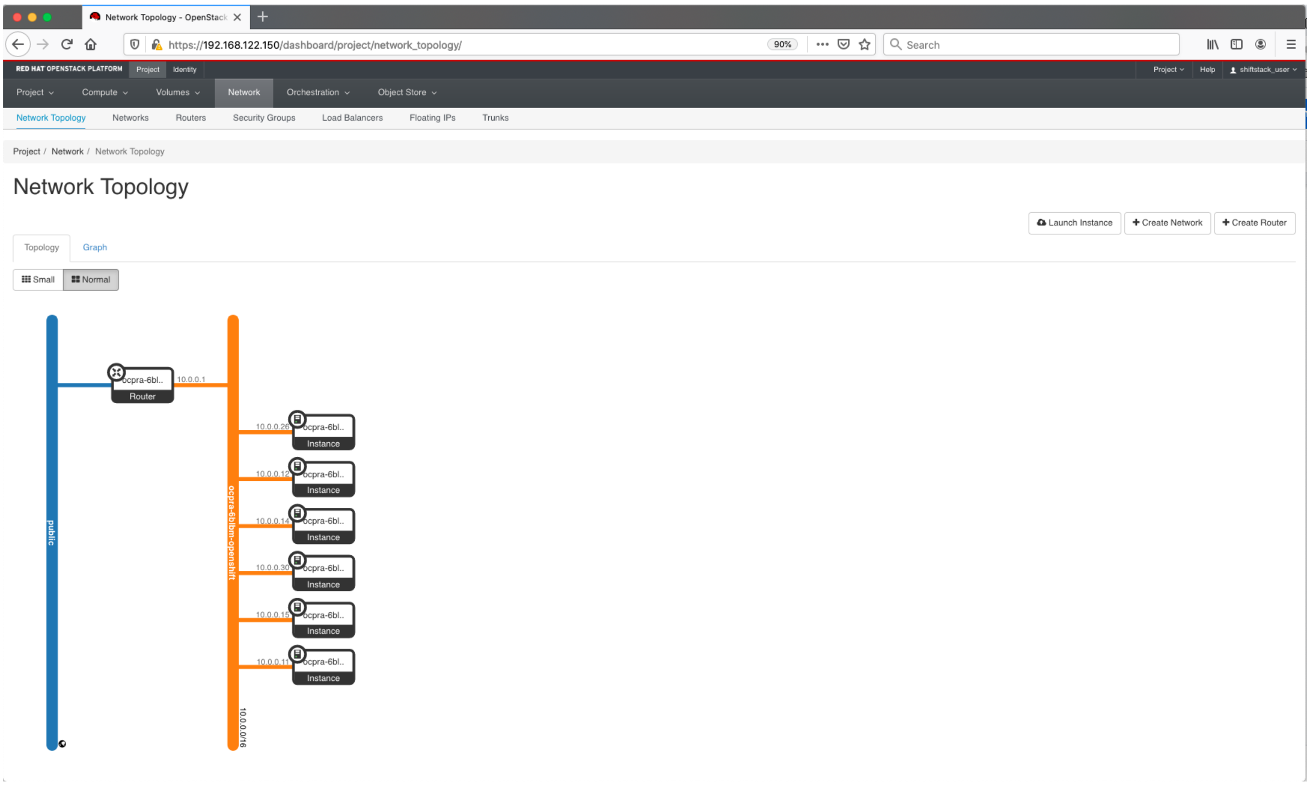
Images

Displaying 1 item

Name	Type	Status	Visibility	Protected	Disk Format	Size
ocpra-6blbm-rhcos	Image	Active	Private	No	QCOW2	16.00 GB

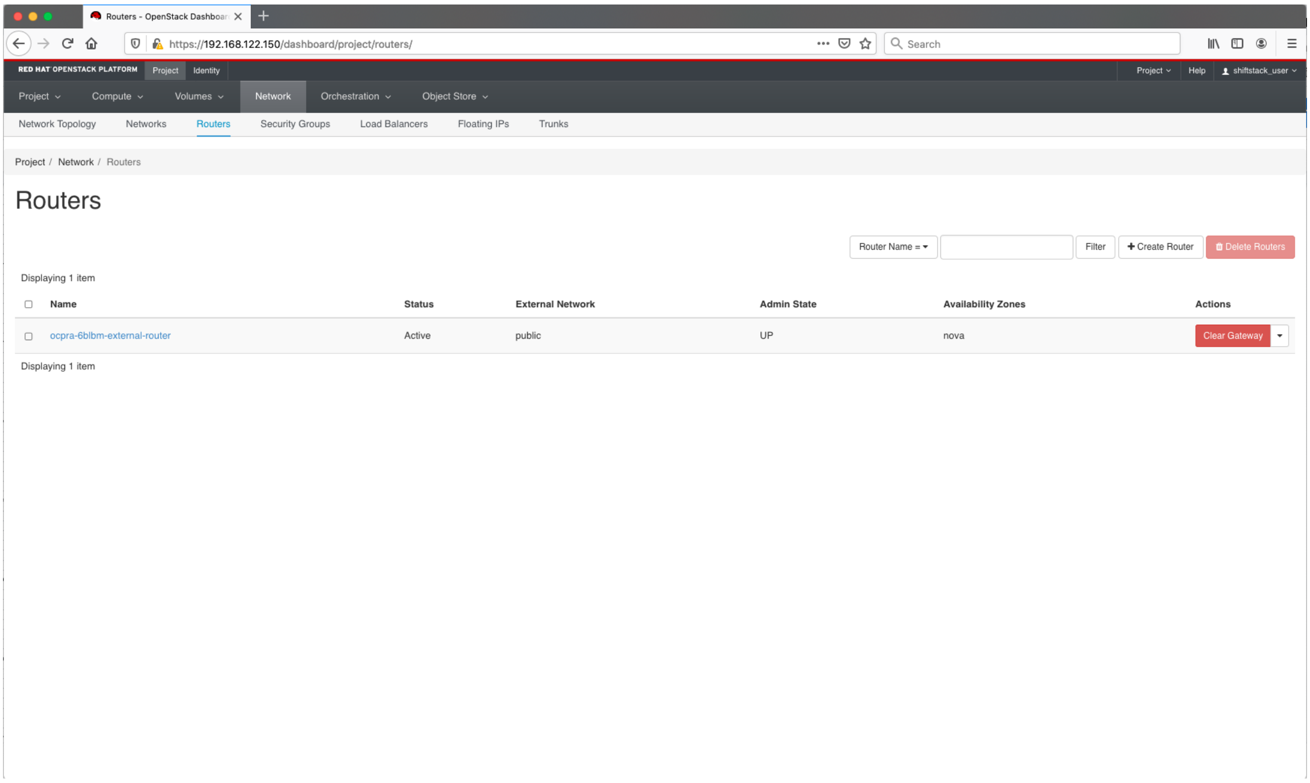
Displaying 1 item

Network



Project / Network / Network Topology

Network Topology



Routers

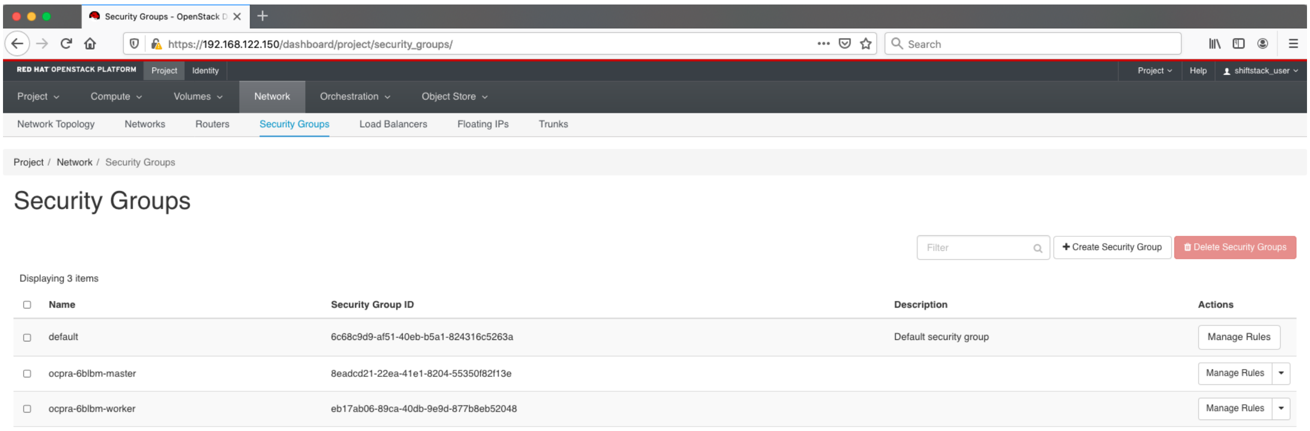
Router Name = | Filter | + Create Router | Delete Routers

Displaying 1 item

□	Name	Status	External Network	Admin State	Availability Zones	Actions
□	ocpra-6blbm-external-router	Active	public	UP	nova	Clear Gateway

Displaying 1 item

Security groups



Security Groups

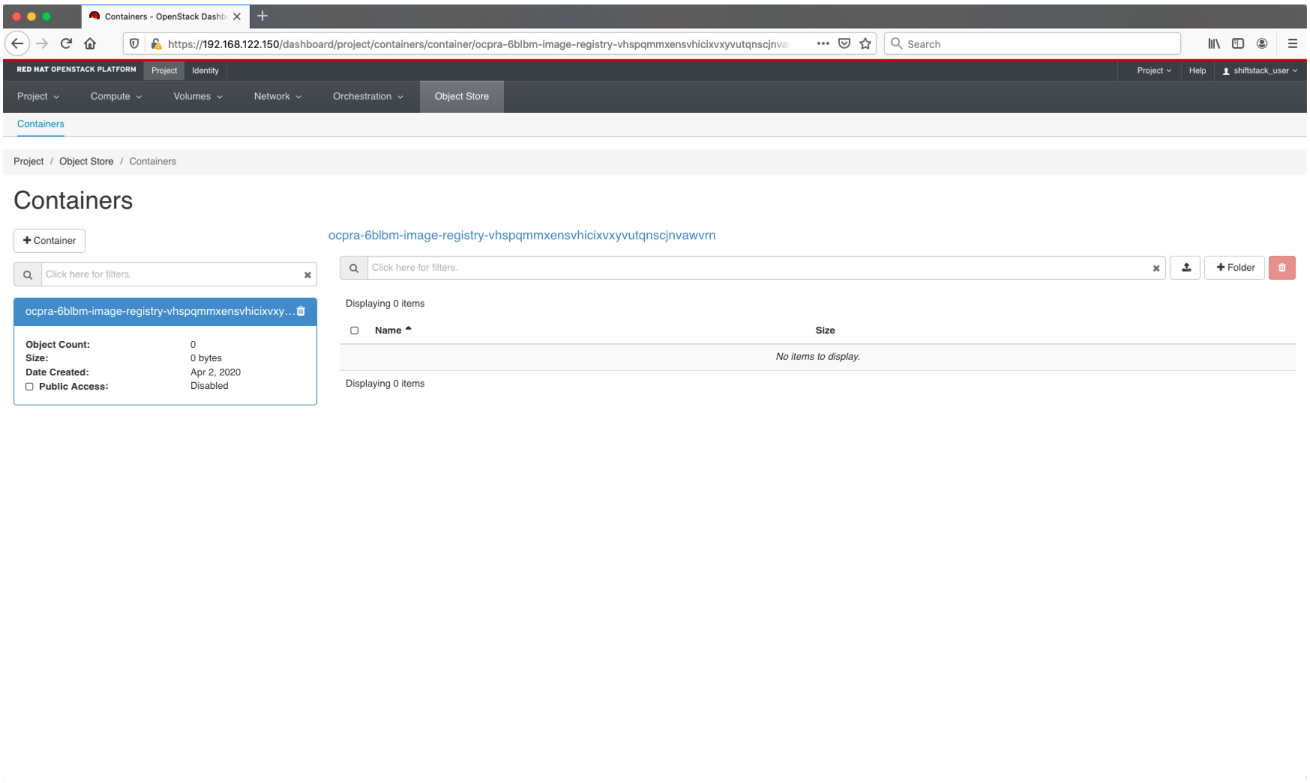
Filter | + Create Security Group | Delete Security Groups

Displaying 3 items

□	Name	Security Group ID	Description	Actions
□	default	6c68c9d9-af51-40eb-b5a1-824316c5263a	Default security group	Manage Rules
□	ocpra-6blbm-master	8eadcd21-22ea-41e1-8204-55350f82f13e		Manage Rules
□	ocpra-6blbm-worker	eb17ab06-89ca-40db-9e9d-877b8eb52048		Manage Rules

Displaying 3 items

Object storage registry


Containers - OpenStack Dashb
https://192.168.122.150/dashboard/project/containers/container/ocpra-6blbm-image-registry-vhspqmmxensvhicixvxyvutqnscjnva
Search
RED HAT OPENSTACK PLATFORM
Project
Identity
Project
Help
shiftstack_user
Project
Compute
Volumes
Network
Orchestration
Object Store
Containers
Project / Object Store / Containers
Containers
+ Container
Click here for filters.
ocpra-6blbm-image-registry-vhspqmmxensvhicixvxy...
ocpra-6blbm-image-registry-vhspqmmxensvhicixvxyvutqnscjnvawvrn
Click here for filters.
+ Folder

Object Count: 0
Size: 0 bytes
Date Created: Apr 2, 2020
□ Public Access: Disabled

Displaying 0 items

□	Name	Size
	No items to display.	

Displaying 0 items

6.5. POST INSTALLATION

To make the RHOCP on RHOSP deployment production ready, the reference architecture performed the following post-installation tasks:

- Attach the ingress floating IP to the ingress port to make it available.
- Place master nodes on separate Compute nodes.
- Verified cluster status

6.5.1. Make the ingress floating IP available

The ingress floating IP, for use by the applications running on RHOCP, was allocated to 192.168.122.167 as part of the installation and has an entry in DNS.

To make the RHOCP ingress access available, you need to manually attach the ingress floating IP to the ingress port once the cluster is created, following the guidance in Configuring application access with floating IP addresses.

NOTE

The ingress IP is a fixed IP address managed by keepalived. It has no record in the OpenStack Network database, and is therefore not visible to RHOSP, therefore it remains in a “DOWN” state when queried.

Run the following command to check the ingress port ID:

```
$ openstack port list | grep ingress
| 28282230-f90e-4b63-a5c3-a6e2faddbd15 | ocpra-6blbm-ingress-port  | fa:16:3e:4e:b8:bc |
ip_address='10.0.0.7', subnet_id='cb3dbf4a-8fb3-4b2e-bc2d-ad12606d849a'  | DOWN   |
```

Run the following command to attach the port to the IP address:

```
$ openstack floating ip set --port 28282230-f90e-4b63-a5c3-a6e2faddbd15 192.168.122.167
```

6.5.2. Place master nodes on separate Compute nodes

The RHOCP installation program does not include support for RHOSP anti-affinity rules or availability zones. Therefore, you need to move each master node to its own Compute node after installation. This reference architecture uses live migration to ensure one master per Compute node, as detailed in the following references:

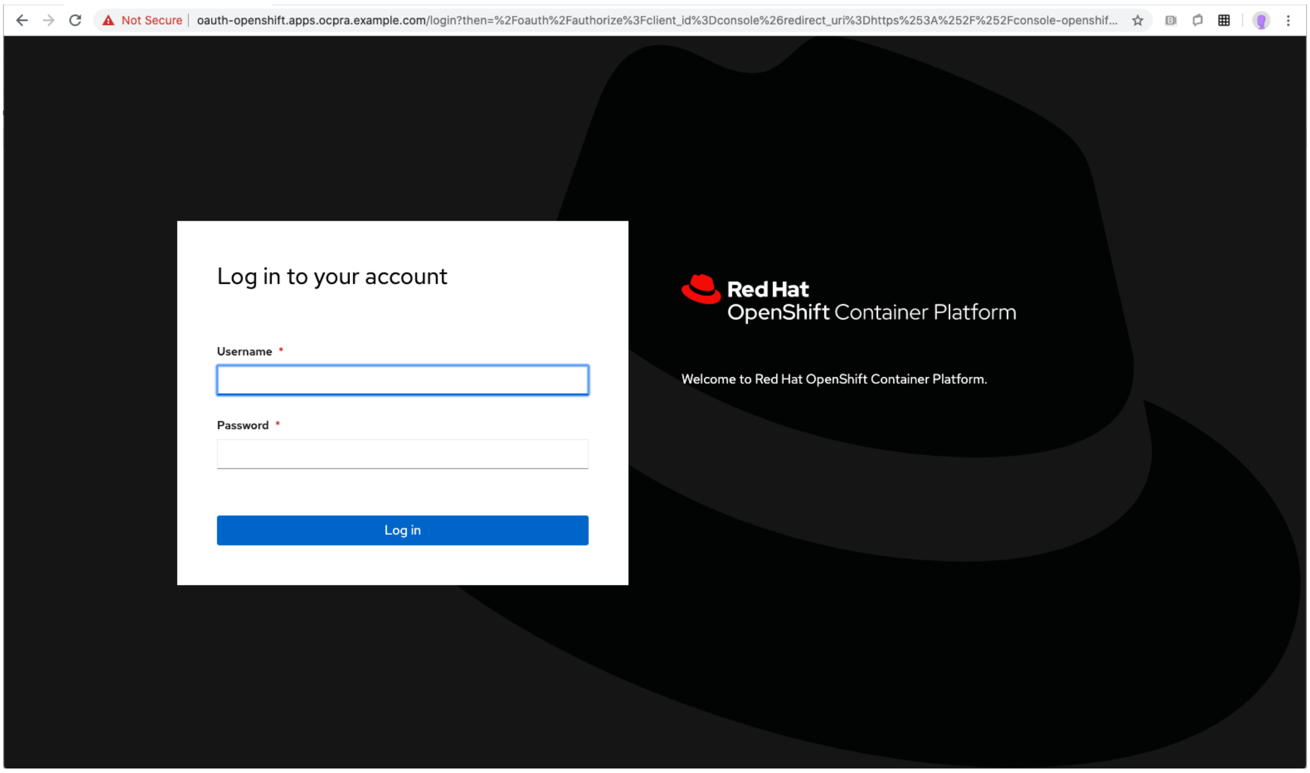
- RHOSP 13: Live Migrate a Virtual Machine
- RHOSP 16.0: Live migrating a virtual machine

Future releases are planned to support RHOSP anti-affinity rules and availability zones.

6.5.3. Verify the cluster status

This reference architecture follows the procedure described in Verifying cluster status to verify that the cluster is running correctly.

You can access the RHOCP console using the URL associated with the DNS. For this reference architecture the console is deployed to: https://console-openshift-console.apps.ocpra.example.com/



Not Secure | oauth-openshift.apps.ocpra.example.com/login?then=%2Foauth%2Fauthorize%3Fclient_id%3Dconsole%26redirect_uri%3Dhttps%253A%252F%252Fconsole-openshif...

Log in to your account

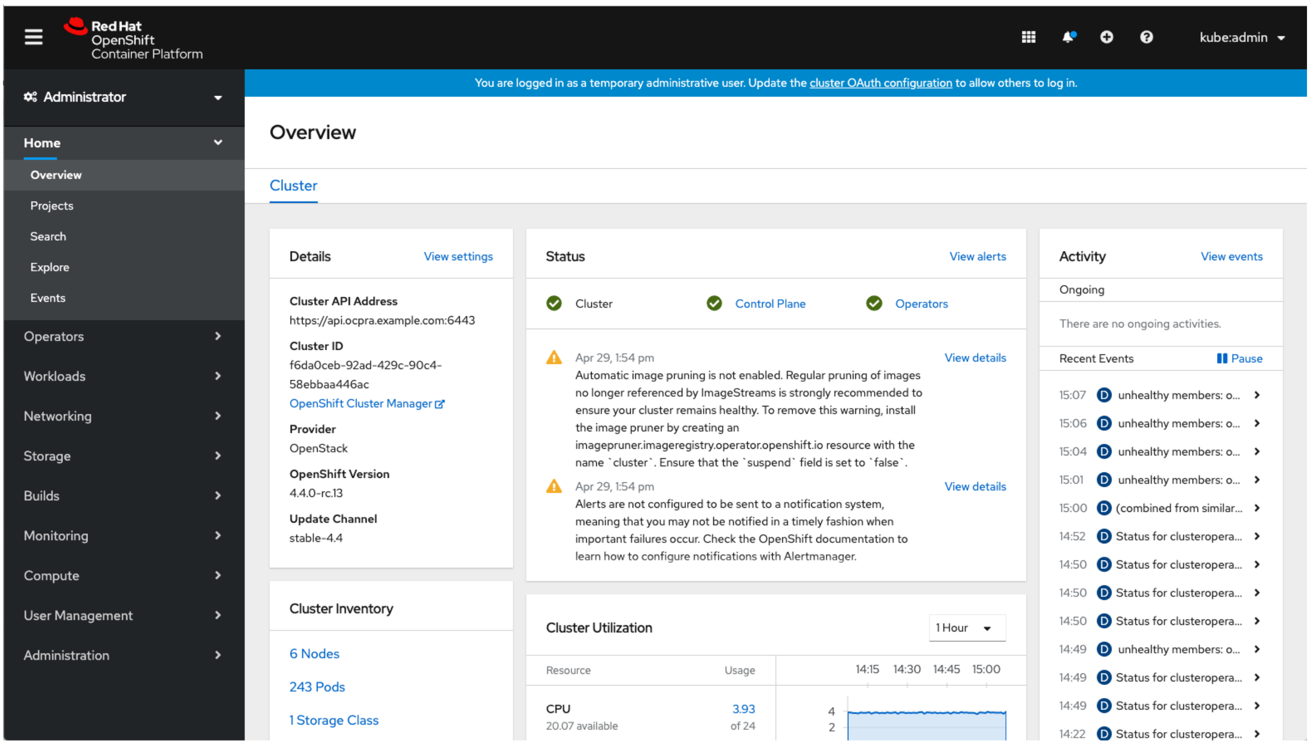
Username *

Password *

Log in

Red Hat
OpenShift Container Platform

Welcome to Red Hat OpenShift Container Platform.



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Overview

Details

Cluster API Address
https://api.ocpra.example.com:6443

Cluster ID
f6da0ceb-92ad-429c-90c4-58ebbaa446ac
OpenShift Cluster Manager

Provider
OpenStack

OpenShift Version
4.4.0-rc.13

Update Channel
stable-4.4

Status

✓ Cluster ✓ Control Plane ✓ Operators

Apr 29, 1:54 pm
Automatic image pruning is not enabled. Regular pruning of images no longer referenced by ImageStreams is strongly recommended to ensure your cluster remains healthy. To remove this warning, install the image pruner by creating an imagepruner.imageregistry.operator.openshift.io resource with the name `cluster`. Ensure that the `suspend` field is set to `false`.

Apr 29, 1:54 pm
Alerts are not configured to be sent to a notification system, meaning that you may not be notified in a timely fashion when important failures occur. Check the OpenShift documentation to learn how to configure notifications with Alertmanager.

Activity

Ongoing

There are no ongoing activities.

Recent Events

15:07 D unhealthy members: o...
15:06 D unhealthy members: o...
15:04 D unhealthy members: o...
15:01 D unhealthy members: o...
15:00 D (combined from similar...
14:52 D Status for clusteropera...
14:50 D Status for clusteropera...
14:50 D Status for clusteropera...
14:50 D Status for clusteropera...
14:49 D unhealthy members: o...
14:49 D Status for clusteropera...
14:49 D Status for clusteropera...
14:22 D Status for clusteropera...

Cluster Inventory

6 Nodes

243 Pods

1 Storage Class

Cluster Utilization

Resource	Usage
CPU 20.07 available	3.93 of 24

CHAPTER 7. SUMMARY

With Red Hat OpenShift Platform 4.4, Red Hat OpenStack Platform 13 and 16.0, and Red Hat Ceph Storage 3 and 4, organisations have access to a comprehensive and prescriptive installation experience for their on-premises container infrastructure.

Built on the already proven and successful integrations of Red Hat OpenStack Platform 13 and 16.0, and Red Hat Ceph Storage 3 and 4, IT Operations can be assured of an OpenShift installation and Day 2 operational experience similar to their public cloud installs. This makes the vision, and delivery, of true Hybrid Cloud even stronger and more compelling by ensuring organisations do not need to accept any compromise regardless of infrastructure solution chosen.

This reference architecture showcases a prescriptive and pre-validated private cloud solution from Red Hat that allows you to run IT as a Service (ITaaS), and provides rapid provisioning and lifecycle management of containerized infrastructure, virtual machines (VMs), and associated application and infrastructure services.

Red Hat OpenShift Container Platform, Red Hat OpenStack Platform, and Red Hat Ceph Storage are the key architectural components of this solution. It can be easily extended to Hybrid and Multi-Cloud with Red Hat OpenShift Container Platform serving as the common container and kubernetes platform across all clouds.

The Red Hat Quality Engineering teams (QE) have tested and validated the implementation as presented in this solution. This ensures organizations seeking to operationalize this reference architecture quickly can be assured that all options represented are both fully tested as well as fully supported by Red Hat.

CHAPTER 8. REFERENCES

Reference architecture:

- Deploying Red Hat OpenShift Container Platform 3.11 on Red Hat OpenStack Platform 13

Red Hat OpenShift Container Platform 4.4

- OpenShift Container Platform 4.4 Documentation
- Installing on OpenStack
- Red Hat Software Portal
- Upstream installer code and documentation

> **NOTE**
>
> Documentation found on docs.openshift.com and access.redhat.com are the same, and both are supported by Red Hat. Documentation on https://github.com/openshift/installer is considered upstream community documentation and not supported by Red Hat.

Red Hat OpenStack Platform 13

- Product Documentation for Red Hat OpenStack Platform 13
- Deploying an Overcloud with Containerized Red Hat Ceph

Red Hat OpenStack Platform 16.0

- Product Documentation for Red Hat OpenStack Platform 16.0
- Deploying an Overcloud with Containerized Red Hat Ceph

Red Hat Ceph Storage 3

- Product Documentation for Red Hat Ceph Storage 3
- Red Hat Ceph Storage Hardware Selection Guide
- Red Hat Ceph Storage 3 Architecture Guide

Red Hat Ceph Storage 4

- Product Documentation for Red Hat Ceph Storage 4
- Hardware Guide
- Architecture Guide

install_config.yaml

```
apiVersion: v1
```

```
baseDomain: augusts.be
compute:
- architecture: amd64
  hyperthreading: Enabled
  name: worker
  platform: {}
  replicas: 3
controlPlane:
  architecture: amd64
  hyperthreading: Enabled
  name: master
  platform: {}
  replicas: 3
metadata:
  creationTimestamp: null
  name: ocpra
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  openstack:
    cloud: openstack
    computeFlavor: m1.large
    externalDNS: null
    externalNetwork: public
    lbFloatingIP: 192.168.122.152
    octaviaSupport: "0"
    region: ""
    trunkSupport: "1"
    externalDNS: ['8.8.8.8']
    clusterOSImage: http://10.11.173.1/pub/rhcos-4.4.0-rc.1-x86_64-openstack.x86_64.raw
publish: External
pullSecret: '{"auths":{"cloud.openshift.com":{"auth":"b3BlbnNoaW … "}}}'
sshKey: |
  ssh-rsa AAAAB3NzaC1yc2 …
```

CHAPTER 9. CONTRIBUTORS

The following individuals contributed to the production of this reference architecture.

August Simonelli	**Project and content lead**
Ramon Acedo Rodriguez	Technical review / Product Management
Luis Tomas Bolivar	Technical review / Engineering
Martin André	Technical review / Engineering
Jon Uriarte	Technical Review / QE
Katherine Dube	Technical review / OpenShift
Eric Duen	Engineering oversight / Shift on Stack
Greg Charot	Technical review / Storage
Tom Barron	Technical review / Storage
Giulio Fidente	Technical review / Storage
Robert Heinzmann	Content review
Benjamin Cohen	Content review
Steven Ellis	Content review
Mohammad Ahmad	Content review
Irina Gallagher	Document reviewer and publisher
Jess Schaefer	Graphics