Hewlett Packard Enterprise | redhat.

HPE Reference Architecture for Red Hat OpenStack Platform on HPE Synergy with Ceph Storage

Contents

Executive summary

Many organizations are implementing private cloud to streamline operations by providing end users with self-service portals where they can provision and maintain virtual machines and application services. OpenStack® has becoming one of the standard for private cloud deployments; providing support for open heterogeneous software, hardware, and operating systems. OpenStack can provide organizations with a platform that can deliver Infrastructure as a Service (IaaS) offering to end user and developer communities. To help accelerate private cloud deployments, Hewlett Packard Enterprise and Red Hat® are collaborating to optimize Red Hat OpenStack on HPE Synergy, the industry's first composable infrastructure. This Reference Architecture will provide an example of an entry level Red Hat OpenStack Platform 13 deployment on the HPE Synergy and can be used as an installation and deployment example for organizations deploying their first Red Hat OpenStack Platform 13 based private cloud.

Red Hat OpenStack Platform provides the foundation to build a private or public Infrastructure-as-a-Service (IaaS) cloud on top of Red Hat Enterprise Linux. It offers a massively scalable, fault-tolerant platform for the development of cloud-enabled workloads. Red Hat OpenStack Platform allows enterprises to combine virtualization, networking, and storage based on your requirements. The cloud is managed using a web-based interface that allows administrators to control, provision, and automate OpenStack resources. Additionally, the OpenStack infrastructure is facilitated through an extensive API, which is also available to end users of the cloud.

Red Hat OpenStack on HPE Synergy provides a flexible composable IaaS cloud to dynamically expand capacity to meet demand peaks using templates for your workloads. This joint offering allows customers to standardize on shared infrastructure with open source and composable infrastructure with its RESTful API for production ready cloud environments and transforms rigid physical systems to flexible virtual resource pools so all resources are instantly available to run the Red Hat OpenStack Platform 13. HPE Synergy lets IT administrators and developers use infrastructure as code to deploy and manage their data center environments. With RHOS 13 and HPE Synergy, customer can modernize their data center and start the journey to digital transformation.

This Reference Architecture provides architectural guidance for deploying and managing a Red Hat OpenStack environment on HPE Synergy Composable Infrastructure. This Reference Architecture demonstrates the following benefits:

- Composable infrastructure for rapid deployment of network, compute, and storage resources
- Scalable infrastructure that allows organizations to scale as demand for private cloud deployments
- Cost effective modern, modular and open private cloud solution that offers a choice to replace existing costly, rigid infrastructure and proprietary virtualization tools.
- Provide flexibility with software-defined storage (Red Hat Ceph Storage)

Target audience: Chief Information Officers (CIOs), Chief Technology Officers (CTOs), data center managers, enterprise architects and implementation personnel wishing to learn more about Red Hat OpenStack Platform on HPE Synergy Composable Infrastructure. Familiarity with HPE Synergy, Red Hat OpenStack Platform, Red Hat Enterprise Linux, and core networking knowledge is assumed.

Document purpose: The purpose of this document is to demonstrate the value of combining Red Hat OpenStack Platform with HPE Composable Infrastructure capabilities offered by HPE Synergy to create a highly manageable, highly available, and performant solution that meet the needs of the business, IT personnel, and the user community.

This Reference Architecture describes solution testing performed in April 2019.

Introduction

Deploying compute at a rapid pace aligned to the changing business needs of the organization has become mandatory. Hewlett Packard Enterprise has invented a groundbreaking composable infrastructure platform, HPE Synergy that matches the IT business needs of any organization. HPE Synergy is a powerful software-defined solution that lets you manage your infrastructure as code (IaC), deploying IT resources quickly and for any workload. HPE Synergy, the first Composable Infrastructure, empowers IT to create and deliver new value instantly and continuously. Through a single interface, HPE Synergy composes compute, storage, and fabric pools into any configuration for any application. HPE OneView is the infrastructure automation engine built with software intelligence. It streamlines provisioning and lifecycle management across compute, storage and fabric resources in the Synergy System.

Red Hat OpenStack is rapidly becoming the standard for private cloud deployments; providing support for heterogeneous software, hardware, and operating systems. OpenStack can provide organizations with a platform that can deliver Infrastructure-as-Service (IaaS) and Platform-as-a

Service (PaaS) offerings to end user and developer communities. Many organizations are implementing private cloud to streamline operations by providing end users with self-service portals where they can provision and maintain virtual machines and application services.

This Reference Architecture will provide an example of an entry level Red Hat OpenStack Platform (RHOSP) 13 private cloud and Ceph Storage deployment on industry leading HPE Synergy servers. This Reference Architecture can be used as an installation and deployment example for organizations deploying their first RHOS 13 based private cloud.

Solution overview

The fundamental building blocks of this Reference Architecture are a suite of core Hewlett Packard Enterprise technologies with Red Hat OpenStack software layered on in order to create the foundation for a robust solution. The solution includes HPE Synergy, a single intelligent composable infrastructure that transforms rigid physical systems to flexible virtual resource pools so all resources are instantly available to run the Red Hat OpenStack Platform 13 based on private cloud. HPE Synergy lets IT administrators and developers use infrastructure as code to deploy and manage their data center environments. HPE Synergy Composer that houses HPE OneView provides precise composed logical infrastructures enabling administrators to provision, control, and manage software-defined data center components. HPE OneView is the infrastructure automation engine built with software intelligence which streamlines provisioning and lifecycle management across compute, storage and fabric resources in the Synergy System. This new approach for composable infrastructure combines true stateless computing with rapid deployment and updates. As customers leverage Red Hat OpenStack Platform to provide private cloud & cloud management services for running enterprise applications, it becomes critical to rely on an underlying platform that allows quick deployment of IT resources at scale and one that provides fluid pools of compute, storage and fabric resources to build a virtualized data center

HPE Synergy combined with Red Hat OpenStack Platform delivers a secure enterprise-ready private cloud that is flexible, simple to deploy and cost efficient.

The configuration uses a single (1) frame HPE Synergy Frame which consists of three (3) Red Hat OpenStack Controller nodes, three (3) Red Hat OpenStack Compute nodes and, three (3) Red Hat OpenStack Ceph Storage nodes.

The Reference Architecture demonstrates an architecture that showcase the value of combining HPE Synergy with Red Hat OpenStack Platform 13 from a deployment and lifecycle perspective in a cost-effective and highly manageable fashion. The purpose of this Reference Architecture is to deliver an experience to the broadest spectrum of end-user types with a minimal set of compromises. Flexible access to compute, storage, and fabric resources allows for use and repurposing.

HPE Synergy Frames, server modules, HPE Synergy D3940 storage module, HPE Virtual Connect SE 40Gb F8 module for Synergy reduce complexity by using intelligent auto-discovery to find all available resources to accelerate workload deployments to provide the resilient and integrated infrastructure that meets the reliability and performance needs of end-user computing architects. This drives IT efficiency as the business grows and delivers balanced performance across resources to increase solution effectiveness.

Figure 1 illustrates a high-level overview of Solution Architecture designed on HPE Synergy with Red Hat OpenStack Platform 13 private cloud deployment and Ceph Storage for private cloud deployment.

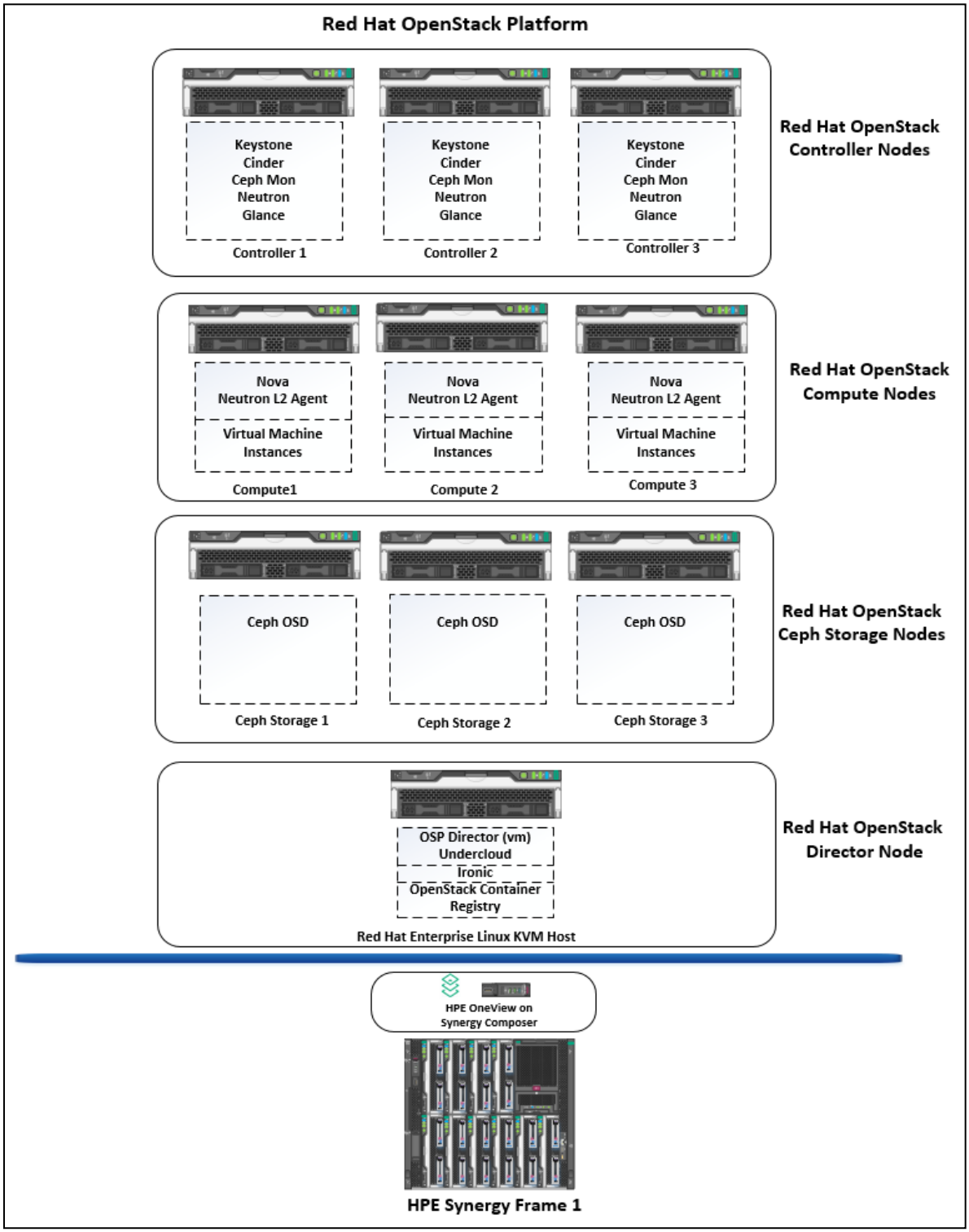


Figure 1. High level architecture for HPE Synergy with Red Hat OpenStack 13 and Ceph Storage

Solution components

Hardware

HPE Synergy is uniquely architected as Composable Infrastructure (CI) to match the powerful 'infrastructure-as-code' capabilities of the HPE intelligent software architecture. Figure 2 shows the Composable Infrastructure of HPE Synergy. The combination of hardware flexibility with embedded intelligence enables auto-discovery of all available resources for quick deployment and use management of hardware by profiles defined in software allows fast re-purposing of compute, storage and fabric resources to meet workload demands.

The Reference Architecture focuses on deploying RHOS 13 on HPE Synergy to showcase the consolidation of private cloud onto a single composable infrastructure. The hardware is viewed as “blocks” of functionality and technology segmentation. This Reference Architecture can be viewed as a series of building blocks which are summarized below.

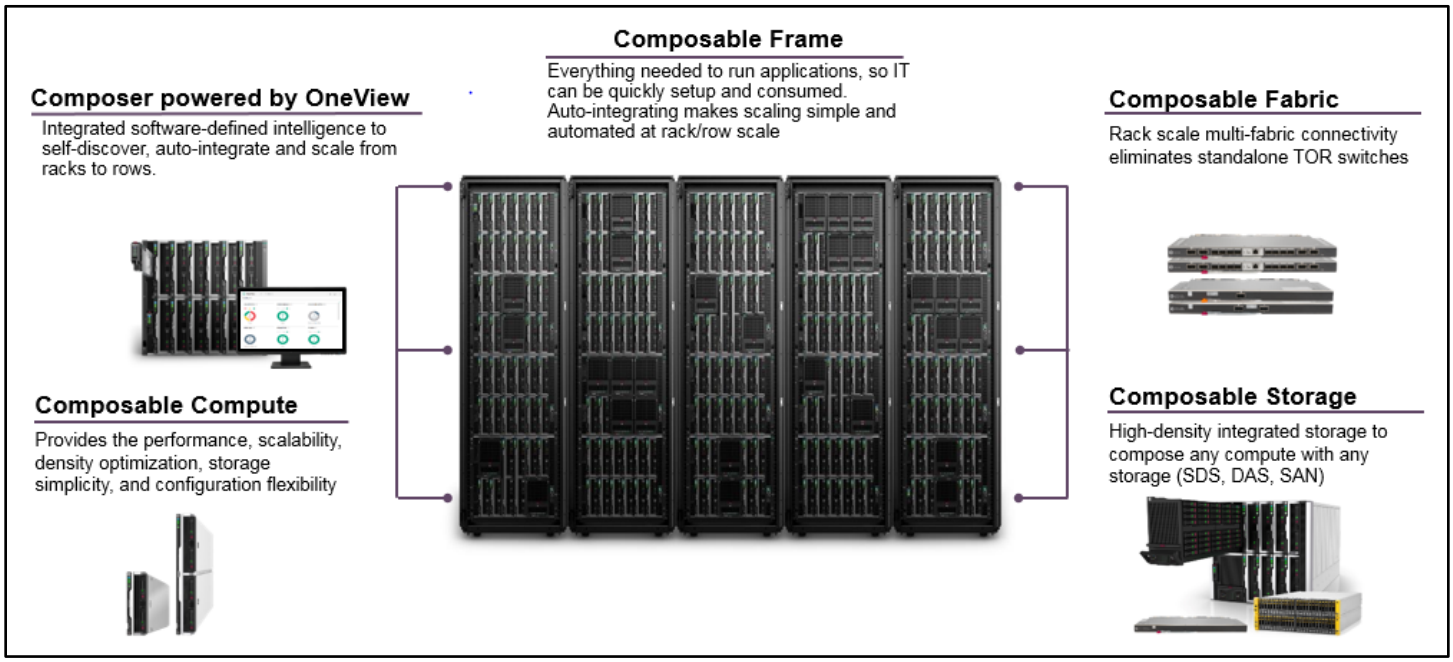


Figure 2. Composable infrastructure of HPE Synergy

HPE Synergy Composer

HPE Synergy Composer provides enterprise-level management to compose and deploy system resources to application needs. This management appliance uses software-defined intelligence with embedded HPE OneView to aggregate compute, storage and fabric resources in a manner that scales to application needs, instead of being restricted to the fixed ratios of traditional resource offerings. HPE OneView server profiles and profile templates capture the entire server configuration in one place, enabling administrators to replicate new server profiles and to modify them as needed to reflect changes in the data center. Resources can be updated, expanded, flexed, and redeployed without service interruptions. With HPE OneView Rest API and automation tools, the entire process of server personality definition and configuration can be automated.

HPE Synergy 12000 Frame

The HPE Synergy 12000 Frame is a key element of HPE Synergy, providing the base for an intelligent infrastructure with embedded management and scalable links for expansion as business demand requires. The HPE Synergy 12000 Frame is the base infrastructure that pools resources of compute, storage, fabric, cooling, power and scalability. With an embedded management solution combining the HPE Synergy Composer and HPE Synergy Frame link modules, IT can manage, assemble and scale resources on demand. The HPE Synergy 12000 Frame is designed for needs now and in the future with expanded compute and fabric bandwidths. HPE Synergy 12000 frame specifications can be found here.

The Synergy 12000 features, a fully automated and managed composer module. HPE OneView handles all the setup, provisioning, and management both at the physical and logical level. HPE Synergy 12000 Frame specifications can be found here.

For more information on HPE Synergy architecture and components, visit the HPE Synergy website.

HPE Synergy 480 Gen 10 Compute Module

The HPE Synergy 480 Compute Module delivers superior capacity, efficiency, and flexibility in a two-socket, half-height, single-wide form factor to support demanding workloads. Powered by Intel® Xeon® scalable family of processors, up to 3TB DDR4, more storage capacity and controllers and a variety of GPU options within a composable architecture. HPE Synergy 480 Gen10 Compute Module is the ideal platform for general-purpose enterprise workload performance now and in the future.

- The most secure server with exclusive HPE silicon root of trust. Protect your applications and assets against downtime associated with hacks and viruses.
- More customer choice for greater performance and flexibility with Intel Xeon Scalable Family of processors on the Synergy 480 Gen10 architecture.
- Intelligent System Tuning with processor smoothing and workloads matching to improve processor throughput/overall performance up to 8% over previous generation.
- Max memory 3TB for large in-memory database and analytic applications.
- New hybrid Smart Array for both RAID and HBA zoning in a single controller; internal M.2 storage options that add boot flexibility and additional local storage capacity.

More information on HPE Synergy 480 Compute Modules can be found at the HPE Synergy Compute Module website.

HPE Synergy D3940 Storage Module

The HPE Synergy D3940 Storage Module holds up to 40 Small Form Factor (SFF) hard drives or SSDs and is designed for use in HPE Synergy 12000 Frames. Through the HPE Synergy 12GB SAS connection module, the HPE Synergy D3940 Storage Module provides composable direct attached storage for up to 10 HPE Synergy 480 Compute Modules in a single HPE Synergy 12000 Frame. HPE Synergy D3940 Storage Module is optimized for use as either a direct attached storage array or similar solutions. Visit the HPE Synergy Storage website for more information.

HPE Virtual Connect SE 40Gb F8 Module for Synergy

The HPE Virtual Connect SE 40Gb F8 Module, master module based on composable fabric, is designed for composable Infrastructure. Its disaggregated, rack-scale design uses a master/satellite architecture to consolidate data center network connections, reduce hardware and scales network bandwidth across multiple HPE Synergy 12000 Frames. The HPE Virtual Connect SE 40Gb F8 module for Synergy eliminates network sprawl at the edge with one device that converges traffic inside the HPE Synergy 12000 Frames, and directly connects to external LANs.

Table 1. HPE Synergy components utilized in this Reference Architecture

Component	Description
HPE Synergy 480 Gen10 Compute Module	10 Nodes
HPE Synergy 12Gb SAS Connection Module	2
HPE Virtual Connect SE 40Gb F8 Module for Synergy	2
HPE Synergy 12000 Frame	1 Frame
HPE Synergy Composer	1
HPE Synergy Storage Module	D3940
HPE Synergy network options	Synergy 3820C 10/20Gb CNA

Software

Red Hat Enterprise Linux

Red Hat® Enterprise Linux® server powers the applications that run your business with the control, confidence, and freedom that comes from a consistent foundation across hybrid cloud deployments. As the premier platform provider for enterprise workloads, Red Hat works side by side

with engineers from major hardware vendors and cloud providers to make sure that the operating system takes full advantage of the latest innovations. This leadership with partners, as well as Red Hat's influence and contributions to upstream communities, provides a stable, secure, and performance driven foundation for the applications that run the business of today and tomorrow. Red Hat Enterprise Linux is at the core of this solution; each of the Red Hat OpenShift container platform control plane nodes running as virtual machines are running Red Hat Enterprise Linux.

Red Hat OpenStack Platform

Red Hat OpenStack Platform provides the foundation to build a private or public Infrastructure-as-a-Service (IaaS) cloud on top of Red Hat Enterprise Linux. Red Hat OpenStack Platform delivers an integrated foundation to create, deploy, and scale a secure and reliable public or private OpenStack cloud.

Figure 3 shows a high-level overview of the OpenStack core services[1].

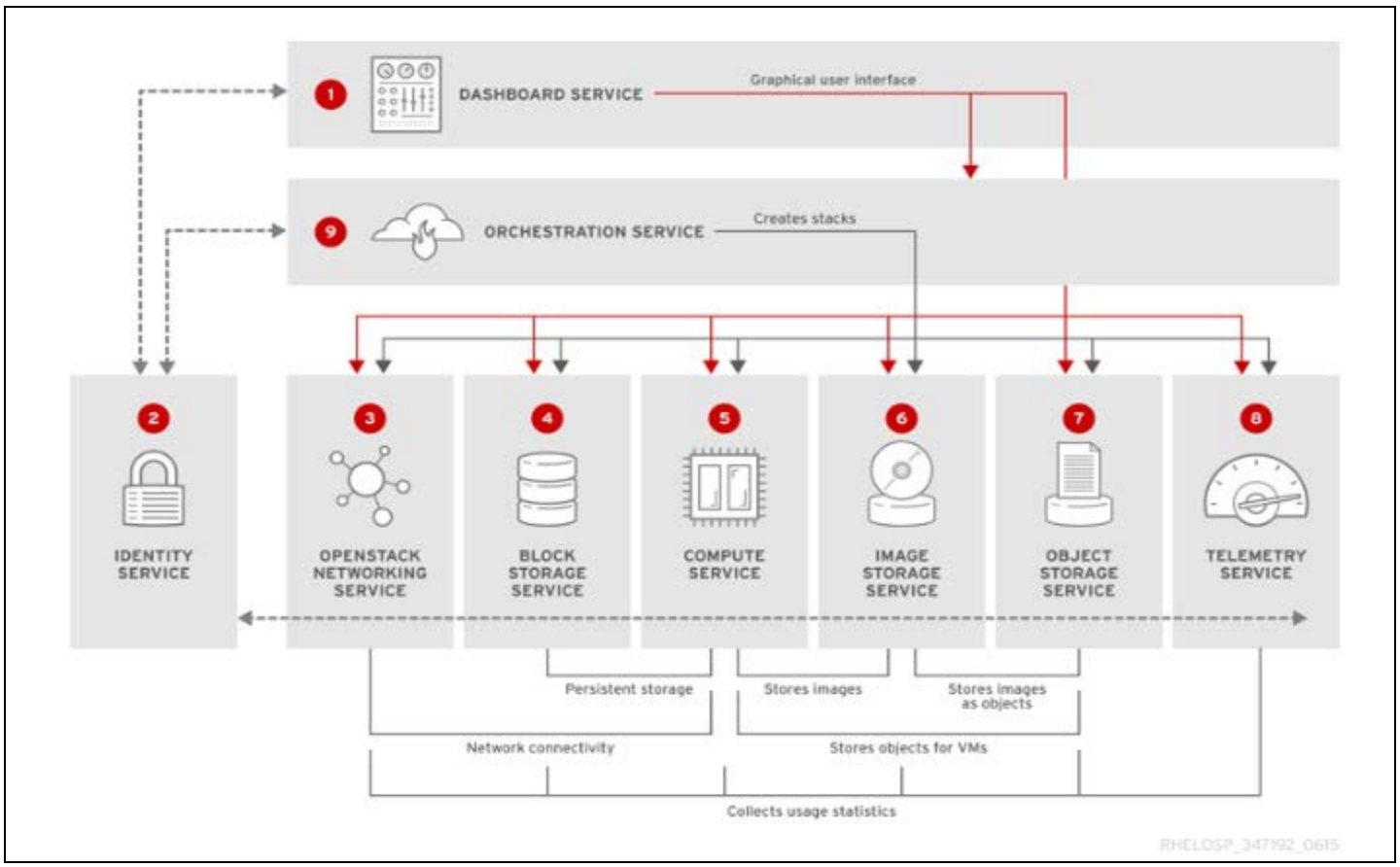


Figure 3. High-level overview of the OpenStack core services and their relationship with each other

Red Hat Ceph Storage

Red Hat Ceph Storage is a distributed data object store designed to provide excellent performance, reliability, and scalability. Distributed object stores are the future of storage, because they accommodate unstructured data, and because clients can use modern object interfaces and legacy interfaces simultaneously. At the heart of every Ceph deployment is the Ceph Storage Cluster, which consists of two types of daemons: Ceph OSD (Object Storage Daemon) and Ceph Monitor.

Red Hat Ceph Storage can provide block storage, Ceph Object Gateway and Ceph File Storage (CephFS) for Red Hat OpenStack.

[1] Source: https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/html/product_guide/ch-rhosp-software

Best practices and configuration guidance for the solution

This section highlights the configuration of the solution at a high level, provides guidance for decision making by providing details about hardware and software pre-requisites and points to further resources as needed.

HPE Synergy solution configuration

HPE Synergy hardware for this Reference Architecture was set up following HPE Synergy Configuration and Compatibility Guide. This section describes the setup of components specific to this solution.

For this Reference Architecture, one HPE Synergy 12000 Frame is configured with one HPE Synergy Composers and HPE Synergy D3940 Storage Modules. The HPE Synergy 12000 Frame has three HPE Synergy 480 which acts as Red Hat OpenStack Compute Nodes, three HPE Synergy 480 which acts as Red Hat OpenStack Controller Nodes and three HPE Synergy 480 which acts as Red Hat OpenStack Ceph Nodes. One HPE Synergy 480 node is being used as KVM host where RHOS 13 director is installed as a virtual machine.

Red Hat OpenStack Platform 13 deployment in HPE Synergy

Red Hat OpenStack Platform 13 director

In this Reference Architecture, the Red Hat OpenStack Platform 13 director is installed as a virtual machine on one of the HPE Synergy 480 Gen10 servers. The host is running Red Hat Enterprise Linux version 7.6 with KVM. The Red Hat OpenStack Platform 13 director virtual machine is running Red Hat Enterprise Linux version 7.3. By running the Red Hat OpenStack Platform 10 director as a virtual machine, the physical server can be used to support additional services and virtual machines. Additionally, running the RHOSP director on a virtualization platform provides the ability to snapshot the virtual machine at various stages of the installation. Virtual machine snapshots can also be useful if it is necessary to revert the system to a previously known good state or configuration. In this Reference Architecture, there are additional Red Hat Enterprise Linux version 7.6 based virtual machines; a logging vm, and monitoring virtual machines.

Network configuration

The network infrastructure is comprised of two HPE 5900AF network switches. One of the HPE 5900AF is used for the management (iLO) and provisioning networks. Other HPE 5900AF is used for the following Red Hat OpenStack Platform 13 networks:

- External
- Internal API
- Tenant
- Storage Management
- Storage network

An overview of the network connections to each of the individual servers is shown in Figure 4 below.

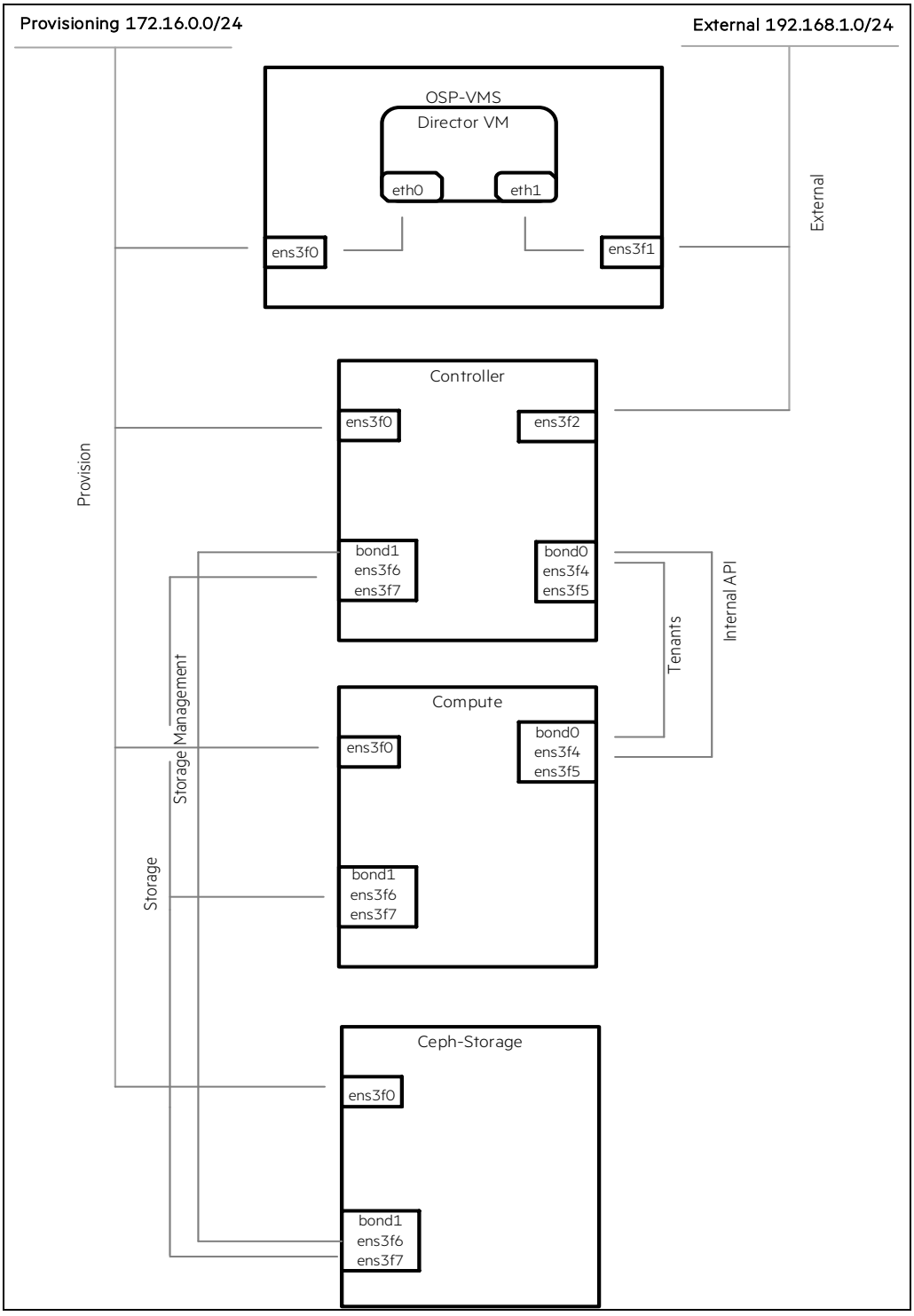


Figure 4. Network topology used for testing

VLAN configuration

HPE OneView manages the infrastructure and is used to define the network and infrastructure related components.

Table 2 describes the configuration of the networks as defined within HPE OneView for Synergy and the bandwidth associated with each network.

Table 2. Networks defined within HPE Synergy Composer for this solution

Network Name	Type	VLAN Name	VLAN Number	Requested Bandwidth (Gb)	Maximum Bandwidth (Gb)
External	Ethernet	hpe-ext-mgmt	104	2.5	20
Internal API	Ethernet	hpe-api	3041	2.5	20
Provisioning	Ethernet	hpe-prov	3040	2.5	20
Tenant	Ethernet	hpe-ten	3044	2.5	20
Storage Management	Ethernet	hpe-stormgmt	3043	2.5	20
Storage network	Ethernet	hpe-stor	3042	2.5	20

Creating Network and Network Sets in HPE OneView

Following networks are created and Figure 5 has the screenshot of the networks with their vlan details:

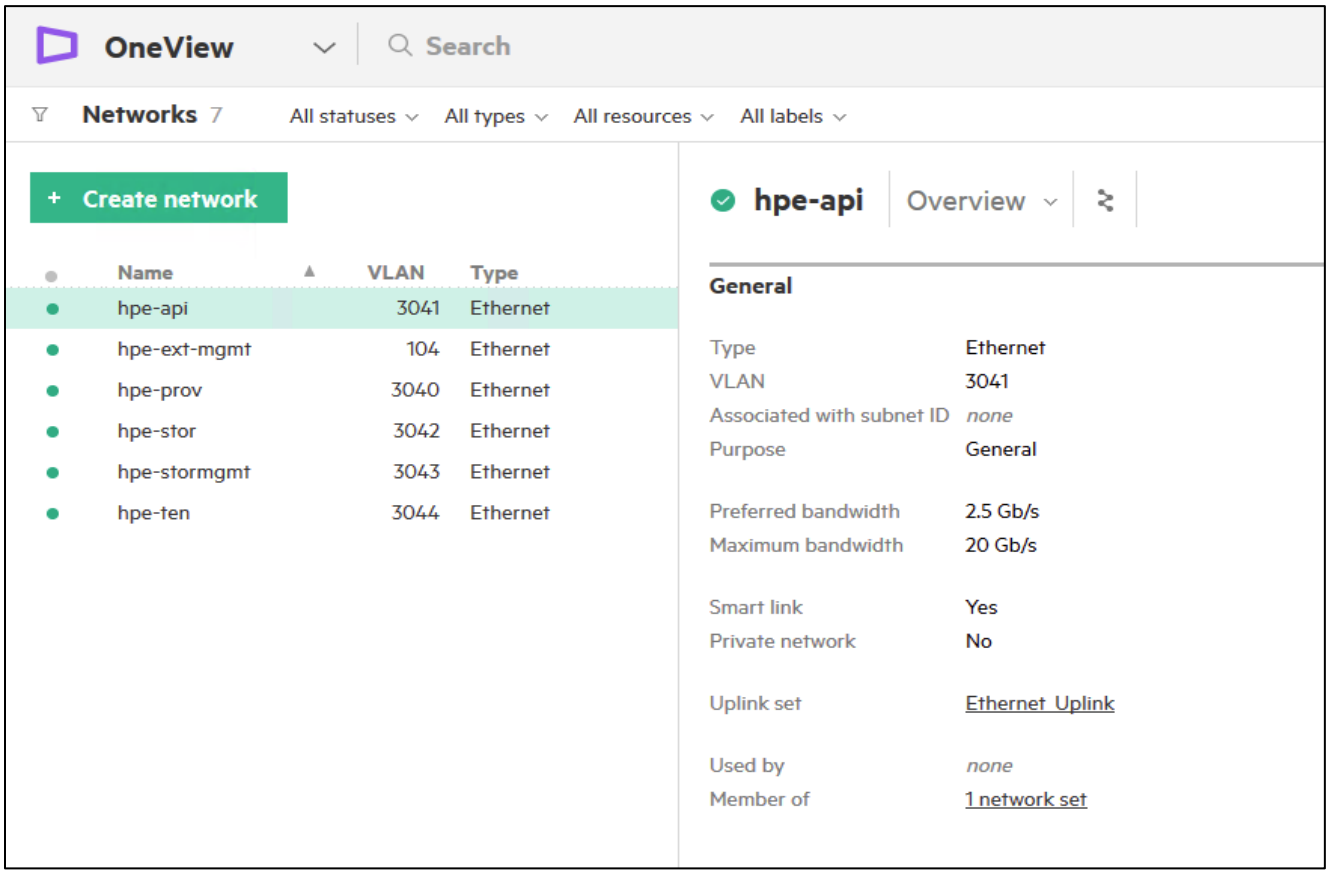


Figure 5. List of networks created

Figure 6 shows an overview of one such network.

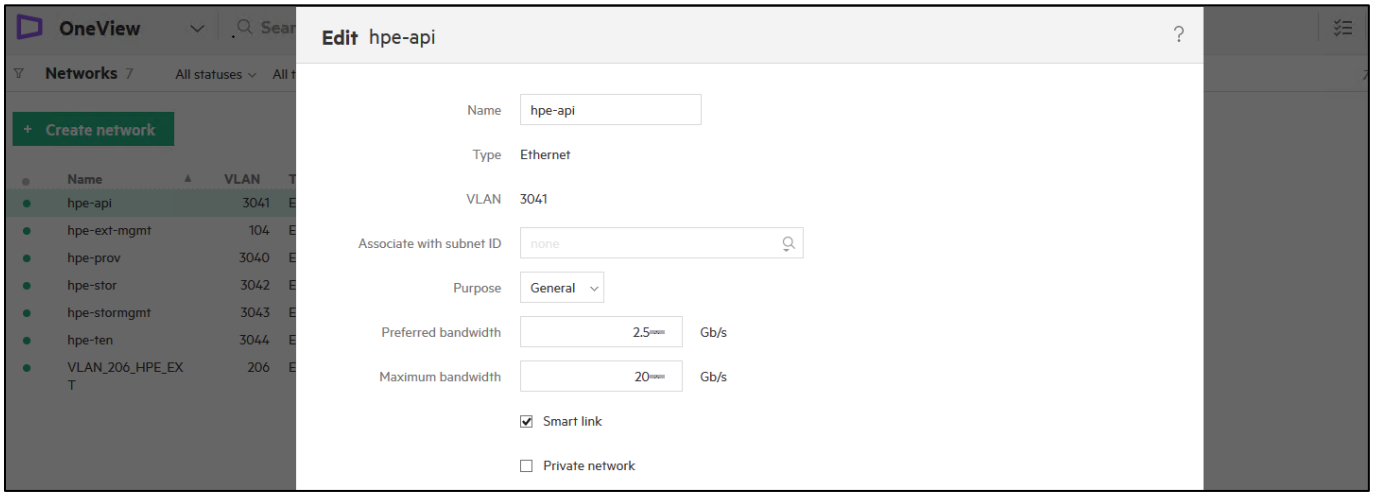


Figure 6. Overview of hpe-api network

Virtual Connect administrators can assign multiple VLANs to a single profile connection. HPE OneView expands on this and allows administrators to configure network sets containing multiple vNets. The network set is then assigned to server profile connections. For our testing, two Network Sets, namely Storage and Tenant-API, are created, as shown in Figure 7.

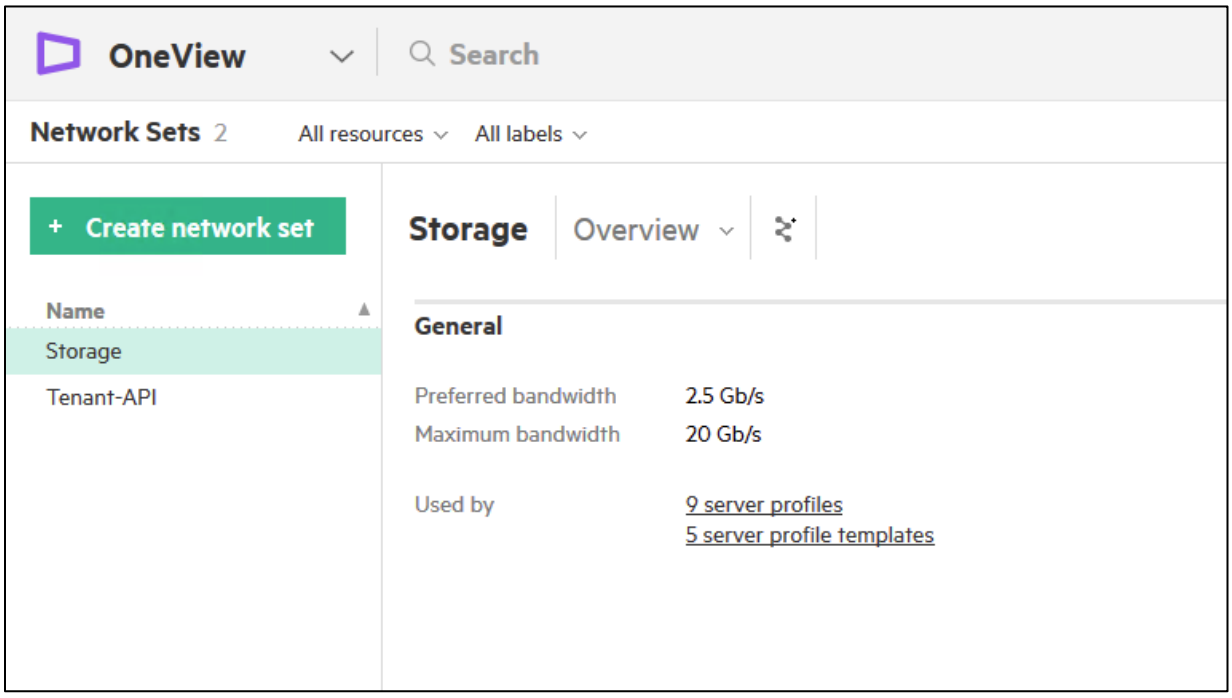


Figure 7. Networks Set used in the set-up

Figure 8 shows overview of one such network set.

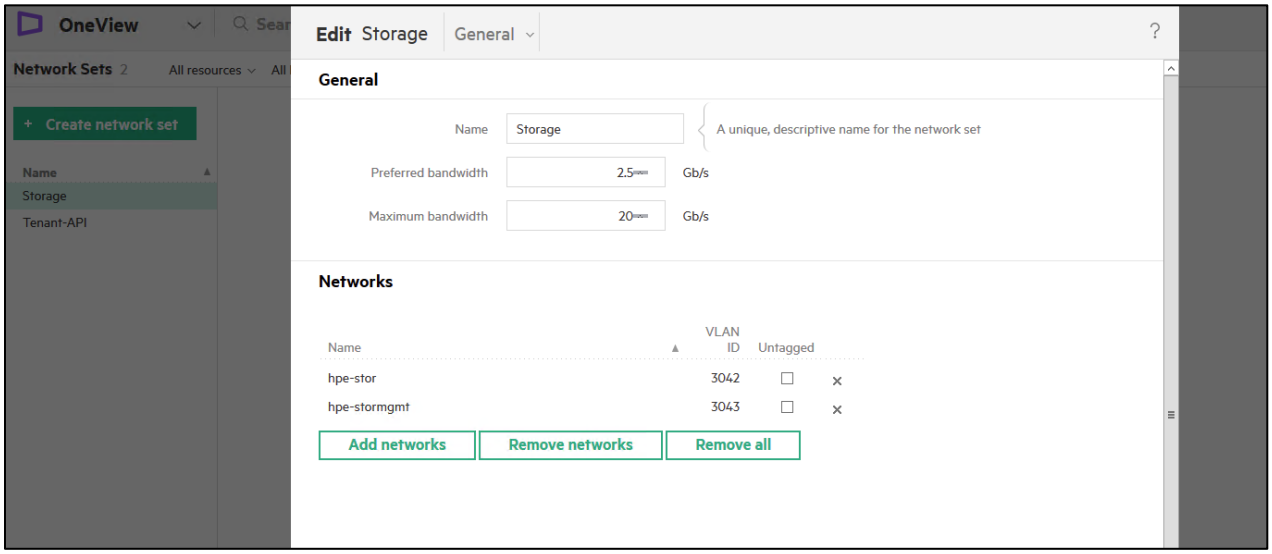


Figure 8. Overview of storage network set

Configure the Red Hat Ceph Storage Nodes

The Red Hat Ceph Storage cluster in this solution is comprised of three HPE Synergy 480 Gen10 servers. A three node cluster is the absolute minimum replicated cluster size. A cluster size of five nodes for replication and seven nodes for ensure coding would provide better availability and increased performance.

The below screenshot shows that the Synergy D3940 Storage Module that is used in the Reference Architecture.

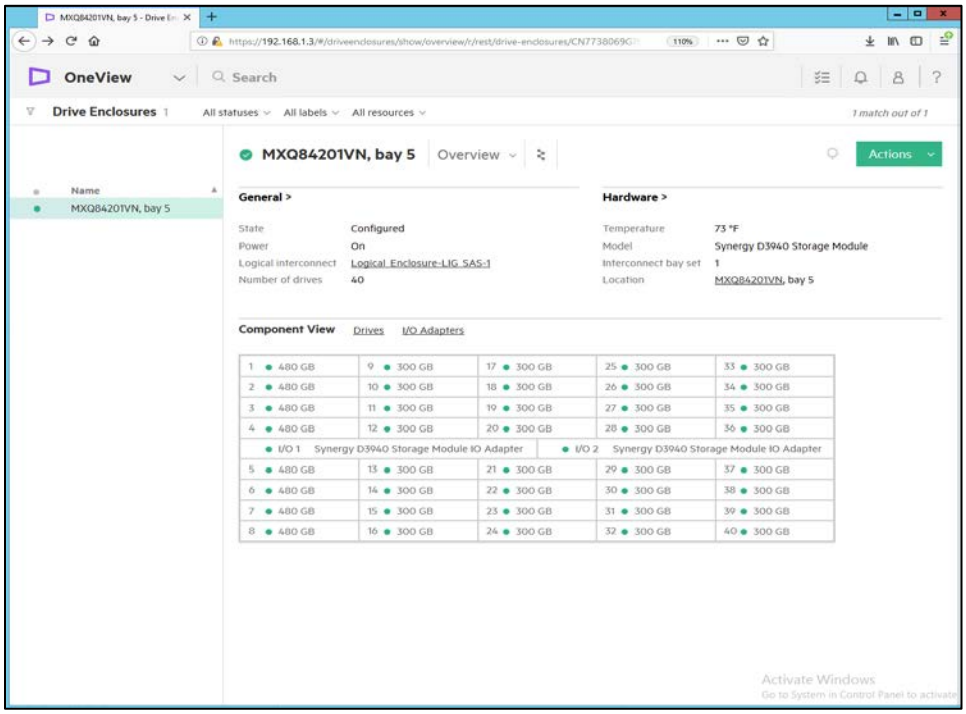


Figure 9. Disk Layout of Synergy D3940 Storage Module

The 480GB drives are of type SATA SSD and used for the Ceph journals, while the 300GB drives are SAS HDD drives which are used for the Ceph OSDs.

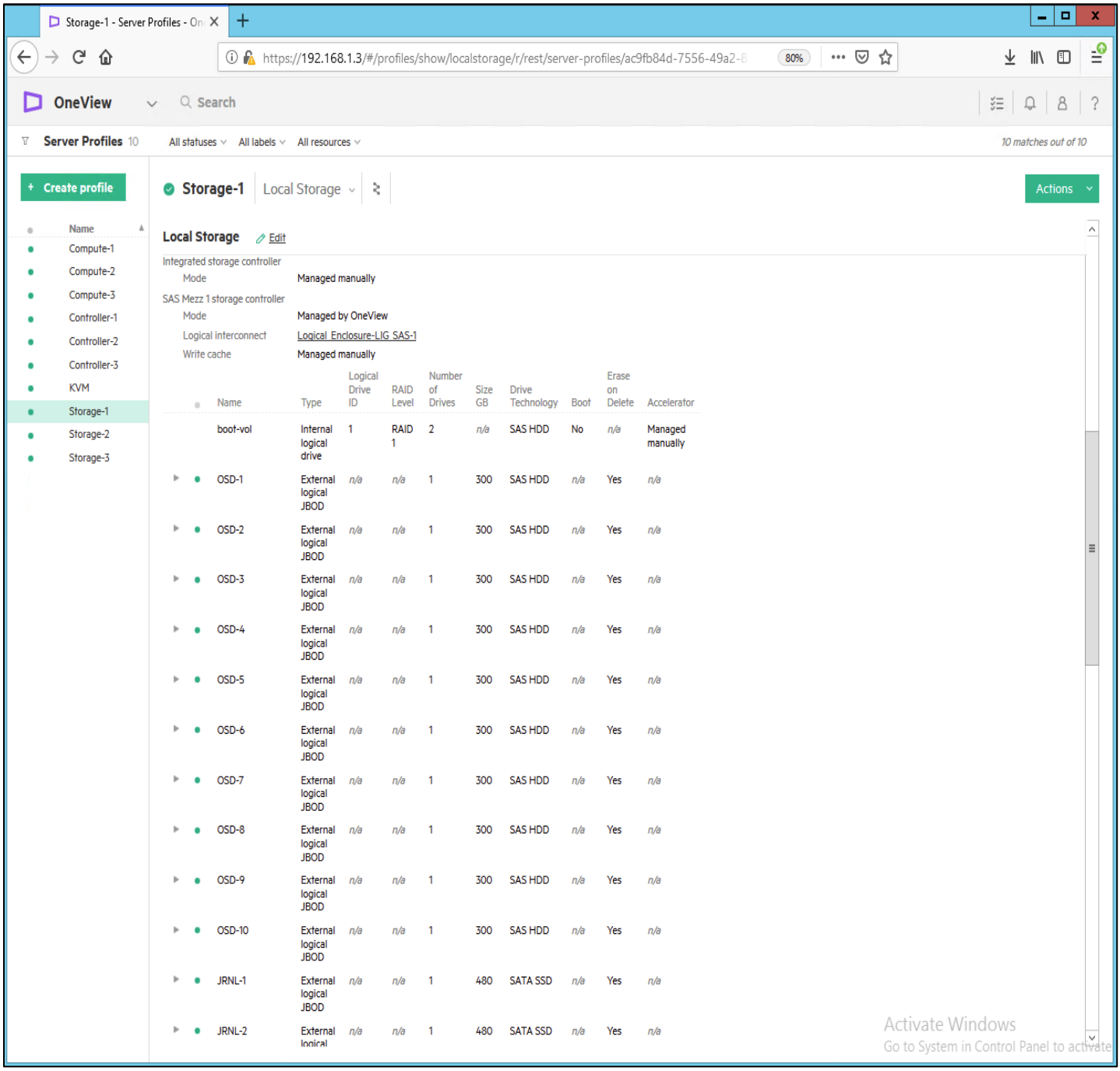


Figure 10. Server profile for the Storage-1 with local storage mapping

Figure 10 above shows the Server Profile for the Storage-1 with Local Storage mapping. The drives OSD-1 to OSD-10 are the SAS HDD, while the JRNL-1 and JRNL-2 are the SATA SSD drives used for the Ceph Journal.

Figure 11 shows the drives from overcloud-storage-0 node perspective. As you see, each of the HDD drives show up twice. This is due to the dual SAS switches in Synergy. While presenting the disks to Ceph, use a unique full path to the disk, example. "/dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b884cf95-lun-0"

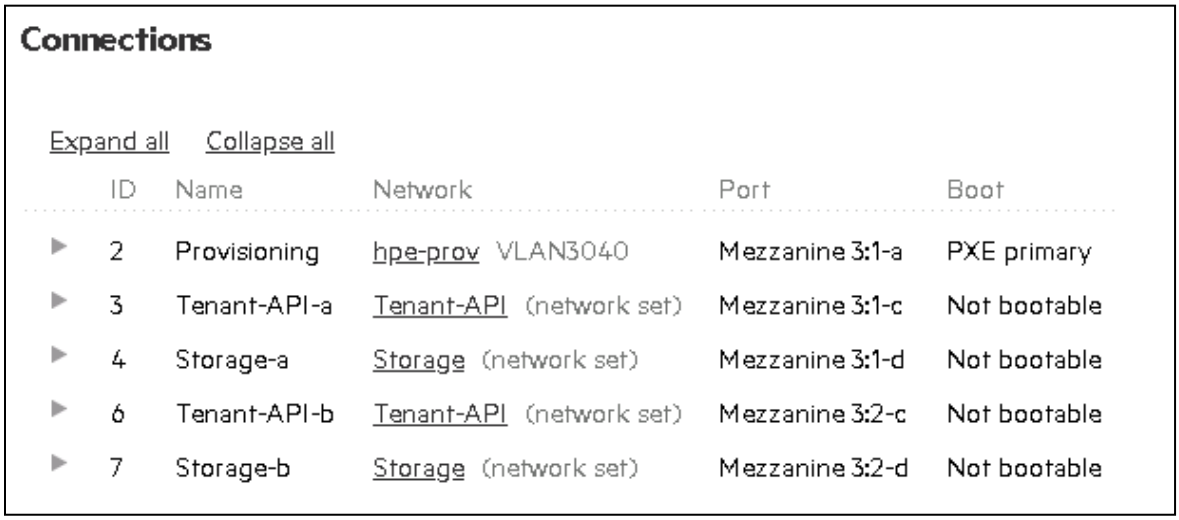
```
[root@overcloud-cephstorage-0 ~]# lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0 447.1G  0 disk
├─sda1   8:1    0     8G  0 part
├─sda2   8:2    0     8G  0 part
├─sda3   8:3    0     8G  0 part
├─sda4   8:4    0     8G  0 part
└─sda5   8:5    0     8G  0 part
sdb      8:16   0 447.1G  0 disk
├─sdb1   8:17   0     8G  0 part
├─sdb2   8:18   0     8G  0 part
├─sdb3   8:19   0     8G  0 part
├─sdb4   8:20   0     8G  0 part
└─sdb5   8:21   0     8G  0 part
sdc      8:32   0 279.4G  0 disk
sdd      8:48   0 279.4G  0 disk
└─sdd1   8:49   0 279.4G  0 part
sde      8:64   0 279.4G  0 disk
sdg      8:96   0 279.4G  0 disk
└─sdg1   8:97   0 279.4G  0 part
sdh      8:112  0 279.4G  0 disk
sdi      8:128  0 279.4G  0 disk
└─sdi1   8:129  0 279.4G  0 part
sdj      8:144  0 279.4G  0 disk
sdk      8:160  0 279.4G  0 disk
└─sdk1   8:161  0 279.4G  0 part
sdl      8:176  0 279.4G  0 disk
sdm      8:192  0 279.4G  0 disk
└─sdm1   8:193  0 279.4G  0 part
sdn      8:208  0 279.4G  0 disk
sdo      8:224  0 279.4G  0 disk
└─sdo1   8:225  0 279.4G  0 part
sdp      8:240  0 279.4G  0 disk
└─sdp1   8:241  0 279.4G  0 part
sdq     65:0    0 279.4G  0 disk
sdr     65:16   0 279.4G  0 disk
sds     65:32   0 279.4G  0 disk
└─sds1  65:33   0 279.4G  0 part
sdt     65:48   0 279.4G  0 disk
sdu     65:64   0 279.4G  0 disk
└─sdu1  65:65   0 279.4G  0 part
sdv     65:80   0 279.4G  0 disk
sdw     65:96   0 279.4G  0 disk
└─sdw1  65:97   0 279.4G  0 part
sdx     65:112  0 279.4G  0 disk
├─sdx1  65:113  0     1M  0 part
└─sdx2  65:114  0 279.4G  0 part /
```

Figure 11. Storage drives

Server profile

HPE Synergy Composable Infrastructure using HPE Virtual Connect provides the construct of a server profile. A server profile allows a suite of configuration parameters, including network and SAN connectivity, BIOS tuning, boot order configuration, local storage configuration and more to be templatized and applied programmatically to compute resources. These templates are the key to delivering the "infrastructure as code" capabilities of the HPE Synergy platform. For the purpose of this solution, separate template was created for Compute, Controller, Director and Storage nodes that was applied to all modules respectively.

Figure 12 and Figure 13 shows the Network interfaces and Storage interfaces of compute server profile template

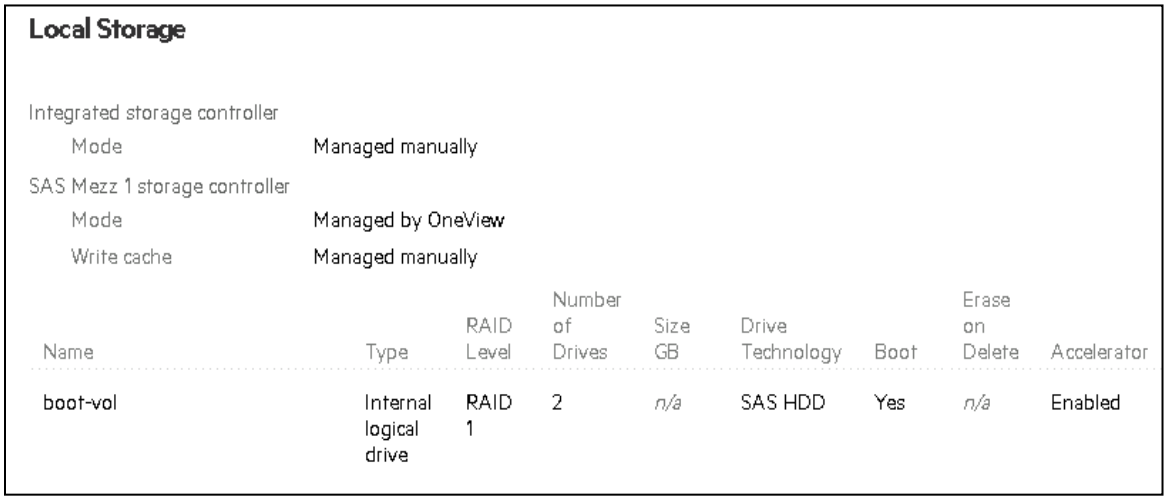


Connections

Expand all Collapse all

	ID	Name	Network	Port	Boot
▶	2	Provisioning	hpe-prov VLAN3040	Mezzanine 3:1-a	PXE primary
▶	3	Tenant-API-a	Tenant-API (network set)	Mezzanine 3:1-c	Not bootable
▶	4	Storage-a	Storage (network set)	Mezzanine 3:1-d	Not bootable
▶	6	Tenant-API-b	Tenant-API (network set)	Mezzanine 3:2-c	Not bootable
▶	7	Storage-b	Storage (network set)	Mezzanine 3:2-d	Not bootable

Figure 12. Network interfaces as defined by the compute server profile template



Local Storage

Integrated storage controller
Mode: Managed manually

SAS Mezz 1 storage controller
Mode: Managed by OneView
Write cache: Managed manually

Name	Type	RAID Level	Number of Drives	Size GB	Drive Technology	Boot	Erase on Delete	Accelerator
boot-vol	Internal logical drive	RAID 1	2	*n/a*	SAS HDD	Yes	*n/a*	Enabled

Figure 13. Storage interfaces as defined by the compute server profile template

Figure 14 and Figure 15 shows the Network interfaces and Storage interfaces of controller server profile template

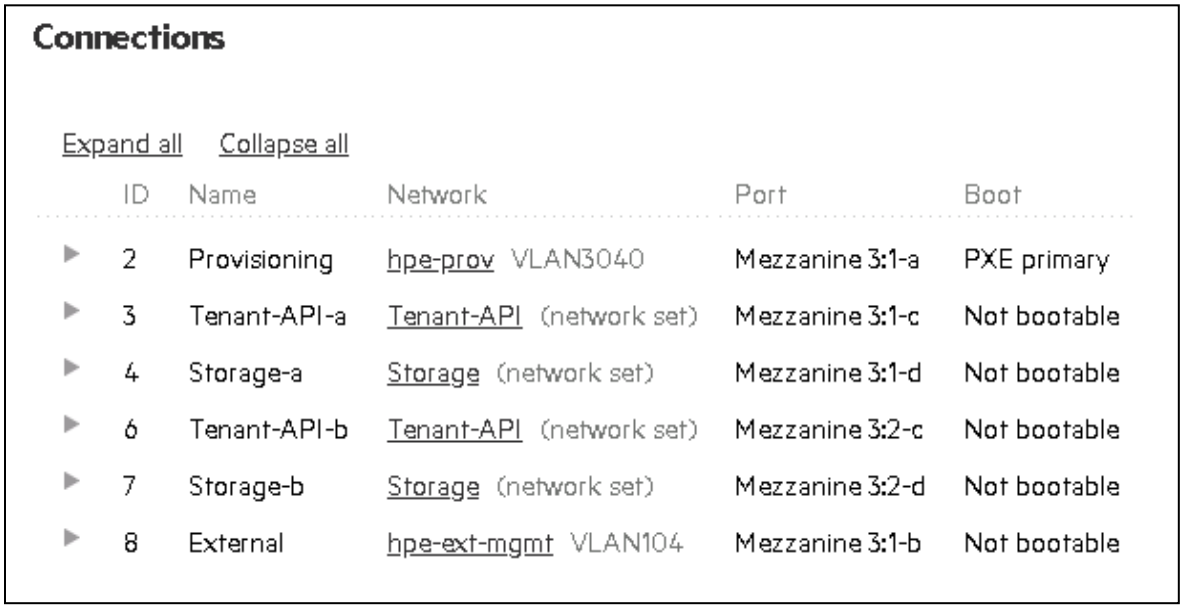


Figure 14. Network interfaces as defined by the controller server profile template

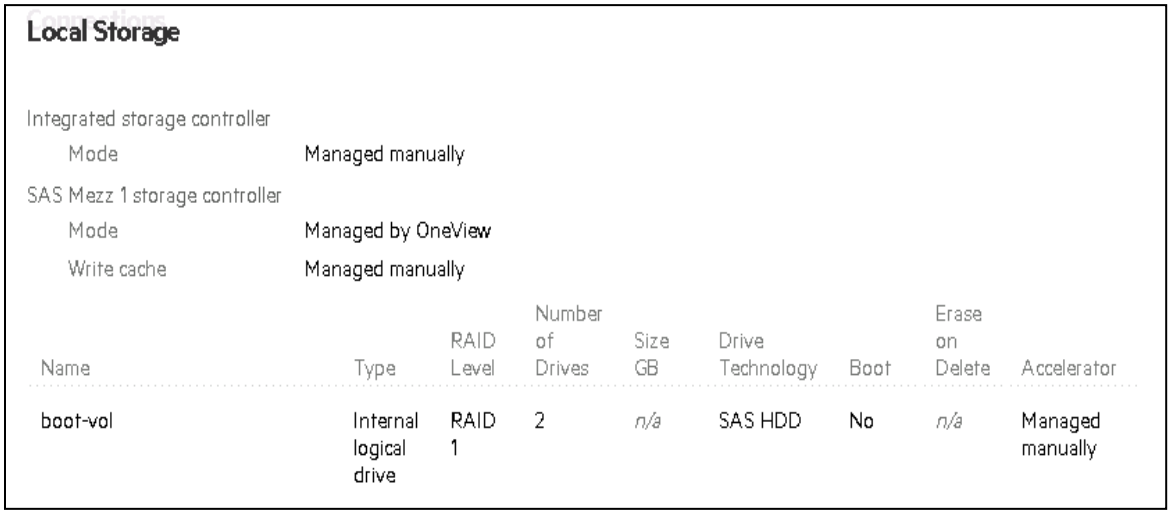


Figure 15. Storage interfaces as defined by the controller server profile template

Figure 16 shows the Network interfaces of Storage server profile template

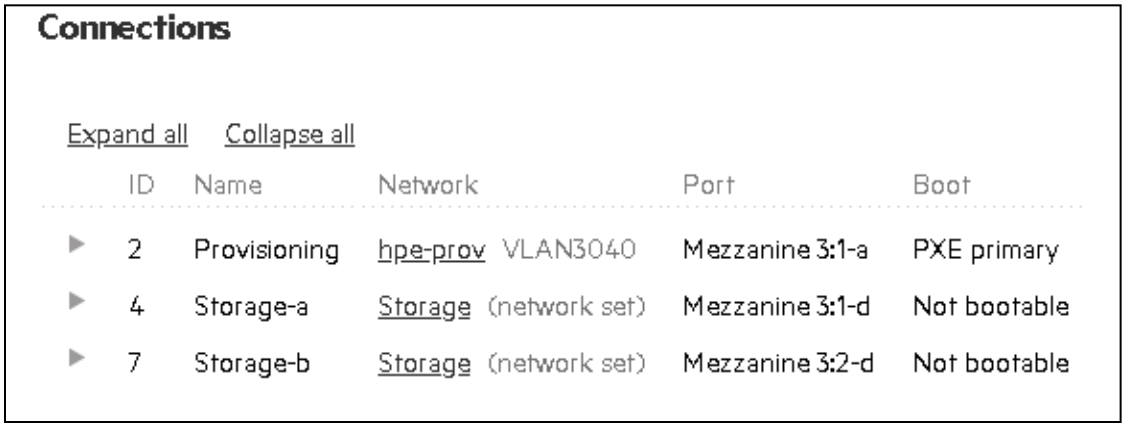


Figure 16. Network interfaces as defined by the storage server profile template

The storage arrays are defined and initialized in the server profiles. The storage connections in the server profile define the volumes that will be created on the host from physical disks presented from the internal storage drive bays and the D3940 Storage Module.

Figure 17 shows the network interfaces of director server profile template.

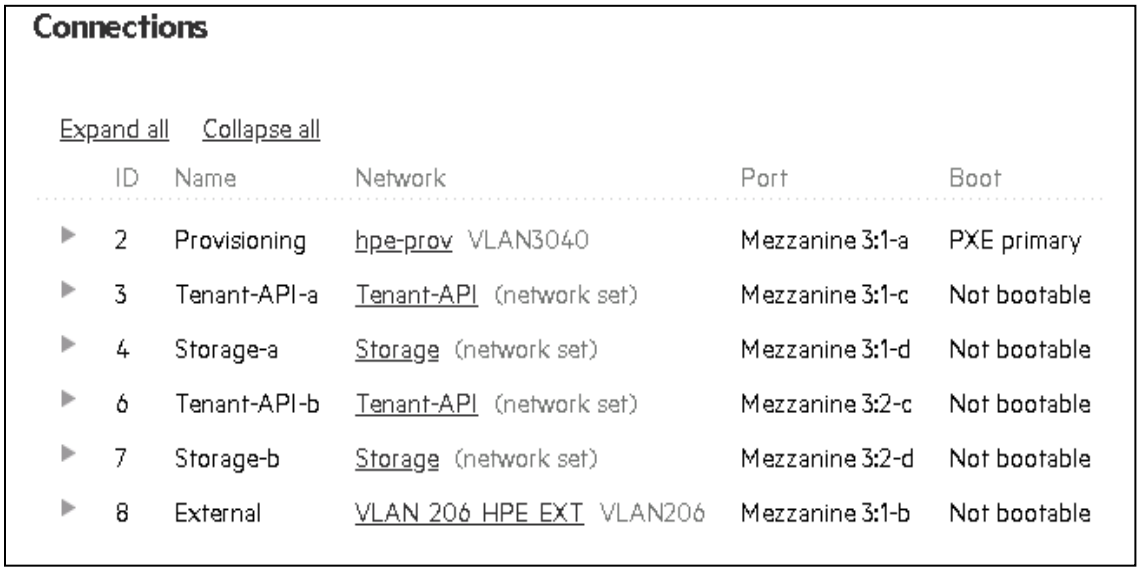
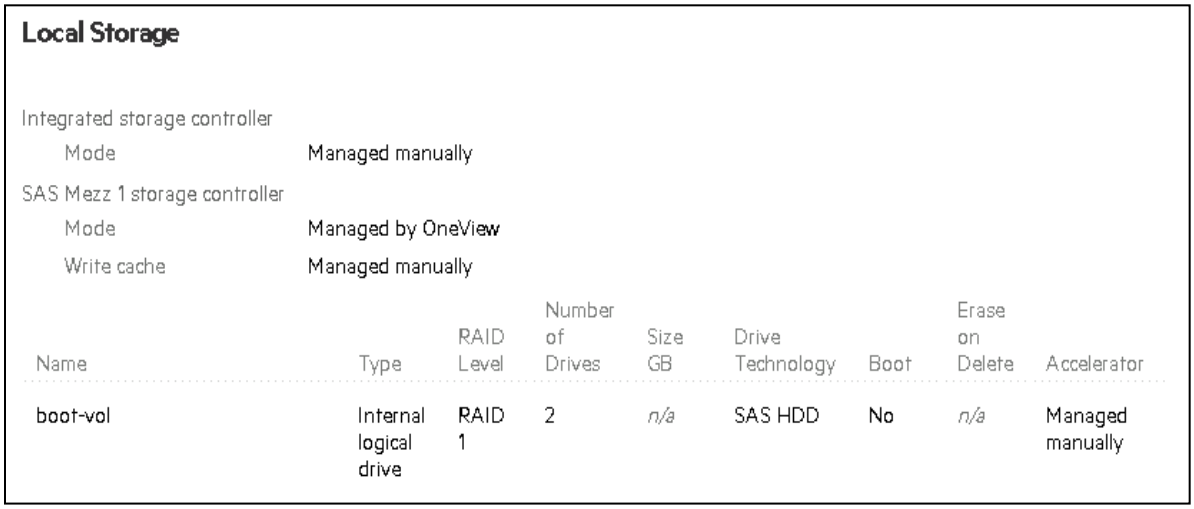


Figure 17. Network interfaces as defined by the director server profile template

Figure 18 shows the storage interfaces of director server profile template.



Local Storage

Integrated storage controller

Mode: Managed manually

SAS Mezz 1 storage controller

Mode: Managed by OneView

Write cache: Managed manually

Name	Type	RAID Level	Number of Drives	Size GB	Drive Technology	Boot	Erase on Delete	Accelerator
boot-vol	Internal logical drive	RAID 1	2	*n/a*	SAS HDD	No	*n/a*	Managed manually

Figure 18. Storage interfaces as defined by the director server profile template

Installation and configuration of the Red Hat OpenStack Platform 13 director

This section provides the installation and configuration details for installing both the Red Hat Open-Stack Platform 13 undercloud and overcloud with Red Hat Ceph Storage. Specifically, the installation and configuration details are provided for installing the RHOSP 13 director, customizing the environment configuration files (heat templates) for the overcloud deployment, and deploying the overcloud. Customization of the Red Hat OpenStack Platform 13 heat templates is performed to provide environment specific details for the overcloud deployment. Additionally, configuration details for the new monitoring and logging environments are provided in this section.

In this Reference Architecture, Red Hat OpenStack Platform 13 director is installed on a virtual machine. The virtual machine is hosted on a HPE Synergy server running Red Hat Enterprise Linux version 7.6 with KVM enabled. A virtual machine is created with the following requirements.

Table 3. Virtual machine specification

Components	Specification
CPU	8 virtual CPU cores
Memory	64 GiB RAM
Storage	125 GiB virtual disk
Two network adapters	Virtio network device
Software selection	Red Hat Enterprise Linux Server release version 7.6 (Maipo) infrastructure server

One network adapter of the VM is connected to the provisioning network and the other to the external network.

Customization of undercloud.conf file is needed prior to installing the undercloud. Refer to appendix B for full undercloud.conf.

```
[DEFAULT]
local_ip = 172.16.0.1/24
undercloud_public_vip = 172.16.0.10
```

```
undercloud_admin_vip = 172.16.0.11
local_interface = eth0
masquerade_network = 172.16.0.0/24
dhcp_start = 172.16.0.20
dhcp_end = 172.16.0.120
network_cidr = 172.16.0.0/24
network_gateway = 172.16.0.1
discovery_iprange = 172.16.0.150,172.16.0.180
inspection_iprange = 172.16.0.150,172.16.0.180
generate_service_certificate = false
inspection_interface = br-ctlplane
```

Execute the following command to install the undercloud:

```
$ openstack undercloud install
```

When the installation script completes source the `stackrc` file and verify the undercloud is operational using the following command:

```
$ openstack service list
```

Configure the undercloud

Install, copy and extract the image file

Install the rhosp-director-images and rhosp-director-images-ipa packages by issuing the following command:

```
$ sudo yum install rhosp-director-images rhosp-director-images-ipa
```

Extract the images archives to the images directory on the stack user's home (/home/stack/images):

```
$ mkdir ~/images
$ cd ~/images
$ for i in /usr/share/rhosp-director-images/overcloud-full-latest-13.0.tar
/usr/share/rhosp-
director-images/ironic-python-agent-latest-13.0.tar; do tar -xvf $i;
done
```

Modify image file

The overcloud image can be modified using virt-customize to add a specific password for the root user. This is an optional step, but can be useful when troubleshooting the overcloud deployment.

Upload the overcloud images to the undercloud glance repository. Update the DNS names servers by passing the UUID of the subnet and the DNS name servers.

Import these images into the director:

```
$ openstack overcloud image upload --image-path /home/stack/images/
```

To check these images have uploaded successfully, issue the below command:

```
$ openstack image list
```

This list will not show the introspection PXE images. The director copies these files to /httpboot.

Set the DNS names server on the undercloud subnet

Source the stackrc file to enable the director's command line tools, and set the name servers for the ctlplane-subnet subnet.

Configuring a container image source

All overcloud services are containerized, which means the overcloud requires access to a registry with the necessary container images. This chapter provides information on how to prepare the registry and your overcloud configuration to use container images for Red Hat OpenStack Platform.

Registry methods

In this Reference Architecture, we configured a local registry on the undercloud to store the Overcloud container images. The undercloud uses the Docker-distribution service to act as a registry. This allows the director to synchronize the images from registry.access.redhat.com and push them to the Docker-distribution registry. When creating, the overcloud pulls the container images from the undercloud's Docker-distribution registry. This method allows a user to store a registry internally, which can speed up the deployment and decrease network congestion. However, the undercloud only acts as a basic registry and provides limited life cycle management for container images.

Refer to section 5.5 of the Red Hat OpenStack Platform 13 director installation and usage guide, https://access.redhat.com/documentation/enus/red_hat_openstack_platform/13/pdf/director_installation_and_usage/Red_Hat_OpenStack_Platform-13-Director_Installation_and_Usage-en-US.pdf

Create instack.json and perform introspection

The introspection process will contact each node to be used in the overcloud deployment and build an inventory that will be stored in the swift data of the undercloud. The first step in performing an introspection is to create an instackenv.json file that contains authentication and connection information for each node. This Reference Architecture was tested with both the generic pxe_impitool driver and the pxe_ilo. The pxe_ilo driver is documented in this Reference Architecture. A new local user was created in iLo named root with the following account privileges; Administer User Accounts, Remote Console Access, Virtual Power and Reset, Virtual Media, and Configure iLO Settings. This account was used to perform the introspection.

Import the instack.json file to register the node with Red Hat OpenStack Platform director using the below command.

```
$ openstack baremetal import
```

This imports the template and registers each node from the template into the director. After the node registration and configuration completes, view a list of these nodes in the CLI using the following command:

```
$ openstack baremetal node list
```

Before performing the introspection using the below command,

```
$ openstack overcloud node introspect
```

Verify the introspection pxe images are installed in /httpboot.

Tagging nodes into profiles

To tag a node into a specific profile, add a profile option to the properties/capabilities parameter for each node. To tag your nodes to use Controller, Compute, and Storage profiles respectively, use the following command:

```
$ openstack baremetal node set
```

After completing node tagging, check the assigned profiles or possible profiles.

Configure Red Hat Ceph Storage node

Retrieving hardware introspection details: The bare metal service hardware inspection extras (in-spection_extras) is enabled by default to retrieve hardware details. You can use these hardware details to get the OSD disk information for Ceph.

For example, use the openstack baremetal introspection data save _UUID_ | jq ".inventory.disks" command to retrieve the disk information for configuring Ceph Storage, with the UUID of the bare metal node.

```
(undercloud) [stack@undercloud doc]$ openstack baremetal introspection data save
9c9262ab-2e92-43b9-b2be-b0c00acbfba5 | jq ".inventory.disks" > overcloud-ceph1.txt

(undercloud) [stack@undercloud doc]$ less overcloud-ceph1.txt
……..
……..
……..
  {
    "size": 3e+11,
    "serial": "5000c500b884cf97",
    "wwn": "0x5000c500b884cf97",
    "rotational": true,
    "vendor": "HP",
    "name": "/dev/sdc",
    "wwn_vendor_extension": null,
    "hctl": "1:0:2:0",
    "wwn_with_extension": "0x5000c500b884cf97",
    "by_path": "/dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b884cf95-lun-0",
    "model": "EG000300JWEBF"
```

```
  },
  {
    "size": 3e+11,
    "serial": "5000c500b884cf97",
    "wwn": "0x5000c500b884cf97",
    "rotational": true,
    "vendor": "HP",
    "name": "/dev/sde",
    "wwn_vendor_extension": null,
    "hctl": "1:0:3:0",
    "wwn_with_extension": "0x5000c500b884cf97",
    "by_path": "/dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b884cf96-lun-0",
    "model": "EG000300JWEBF"
  },
```

In this Reference Architecture, we use the disk path information in the /home/stack/templates/50-storage-environment.yaml as follows. Refer to appendix B for full 50-storage-environment.yaml.

```
parameter_defaults:
  CephAnsibleDisksConfig:
    osd_scenario: non-collocated
    devices:
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b884cf95-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b897276d-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8972899-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8973361-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b896a431-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8968b35-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8972826-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b89688f1-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x50000398e830d0da-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c419ed-lun-0
```

```
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c288aa-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x50000398e831d8d7-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c478aa-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c535e6-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c41eb6-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c4e5ae-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c4e6f5-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c50f6a-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c51bea-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c52f5e-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c3c74d-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c50b21-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c52579-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c16a7d-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c5265d-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c4e1d5-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c57df9-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c4f041-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c4ed59-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c56c8d-lun-0
  dedicated_devices:
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608900-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608900-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608900-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608900-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608900-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608901-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608901-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608901-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608901-lun-0
```

```
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608901-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608902-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608902-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608902-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608902-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608902-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608903-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608903-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608903-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608903-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608903-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608904-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608904-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608904-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608904-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608904-lun-0
    - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608905-lun-0
```

Note

As you see from the introspection data for the drives, there are two paths to the same drive. This is due to the dual SAS switches on Synergy. In Ceph the High Availability of the data is taken care by the replication or erasure coding features of Ceph, so while configuring Ceph we used the unique “by_path” instead of the “name” of the disk.

Heat template

In the stack user’s home directory, create a directory structure to hold any customized heat templates. In this Reference Architecture, there are ten files that hold our customized configuration templates and scripts.

Those files include:

```
/home/stack/templates/00-global-config.yaml
/usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml
/home/stack/templates/network-environment.yaml
/home/stack/templates/30-overcloud_images.yaml
/home/stack/templates/50-storage-environment.yaml
```

```
/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-
ansible.yaml
/usr/share/openstack-tripleo-heat-templates/environments/services-docker/octavia.yaml
/home/stack/templates/logging-environment.yaml
/home/stack/templates/monitoring-environment.yaml
/home/stack/templates/collectd-environment.yaml
```

The NTP server information along with the overcloud node counts and flavors is in the 00-global-config.yaml. The nic-configs directory mentioned in the network-isolation.yaml contains the network configuration files for the Controller, Compute, and Ceph-storage nodes. These files provide the configuration information for creating the bond interfaces, assigning the bonded interfaces to their respective bridge, and assigning the VLANs to the interface for the storage and cloud management trunks.

OpenStack is deployed onto the overcloud using the below command along with the heat templates:

```
$ openstack overcloud deploy

openstack overcloud deploy   --templates
  -e /home/stack/templates/00-global-config.yaml
  -e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml
  -e /home/stack/templates/network-environment.yaml
  -e /home/stack/templates/30-overcloud_images.yaml
  -e /home/stack/templates/50-storage-environment.yaml
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-
ansible.yaml
  -e /usr/share/openstack-tripleo-heat-templates/environments/services-
docker/octavia.yaml
  -e /home/stack/templates/logging-environment.yaml
  -e /home/stack/templates/monitoring-environment.yaml
  -e /home/stack/templates/collectd-environment.yaml
```

The overcloud creation process begins and the director provisions your nodes. This process takes some time to complete. To view the status of the overcloud creation, open a separate terminal as the stack user and run the following command:

```
$ openstack stack list –nested
```

Accessing the overcloud

After the OpenStack deployment completes successfully, users access the director's web UI through SSL. For example, in this deployment, the address to access the UI is http://192.168.1.121/

The web UI initially presents a login screen with fields for the following:

Username: The administration user for the director. The default is admin.

Password: The password for the administration user. Check the overcloudrc file for the password.

The below image show the admin view from the OpenStack dashboard horizon after the OpenStack deployment. You can see that two instances were created by the admin user to validate the OpenStack deployment.

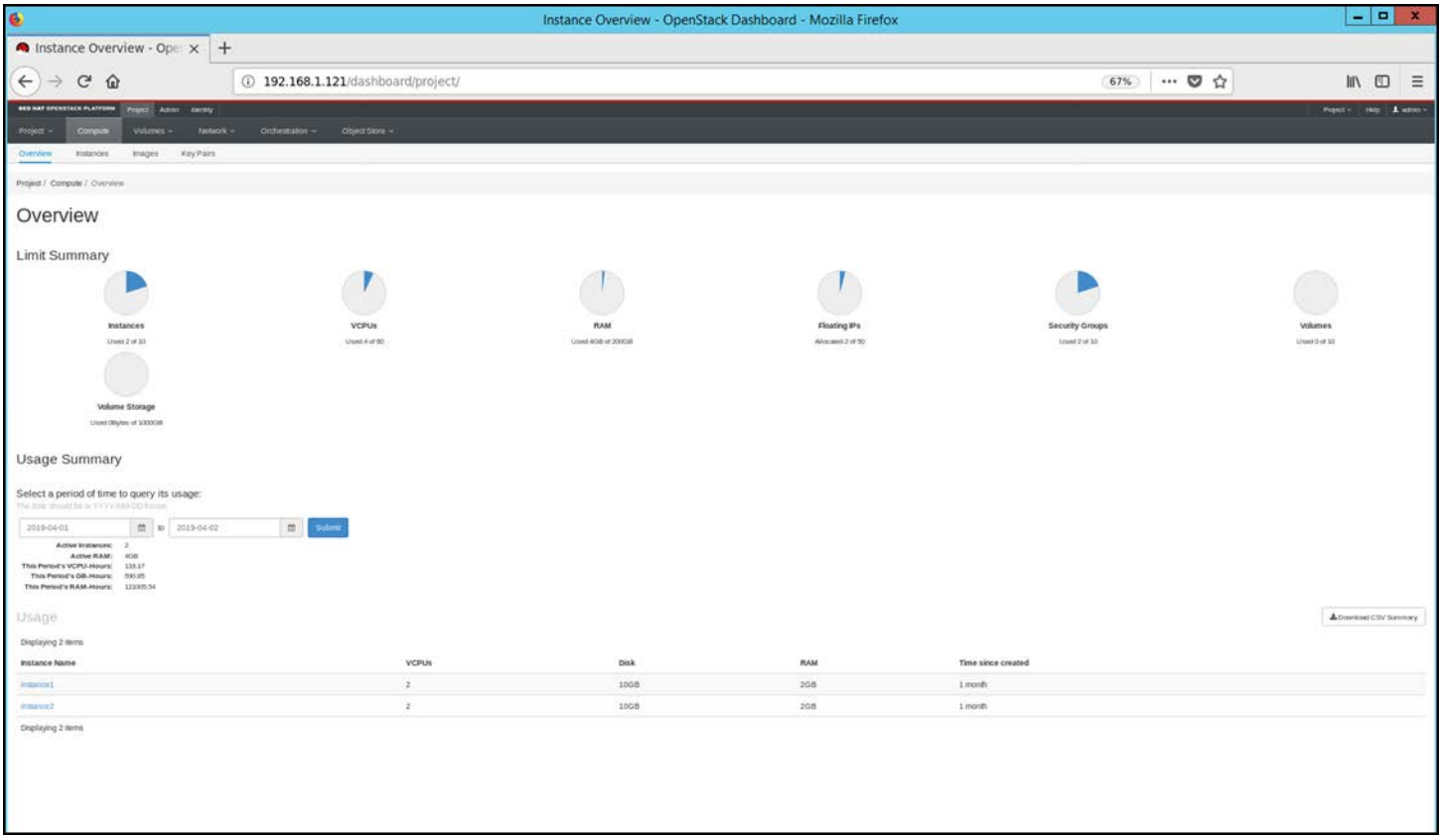


Figure 19. OpenStack dashboard

Monitoring and logging configuration

Monitoring tools are an optional suite of tools designed to help operators maintain an OpenStack environment. The tools perform the following functions:

Centralized logging: Allows you gather logs from all components in the OpenStack environment in one central location. You can identify problems across all nodes and services, and optionally, export the log data to Red Hat for assistance in diagnosing problems.

Availability monitoring: Allows you to monitor all components in the OpenStack environment and determine if any components are currently experiencing outages or are otherwise not functional. You can also configure the system to alert you when problems are identified.

Monitoring tools use a client-server model with the client deployed onto the Red Hat OpenStack Platform overcloud nodes. The Fluentd service provides client-side centralized logging (CL) and the Sensu client service provides client-side availability monitoring (AM).

To install the client side tools, copy environment files from the director's heat template collection and modify them to suit your environment. To set centralized logging client parameters set the fluent configuration in the logging-environment.yaml. For availability monitoring client set Sensu configuration setting in the monitoring-environment.

The yaml files for the operational tools are included with the OpenStack overcloud deploy command to install Sensu client and Fluentd tools on the overcloud nodes.

Install the server-side components

You can use the opstools ansible playbook to install the server-side components onto Red Hat Enterprise Linux version 7.

Note
Red Hat does not support the server-side components and their deployment process. These server-side components include availability monitoring and centralized logging services that complement the Red Hat supported client-side components.

For this Reference Architecture, the server-side components were installed on two Virtual Machines running RHEL 7.6 on the External Network (192.168.1.0/24). Each of the VMs had 16 GB RAM, 50 GB disk, and 8 VCPUs.

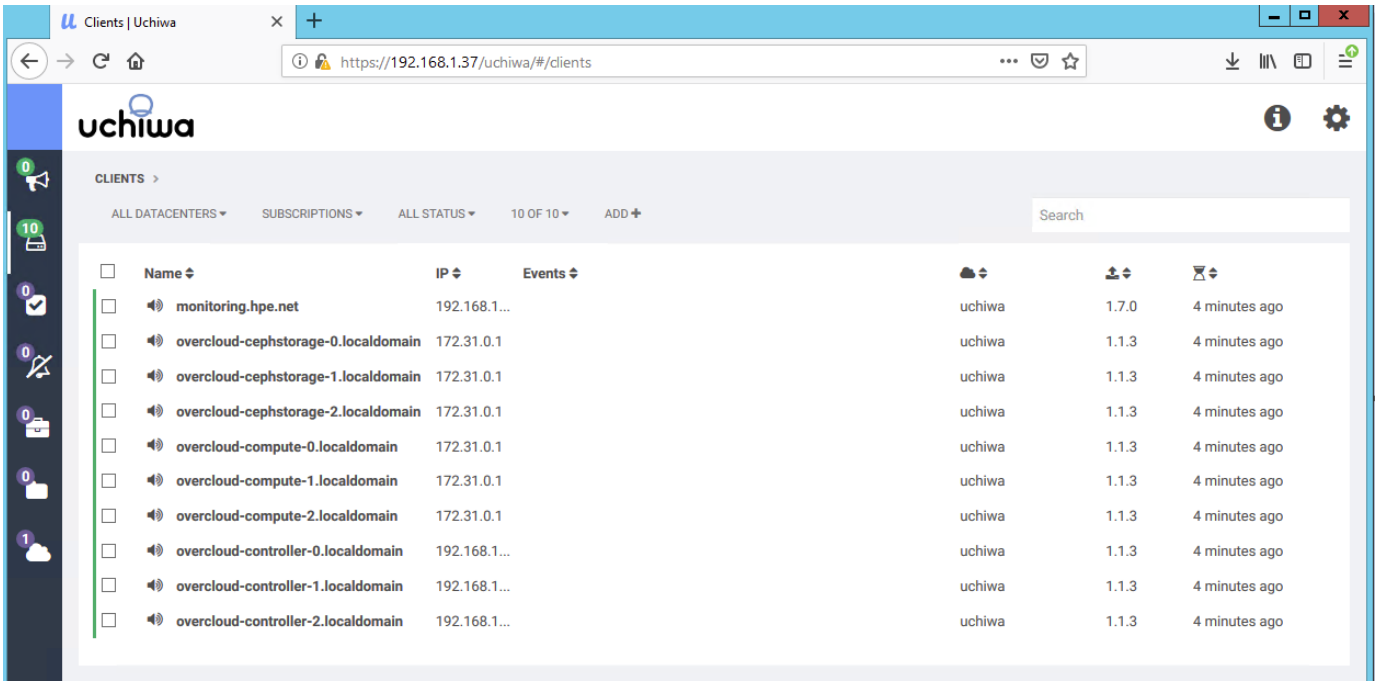


Figure 20. Monitoring and logging dashboard

For the details on the monitoring and logging environment, refer the "Red Hat OpenStack Platform 13 monitoring tools configuration guide", https://access.redhat.com/documentation/enus/red_hat_openstack_platform/13/pdf/monitoring_tools_configuration_guide/Red_Hat_OpenStack_Platform-13-Monitoring_Tools_Configuration_Guide-en-US.pdf

Post deployment validation

Post OpenStack deployment validation was performed by installing and configuring Red Hat OpenShift. When deployed on OpenStack, OpenShift Container Platform can be configured to access the OpenStack infrastructure, including using OpenStack Cinder volumes as persistent storage for application data. The following link provides the details for installing Red Hat OpenShift on Red Hat OpenStack Platform, https://docs.openshift.com/container-platform/3.11/install_config/configuring_openstack.html#inventory_provision

Figure 21 shows networks, instances, and routers created for an OpenShift deployment on OpenStack.

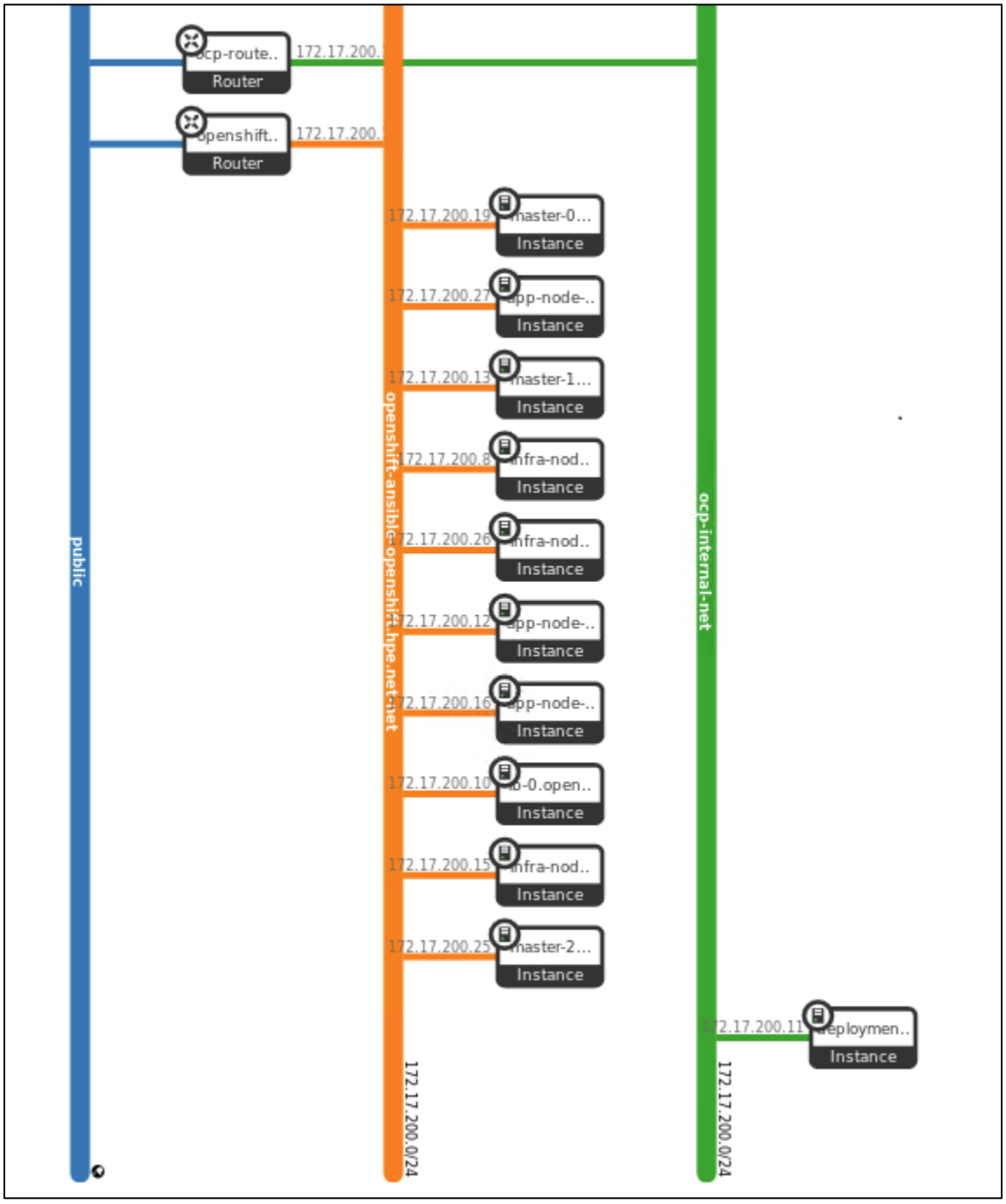


Figure 21. Instances, network and router created for an OpenShift deployment on OpenStack

Capacity and sizing

Analysis and recommendations

HPE Synergy deployment analysis

HPE Synergy Composer powered by HPE OneView can initialize the HPE Synergy Infrastructure ground up in a span of less than an hour. Using HPE OneView, the logical infrastructure can be created in the following manner that includes creation of network, Network Sets, logical interconnect groups, enclosure group, and logical enclosure. The entire process takes less than an hour that is also inclusive of applying the latest HPE Synergy firmware baseline across the frame.

Red Hat OpenStack Platform with an HPE Synergy three frame configuration

The solution presented in this document is based on a single HPE Synergy 12000 frame. This solution can easily be deployed in a three frame Synergy configuration. A three-frame configuration improves scalability and reliability of the solution. Distributing the solution across three HPE Synergy frames improves reliability of the solution and scalability of the solution.

In a three-frame HPE Synergy configuration, each Synergy 12000 frame is configured with a single OpenStack Controller Node, a single OpenStack Ceph Storage Node, and a single D3940 Storage Module, and one or more OpenStack Compute Nodes. This configuration spreads the OpenStack infrastructure across the three HPE Synergy frames which can reduce the possibility of an outage should one of the Synergy frames become degraded. Three Synergy 12000 enclosures provides 20 additional enclosure bays that allow for scaling out of OpenStack Compute Nodes from 3 to 23. Each Synergy 12000 controller will also require an additional D3940 Storage Module to provide direct attached storage for the OpenStack Ceph Storage Node. Going from a single D3940 Storage Module in a single frame solution to three D3940 Storage Modules in a three-frame solution increases the number of hard disk drives that can be used for Ceph Storage from 40 to 120. Increased storage and compute bays allow the Ceph Storage cluster to scale up and scale out, additional disks can be added to the existing three nodes Ceph cluster to scale up the cluster up by adding additional capacity. In addition, the Ceph cluster can be scaled out by adding additional Ceph storage nodes to the Ceph cluster providing both scale out and scale up of the Ceph cluster to increase performance and capacity of the Ceph cluster.

Server profiles and Red Hat OpenStack Platform Heat templates used for deployment in a single frame configuration can be used when going from a single frame solution to a multi-frame solution. The composable nature of networking provided by HPE Synergy and HPE OneView allows the network configuration defined for a single HPE Synergy Frame to span a multi-frame solution. The OpenStack networks defined for use by the Server profiles in a single frame solution can also be used and applied to server profiles in a multi-frame solution.

In a three-frame configuration the OpenStack Controller nodes, OpenStack Compute nodes, and OpenStack Ceph Storage nodes are equally distributed across the three frames as shown in Figure 22 below.

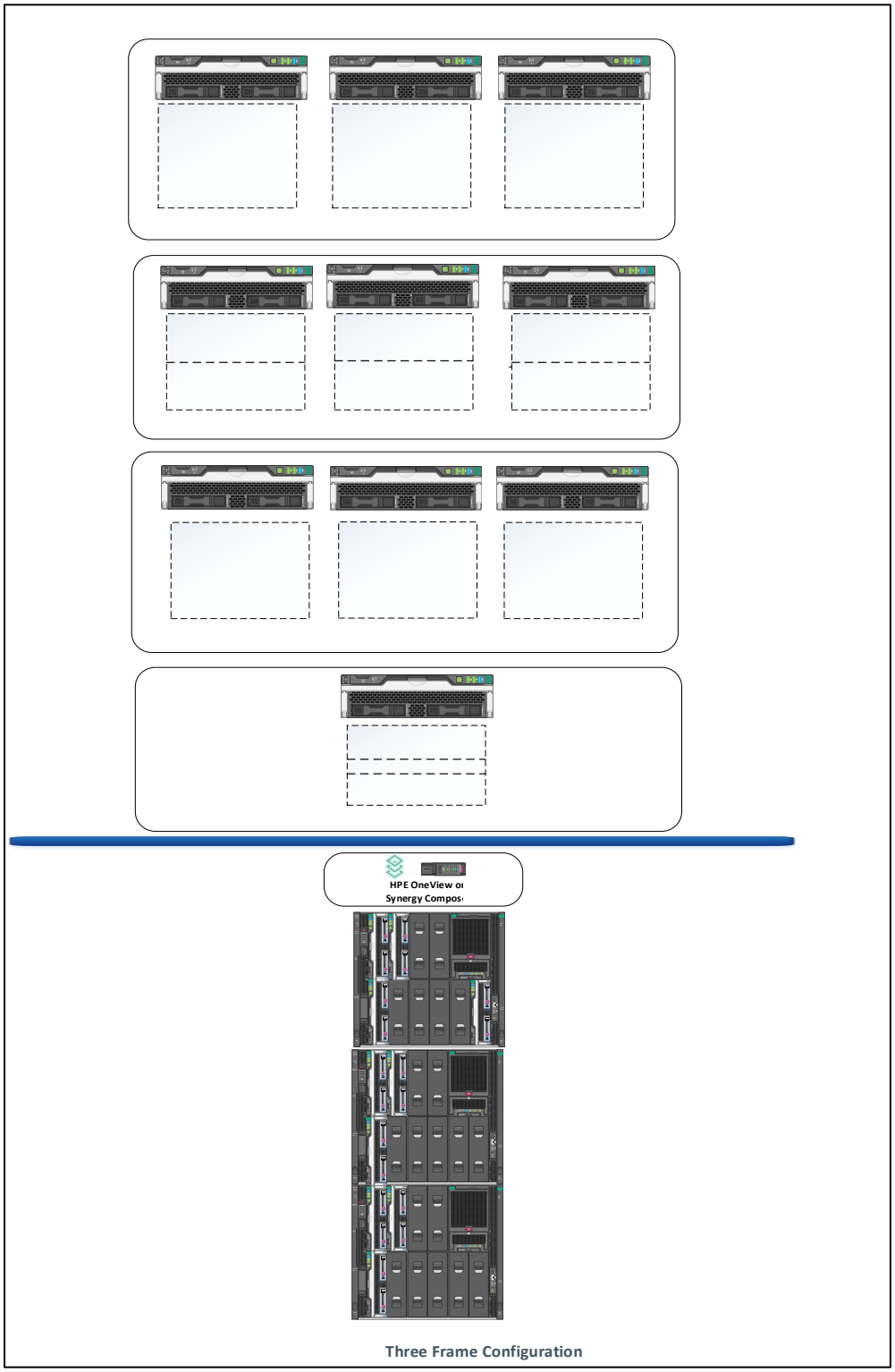


Figure 22. Red Hat OpenStack Platform with an HPE Synergy 3-frame configuration

Summary

With RHOS on HPE Synergy, HPE and Red Hat presents the composable infrastructure and modular design for running private cloud deployments. This Reference Architecture serves as a proof point as to how HPE Synergy combined with RHOS delivers an ideal platform for secure enterprise-ready private cloud.

For customers looking to build private cloud and cloud management services, HPE Synergy combined with RHOS provides a great platform to build a secure private cloud based on RHOS 13 and Ceph on HPE Synergy composable infrastructure. The combination has many benefits which customers have been excited about; among which some of them are mentioned below:

- **Cost effective** modern, modular and open solution that offers a choice to replace existing costly, rigid infrastructure and proprietary virtualization tools. Modular infrastructure to run RHOS 13 private cloud deployment not only reduces hardware complexity but also increases flexibility.
- **Open cloud** standardize on shared infrastructure with open source optimized and composable infrastructure with its RESTful API for production ready cloud environments
- **Flexible** composable IaaS cloud to dynamically expand capacity to meet demand peaks using templates for your workloads
- **Simplified infrastructure management:** The Simplified infrastructure management with HPE OneView to provision and manage physical and virtual infrastructure respectively is one of the biggest advantage of this combination. The workflow is to leverage OneView to create server profile templates with all physical compute, storage and networking attributes configured. This template is used to instantiate individual server profiles, assembling ratios of compute, storage and fabric.
- **Data center modernization:** With RHOS 13 and HPE Synergy, you can expand and contract both physical and virtual infrastructure to meet the changing business needs. This allows you to transform your data center by delivering quick results to line of businesses deploying a wide range of applications providing future proofing for any application – traditional or cloud-native.

Appendix A: Bill of materials

Note

Part numbers are at time of publication/testing and subject to change. The bill of materials does not include complete support options or other rack and power requirements. If you have questions regarding ordering, please consult with your Hewlett Packard Enterprise reseller or a Hewlett Packard Enterprise sales representative for more details. hpe.com/us/en/services/consulting.html

Table 4. Bill of materials

Qty	Part number	Description
		Rack and Network Infrastructure
1	P9K10A	HPE 42U 600x1200mm Enterprise Shock Rack
4	AF522A	HPE Intelligent 8.6kVA/L15-30P/NA/J PDU
1	HC790A	HPE Integration Center Routg Service FIO
1	BW932A	HPE 600mm Rack Stabilizer Kit
1	BW909A	HPE 42U 1200mm Side Panel Kit
1	JG505A	HPE 59xx CTO Switch Solution
2	JG510A	HPE 5900AF 48G 4XG 2QSFP+ Switch
4	JD096C	HPE X240 10G SFP+ SFP+ 1.2m DAC Cable
4	JC680A	HPE 58x0AF 650W AC Power Supply
4	JC682A	HPE 58x0AF Bck(pwr) Frt(prt) Fan Tray
1	797740-B21	HPE Synergy12000 CTO Frame 1xFLM 10x Fan

Qty	Part number	Description
1	798096-B21	HPE Synergy 12000F 6x 2650W AC Ti FIO PS
1	804353-B21	HPE Synergy Composer
1	804942-B21	HPE Synergy Frame Link Module
1	804938-B21	HPE Synergy 12000 Frame Rack Rail Option
1	804943-B21	HPE Synergy 12000 Frame 4x Lift Handle
6	Q0P72A	HPE 2.0m 250V 16A C19-C20 Sgl IPD Jpr Crd
10	871942-B21	HPE SY 480 Gen10 CTO Premium Cmpt Mdl (64Gb Memory)
10	872138-B21	HPE Synergy 480 Gen10 6142 Kit
10	872138-L21	HPE Synergy 480 Gen10 6142 FIO Kit
40	815100-B21	HPE 32GB 2Rx4 PC4-2666V-R Smart Kit
10	804428-B21	HPE Smart Array P416ie-m SR Gen10 Ctrlr
10	871573-B21	HPE SAS Cable for P416ie-m SR G10 Ctrlr
10	777430-B21	HHPE Synergy 3820C 10/20Gb CNA
10	P01367-B21	HPE 96W Smart Storage Battery 260mm Cbl
20	872475-B21	HPE 300GB SAS 10K SFF SC DS HDD
2	794502-B23	HPE VC SE 40Gb F8 Module
4	779218-B21	HPE Synergy 20Gb Interconnect Link Mod
2	755985-B21	HPE Synergy 12Gb SAS Connection Module
1	835386-B21	HPE Synergy D3940 CTO Storage Module
1	757323-B21	HPE Synergy D3940 IO Adapter
32	785067-B21	HPE 300GB 12G SAS 10K 2.5in SC ENT HDD
8	875470-B21	HPE 48680GB SATA MU SFF SC DS SSD
1	120672-B21	HPE Rack Ballast Kit
1	H6J85A	HPE Rack Hardware Kit
8	804101-B21	HPE Synergy Interconnect Link 3m AOC
2	P9S21A	HPE G2 Mtrd/Swtd 3P 14.4kVA/C13 NA/J PDU
4	720199-B21	HPE BLc 40G QSFP+ QSFP+ 3m DAC Cable
8	720193-B21	HPE BLc QSFP+ to SFP+ Adapter
8	455883-B21	HPE BLc 10G SFP+ SR Transceiver
8	AJ837A	HPE 15m Multi-mode OM3 LC/LC FC Cable
9	861412-B21	HPE CAT6A 4ft Cbl
2	838327-B21	HPE Synergy Dual 10GBASE-T QSFP 30m RJ45 Transceiver

Appendix B: Templates and Configuration files

B.1. undercloud.conf

```
[DEFAULT]
local_ip = 172.16.0.1/24
undercloud_public_vip = 172.16.0.10
undercloud_admin_vip = 172.16.0.11
local_interface = eth0
masquerade_network = 172.16.0.0/24
dhcp_start = 172.16.0.20
dhcp_end = 172.16.0.120
network_cidr = 172.16.0.0/24
network_gateway = 172.16.0.1
discovery_iprange = 172.16.0.150,172.16.0.180
inspection_iprange = 172.16.0.150,172.16.0.180
generate_service_certificate = false
inspection_interface = br-ctlplane

#
# From instack-undercloud
#

# Fully qualified hostname (including domain) to set on the
# Undercloud. If left unset, the current hostname will be used, but
# the user is responsible for configuring all system hostname settings
# appropriately.  If set, the undercloud install will configure all
# system hostname settings. (string value)
#undercloud_hostname = <None>

# IP information for the interface on the Undercloud that will be
# handling the PXE boots and DHCP for Overcloud instances.  The IP
# portion of the value will be assigned to the network interface
# defined by local_interface, with the netmask defined by the prefix
# portion of the value. (string value)
#local_ip = 192.168.24.1/24

# Virtual IP or DNS address to use for the public endpoints of
# Undercloud services. Only used with SSL. (string value)
# Deprecated group/name - [DEFAULT]/undercloud_public_vip
#undercloud_public_host = 192.168.24.2

# Virtual IP or DNS address to use for the admin endpoints of
# Undercloud services. Only used with SSL. (string value)
```

```
# Deprecated group/name - [DEFAULT]/undercloud_admin_vip
#undercloud_admin_host = 192.168.24.3

# DNS nameserver(s) to use for the undercloud node. (list value)
#undercloud_nameservers =

# List of ntp servers to use. (list value)
#undercloud_ntp_servers =

# DNS domain name to use when deploying the overcloud. The overcloud
# parameter "CloudDomain" must be set to a matching value. (string
# value)
#overcloud_domain_name = localdomain

# List of routed network subnets for provisioning and introspection.
# Comma separated list of names/tags. For each network a section/group
# needs to be added to the configuration file with these parameters
# set: cidr, dhcp_start, dhcp_end, inspection_iprange, gateway and
# masquerade. Note: The section/group must be placed before or after
# any other section. (See the example section [ctlplane-subnet] in the
# sample configuration file.) (list value)
#subnets = ctlplane-subnet

# Name of the local subnet, where the PXE boot and DHCP interfaces for
# overcloud instances is located. The IP address of the
# local_ip/local_interface should reside in this subnet. (string
# value)
#local_subnet = ctlplane-subnet

# Certificate file to use for OpenStack service SSL connections.
# Setting this enables SSL for the OpenStack API endpoints, leaving it
# unset disables SSL. (string value)
#undercloud_service_certificate =

# When set to True, an SSL certificate will be generated as part of
# the undercloud install and this certificate will be used in place of
# the value for undercloud_service_certificate.  The resulting
# certificate will be written to
# /etc/pki/tls/certs/undercloud-[undercloud_public_host].pem.  This
# certificate is signed by CA selected by the
# "certificate_generation_ca" option. (boolean value)
#generate_service_certificate = false

# The certmonger nickname of the CA from which the certificate will be
```

```
# requested. This is used only if the generate_service_certificate
# option is set. Note that if the "local" CA is selected the
# certmonger's local CA certificate will be extracted to /etc/pki/ca-
# trust/source/anchors/cm-local-ca.pem and subsequently added to the
# trust chain. (string value)
#certificate_generation_ca = local

# The kerberos principal for the service that will use the
# certificate. This is only needed if your CA requires a kerberos
# principal. e.g. with FreeIPA. (string value)
#service_principal =

# Network interface on the Undercloud that will be handling the PXE
# boots and DHCP for Overcloud instances. (string value)
#local_interface = eth1

# MTU to use for the local_interface. (integer value)
#local_mtu = 1500

# DEPRECATED: Network that will be masqueraded for external access, if
# required. This should be the subnet used for PXE booting. (string
# value)
# This option is deprecated for removal.
# Its value may be silently ignored in the future.
# Reason: With support for routed networks, masquerading of the
# provisioning networks is moved to a boolean option for each subnet.
#masquerade_network = 192.168.24.0/24

# Path to hieradata override file. If set, the file will be copied
# under /etc/puppet/hieradata and set as the first file in the hiera
# hierarchy. This can be used to custom configure services beyond what
# undercloud.conf provides (string value)
#hieradata_override =

# Path to network config override template. If set, this template will
# be used to configure the networking via os-net-config. Must be in
# json format. Templated tags can be used within the template, see
# instack-undercloud/elements/undercloud-stack-config/net-
# config.json.template for example tags (string value)
#net_config_override =

# Network interface on which inspection dnsmasq will listen.  If in
# doubt, use the default value. (string value)
# Deprecated group/name - [DEFAULT]/discovery_interface
```

```
#inspection_interface = br-ctlplane

# Whether to enable extra hardware collection during the inspection
# process. Requires python-hardware or python-hardware-detect package
# on the introspection image. (boolean value)
#inspection_extras = true

# Whether to run benchmarks when inspecting nodes. Requires
# inspection_extras set to True. (boolean value)
# Deprecated group/name - [DEFAULT]/discovery_runbench
#inspection_runbench = false

# Whether to support introspection of nodes that have UEFI-only
# firmware. (boolean value)
#inspection_enable_uefi = true

# Makes ironic-inspector enroll any unknown node that PXE-boots
# introspection ramdisk in Ironic. By default, the "fake" driver is
# used for new nodes (it is automatically enabled when this option is
# set to True). Set discovery_default_driver to override.
# Introspection rules can also be used to specify driver information
# for newly enrolled nodes. (boolean value)
#enable_node_discovery = false

# The default driver or hardware type to use for newly discovered
# nodes (requires enable_node_discovery set to True). It is
# automatically added to enabled_drivers or enabled_hardware_types
# accordingly. (string value)
#discovery_default_driver = ipmi

# Whether to enable the debug log level for Undercloud OpenStack
# services. (boolean value)
#undercloud_debug = true

# Whether to update packages during the Undercloud install. (boolean
# value)
#undercloud_update_packages = true

# Whether to install Tempest in the Undercloud. (boolean value)
#enable_tempest = true

# Whether to install Telemetry services (ceilometer, gnocchi, aodh,
# panko ) in the Undercloud. (boolean value)
#enable_telemetry = false
```

```
# Whether to install the TripleO UI. (boolean value)
#enable_ui = true

# Whether to install requirements to run the TripleO validations.
# (boolean value)
#enable_validations = true

# Whether to install the Volume service. It is not currently used in
# the undercloud. (boolean value)
#enable_cinder = false

# Whether to install novajoin metadata service in the Undercloud.
# (boolean value)
#enable_novajoin = false

# Array of host/port combiniations of docker insecure registries.
# (list value)
#docker_insecure_registries =

# One Time Password to register Undercloud node with an IPA server.
# Required when enable_novajoin = True. (string value)
#ipa_otp =

# Whether to use iPXE for deploy and inspection. (boolean value)
# Deprecated group/name - [DEFAULT]/ipxe_deploy
#ipxe_enabled = true

# Maximum number of attempts the scheduler will make when deploying
# the instance. You should keep it greater or equal to the number of
# bare metal nodes you expect to deploy at once to work around
# potential race condition when scheduling. (integer value)
# Minimum value: 1
#scheduler_max_attempts = 30

# Whether to clean overcloud nodes (wipe the hard drive) between
# deployments and after the introspection. (boolean value)
#clean_nodes = false

# DEPRECATED: List of enabled bare metal drivers. (list value)
# This option is deprecated for removal.
# Its value may be silently ignored in the future.
# Reason: Please switch to hardware types and the
# enabled_hardware_types option.
```

```
#enabled_drivers = pxe_ipmitool,pxe_drac,pxe_ilo

# List of enabled bare metal hardware types (next generation drivers).
# (list value)
#enabled_hardware_types = ipmi,redfish,ilo,idrac

# An optional docker 'registry-mirror' that will beconfigured in
# /etc/docker/daemon.json. (string value)
#docker_registry_mirror =

# List of additional architectures enabled in your cloud environment.
# The list of supported values is: ppc64le (list value)
#additional_architectures =

# Enable support for routed ctlplane networks. (boolean value)
#enable_routed_networks = false


[auth]

#
# From instack-undercloud
#

# Password used for MySQL root user. If left unset, one will be
# automatically generated. (string value)
#undercloud_db_password = <None>

# Keystone admin token. If left unset, one will be automatically
# generated. (string value)
#undercloud_admin_token = <None>

# Keystone admin password. If left unset, one will be automatically
# generated. (string value)
#undercloud_admin_password = <None>

# Glance service password. If left unset, one will be automatically
# generated. (string value)
#undercloud_glance_password = <None>

# Heat db encryption key(must be 16, 24, or 32 characters. If left
# unset, one will be automatically generated. (string value)
#undercloud_heat_encryption_key = <None>
```

```
# Heat service password. If left unset, one will be automatically
# generated. (string value)
#undercloud_heat_password = <None>

# Heat cfn service password. If left unset, one will be automatically
# generated. (string value)
#undercloud_heat_cfn_password = <None>

# Neutron service password. If left unset, one will be automatically
# generated. (string value)
#undercloud_neutron_password = <None>

# Nova service password. If left unset, one will be automatically
# generated. (string value)
#undercloud_nova_password = <None>

# Ironic service password. If left unset, one will be automatically
# generated. (string value)
#undercloud_ironic_password = <None>

# Aodh service password. If left unset, one will be automatically
# generated. (string value)
#undercloud_aodh_password = <None>

# Gnocchi service password. If left unset, one will be automatically
# generated. (string value)
#undercloud_gnocchi_password = <None>

# Ceilometer service password. If left unset, one will be
# automatically generated. (string value)
#undercloud_ceilometer_password = <None>

# Panko service password. If left unset, one will be automatically
# generated. (string value)
#undercloud_panko_password = <None>

# Ceilometer metering secret. If left unset, one will be automatically
# generated. (string value)
#undercloud_ceilometer_metering_secret = <None>

# Ceilometer snmpd read-only user. If this value is changed from the
# default, the new value must be passed in the overcloud environment
# as the parameter SnmpdReadonlyUserName. This value must be between 1
# and 32 characters long. (string value)
```

```
#undercloud_ceilometer_snmpd_user = ro_snmp_user

# Ceilometer snmpd password. If left unset, one will be automatically
# generated. (string value)
#undercloud_ceilometer_snmpd_password = <None>

# Swift service password. If left unset, one will be automatically
# generated. (string value)
#undercloud_swift_password = <None>

# Mistral service password. If left unset, one will be automatically
# generated. (string value)
#undercloud_mistral_password = <None>

# Rabbitmq cookie. If left unset, one will be automatically generated.
# (string value)
#undercloud_rabbit_cookie = <None>

# Rabbitmq password. If left unset, one will be automatically
# generated. (string value)
#undercloud_rabbit_password = <None>

# Rabbitmq username. If left unset, one will be automatically
# generated. (string value)
#undercloud_rabbit_username = <None>

# Heat stack domain admin password. If left unset, one will be
# automatically generated. (string value)
#undercloud_heat_stack_domain_admin_password = <None>

# Swift hash suffix. If left unset, one will be automatically
# generated. (string value)
#undercloud_swift_hash_suffix = <None>

# HAProxy stats password. If left unset, one will be automatically
# generated. (string value)
#undercloud_haproxy_stats_password = <None>

# Zaqar password. If left unset, one will be automatically generated.
# (string value)
#undercloud_zaqar_password = <None>

# Horizon secret key. If left unset, one will be automatically
# generated. (string value)
```

```
#undercloud_horizon_secret_key = <None>

# Cinder service password. If left unset, one will be automatically
# generated. (string value)
#undercloud_cinder_password = <None>

# Novajoin vendordata plugin service password. If left unset, one will
# be automatically generated. (string value)
#undercloud_novajoin_password = <None>


[ctlplane-subnet]

#
# From instack-undercloud
#

# Network CIDR for the Neutron-managed subnet for Overcloud instances.
# (string value)
# Deprecated group/name - [DEFAULT]/network_cidr
#cidr = 192.168.24.0/24

# Start of DHCP allocation range for PXE and DHCP of Overcloud
# instances on this network. (string value)
# Deprecated group/name - [DEFAULT]/dhcp_start
#dhcp_start = 192.168.24.5

# End of DHCP allocation range for PXE and DHCP of Overcloud instances
# on this network. (string value)
# Deprecated group/name - [DEFAULT]/dhcp_end
#dhcp_end = 192.168.24.24

# Temporary IP range that will be given to nodes on this network
# during the inspection process. Should not overlap with the range
# defined by dhcp_start and dhcp_end, but should be in the same ip
# subnet. (string value)
# Deprecated group/name - [DEFAULT]/inspection_iprange
#inspection_iprange = 192.168.24.100,192.168.24.120

# Network gateway for the Neutron-managed network for Overcloud
# instances on this network. (string value)
# Deprecated group/name - [DEFAULT]/network_gateway
#gateway = 192.168.24.1
```

```
# The network will be masqueraded for external access. (boolean value)
#masquerade = false
```

B.2. run_deploy.sh

```
#!/bin/bash

set -o verbose
source stackrc
pushd /home/stack

 openstack overcloud deploy \
  --templates \
  -e /home/stack/templates/00-global-config.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml
\
  -e /home/stack/templates/network-environment.yaml \
  -e /home/stack/templates/30-overcloud_images.yaml \
  -e /home/stack/templates/50-storage-environment.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-
ansible.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/services-
docker/octavia.yaml \
  -e /home/stack/templates/logging-environment.yaml \
  -e /home/stack/templates/monitoring-environment.yaml \
  -e /home/stack/templates/collectd-environment.yaml

echo DONE
```

B.3. 00-global-config.yaml

```
parameter_defaults:
  NodeRootPassword: redhat
  ControllerCount: 3
  OvercloudControllerFlavor: control
  ComputeCount: 3
  OvercloudComputeFlavor: compute
  CephStorageCount: 3
  OvercloudCephStorageFlavor: ceph-storage

# Misc Config
  NtpServer: '15.226.47.253'
```

```
  NeutronNetworkType: 'vxlan'
  NeutronTunnelTypes: 'vxlan'
```

B.4. network-environment.yaml

```
resource_registry:
  OS::TripleO::Compute::Net::SoftwareConfig: /home/stack/templates/nic-
configs/compute.yaml
  OS::TripleO::Controller::Net::SoftwareConfig: /home/stack/templates/nic-
configs/controller.yaml
  OS::TripleO::CephStorage::Net::SoftwareConfig: /home/stack/templates/nic-
configs/ceph-storage.yaml

parameter_defaults:
  # This sets 'external_network_bridge' in l3_agent.ini to an empty string
  # so that external networks act like provider bridge networks (they
  # will plug into br-int instead of br-ex)
  NeutronExternalNetworkBridge: "''"

  # Internal API used for private OpenStack Traffic
  InternalApiNetCidr: 172.17.1.0/24
  InternalApiAllocationPools: [{'start': '172.17.1.10', 'end': '172.17.1.200'}]
  InternalApiNetworkVlanID: 3041
# InternalApiNetworkVlanID: 101

  # Tenant Network Traffic - will be used for VXLAN over VLAN
  TenantNetCidr: 172.17.2.0/24
  TenantAllocationPools: [{'start': '172.17.2.10', 'end': '172.17.2.200'}]
  TenantNetworkVlanID: 3044
# TenantNetworkVlanID: 201

  # Public Storage Access - e.g. Nova/Glance <--> Ceph
  StorageNetCidr: 172.17.3.0/24
  StorageAllocationPools: [{'start': '172.17.3.10', 'end': '172.17.3.200'}]
  StorageNetworkVlanID: 3042
#  StorageNetworkVlanID: 301

  # Private Storage Access - i.e. Ceph background cluster/replication
  StorageMgmtNetCidr: 172.17.4.0/24
  StorageMgmtAllocationPools: [{'start': '172.17.4.10', 'end': '172.17.4.200'}]
  StorageMgmtNetworkVlanID: 3043
#  StorageMgmtNetworkVlanID: 401

  # External Networking Access - Public API Access
```

```
  ExternalNetCidr: 192.168.1.0/24
  # Leave room for floating IPs in the External allocation pool (if required)
  ExternalAllocationPools: [{'start': '192.168.1.100', 'end': '192.168.1.129'}]
  # Set to the router gateway on the external network
  ExternalInterfaceDefaultRoute: 192.168.1.11

  # Add in configuration for the Control Plane
  ControlPlaneSubnetCidr: "24"
  ControlPlaneDefaultRoute: 172.16.0.1
  EC2MetadataIp: 172.16.0.1
  DnsServers: ['15.226.47.253', '8.8.8.8']

  # Bonding options (only active/backup works in a virtual environment)
  BondInterfaceOvsOptions: 'mode=1 miimon=150'
```

B.5. 30-overcloud_images.yaml

```
# Generated with the following on 2019-02-28T01:14:56.478907
#
#   openstack overcloud container image prepare --
namespace=registry.access.redhat.com/rhosp13 --push-destination 172.16.0.1:8787 --
prefix=openstack- --tag-from-label {version}-{release} --output-env-
file=/home/stack/templates/overcloud_images.yaml --output-images-file
/home/stack/local_registry_images.yaml -e /usr/share/openstack-tripleo-heat-
templates/environments/services-docker/octavia.yaml -e /usr/share/openstack-tripleo-
heat-templates/environments/ceph-ansible/ceph-ansible.yaml -e
/home/stack/templates/50-storage-environment.yaml
#

parameter_defaults:
  DockerAodhApiImage: 172.16.0.1:8787/rhosp13/openstack-aodh-api:13.0-66
  DockerAodhConfigImage: 172.16.0.1:8787/rhosp13/openstack-aodh-api:13.0-66
  DockerAodhEvaluatorImage: 172.16.0.1:8787/rhosp13/openstack-aodh-evaluator:13.0-66
  DockerAodhListenerImage: 172.16.0.1:8787/rhosp13/openstack-aodh-listener:13.0-65
  DockerAodhNotifierImage: 172.16.0.1:8787/rhosp13/openstack-aodh-notifier:13.0-66
  DockerCeilometerCentralImage: 172.16.0.1:8787/rhosp13/openstack-ceilometer-
central:13.0-63
  DockerCeilometerComputeImage: 172.16.0.1:8787/rhosp13/openstack-ceilometer-
compute:13.0-65
  DockerCeilometerConfigImage: 172.16.0.1:8787/rhosp13/openstack-ceilometer-
central:13.0-63
  DockerCeilometerNotificationImage: 172.16.0.1:8787/rhosp13/openstack-ceilometer-
notification:13.0-65
```

```
  DockerCephDaemonImage: 172.16.0.1:8787/rhceph/rhceph-3-rhel7:3-20
  DockerCinderApiImage: 172.16.0.1:8787/rhosp13/openstack-cinder-api:13.0-68
  DockerCinderConfigImage: 172.16.0.1:8787/rhosp13/openstack-cinder-api:13.0-68
  DockerCinderSchedulerImage: 172.16.0.1:8787/rhosp13/openstack-cinder-
scheduler:13.0-70
  DockerCinderVolumeImage: 172.16.0.1:8787/rhosp13/openstack-cinder-volume:13.0-68
  DockerClustercheckConfigImage: 172.16.0.1:8787/rhosp13/openstack-mariadb:13.0-67
  DockerClustercheckImage: 172.16.0.1:8787/rhosp13/openstack-mariadb:13.0-67
  DockerCrondConfigImage: 172.16.0.1:8787/rhosp13/openstack-cron:13.0-70
  DockerCrondImage: 172.16.0.1:8787/rhosp13/openstack-cron:13.0-70
  DockerGlanceApiConfigImage: 172.16.0.1:8787/rhosp13/openstack-glance-api:13.0-69
  DockerGlanceApiImage: 172.16.0.1:8787/rhosp13/openstack-glance-api:13.0-69
  DockerGnocchiApiImage: 172.16.0.1:8787/rhosp13/openstack-gnocchi-api:13.0-66
  DockerGnocchiConfigImage: 172.16.0.1:8787/rhosp13/openstack-gnocchi-api:13.0-66
  DockerGnocchiMetricdImage: 172.16.0.1:8787/rhosp13/openstack-gnocchi-metricd:13.0-
67
  DockerGnocchiStatsdImage: 172.16.0.1:8787/rhosp13/openstack-gnocchi-statsd:13.0-66
  DockerHAProxyConfigImage: 172.16.0.1:8787/rhosp13/openstack-haproxy:13.0-67
  DockerHAProxyImage: 172.16.0.1:8787/rhosp13/openstack-haproxy:13.0-67
  DockerHeatApiCfnConfigImage: 172.16.0.1:8787/rhosp13/openstack-heat-api-cfn:13.0-64
  DockerHeatApiCfnImage: 172.16.0.1:8787/rhosp13/openstack-heat-api-cfn:13.0-64
  DockerHeatApiConfigImage: 172.16.0.1:8787/rhosp13/openstack-heat-api:13.0-65
  DockerHeatApiImage: 172.16.0.1:8787/rhosp13/openstack-heat-api:13.0-65
  DockerHeatConfigImage: 172.16.0.1:8787/rhosp13/openstack-heat-api:13.0-65
  DockerHeatEngineImage: 172.16.0.1:8787/rhosp13/openstack-heat-engine:13.0-63
  DockerHorizonConfigImage: 172.16.0.1:8787/rhosp13/openstack-horizon:13.0-64
  DockerHorizonImage: 172.16.0.1:8787/rhosp13/openstack-horizon:13.0-64
  DockerInsecureRegistryAddress:
  - 172.16.0.1:8787
  DockerIscsidConfigImage: 172.16.0.1:8787/rhosp13/openstack-iscsid:13.0-64
  DockerIscsidImage: 172.16.0.1:8787/rhosp13/openstack-iscsid:13.0-64
  DockerKeystoneConfigImage: 172.16.0.1:8787/rhosp13/openstack-keystone:13.0-64
  DockerKeystoneImage: 172.16.0.1:8787/rhosp13/openstack-keystone:13.0-64
  DockerMemcachedConfigImage: 172.16.0.1:8787/rhosp13/openstack-memcached:13.0-66
  DockerMemcachedImage: 172.16.0.1:8787/rhosp13/openstack-memcached:13.0-66
  DockerMysqlClientConfigImage: 172.16.0.1:8787/rhosp13/openstack-mariadb:13.0-67
  DockerMysqlConfigImage: 172.16.0.1:8787/rhosp13/openstack-mariadb:13.0-67
  DockerMysqlImage: 172.16.0.1:8787/rhosp13/openstack-mariadb:13.0-67
  DockerNeutronApiImage: 172.16.0.1:8787/rhosp13/openstack-neutron-server:13.0-72
  DockerNeutronConfigImage: 172.16.0.1:8787/rhosp13/openstack-neutron-server:13.0-72
  DockerNeutronDHCPImage: 172.16.0.1:8787/rhosp13/openstack-neutron-dhcp-agent:13.0-
74.1549896444
  DockerNeutronL3AgentImage: 172.16.0.1:8787/rhosp13/openstack-neutron-l3-agent:13.0-
72.1549896439
```

```
  DockerNeutronMetadataImage: 172.16.0.1:8787/rhosp13/openstack-neutron-metadata-
agent:13.0-75
  DockerNovaApiImage: 172.16.0.1:8787/rhosp13/openstack-nova-api:13.0-73
  DockerNovaComputeImage: 172.16.0.1:8787/rhosp13/openstack-nova-compute:13.0-
78.1548959796
  DockerNovaConductorImage: 172.16.0.1:8787/rhosp13/openstack-nova-conductor:13.0-71
  DockerNovaConfigImage: 172.16.0.1:8787/rhosp13/openstack-nova-api:13.0-73
  DockerNovaConsoleauthImage: 172.16.0.1:8787/rhosp13/openstack-nova-
consoleauth:13.0-71
  DockerNovaLibvirtConfigImage: 172.16.0.1:8787/rhosp13/openstack-nova-compute:13.0-
78.1548959796
  DockerNovaLibvirtImage: 172.16.0.1:8787/rhosp13/openstack-nova-libvirt:13.0-
79.1548959794
  DockerNovaMetadataImage: 172.16.0.1:8787/rhosp13/openstack-nova-api:13.0-73
  DockerNovaPlacementConfigImage: 172.16.0.1:8787/rhosp13/openstack-nova-placement-
api:13.0-72
  DockerNovaPlacementImage: 172.16.0.1:8787/rhosp13/openstack-nova-placement-
api:13.0-72
  DockerNovaSchedulerImage: 172.16.0.1:8787/rhosp13/openstack-nova-scheduler:13.0-73
  DockerNovaVncProxyImage: 172.16.0.1:8787/rhosp13/openstack-nova-novncproxy:13.0-74
  DockerOctaviaApiImage: 172.16.0.1:8787/rhosp13/openstack-octavia-api:13.0-64
  DockerOctaviaConfigImage: 172.16.0.1:8787/rhosp13/openstack-octavia-api:13.0-64
  DockerOctaviaHealthManagerImage: 172.16.0.1:8787/rhosp13/openstack-octavia-health-
manager:13.0-66
  DockerOctaviaHousekeepingImage: 172.16.0.1:8787/rhosp13/openstack-octavia-
housekeeping:13.0-66
  DockerOctaviaWorkerImage: 172.16.0.1:8787/rhosp13/openstack-octavia-worker:13.0-66
  DockerOpenvswitchImage: 172.16.0.1:8787/rhosp13/openstack-neutron-openvswitch-
agent:13.0-73
  DockerPankoApiImage: 172.16.0.1:8787/rhosp13/openstack-panko-api:13.0-66
  DockerPankoConfigImage: 172.16.0.1:8787/rhosp13/openstack-panko-api:13.0-66
  DockerRabbitmqConfigImage: 172.16.0.1:8787/rhosp13/openstack-rabbitmq:13.0-68
  DockerRabbitmqImage: 172.16.0.1:8787/rhosp13/openstack-rabbitmq:13.0-68
  DockerRedisConfigImage: 172.16.0.1:8787/rhosp13/openstack-redis:13.0-69
  DockerRedisImage: 172.16.0.1:8787/rhosp13/openstack-redis:13.0-69
  DockerSwiftAccountImage: 172.16.0.1:8787/rhosp13/openstack-swift-account:13.0-65
  DockerSwiftConfigImage: 172.16.0.1:8787/rhosp13/openstack-swift-proxy-server:13.0-
67
  DockerSwiftContainerImage: 172.16.0.1:8787/rhosp13/openstack-swift-container:13.0-
68
  DockerSwiftObjectImage: 172.16.0.1:8787/rhosp13/openstack-swift-object:13.0-65
  DockerSwiftProxyImage: 172.16.0.1:8787/rhosp13/openstack-swift-proxy-server:13.0-67
```

B.6. 50-storage-environment.yaml

```
resource_registry:
  OS::TripleO::NodeUserData: /home/stack/templates/firstboot/wipe-disks.yaml

parameter_defaults:
  CephAnsibleDisksConfig:
    osd_scenario: non-collocated
    devices:
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b884cf95-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b897276d-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8972899-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8973361-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b896a431-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8968b35-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8972826-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b89688f1-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x50000398e830d0da-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c419ed-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c288aa-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x50000398e831d8d7-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c478aa-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c535e6-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c41eb6-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c4e5ae-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c4e6f5-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c50f6a-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c51bea-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c52f5e-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c3c74d-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c50b21-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c52579-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c16a7d-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c5265d-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c4e1d5-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c57df9-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c4f041-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c4ed59-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5000c500b8c56c8d-lun-0
    dedicated_devices:
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608900-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608900-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608900-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608900-lun-0
```

```
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608900-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608901-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608901-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608901-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608901-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608901-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608902-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608902-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608902-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608902-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608902-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608903-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608903-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608903-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608903-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608903-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608904-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608904-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608904-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608904-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608904-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608905-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608905-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608905-lun-0
      - /dev/disk/by-path/pci-0000:5c:00.0-sas-0x5001438040608905-lun-0
    journal_size: 8192
  CephConfigOverrides:
    mon_max_pg_per_osd: 2048
```

B.7. ceph-ansible.yaml

```
resource_registry:
  OS::TripleO::Services::CephMgr: ../../docker/services/ceph-ansible/ceph-mgr.yaml
  OS::TripleO::Services::CephMon: ../../docker/services/ceph-ansible/ceph-mon.yaml
  OS::TripleO::Services::CephOSD: ../../docker/services/ceph-ansible/ceph-osd.yaml
  OS::TripleO::Services::CephClient: ../../docker/services/ceph-ansible/ceph-
client.yaml

parameter_defaults:
  CinderEnableIscsiBackend: false
  CinderEnableRbdBackend: true
  CinderBackupBackend: ceph
```

```
  NovaEnableRbdBackend: true
  GlanceBackend: rbd
  GnocchiBackend: rbd
```

B.8. octavia.yaml

```
resource_registry:
  OS::TripleO::Services::OctaviaApi: ../../docker/services/octavia-api.yaml
  OS::TripleO::Services::OctaviaHousekeeping: ../../docker/services/octavia-
housekeeping.yaml
  OS::TripleO::Services::OctaviaHealthManager: ../../docker/services/octavia-health-
manager.yaml
  OS::TripleO::Services::OctaviaWorker: ../../docker/services/octavia-worker.yaml
  OS::TripleO::Services::OctaviaDeploymentConfig:
../../docker/services/octavia/octavia-deployment-config.yaml

parameter_defaults:
    NeutronServicePlugins: "qos,router,trunk"
    NeutronEnableForceMetadata: true

    # This flag enables internal generation of certificates for communication
    # with amphorae. Use OctaviaCaCert, OctaviaCaKey, OctaviaCaKeyPassphrase
    # and OctaviaClient cert to configure secure production environments.
    OctaviaGenerateCerts: true
```

B.9. logging-environment.yaml

```
## A Heat environment file which can be used to set up
## logging agents

resource_registry:
  OS::TripleO::Services::Fluentd: /usr/share/openstack-tripleo-heat-
templates/docker/services/fluentd.yaml

parameter_defaults:
  LoggingServers:
    - host: 192.168.1.37
      port: 24224
#   - host: log1.example.com
#     port: 24224
#
## Example SSL configuration
```

```
## (note the use of port 24284 for ssl connections)
#
# LoggingServers:
#   - host: 192.168.24.11
#     port: 24284
# LoggingUsesSSL: true
# LoggingSharedKey: secret
# LoggingSSLCertificate: |
#   -----BEGIN CERTIFICATE-----
#   ...certificate data here...
#   -----END CERTIFICATE-----
```

B.10. monitoring-environment.yaml

```
## A Heat environment file which can be used to set up monitoring agents

resource_registry:
  OS::TripleO::Services::SensuClient: /usr/share/openstack-tripleo-heat-
templates/docker/services/sensu-client.yaml

parameter_defaults:
  MonitoringRabbitHost: 192.168.1.37
  MonitoringRabbitPort: 5672
  MonitoringRabbitUserName: sensu
  MonitoringRabbitPassword: sensu
#  MonitoringRabbitUseSSL: false
#  MonitoringRabbitVhost: "/sensu"
#  SensuClientCustomConfig:
#    api:
#      warning: 10
#      critical: 20
```

B.11. collectd-environment.yaml

```
resource_registry:
  OS::TripleO::Services::Collectd: /usr/share/openstack-tripleo-heat-
templates/docker/services/collectd.yaml

parameter_defaults:
#
## Collectd server configuration
```

```
   CollectdServer: 192.168.1.37
#
################
#### Other config parameters, the values shown here are the defaults
################
#
#   CollectdServerPort: 25826
#   CollectdSecurityLevel: None
#
################
#### If CollectdSecurityLevel is set to Encrypt or Sign
#### the following parameters are also needed
###############
#
#   CollectdUsername: user
#   CollectdPassword: password
#
## CollectdDefaultPlugins, These are the default plugins used by collectd
#
   CollectdDefaultPlugins:
     - disk
     - interface
     - load
     - memory
     - processes
     - tcpconns
#
## Extra plugins can be enabled by the CollectdExtraPlugins parameter:
## All the plugins availables are:
#
   CollectdExtraPlugins:
     - disk
     - df
#
## You can use ExtraConfig (or one of the related *ExtraConfig keys)
## to configure collectd.  See the documentation for puppet-collectd at
## https://github.com/voxpupuli/puppet-collectd for details.
#
   ExtraConfig:
     collectd::plugin::disk::disks:
       - "/^[vhs]d[a-f][0-9]?$/"
     collectd::plugin::df::mountpoints:
       - "/"
     collectd::plugin::df::ignoreselected: false
```

```
    collectd::plugin::cpu::valuespercentage: true
```

B.12. compute.yaml

```
heat_template_version: pike
description: >
  Software Config to drive os-net-config with 2 bonded nics on a bridge with VLANs
attached for the compute role.
parameters:
  ControlPlaneIp:
    default: ''
    description: IP address/subnet on the ctlplane network
    type: string
  ExternalIpSubnet:
    default: ''
    description: IP address/subnet on the external network
    type: string
  InternalApiIpSubnet:
    default: ''
    description: IP address/subnet on the internal_api network
    type: string
  StorageIpSubnet:
    default: ''
    description: IP address/subnet on the storage network
    type: string
  StorageMgmtIpSubnet:
    default: ''
    description: IP address/subnet on the storage_mgmt network
    type: string
  TenantIpSubnet:
    default: ''
    description: IP address/subnet on the tenant network
    type: string
  ManagementIpSubnet: # Only populated when including environments/network-
management.yaml
    default: ''
    description: IP address/subnet on the management network
    type: string
  BondInterfaceOvsOptions:
    default: ''
    description: 'The ovs_options or bonding_options string for the bond
      interface. Set things like lacp=active and/or bond_mode=balance-slb
      for OVS bonds or like mode=4 for Linux bonds using this option.'
```

```
    type: string
    constraints:
    - allowed_pattern: ^((?!balance.tcp).)*$
      description: 'The balance-tcp bond mode is known to cause packet loss and
        should not be used in BondInterfaceOvsOptions.'
  ExternalNetworkVlanID:
    default: 10
    description: Vlan ID for the external network traffic.
    type: number
  InternalApiNetworkVlanID:
    default: 20
    description: Vlan ID for the internal_api network traffic.
    type: number
  StorageNetworkVlanID:
    default: 30
    description: Vlan ID for the storage network traffic.
    type: number
  StorageMgmtNetworkVlanID:
    default: 40
    description: Vlan ID for the storage mgmt network traffic.
    type: number
  TenantNetworkVlanID:
    default: 50
    description: Vlan ID for the tenant network traffic.
    type: number
  ManagementNetworkVlanID:
    default: 60
    description: Vlan ID for the management network traffic.
    type: number
  ControlPlaneSubnetCidr: # Override this via parameter_defaults
    default: '24'
    description: The subnet CIDR of the control plane network.
    type: string
  ControlPlaneDefaultRoute: # Override this via parameter_defaults
    description: The default route of the control plane network.
    type: string
  ExternalInterfaceDefaultRoute: # Not used by default in this template
    default: 10.0.0.1
    description: The default route of the external network.
    type: string
  ManagementInterfaceDefaultRoute: # Commented out by default in this template
    default: unset
    description: The default route of the management network.
    type: string
```

```
  DnsServers: # Override this via parameter_defaults
    default: []
    description: A list of DNS servers (2 max for some implementations) that will be
added to resolv.conf.
    type: comma_delimited_list
  EC2MetadataIp: # Override this via parameter_defaults
    description: The IP address of the EC2 metadata server.
    type: string
resources:
  OsNetConfigImpl:
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      config:
        str_replace:
          template:
            get_file: /usr/share/openstack-tripleo-heat-
templates/network/scripts/run-os-net-config.sh
          params:
            $network_config:
              network_config:
              - type: interface
                name: ens3f0
                use_dhcp: false
                dns_servers:
                  get_param: DnsServers
                addresses:
                - ip_netmask:
                    list_join:
                    - /
                    - - get_param: ControlPlaneIp
                      - get_param: ControlPlaneSubnetCidr
                routes:
                - ip_netmask: 169.254.169.254/32
                  next_hop:
                    get_param: EC2MetadataIp
                - default: true
                  next_hop:
                    get_param: ControlPlaneDefaultRoute
              - type: ovs_bridge
                name: bridge_name
                members:
                - type: linux_bond
                  name: bond0
```

```
            bonding_options:
              get_param: BondInterfaceOvsOptions
            members:
            - type: interface
              name: ens3f4
              primary: true
            - type: interface
              name: ens3f5
          - type: vlan
            device: bond0
            vlan_id:
              get_param: InternalApiNetworkVlanID
            addresses:
            - ip_netmask:
                get_param: InternalApiIpSubnet
          - type: vlan
            device: bond0
            vlan_id:
              get_param: TenantNetworkVlanID
            addresses:
            - ip_netmask:
                get_param: TenantIpSubnet
        # ---------------------------------------------------
        - type: ovs_bridge
          name: br-storage
          members:
          - type: linux_bond
            name: bond1
            bonding_options:
              get_param: BondInterfaceOvsOptions
            members:
            - type: interface
              name: ens3f6
              primary: true
            - type: interface
              name: ens3f7
          - type: vlan
            device: bond1
            vlan_id:
              get_param: StorageNetworkVlanID
            addresses:
            - ip_netmask:
                get_param: StorageIpSubnet
          - type: vlan
```

```
                  device: bond1
                  vlan_id: {get_param: StorageMgmtNetworkVlanID}
                  addresses:
                    - ip_netmask: {get_param: StorageMgmtIpSubnet}
                # ----------------------------------------
                # Uncomment when including environments/network-management.yaml
                # If setting default route on the Management interface, comment
                # out the default route on the Control Plane.
                #- type: vlan
                #  device: bond0
                #  vlan_id: {get_param: ManagementNetworkVlanID}
                #  addresses:
                #    - ip_netmask: {get_param: ManagementIpSubnet}
                #  routes:
                #    - default: true
                #      next_hop: {get_param: ManagementInterfaceDefaultRoute}
outputs:
  OS::stack_id:
    description: The OsNetConfigImpl resource.
    value:
      get_resource: OsNetConfigImpl
```

B.13. ceph-storage.yaml

```
heat_template_version: pike
description: >
  Software Config to drive os-net-config with 2 bonded nics on a bridge with VLANs
attached for the ceph storage role.
parameters:
  ControlPlaneIp:
    default: ''
    description: IP address/subnet on the ctlplane network
    type: string
  ExternalIpSubnet:
    default: ''
    description: IP address/subnet on the external network
    type: string
  InternalApiIpSubnet:
    default: ''
    description: IP address/subnet on the internal_api network
    type: string
  StorageIpSubnet:
    default: ''
```

```
    description: IP address/subnet on the storage network
    type: string
  StorageMgmtIpSubnet:
    default: ''
    description: IP address/subnet on the storage_mgmt network
    type: string
  TenantIpSubnet:
    default: ''
    description: IP address/subnet on the tenant network
    type: string
  ManagementIpSubnet: # Only populated when including environments/network-
management.yaml
    default: ''
    description: IP address/subnet on the management network
    type: string
  BondInterfaceOvsOptions:
    default: ''
    description: 'The ovs_options or bonding_options string for the bond
      interface. Set things like lacp=active and/or bond_mode=balance-slb
      for OVS bonds or like mode=4 for Linux bonds using this option.'
    type: string
    constraints:
    - allowed_pattern: ^((?!balance.tcp).)*$
      description: 'The balance-tcp bond mode is known to cause packet loss and
        should not be used in BondInterfaceOvsOptions.'
  ExternalNetworkVlanID:
    default: 10
    description: Vlan ID for the external network traffic.
    type: number
  InternalApiNetworkVlanID:
    default: 20
    description: Vlan ID for the internal_api network traffic.
    type: number
  StorageNetworkVlanID:
    default: 30
    description: Vlan ID for the storage network traffic.
    type: number
  StorageMgmtNetworkVlanID:
    default: 40
    description: Vlan ID for the storage mgmt network traffic.
    type: number
  TenantNetworkVlanID:
    default: 50
    description: Vlan ID for the tenant network traffic.
```

```
    type: number
  ManagementNetworkVlanID:
    default: 60
    description: Vlan ID for the management network traffic.
    type: number
  ControlPlaneSubnetCidr: # Override this via parameter_defaults
    default: '24'
    description: The subnet CIDR of the control plane network.
    type: string
  ControlPlaneDefaultRoute: # Override this via parameter_defaults
    description: The default route of the control plane network.
    type: string
  ExternalInterfaceDefaultRoute: # Not used by default in this template
    default: 10.0.0.1
    description: The default route of the external network.
    type: string
  ManagementInterfaceDefaultRoute: # Commented out by default in this template
    default: unset
    description: The default route of the management network.
    type: string
  DnsServers: # Override this via parameter_defaults
    default: []
    description: A list of DNS servers (2 max for some implementations) that will be
added to resolv.conf.
    type: comma_delimited_list
  EC2MetadataIp: # Override this via parameter_defaults
    description: The IP address of the EC2 metadata server.
    type: string
resources:
  OsNetConfigImpl:
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      config:
        str_replace:
          template:
            get_file: /usr/share/openstack-tripleo-heat-
templates/network/scripts/run-os-net-config.sh
          params:
            $network_config:
              network_config:
              - type: interface
                name: ens3f0
                use_dhcp: false
```

```
      dns_servers:
        get_param: DnsServers
      addresses:
      - ip_netmask:
          list_join:
          - /
          - - get_param: ControlPlaneIp
            - get_param: ControlPlaneSubnetCidr
      routes:
      - ip_netmask: 169.254.169.254/32
        next_hop:
          get_param: EC2MetadataIp
      - default: true
        next_hop:
          get_param: ControlPlaneDefaultRoute
    - type: ovs_bridge
      name: br-storage
      members:
      - type: linux_bond
        name: bond1
        bonding_options:
          get_param: BondInterfaceOvsOptions
        members:
        - type: interface
          name: ens3f6
          primary: true
        - type: interface
          name: ens3f7
      - type: vlan
        device: bond1
        vlan_id:
          get_param: StorageNetworkVlanID
        addresses:
        - ip_netmask:
            get_param: StorageIpSubnet
      - type: vlan
        device: bond1
        vlan_id:
          get_param: StorageMgmtNetworkVlanID
        addresses:
        - ip_netmask:
            get_param: StorageMgmtIpSubnet
      # Uncomment when including environments/network-management.yaml
      # If setting default route on the Management interface, comment
```

```
                # out the default route on the Control Plane.
                #-
                #  type: vlan
                #  device: bond1
                #  vlan_id: {get_param: ManagementNetworkVlanID}
                #  addresses:
                #    -
                #      ip_netmask: {get_param: ManagementIpSubnet}
                #  routes:
                #    -
                #      default: true
                #      next_hop: {get_param: ManagementInterfaceDefaultRoute}
outputs:
  OS::stack_id:
    description: The OsNetConfigImpl resource.
    value:
      get_resource: OsNetConfigImpl
```

B.14. controller.yaml

```
heat_template_version: pike
description: >
  Software Config to drive os-net-config with 2 bonded nics on a bridge with VLANs
attached for the controller role.
parameters:
  ControlPlaneIp:
    default: ''
    description: IP address/subnet on the ctlplane network
    type: string
  ExternalIpSubnet:
    default: ''
    description: IP address/subnet on the external network
    type: string
  InternalApiIpSubnet:
    default: ''
    description: IP address/subnet on the internal_api network
    type: string
  StorageIpSubnet:
    default: ''
    description: IP address/subnet on the storage network
    type: string
  StorageMgmtIpSubnet:
    default: ''
```

```
    description: IP address/subnet on the storage_mgmt network
    type: string
  TenantIpSubnet:
    default: ''
    description: IP address/subnet on the tenant network
    type: string
  ManagementIpSubnet: # Only populated when including environments/network-
management.yaml
    default: ''
    description: IP address/subnet on the management network
    type: string
  BondInterfaceOvsOptions:
    default: bond_mode=active-backup
    description: 'The ovs_options or bonding_options string for the bond
      interface. Set things like lacp=active and/or bond_mode=balance-slb
      for OVS bonds or like mode=4 for Linux bonds using this option.'
    type: string
    constraints:
    - allowed_pattern: ^((?!balance.tcp).)*$
      description: 'The balance-tcp bond mode is known to cause packet loss and
        should not be used in BondInterfaceOvsOptions.'
  ExternalNetworkVlanID:
    default: 10
    description: Vlan ID for the external network traffic.
    type: number
  InternalApiNetworkVlanID:
    default: 20
    description: Vlan ID for the internal_api network traffic.
    type: number
  StorageNetworkVlanID:
    default: 30
    description: Vlan ID for the storage network traffic.
    type: number
  StorageMgmtNetworkVlanID:
    default: 40
    description: Vlan ID for the storage mgmt network traffic.
    type: number
  TenantNetworkVlanID:
    default: 50
    description: Vlan ID for the tenant network traffic.
    type: number
  ManagementNetworkVlanID:
    default: 60
    description: Vlan ID for the management network traffic.
```

```
    type: number
  ControlPlaneDefaultRoute: # Override this via parameter_defaults
    description: The default route of the control plane network.
    type: string
  ExternalInterfaceDefaultRoute:
    default: 10.0.0.1
    description: default route for the external network
    type: string
  ManagementInterfaceDefaultRoute: # Commented out by default in this template
    default: unset
    description: The default route of the management network.
    type: string
  ControlPlaneSubnetCidr: # Override this via parameter_defaults
    default: '24'
    description: The subnet CIDR of the control plane network.
    type: string
  DnsServers: # Override this via parameter_defaults
    default: []
    description: A list of DNS servers (2 max for some implementations) that will be
added to resolv.conf.
    type: comma_delimited_list
  EC2MetadataIp: # Override this via parameter_defaults
    description: The IP address of the EC2 metadata server.
    type: string
resources:
  OsNetConfigImpl:
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      config:
        str_replace:
          template:
            get_file: /usr/share/openstack-tripleo-heat-
templates/network/scripts/run-os-net-config.sh
          params:
            $network_config:
              network_config:
              - type: interface
                name: ens3f0
                use_dhcp: false
                addresses:
                - ip_netmask:
                    list_join:
                    - /
```

```
            - - get_param: ControlPlaneIp
              - get_param: ControlPlaneSubnetCidr
          routes:
          - ip_netmask: 169.254.169.254/32
            next_hop:
              get_param: EC2MetadataIp
        - type: ovs_bridge
          name: bridge_name
          dns_servers:
            get_param: DnsServers
          use_dhcp: false
          addresses:
          - ip_netmask:
              get_param: ExternalIpSubnet
          routes:
          - default: true
            next_hop:
              get_param: ExternalInterfaceDefaultRoute
          members:
          - type: interface
            name: ens3f2
            primary: true
        # -----------------------------------------------
        - type: ovs_bridge
          name: bridge_name
          members:
          - type: linux_bond
            name: bond0
            bonding_options:
              get_param: BondInterfaceOvsOptions
            members:
            - type: interface
              name: ens3f4
              primary: true
            - type: interface
              name: ens3f5
          - type: vlan
            device: bond0
            vlan_id:
              get_param: InternalApiNetworkVlanID
            addresses:
            - ip_netmask:
                get_param: InternalApiIpSubnet
          - type: vlan
```

```
      device: bond0
      vlan_id:
        get_param: TenantNetworkVlanID
      addresses:
      - ip_netmask:
          get_param: TenantIpSubnet
    # ------------------------------------------
    - type: ovs_bridge
      name: br-storage
      members:
      - type: linux_bond
        name: bond1
        bonding_options:
          get_param: BondInterfaceOvsOptions
        members:
        - type: interface
          name: ens3f6
          primary: true
        - type: interface
          name: ens3f7
      - type: vlan
        device: bond1
        vlan_id:
          get_param: StorageNetworkVlanID
        addresses:
        - ip_netmask:
            get_param: StorageIpSubnet
      - type: vlan
        device: bond1
        vlan_id:
          get_param: StorageMgmtNetworkVlanID
        addresses:
        - ip_netmask:
            get_param: StorageMgmtIpSubnet

      # Uncomment when including environments/network-management.yaml
      # If setting default route on the Management interface, comment
      # out the default route on the External interface. This will
      # make the External API unreachable from remote subnets.
      #-
      #  type: vlan
      #  device: bond1
      #  vlan_id: {get_param: ManagementNetworkVlanID}
      #  addresses:
```

```
                #    -
                #      ip_netmask: {get_param: ManagementIpSubnet}
                #  routes:
                #    -
                #      default: true
                #      next_hop: {get_param: ManagementInterfaceDefaultRoute}
outputs:
  OS::stack_id:
    description: The OsNetConfigImpl resource.
    value:
      get_resource: OsNetConfigImpl
```

B.15. clean-disks.sh

```
#!/usr/bin/bash


for node in $(ironic node-list  | grep ceph | awk '{print $2}')
do
  openstack baremetal node manage $node
  openstack baremetal node clean $node --clean-steps '[{"interface": "deploy",
"step": "erase_devices_metadata"}]'
  #openstack baremetal node provide $node
done
```

B.16. instackenv.json

```
{
  "nodes": [
    {
      "pm_user": "admin",
      "arch": "x86_64",
      "name": "overcloud-compute1",
      "capabilities": "profile:compute,boot_option:local",
      "pm_addr": "192.168.1.43",
      "pm_password": "password",
      "pm_type": "pxe_ilo",
      "mac": [
        "98:F2:B3:22:35:3C"
      ],
      "cpu": "1",
      "memory": "1024",
      "disk": "10"
```

```
    },
    {
      "pm_user": "admin",
      "arch": "x86_64",
      "name": "overcloud-compute2",
      "capabilities": "profile:compute,boot_option:local",
      "pm_addr": "192.168.1.49",
      "pm_password": "password",
      "pm_type": "pxe_ilo",
      "mac": [
        "98:F2:B3:22:35:26"
      ],
      "cpu": "1",
      "memory": "1024",
      "disk": "10"
    },
    {
      "pm_user": "admin",
      "arch": "x86_64",
      "name": "overcloud-compute3",
      "capabilities": "profile:compute,boot_option:local",
      "pm_addr": "192.168.1.50",
      "pm_password": "password",
      "pm_type": "pxe_ilo",
      "mac": [
        "98:F2:B3:22:55:5A"
      ],
      "cpu": "1",
      "memory": "1024",
      "disk": "10"
    },
    {
      "pm_user": "admin",
      "arch": "x86_64",
      "name": "overcloud-controller1",
      "capabilities": "profile:control,boot_option:local",
      "pm_addr": "192.168.1.46",
      "pm_password": "password",
      "pm_type": "pxe_ilo",
      "mac": [
        "98:F2:B3:22:45:A4"
      ],
      "cpu": "1",
      "memory": "1024",
```

```
    "disk": "10"
  },
  {
    "pm_user": "admin",
    "arch": "x86_64",
    "name": "overcloud-controller2",
    "capabilities": "profile:control,boot_option:local",
    "pm_addr": "192.168.1.47",
    "pm_password": "password",
    "pm_type": "pxe_ilo",
    "mac": [
      "98:F2:B3:22:45:98"
    ],
    "cpu": "1",
    "memory": "1024",
    "disk": "10"
  },
  {
    "pm_user": "admin",
    "arch": "x86_64",
    "name": "overcloud-controller3",
    "capabilities": "profile:control,boot_option:local",
    "pm_addr": "192.168.1.42",
    "pm_password": "password",
    "pm_type": "pxe_ilo",
    "mac": [
      "98:F2:B3:21:1F:BE"
    ],
    "cpu": "1",
    "memory": "1024",
    "disk": "10"
  },
  {
    "pm_user": "admin",
    "arch": "x86_64",
    "name": "overcloud-ceph1",
    "capabilities": "profile:ceph-storage,boot_option:local",
    "pm_addr": "192.168.1.44",
    "pm_password": "password",
    "pm_type": "pxe_ilo",
    "mac": [
      "98:F2:B3:22:35:00"
    ],
    "cpu": "1",
```


```
      "memory": "1024",
      "disk": "10"
    },
    {
      "pm_user": "admin",
      "arch": "x86_64",
      "name": "overcloud-ceph2",
      "capabilities": "profile:ceph-storage,boot_option:local",
      "pm_addr": "192.168.1.51",
      "pm_password": "password",
      "pm_type": "pxe_ilo",
      "mac": [
        "98:F2:B3:22:35:58"
      ],
      "cpu": "1",
      "memory": "1024",
      "disk": "10"
    },
    {
      "pm_user": "admin",
      "arch": "x86_64",
      "name": "overcloud-ceph3",
      "capabilities": "profile:ceph-storage,boot_option:local",
      "pm_addr": "192.168.1.48",
      "pm_password": "password",
      "pm_type": "pxe_ilo",
      "mac": [
        "98:F2:B3:22:35:7A"
      ],
      "cpu": "1",
      "memory": "1024",
      "disk": "10"
    }
  ]
}
```

Resources and additional links

HPE Reference Architectures, hpe.com/info/ra

HPE Servers, hpe.com/servers

HPE Storage, hpe.com/storage

HPE Networking, hpe.com/networking

HPE Technology Consulting Services, hpe.com/us/en/services/consulting.html

HPE Synergy, hpe.com/synergy

Red Hat OpenStack Platform 13 Director Installation and Usage, https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/pdf/director_installation_and_usage/Red_Hat_OpenStack_Platform-13-Director_Installation_and_Usage-en-US.pdf

Red Hat OpenStack Platform 13 Monitoring Tools Configuration Guide, https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/pdf/monitoring_tools_configuration_guide/Red_Hat_OpenStack_Platform-13-Monitoring_Tools_Configuration_Guide-en-US.pdf

To help us improve our documents, please provide feedback at hpe.com/contact/feedback

a00071087enw, May 2019