Hewlett Packard Enterprise

Deployment Guide

Accelerate container deployment with Red Hat OpenShift Container Platform and HPE Synergy

Entry level POC system for Red Hat OpenShift Container Platform on HPE Synergy and Red Hat Hyperconverged Infrastructure

Contents

Executive summary

Organizations are going through a digital transformation that substantially changes the way applications are developed, deployed, and maintained. At the heart of this transformation are microservices and containers that allow developers to quickly create and deploy applications in a Continuous Integration / Continuous Deployment (CI / CD) model. This CI / CD model brings together the developed communities and the operations communities within an organization, commonly referred to as “DevOps”. Red Hat® OpenShift Container Platform is a proven technology for enabling container-based DevOps and CI / CD processes and providing container orchestration and scheduling. Containers provide an additional layer of abstraction between the virtual and physical resources that exist in the infrastructure. This fluid deployment model requires an infrastructure that is just as flexible and can seamlessly add, remove, and repair the underlying physical and virtual resources. HPE Synergy composable infrastructure provides a secure, highly available, and flexible infrastructure that is ideal to meet the requirements of Red Hat OpenShift Container Platform. HPE Synergy provides composable network, storage, and compute resources to customers implementing container-based solutions.

Target audience: This document is intended for systems administrators, architects, and DevOps leads. The reader should have a working knowledge of Red Hat Virtualization, Red Hat OpenShift Container Platform, HPE Synergy, Red Hat Hyperconverged Infrastructure, and Red Hat Ansible.

Solution overview

Many container projects start on a small proof-of-concept (POC) scale and rapidly grow from there. This document describes such a solution using a small entry level HPE Synergy that can easily scale to accommodate additional containerized workloads. Such a configuration is highly suitable for customers looking for a cost effective POC system that can be rapidly deployed in hours, or customers needing a container DevOps platform for their development teams. This document outlines the steps required to create a Red Hat OpenShift Container Platform environment running on HPE Synergy, Red Hat Hyperconverged Infrastructure and Red Hat OpenShift Container Storage. It is meant to be used in conjunction with files and Ansible playbooks found at https://github.com/HewlettPackard/hpe-solutions-openshift. Note that the scripts described in this document and provided on the GitHub site are not supported by Hewlett Packard Enterprise or Red Hat. HPE plans to update this document over time with enhancements to deployment methodologies as well as new software versions, features, and functions. Check for the latest document at https://github.com/HewlettPackard/hpe-solutions-openshift. It is recommended that the installer review this document in its entirety and understand all prerequisites prior to beginning an installation.

Figure 1 displays the configuration for this entry level OpenShift solution on HPE Synergy. The OpenShift master, infrastructure, network load balancer, and application nodes are all deployed as VMs to optimize resource usage and eliminate the need to allocate dedicated physical compute modules for each individual OpenShift and management component. OpenShift Container Storage is used to provide software defined storage and leverage the direct attached storage of the HPE Synergy Compute Modules and the HPE D3940 Storage Module. Three (3) HPE Synergy Compute Modules are deployed in a Red Hat Hyperconverged Infrastructure cluster configuration to provide both HA and resources to support initial workload deployments.

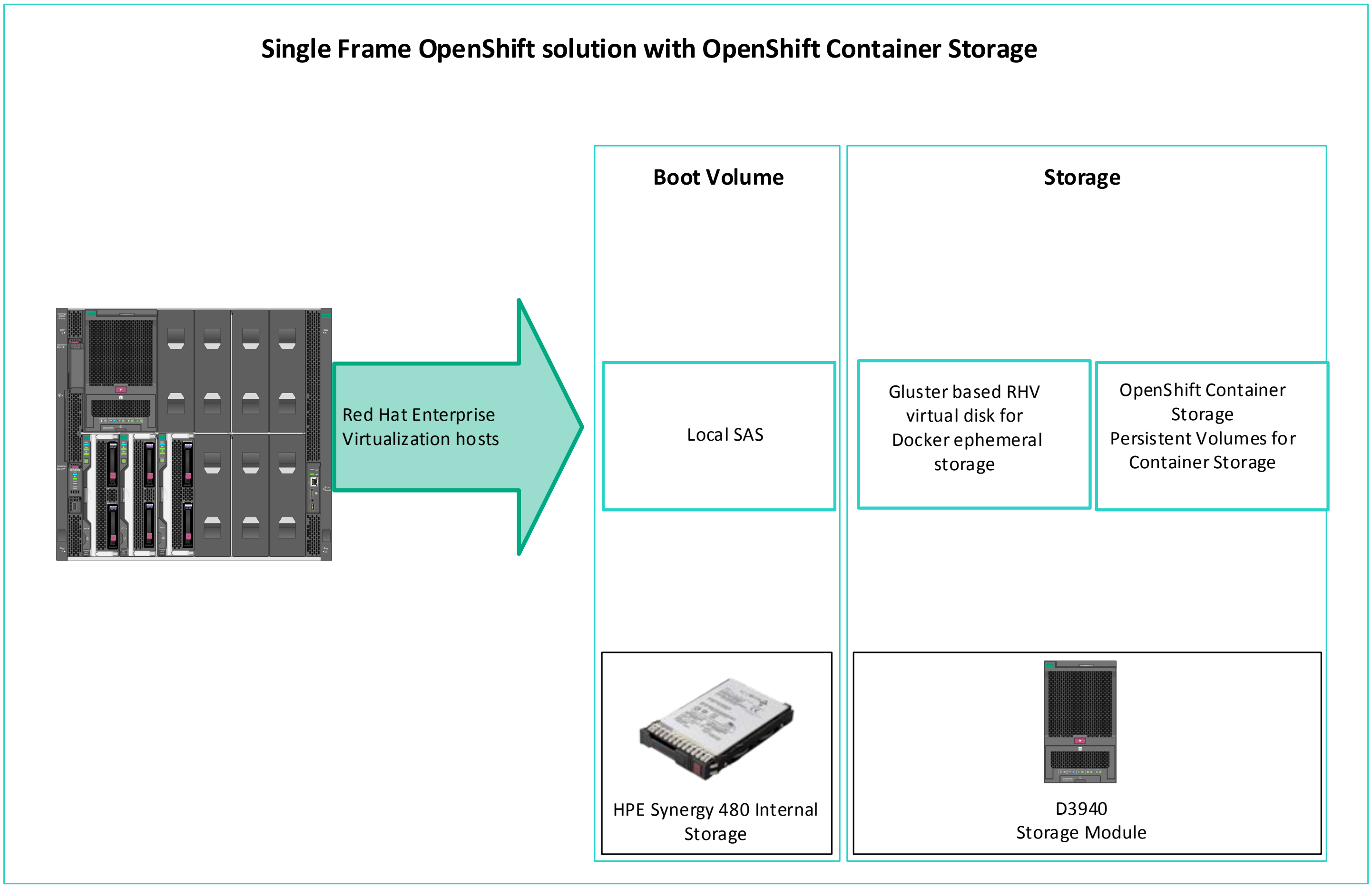

Figure 1: Solution layout

Note
Scripts and files are provided as is and are examples of how to build out your infrastructure. It is expected that they will need to be adapted to work in specific customer environments.

Solution components

This solution is based on HPE Synergy and Red Hat Hyperconverged Infrastructure (RHHI) running on three HPE SY480 Compute Modules in a single HPE Synergy 12000 Frame. The storage in this solution is provided by an HPE Synergy D3940 Storage Module. Red Hat OpenShift Container Platform is deployed entirely on virtual machines running on the Red Hat Hyperconverged Infrastructure for Virtualization cluster. The storage provided by the HPE Synergy D3940 Storage Module is consumed by Red Hat Gluster Storage, providing storage for the virtualization environment. The HPE Synergy D3940 Storage Module provides additional storage for the Gluster based Red Hat OpenShift Container Storage (OCS). Red Hat OpenShift Container Storage provides persistent storage for container applications running on Red Hat OpenShift Container Platform. Figure 2 illustrates the logical design of the solution.

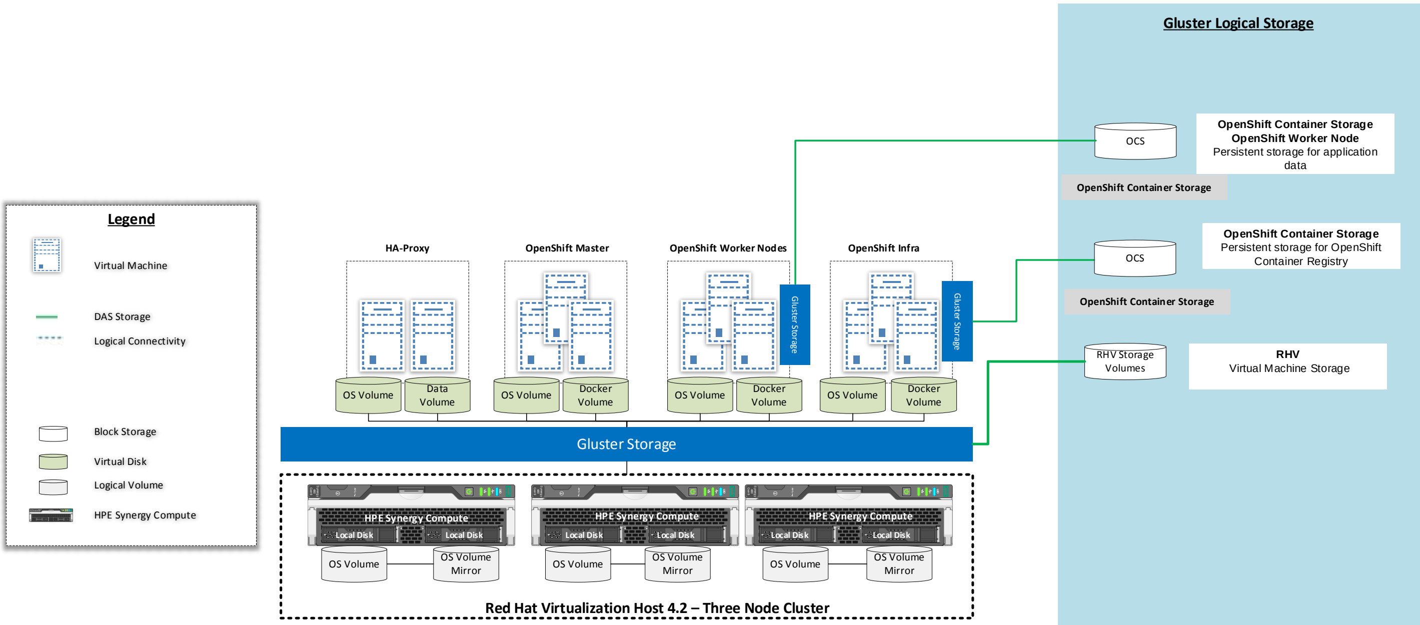

Figure 2: Logical design

Sizing considerations

Red Hat OpenShift Container Platform sizing

Sizing for a Red Hat OpenShift Container Platform environment varies depending upon the requirements of the specific organization and type of deployment. In this section we will discuss sizing considerations for Red OpenShift Container Platform, host requirements, and cluster sizing.

Red Hat OpenShift Container Platform role sizing

Master – The minimum size for a physical or virtual machine running the master node is 4 vCPU and 16 GB RAM with a 40 GB disk space for /var, 1 GB disk space for /usr/local/bin, and 1 GB disk space for the system's temporary directory. Master nodes should be configured with an additional 1 CPU core and 1.5 GB RAM for each additional 1,000 pods

Nodes – Application and Infrastructure nodes require a minimum of 1 vCPU and 8 GB RAM with a disk with at least 15 GB of space for /var/, 1 GB disk space for /usr/local/bin, and 1 GB disk space for the system's temporary directory and an additional 15 GB for Docker storage. Sizing for worker nodes is ultimately dependent on the container workloads and their CPU, memory, and disk requirements.

Etcd – Etcd nodes should be configured with a minimum of 4 vCPU and 15 GB RAM and 20 GB for etcd data.

Red OpenShift Container Platform cluster sizing

The number of application nodes in an OpenShift cluster depends on the number of pods that an organization is planning on deploying. Red Hat OpenShift Container Platform can support the following maximums:

- Maximum of 2,000 nodes per cluster
- Maximum of 120,000 pods per cluster
- Maximum of 250 pods per node
- Maximum of 10 pods per CPU core

To determine the number of nodes required in a cluster estimate the number of pods the organization is planning on deploying and divide by the maximum number of pods per node. For example, if the organization expects to deploy 5000 pods, then the organization should expect to deploy 20 application nodes with 250 pods per node (5000 / 250 =20). In this environment with a default configuration of three physical application nodes, the Red Hat OpenShift cluster should be expected to support a maximum of 750 pods (250 pods x 3 nodes = 750 pods).

For more information about Red Hat OpenShift Container Platform sizing, refer to the Red Hat OpenShift Container Platform product documentation at:
https://access.redhat.com/documentation/en-us/openshift_container_platform/3.10/html/installing_clusters/install-planning#sizing

https://access.redhat.com/documentation/en-us/openshift_container_platform/3.10/html-single/scaling_and_performance_guide/#scaling-performance-cluster-limits

Software versions

Table 1 lists the versions of important software utilized in the creation of this solution. The installer should insure they have downloaded or have access to this software.

Table 1. Major software versions used in solution creation

Component	Version
Red Hat Virtualization / Red Hat Hyperconverged Infrastructure for Virtualization	4.2
Red Hat OpenShift Container Platform	3.10
Red Hat OpenShift Container Storage	3.10

Required Repositories – OpenShift nodes

- rhel-7-server-ansible-2.4-rpms
- rhel-7-fast-datapath-rpms
- rhel-7-server-extras-rpms
- rhel-7-server-rpms
- rhel-7-server-ose-3.10-rpms

Deployment environment

This document makes assumptions about services and networks available within the implementation environment. This section discusses those assumptions and, where applicable, provides details on how they should be configured. If a service is optional it will be noted in the description.

Services

Table 2 lists the services utilized in the creation of this solution and provides a high-level explanation of their function and whether or not they are required.

Table 2. Services used in the creation of this solution

Service	Required/Optional	Description/Notes
DNS	Required	Provides name resolution on management and data center networks.
DHCP	Required	Provides IP address leases on PXE.
TFTP/PXE	Required	Required to provide network boot capabilities to hosts that will install via a Kickstart file.
NTP	Required	Required to insure consistent time across the solution stack.
Active Directory/LDAP	Optional	May be used for authentication functions on various networks. This solution utilized local authentication.

DNS

Name services must be in place for management and data center networks. Once a host has become active ensure that both forward and reverse lookups are working on the management and data center networks. All virtual machines used for the Red Hat OpenShift Container platform deployment must be registered in DNS.

DHCP

DHCP services must be in place for the PXE and management networks. DHCP services are generally in place on data center networks. Because Virtual Connect exposes the MAC address of the network interfaces before installation has begun it is easy to create address reservations for the hosts. A reservation is required for a single adapter on the PXE network of each physical server. This facilitates post-deployment configuration over SSH as well as a secure communication channel for running Ansible scripts.

TFTP/PXE

The hosts in this configuration were deployed via a combination of Kickstart files and manual configuration. HPE plans updates to this document with a focus on enhanced levels of installation automation for the hosts. In order to successfully complete the necessary portions of a Kickstart install you will need a host that is capable of providing HTTP, TFTP, and network boot services. In the solution environment, PXE services existed on a tertiary network beyond the traditional data center and management networks. It is beyond the scope of this document to provide instructions for building a PXE server host. It is assumed that TFTP and network boot services are being provided from a Linux-based host.

In order to boot the hosts, you will need to create the PXE boot menu. This menu will provide a means to select whether to install Red Hat® Enterprise Linux® or Red Hat Enterprise Virtualization on a particular host. To configure the menu, SSH into the PXE server host or connect locally. Edit the file /var/lib/tftpboot/pxelinux.cfg/default using vi or a similar text editor. The specified URL will point to your web server and the location of the required files.

Kickstart options are covered under the "Red Hat Virtualization hosts" section of this document.

NTP

A Network Time Protocol server should be available to hosts within the solution environment.

Installer laptop

A laptop system with the ability to connect to the various components within the solution stack is required.

Ansible Engine

This document assumes that Ansible Engine 2.4 exists within the deployment environment and is accessible to the installer. HPE built this solution using Ansible version 2.4.6.

Ansible is used extensively throughout this solution to automate manual configuration tasks. The Ansible playbook is used to automate the configuration as described in Table 3 below.

Table 3. Ansible playbooks used in this solution

Playbook	Description/Notes
rhhi.yml	Prepare the RHV hosts for Red Hat Hyperconverged Infrastructure installation
deploytemplate.yml	Download a rhel7.5 qcow2 image and prepare a virtual machine template
deployvm.yml	Deploy the virtual machines that will comprise the OpenShift environment
preparehosts.yml	Prepare the virtual machines for the OpenShift deployment
prerequsites.yml	Check the OpenShift prerequisites
deploycluster.yml	Installs OpenShift

The Ansible playbooks in this solution are available from the HPE GitHub repository located at https://github.com/HewlettPackard/hpe-solutions-openshift.

The solution uses the playbooks described in the above table to call Ansible roles. The Ansible roles, associated tasks, and variable files can be found in the roles subdirectory.

The openshift-ansible playbooks, prerequisites.yml and deploycluster.yml, can be found in the /usr/share/ansible/openshift-ansible/playbooks/ subdirectory.

Additional information on running the playbooks and configuring the variable files are described later in this document in the section titled "OpenShift deployment". Refer to Appendix C: Configure the Ansible Host, for detailed information on the Ansible host configuration.

Physical environment

The configuration deployed for this solution is described in greater detail in this section. Figure 3, below, illustrates the various components in the solution as viewed through the HPE OneView interface. At a high level, Hewlett Packard Enterprise and Red Hat deployed:

- One (1) HPE Synergy 12000 Frame with HPE Virtual Connect SE 40 Gb F8 Modules for Synergy
- Two (2) HPE FlexFabric 5940 switches
- Three (3) HPE Synergy 480 Gen10 Compute Modules, with three (3) virtualized hosts running the required Red Hat OpenShift control plane, infrastructure, and application nodes as virtual machines
- One (1) HPE Synergy D3940 Storage Module providing direct attached storage for RHV virtual machine and container storage

This configuration was built on an HPE Converged Architecture 750 which offers an improved time to deployment and tested firmware recipe. That baseline can be retrieved at the HPE Information Library and then selecting ConvergedSystem 750 from the Models/Subcategories menu. The user also has the flexibility of customizing the HPE components throughout this stack per their unique IT and workload requirements or building with individual components. Figure 4 shows the physical configuration of the rack used in this solution.

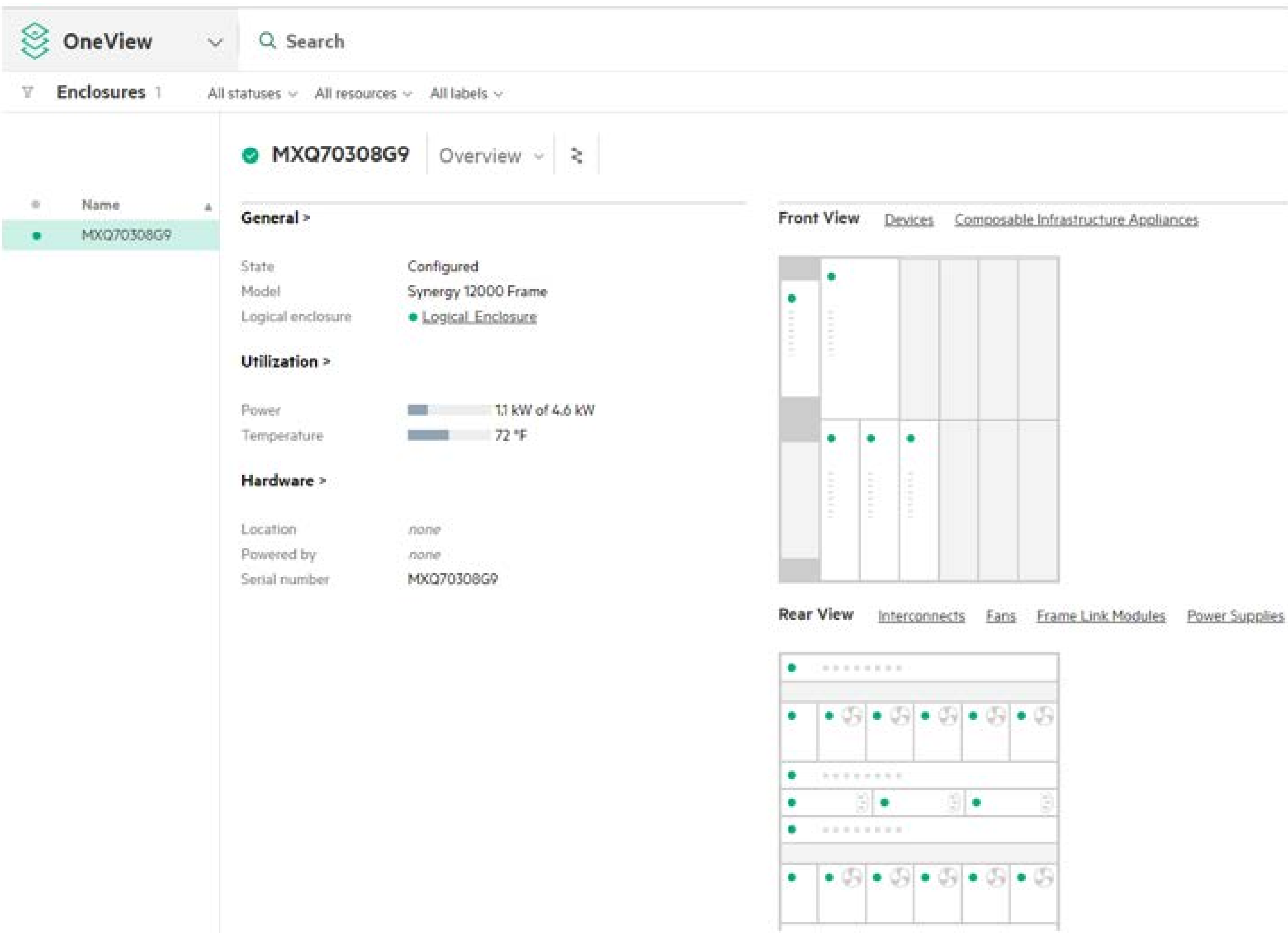

Figure 3: HPE OneView Synergy Frame enclosure

Figure 4 below shows the physical rack layout of the solution. This solution contains one (1) HPE Synergy 12000 Frame with three (3) HPE SY480 Gen10 Compute Modules and one (1) HPE D3940 Storage Module.

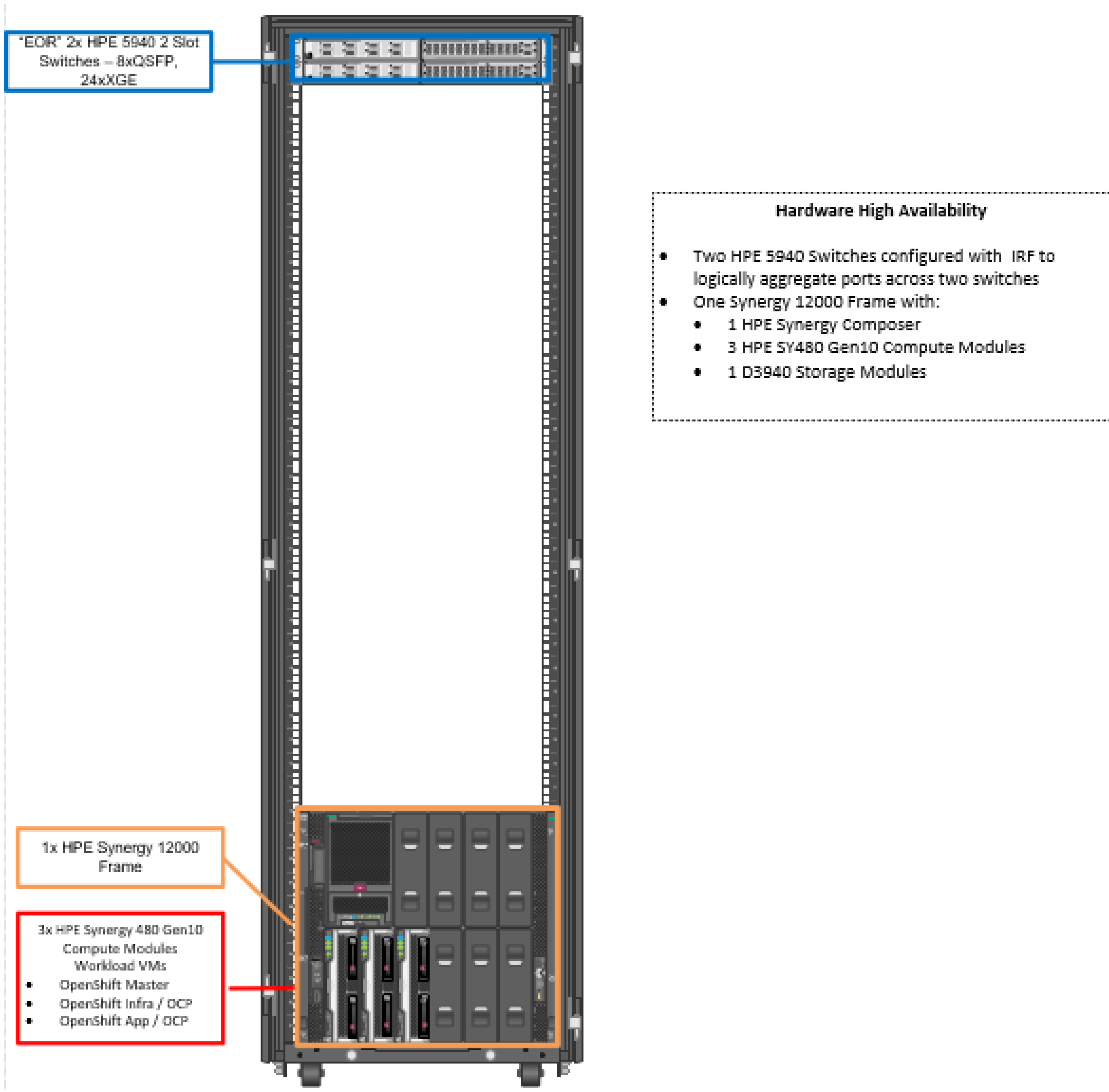

Figure 4: HPE Synergy 12000 rack components

Note

Firmware recipes for the individual components adhere to HPE Converged Solution 750 specifications which can be found in the Firmware and Software Compatibility Matrix.

Table 4. Components utilized in the creation of this solution

Component	Qty	Description
HPE Synergy 12000 Frame	1	One (1) HPE Synergy 12000 Frame house the infrastructure used for the solution
HPE Synergy Composer	1	Core configuration and lifecycle management for the Synergy components
HPE Virtual Connect SE 40 Gb F8 Module	2	Two (2) HPE Virtual Connect SE 40 Gb F8 Modules provide network connectivity into and out of the frames
HPE Synergy 480 Gen10 Compute Module	3	Three (3) virtualized bare metal hosts as described later in this document
HPE Synergy D3940 Storage Module	1	One (1) HPE Synergy D3940 Storage Module to provide solution storage
HPE Synergy 12 Gb SAS Connection Module	2	Two (2) SAS Connection Modules providing connectivity for the D3940 Storage Module
HPE FlexFabric 5940 2-Slot Switch	2	Each HPE FF 5940 switch contains two (2) of the 5930 modules listed below
HPE 5930 24p SFP+ and 2p QSFP+ Module	2	One module per HPE FlexFabric 2-Slot Switch
HPE 5930 8-port QSFP+ Module	2	One module per HPE FlexFabric 2-Slot Switch

Cabling the HPE Synergy 12000 Frame and HPE Virtual Connect SE 40 Gb F8 Modules for Synergy

This section shows the physical cabling between enclosures as well as the physical connectivity to the switching. It is intended to provide an understanding of how the infrastructure was interconnected during testing and a guide for the installer to base their configuration on.

Figure 5 below shows the cabling of the Synergy frame to the network switches. The specific networks contained within the Bridge-Aggregation Groups are described in more detail later in this section. At the lowest level there are two (2) 40 GbE connections dedicated to carrying redundant, production network traffic to the first layer switch where it is further distributed.

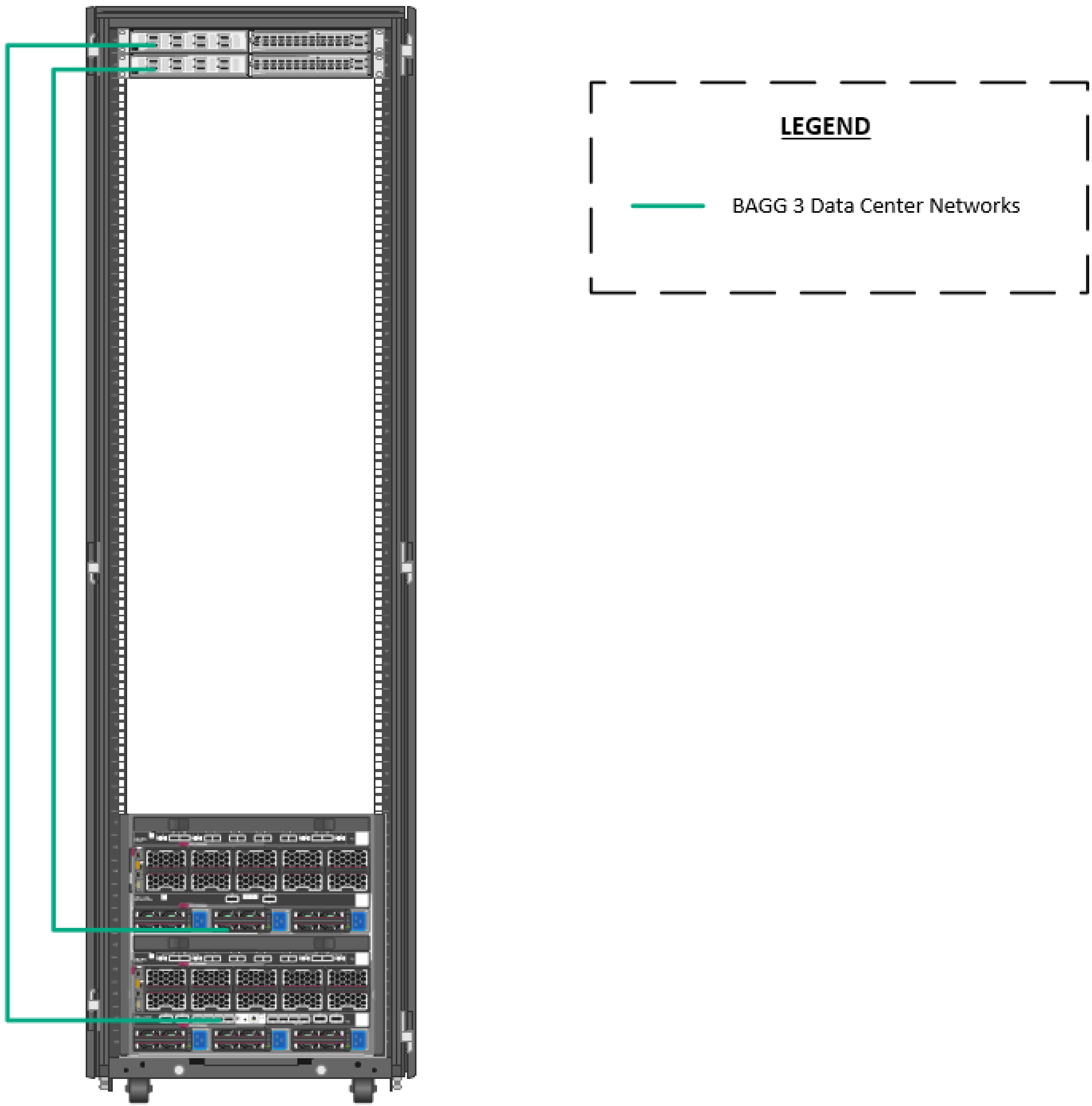

Figure 5: Cabling of the HPE Synergy 12000 Frames to the HPE FF 5940 switches

Table 5 explains the cabling of the Virtual Connect interconnect modules to the HPE FF 5940 switching.

Table 5. Networks used in this solution

Uplink Set	Synergy Source	Switch Destination
Network	Enclosure 1 Port Q5	FortyGigE1/1/1
	Enclosure 1 Port Q6	FortyGigE2/1/1

Configuring the solution switching

The solution described in this document utilized HPE FlexFabric 5940 switches. The HPE FlexFabric 5940 switches are configured per the configuration parameters below. Individual port configurations are described elsewhere in this section. The switches should be configured with an HPE Intelligent Resilient Fabric (IRF). To understand the process of configuring IRF consult the HPE FlexFabric 5940 Switch Series Installation Guide. Consult this guide for initial installation of switching as well as creation of user accounts and access methods. The remainder of this section assumes that the switches have been installed, configured for IRF, hardened, and are accessible over SSH.

Physical cabling

Table 6 below is a map of source ports to ports on the HPE FlexFabric 5940 switches.

Table 6. HPE FlexFabric 5940 port map

Source Port	Switch Port
Virtual Connect Frame U30, Q5	FortyGigE1/1/1
Virtual Connect Frame U30, Q6	FortyGigE2/1/1
To Upstream Switching	Customer Choice

It is recommended that you log onto the switch post-configuration and provide a description for each of these ports.

Network definitions

Table 7 defines the networks configured using HPE Synergy Composer in the creation of this solution. These networks should be defined at both the first layer switch as well as within Composer. This solution utilizes one VLAN for the Data Center and Solution Management segments. Actual VLANs and network counts will be determined by the requirements of your production environment.

Table 7. Networks used in this solution

Network Function	VLAN Number	Bridge Aggregation Group
PXE	10	3
Public	194	3
Management	20	3
Storage	30	3

To add these networks to the switch log on to the console over SSH and run the following commands.

```
# sys
# vlan 10 194 20 30
```

For each of these VLANs perform the following steps.

```
# interface vlan-interface ####
# name VLAN Name per table above
# description Add text that describes the purpose of the VLAN
# q
```

It is recommended that you configure a dummy VLAN on the switches and assign unused ports to that VLAN.

The switches should be configured with separate bridge aggregation groups for the different links to the HPE Synergy frame connections. To configure the three (3) bridge-aggregation groups and ports as described in the tables above, run the following commands.

For the data center and management VLANs run the following commands.

```
# interface Bridge-Aggregation3
# link-aggregation mode dynamic
# description <FrameNameU30>-ICM
# quit

# interface range name <FrameNameU30>-ICM interface Bridge-Aggregation3
# quit

# interface range FortyGigE 1/1/1 to FortyGigE 2/1/1
# port link-aggregation group 3
# quit

# interface range name <FrameNameU30>-ICM
# port link-type trunk
# undo port trunk permit vlan 1
# port trunk permit vlan 10 194 20 30
# quit
```

Optionally, you can enter a description on a per interface basis by running the following command.

```
# description text you want to describe the interface with
```

When you have completed configuration of the switches insure you save your state and apply it by typing “save” and following the prompts.

HPE Synergy 480 Gen10 Compute Modules

This section describes the connectivity of the HPE Synergy 480 Gen10 Compute Modules used in the creation of this solution. The compute modules, regardless of function, were all configured identically. Table 8 describes the individual components. Individual server sizing should be based on customer needs and may not align with the configuration outlined in this document.

Table 8. Host configuration

Component	Quantity
HPE Synergy 480/660 Gen10 Intel® Xeon®-Gold 6142 (2.6GHz/16-core/150W) FIO Processor Kit	2 per server
HPE 32GB 2Rx4 PC4-2666V-R Smart Kit	8 per server
HPE 300GB SAS 10K SFF	2 per server

Each compute module as shown in the table above is shipped with two (2) Intel Xeon-Gold 6142 (2.6GHz/16-core/150W) processors and an HPE Smart Array P416ie-m SR Gen10 12G SAS controller.

Red Hat Virtualization hosts

The solution calls for the installation of Red Hat Virtualization 4.2 (RHVH) on three (3) HPE Synergy 480 Gen10 Compute Modules. These three hosts will be configured in a Red Hat Hyperconverged Infrastructure for Virtualization cluster based on Red Hat Gluster Storage. These hosts provide the virtualization infrastructure for the OpenShift deployment installed on virtual machines. Figure 6 below highlights the connectivity of these hosts. Networks that are carried on the individual bridge aggregation groups are shown in Table 7 in this document. Each host is presented with access to a local RAID 6 volume that will be part of a Gluster cluster to provide storage for the virtual machines used in the solution.

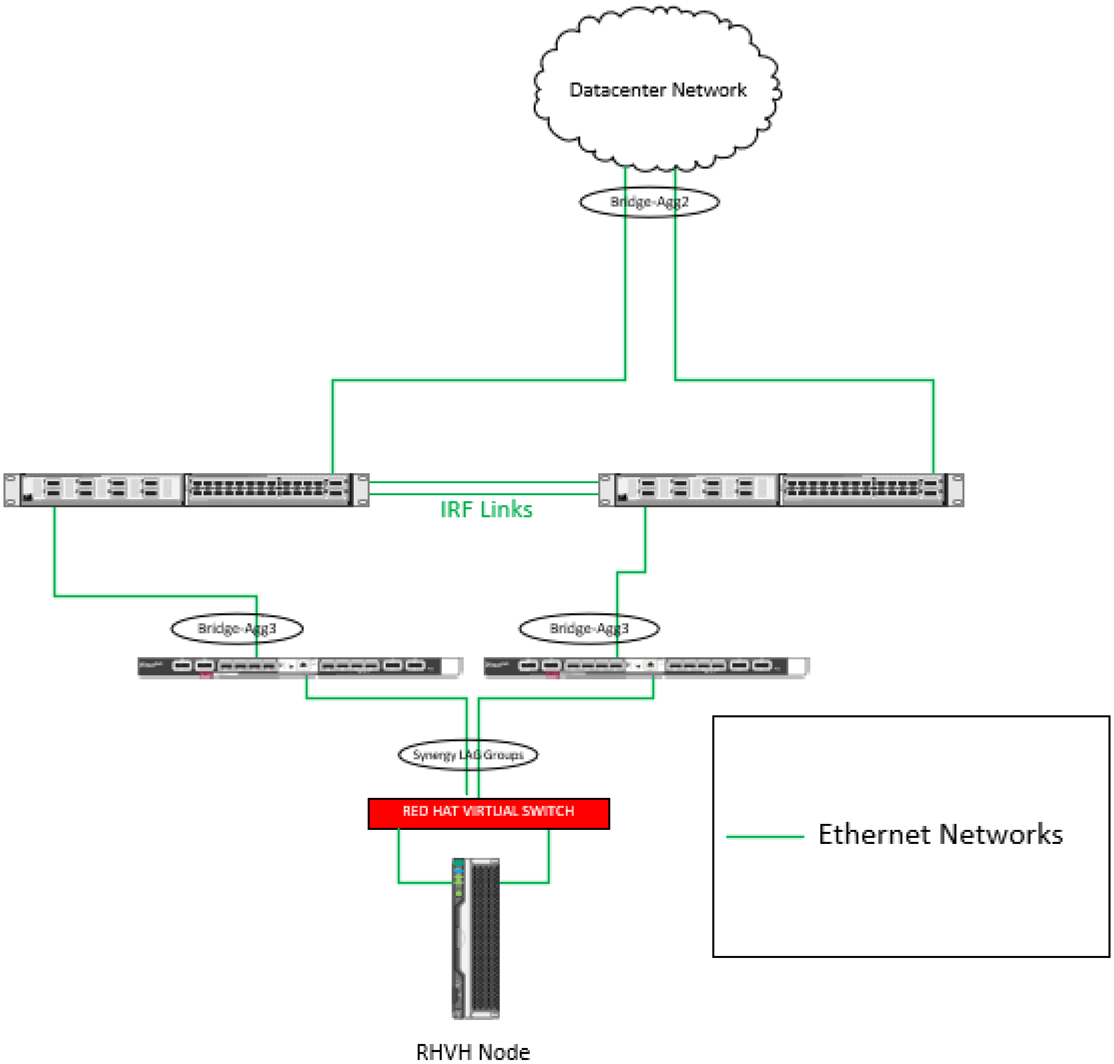

Figure 6: Red Hat OpenShift worker node network connectivity

HPE Synergy Composer

At the core of the management of the Synergy environment is HPE Synergy Composer. HPE Synergy Composer is used to configure the environment prior to the deployment of the operating systems and applications.

This section walks the installer through the process of installing and configuring the Synergy Composer.

Configure the HPE Synergy Composer

To configure HPE Synergy Composer with the installer laptop, follow these steps.

1. Configure the installer laptop physical NIC with the IP address 192.168.10.2/24. No gateway is required.
2. Connect a CAT5e cable from the laptop NIC to the laptop port on the front of a Synergy Composer.
3. Once connected, open a browser and point it to http://192.168.10.1:5800. This will open the HPE Synergy Console on the installer laptop.
4. Once the console comes up click **Connect** to start HPE OneView for Synergy.

5. Click on **Hardware Setup** and enter the following information when prompted. Note that this solution places the HPE Synergy Composer on the management network. Pre-populating DNS with IP information is recommended.
   - Appliance host name: **Fully qualified name of the HPE Synergy Composer**
   - Address assignment: **Manual**
   - IP address: **Enter an IP address on the management network**
   - Subnet mask: **Enter the subnet mask of the management network**
   - Gateway address: **Enter the gateway for the network**
   - Maintenance IP address 1: **Enter a maintenance IP address on the management network**
   - Maintenance IP address 2: **Enter a secondary maintenance IP address on the management network**
   - Preferred DNS server: **Enter the DNS server**
   - IPv6 Address assignment: **Unassign**
6. Once you have entered all information click on **OK** to proceed. This will start a hardware discovery process which may take some time to complete. Once the process has finished, check for issues and correct them. The "HPE Synergy 12000 Frame Setup and Installation Guide" available at hpe.com/info/synergy-docs offers suggestions to fix common issues.
7. Select the **OneView** menu at the top of the screen and click on **Settings** and then on **Appliance**. Validate that both appliances are connected and show a green checkmark.

Configure appliance credentials
Log into the **HPE OneView for Synergy appliance**. At first login you will be asked to define credentials for the Administrator user. To do this, accept the EULA and in the **HPE OneView Support** box insure that **Authorized Service** is **Enabled**. Log in as **Administrator** with the password **admin**. You will be prompted to enter a new password.

Configure solution firmware
This solution adheres to the firmware recipe specified with the HPE Converged Solutions 750 specifications which can be found in the Firmware and Software Compatibility Matrix. The solution used the firmware recipe from June of 2018.

1. Select the **OneView** menu and click on **Settings**.
2. Under **Appliance** select **Update Appliance** and **Update Composer**.

Once the update process completes, validate that the Composer module is connected and there is a green checkmark.

Solution configuration
The installer should utilize the Synergy Guided Setup to complete the following solution configuration details.

NTP
Configure the use of a network time server in the environment.

Create additional users.

It is recommended that you create a Read Only user and an Administrator account with a different username than Administrator.

Firmware
Upload a firmware bundle based on the aforementioned Converged Solutions 750 recipe. Once the bundle starts uploading you can proceed to additional steps without disrupting the upload.

Create an IP Pool on the management network.

Follow the guidance to create an IP pool on the management network. This IP pool will provide IP addresses to management IPs and HPE device iLOs within the solution. Ensure that the pool is enabled prior to proceeding.

Configure networks

As explained in the Network Configuration portion of the switch configuration section of this document, the solution utilizes at least four (4) network segments. Use the Create networks section of the Guided Setup wizard to define the networks shown in Table 9 at a minimum. Your VLAN values will generally differ from those described below.

Table 9. Networks defined within HPE Synergy Composer for this solution

Network Name	Type	VLAN Number	Purpose	Requested Bandwidth (Gb)	Maximum Bandwidth (Gb)
Management	Ethernet	10	Solution management	8	20
Public	Ethernet	194	Application, authentication and other user networks	8	20
Storage	Ethernet	20	Gluster Storage	8	20
PXE_Boot	Ethernet	30	PXE boot for compute	1	20

The management network should be associated with the management network IP pool the installer specified in the prior step. The installer should create any additional required networks for the solution.

Create Logical Interconnect Groups

Within Composer, use the Guided Setup to create a Logical Interconnect Group with two (2) Uplink Sets defined. The Uplink Set called Network carries all other networks defined for the solution.

Table 10 below defines the ports used to carry the Uplink Sets.

Table 10. Uplink Sets and Port Definitions

Uplink Set	Synergy Source
Network	Enclosure 1, Bay 3, Port Q5
	Enclosure 1, Bay 3, Port Q6

Create Enclosure Group

1. From the Guided Setup choose to **Create enclosure group**.
2. Provide a name and enter the number of frames.
3. Choose **Use address pool** and utilize the management pool defined earlier.
4. Use the Logical Interconnect Group from the prior step in the creation of the Enclosure Group.
5. Click on **Create** when ready.

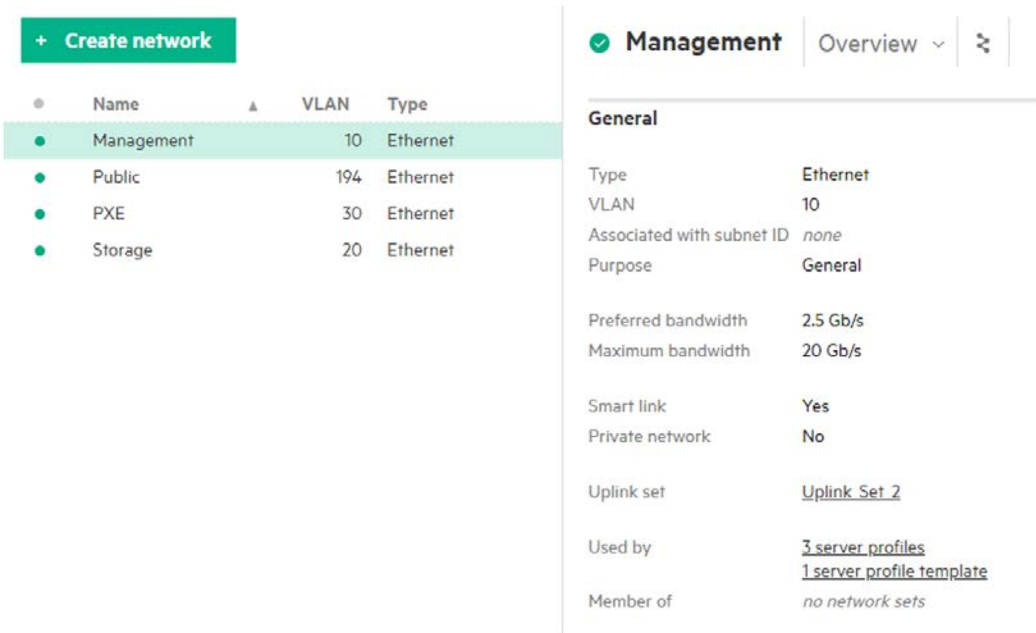

Figure 7: HPE OneView Synergy Networks

Create Logical Enclosure

Use the Guided Setup to create a logical enclosure. Select the firmware you uploaded earlier as a baseline. It can take some time for the firmware to update across the solution stack. Ensure that firmware complies with the baseline by selecting **Actions** and then **Update Firmware**. **Cancel** to exit.

Solution storage

Each server in this solution is configured with access to 12 physical disks, two from the internal drive bays and 10 from the D3940 Storage Module. This is a minimum requirement of 36 disks for the solution. The D3940 Storage Module can hold 40 disks; the storage requirements for the entire solution is provided by a single D3940 Storage Module. Additional D3940 Storage Modules can be added to the solution to provide space for additional disks to meet increased capacity, performance, and scalability requirements.

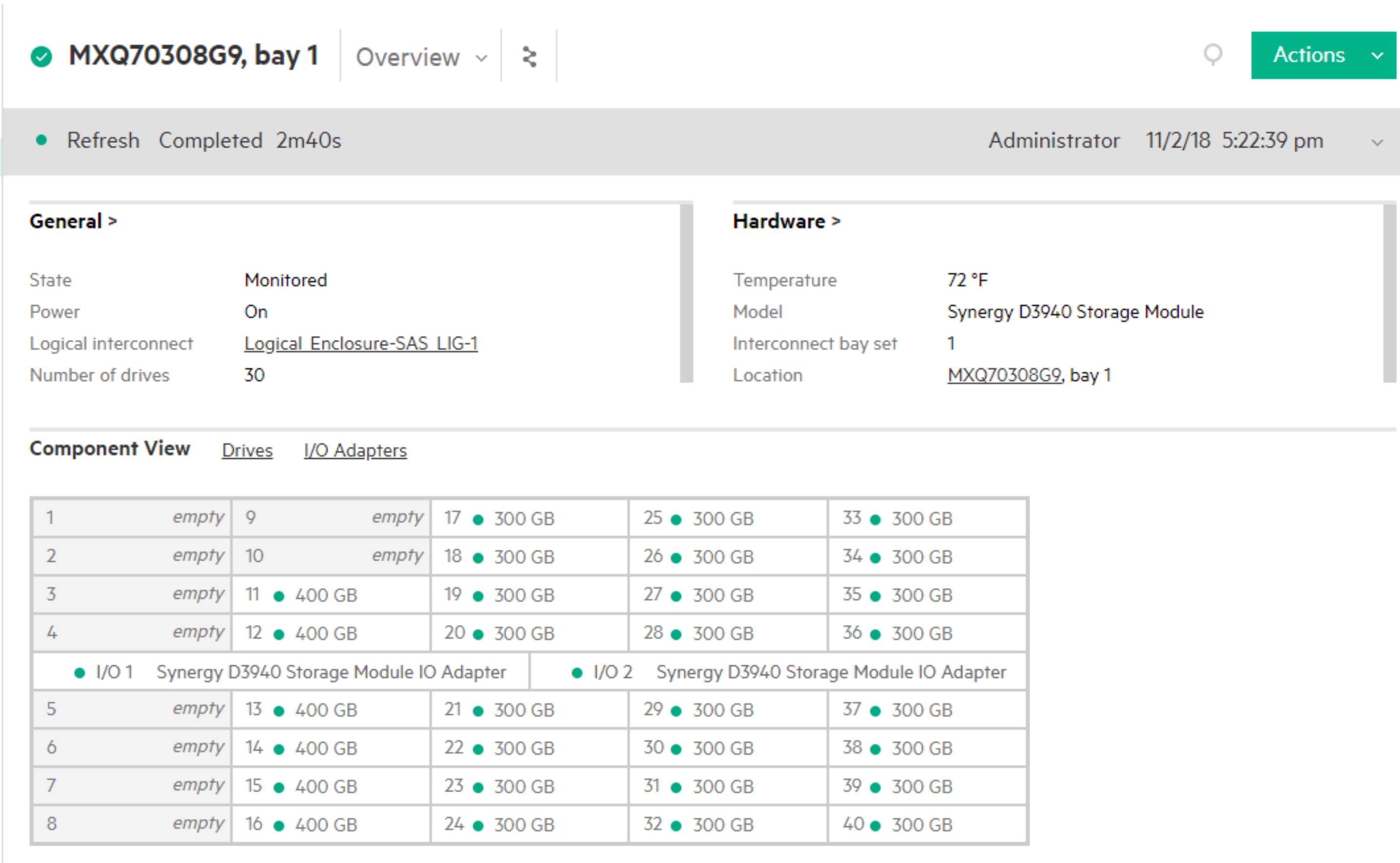

Figure 8: D3940 Storage Module physical disk layout

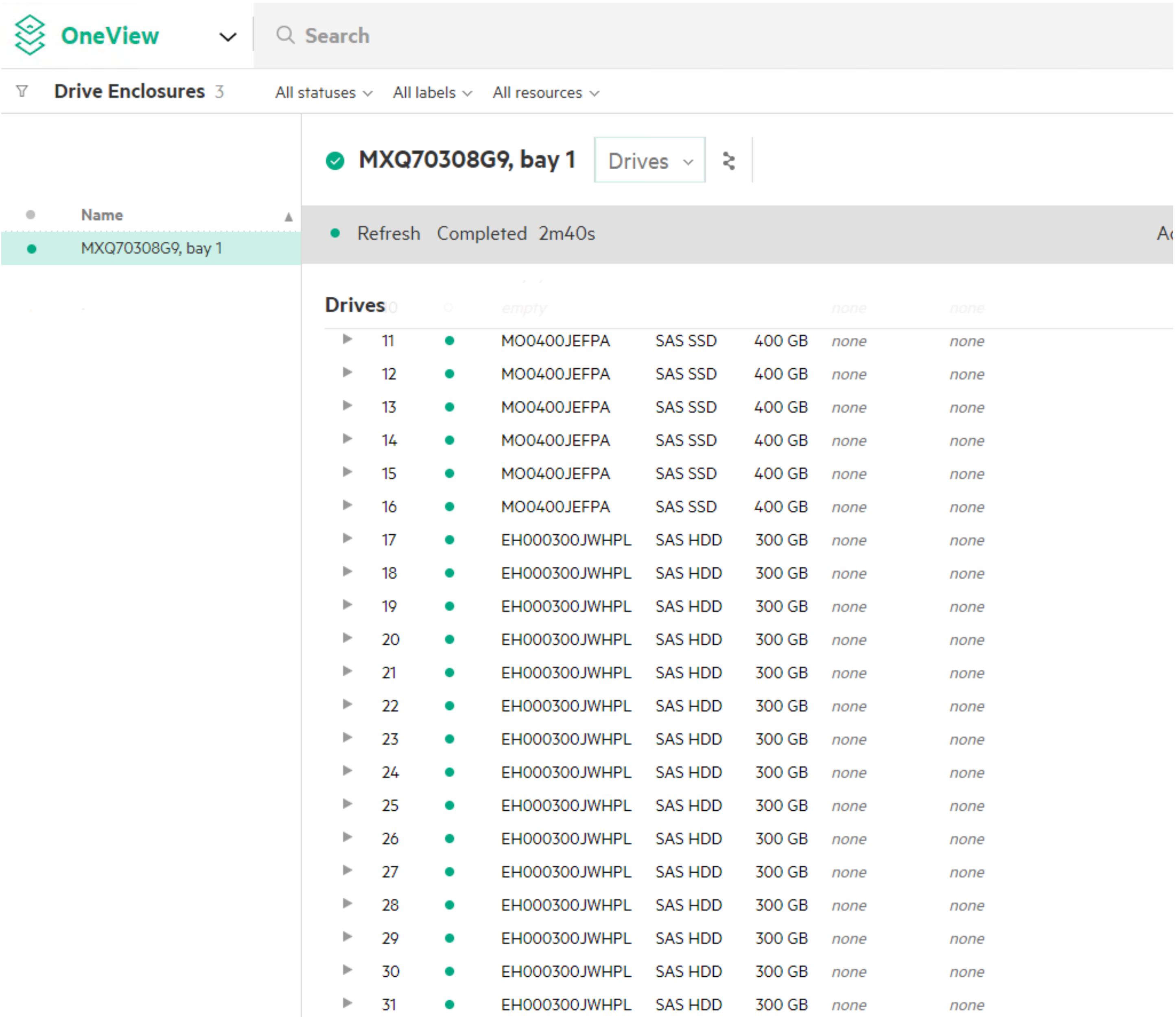

Figure 9: OneView Physical Drives in D3940 Storage Module

Information about storage volumes/disks is described in Table 11 below. All storage in this solution is local direct attached storage. Table 11 describes the storage volumes on a per-node basis.

Table 11. Volumes and sources used in this solution

Volume/Disk Function	Qty	Size	Source	Hosts	Shared/Dedicated
Operating System	1	300GB	Internal drive bays	1 RAID1 per RHVH host	Dedicated
Virtual Machine Hosting / RHVM – Gluster	1	1.8TB	D3940	1 RAID6 per RHVH host	Replicated
Persistent Application Data – OCS	1	400GB	D3940	1 RAID0 per RHVH host / OpenShift worker node	Replicated
OpenShift Container Registry – OCS	1	400GB	D3940	1 RAID0 per RHVH host / Infrastructure node	Replicated

Compute Module configuration

This section describes the configuration of the compute modules and is separated into sections that disseminate universal configuration parameters, options exclusively for virtualized master nodes, and options exclusively for bare metal worker nodes. Required configuration steps are outlined. These may be in the form of Kickstart file examples, pointers to code, or command line options. It is up to the installer to decide how to reach the desired end state outlined in this solution within their compute environment.

Server profiles

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. These templates are the key to delivering the "infrastructure as code" capabilities of the HPE Synergy platform. For the purpose of this solution, a single template was created that was applied to all compute modules.

The critical items configured as part of the template were the connections and storage. Figure 10 describes the configuration of the network interfaces as part of the profile template. There are 8 redundant networks that are defined.

Network connections

Connections

Expand all Collapse all

ID	Name	Network	Port	Boot
1	pxe	PXE VLAN30	Mezzanine 3:1-a	PXE primary
2	mgmt1	Management VLAN10	Mezzanine 3:1-b	Not bootable
4	pub1	Public VLAN194	Mezzanine 3:1-c	Not bootable
6	stor1	Storage VLAN20	Mezzanine 3:1-d	Not bootable
7	mgmt2	Management VLAN10	Mezzanine 3:2-a	Not bootable
8	pub2	Public VLAN194	Mezzanine 3:2-b	Not bootable
9	stor2	Storage VLAN20	Mezzanine 3:2-c	Not bootable

Figure 10: Server network connections as part of the profile template

---

Note

The MAC addresses are defined and available even if a compute module has not become active. The MAC address for the interface connected to the PXE network is required to create a DHCP reservation as describe in Appendix B: PXE Configuration.

Storage connections

The storage for this solution is local storage provided by the D3940 Storage Module and the internal drive bays of the SY480 Compute Modules. 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.

Each of the RHV / RHHI hosts have the following volumes:

- Bootvol: RAID1 volume provide by the two internal drives in the SY480 Compute Module
- Gluster: RAID6 volume comprised of eight physical disks presented from the D3940 Storage Module
- OCS-1: RAID0 volume single SSD disk presented from the D3940 Storage Module used for Open Container Storage
- OCS-2: RAID0 volume single SSD disk presented from the D3940 Storage Module used for Open Container Storage

Figure 11 shows the configuration of the local storage from within the profile.

Local Storage

Managed manually

SAS Mezz 1 storage controller

Managed by OneView

Initialization will occur on next assignment to server hardware

Name	Type	RAID Level	Number of Drives	Size GB	Drive Technology	Boot	Erase on Delete	
bootvol	Internal logical drive	RAID 1	2	*n/a*	not specified	☑	*n/a*	×
gluster	External logical drive	RAID 6	8	300	SAS HDD	☐	No	×
ocs-1	External logical drive	RAID 0	1	400	SAS SSD	☐	No	×
ocs-2	External logical drive	RAID 0	1	400	SAS SSD	☐	No	×

Figure 11: Local storage used for this solution

Note

In order to complete the installation of the required software in the following sections, internet access is required and should be enabled on at least one active adapter.

Red Hat Virtualization hosts

This section outlines the installation and configuration of the Red Hat Virtualization Hosts in a Red Hat Hyperconverged Infrastructure (RHHI). The Red Hat Hyperconverged Infrastructure configures the three Red Hat Virtualization hosts in a three node RHV cluster using Gluster storage for the virtual machines and a self-hosted engine deployment.

Red Hat Virtualization host installation

This solution utilizes a combination of Kickstart and manual processes to configure Red Hat Virtualization on three (3) compute modules.

In our example the NICs belong to the following networks.

- ens3f0: PXE
- ens3f1 and ens3f2: bond0: Management
- ens3f3 and ens3f4: bond1: Public
- ens3f5 and ens3f6: bond2: Storage

The installer will need to understand what NIC belongs to each of your networks. This can again be accomplished via aligning MAC addresses with the Server Profile in Composer.

Insure the Management interface for each RHVH host is registered in DNS prior to proceeding.

The Kickstart file (shown below) is hosted over HTTP. The file can be created with a text editor such as vim or nano. Refer to Appendix B: PXE Configuration for an example of setting up a server that will provide PXE, HTTP, and DHCP services to enable the PXE boot environment.

```
%pre --log=/tmp/pre.log
%end

autopart --type=thinp
zerombr
clearpart --all --initlabel
liveimg --url=http://192.168.10.101/sqimage/squashfs.img
rootpw --plaintext changeme
ignoredisk --only-use=sdb
timezone --utc America/Chicago
network --device=ens3f1 --bootproto=static
network --device=ens3f2 --bootproto=static
network --device=ens3f3 --bootproto=static
network --device=ens3f4 --bootproto=static
network --device=ens3f5 --bootproto=static
network --device=ens3f6 --bootproto=static
network --device=bond0 --bondslaves=ens3f1,ens3f2 --bondopts=mode=802.3ad --bootproto=static --gateway=192.168.1.3 --
nameserver=192.168.1.101,15.226.47.253 --activate
network --device=bond1 --bondslaves=ens3f3,ens3f4 --bondopts=mode=802.3ad --bootproto=static --nodefroute --activate
network --device=bond2 --bondslaves=ens3f5,ens3f6 --bondopts=mode=802.3ad --bootproto=static --nodefroute --activate
text
reboot

%post --erroronfail
imgbase layout --init
nodectl init
%end
```

Post Kickstart configuration

This section outlines the steps required to configure compute modules once the hypervisor has been installed. Ansible is used to prepare the hosts for the RHHI installation. The configuration of RHHI will be accomplished manually using the Hyperconverged Cockpit user interface.

The first step after the servers have been kickstarted and installed with the RHVH operating system image is to run the Ansible RHHI playbook. This playbook will perform the following tasks:

- Create and distribute ssh keys from the first RHHI host.
- Set the hostname on each host
- Register the hosts with Red Hat CDN
- Configure the IP Addresses on the bonded interfaces
- Create an additional user

Refer to Appendix C: Configure Ansible for instructions on configuring the Ansible environment.

1. On the Ansible host, if not already complete, clone the git repository for this solution:
   a. Git clone <URL of repo>
2. Configure the variables in the following files to reflect values specific to your environment:

   hosts
   vault_pass.yml
   roles/rhhi/vars/main.yml
   roles/deployvm/vars/main.yml
   roles/deploytemplate/vars/main.yml
   roles/preparehosts/vars/main.yml

3. Run the RHHI playbook to configure the RHHI hosts
   a. ansible-playbook -e@vault_pass.yml playbooks/rhhi.yaml

Configuring the virtualization environment

Once the rhhi.yaml playbook has completed, the three RHV hosts are ready for the Red Hat Virtualization Hosted Engine Setup using the Red Hat Virtualization Cockpit interface. This will create the Red Hat Hyperconverged Infrastructure. The first step in creating the cluster is to deploy the Hosted Engine in a self-hosted mode on one of the RHVH hosts using the Red Hat Virtualization Cockpit User Interface. If not logged on, access the Cockpit user interface by connecting a browser interface to https://<FQDN rhvh host:9090> of the first RHVH server. This step should only be carried out on one host. Log into the Cockpit user interface with the host's local account credentials. The resulting screen appears in Figure 12.

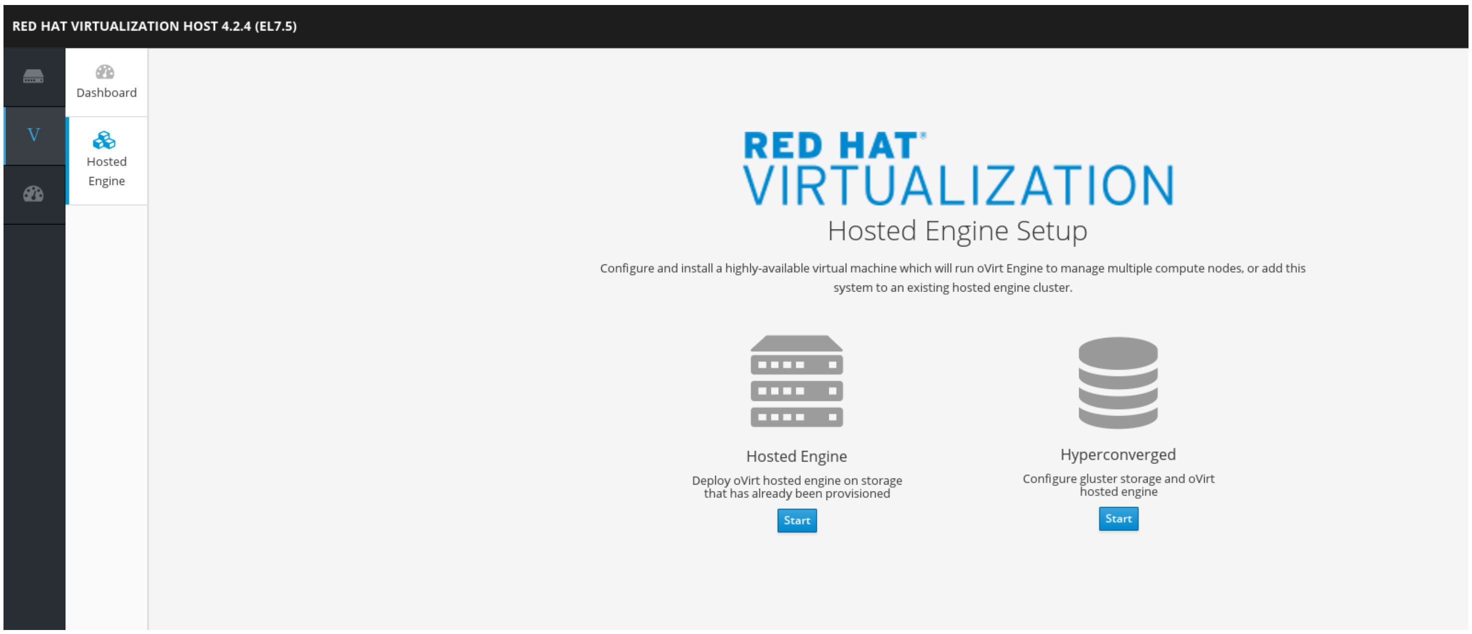

Figure 12: Hosted Engine Setup screen of the Cockpit interface

Deploying Hosted Engine

1. Select the **Hyperconverged Start** Icon to begin the Gluster deployment and then continue to the Hosted Engine deployment. The Hyperconverged deployment is divided into three sections:
   a. Gluster Deployment
   b. Preparing the Hosted Engine Virtual Machine
   c. Preparing the Hosted Engine storage
2. Complete the Gluster Deployment Wizard. In the Hosts screen enter the IP addresses RHVH hosts. The IP addresses entered here are the IP addresses for the Storage network, these are the addresses assigned to storage_ip variable in the Ansible hosts file and assigned to bond2 during the configuration of the RHHI hosts using the Ansible RHHI playbook. In this solution, this corresponds to the IP address assigned to the bond2 interface on each RHVH host. Select **Next** and proceed to the FQDNs form.

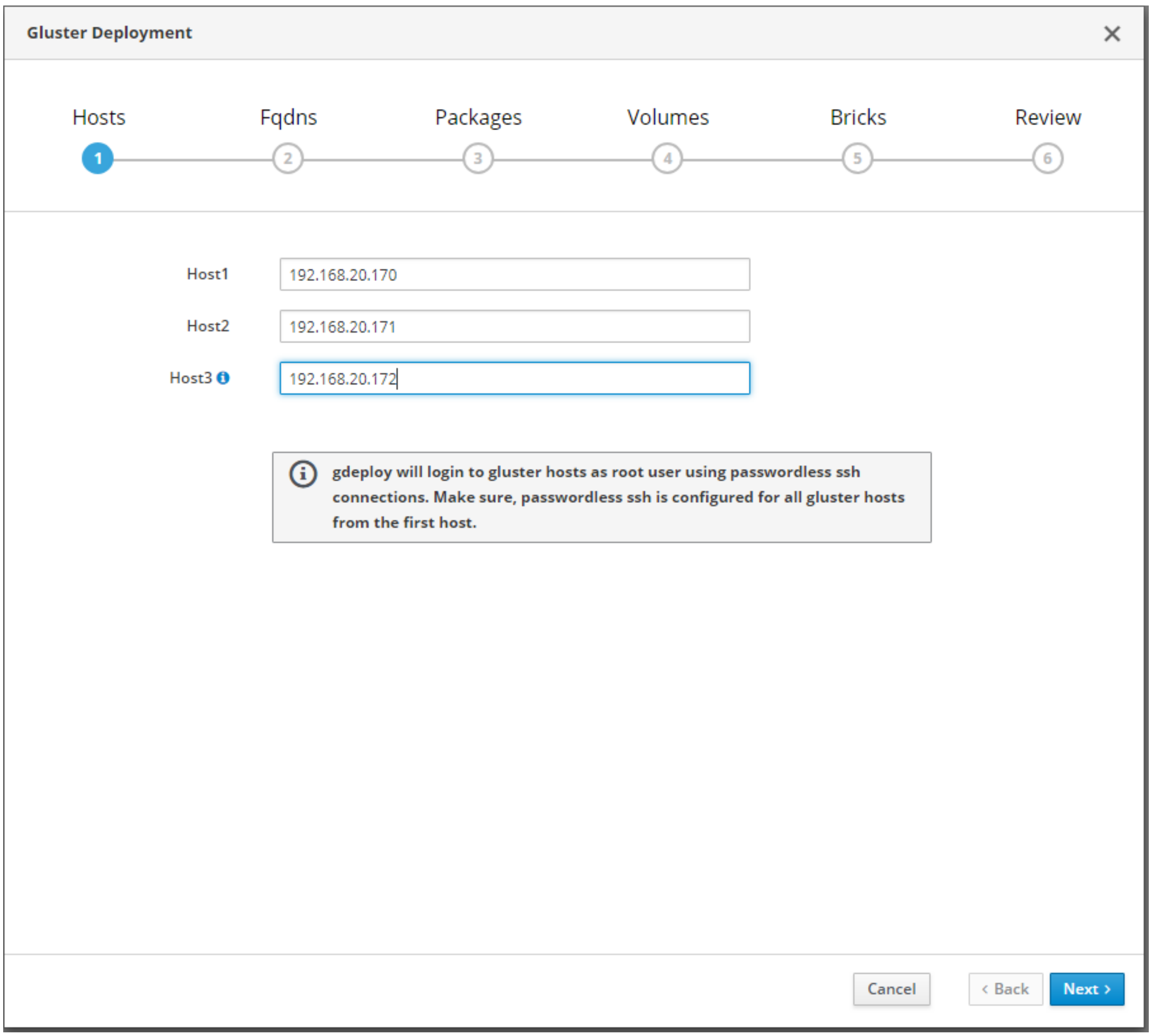

Figure 13: Gluster Deployment Hosts

3. In the FQDN form enter the fully qualified domain names of the additional RHVH hosts that will be members of the RHV cluster. These hosts will automatically be added to the Hosted Engine as members of the RHV default cluster available to run the Hosted Engine virtual machine. Select **Next** to continue to the Packages form.

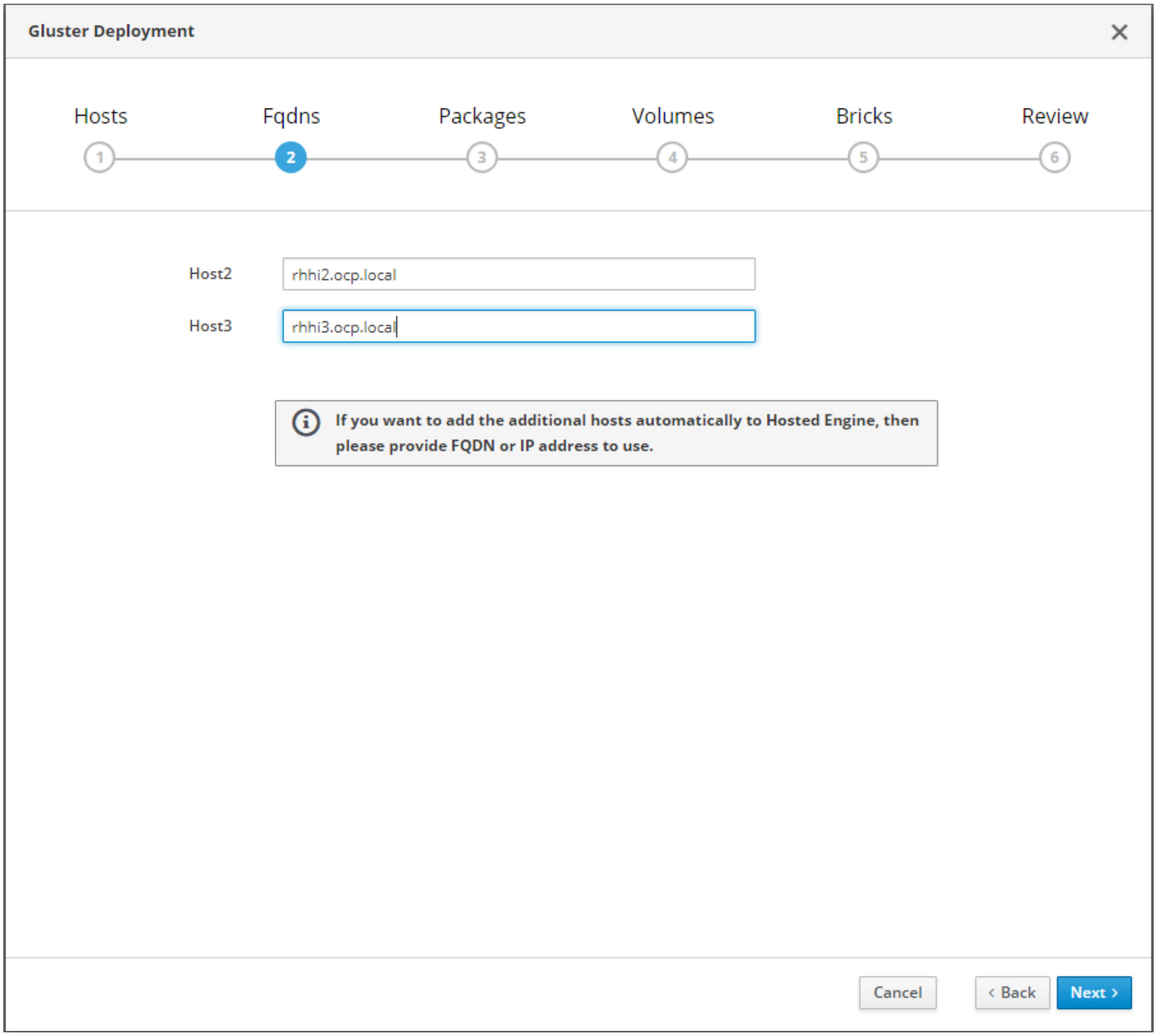

Figure 14: Gluster Deployment FQDN

4. The Packages form allows the installer to select additional repositories to be enabled and packages to be installed. In this solution, no additional packages are required. Leave the Packages and Repositories fields blank, and select **Next** to continue to the Volumes form.

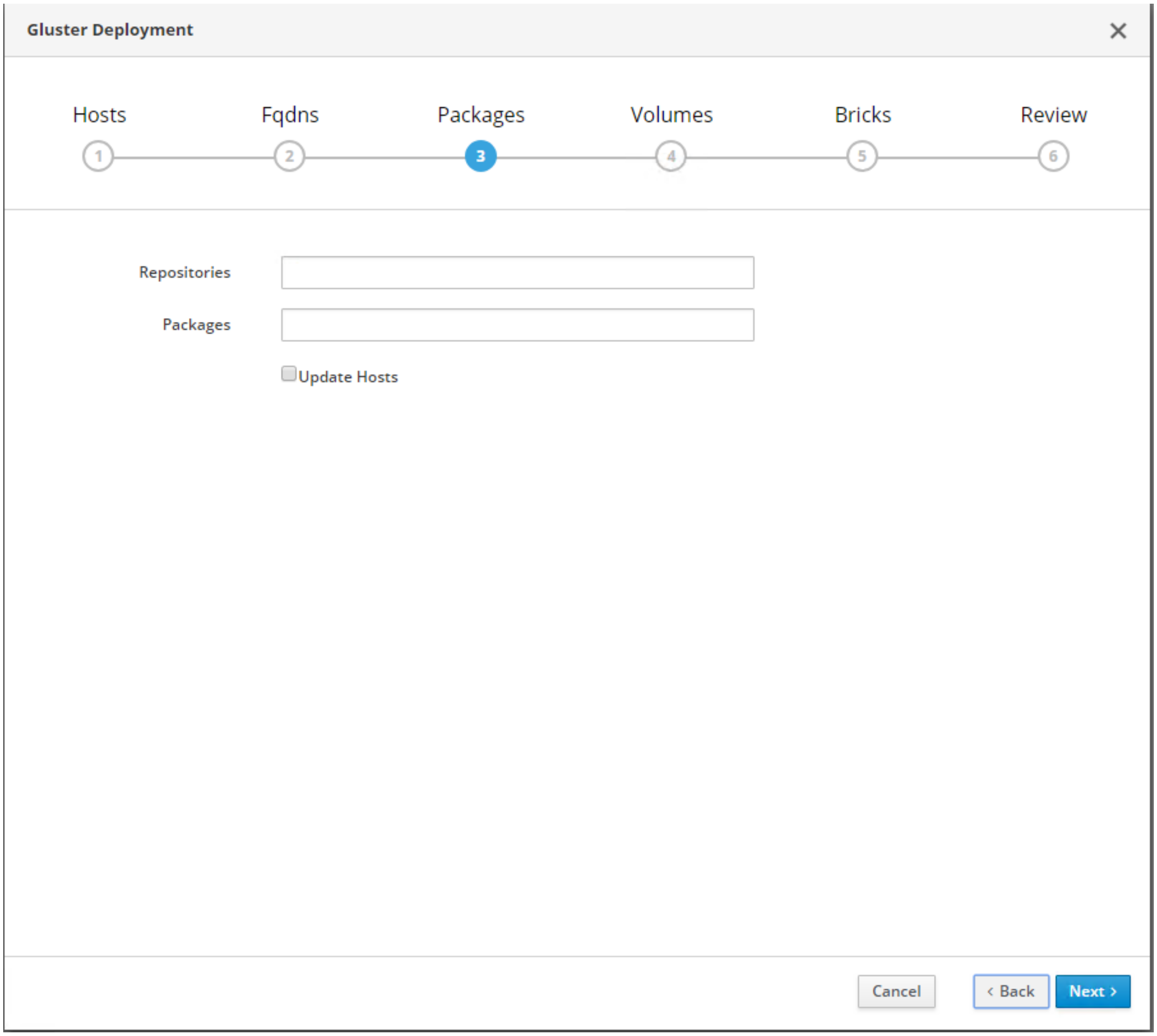

Figure 15: Gluster Deployment Packages

5. The Volumes form provides input fields for the volume Name, Volume Type, and Brick Directories for the Gluster volumes that will be created by the installation script. Select **Next** and continue to the Bricks form.

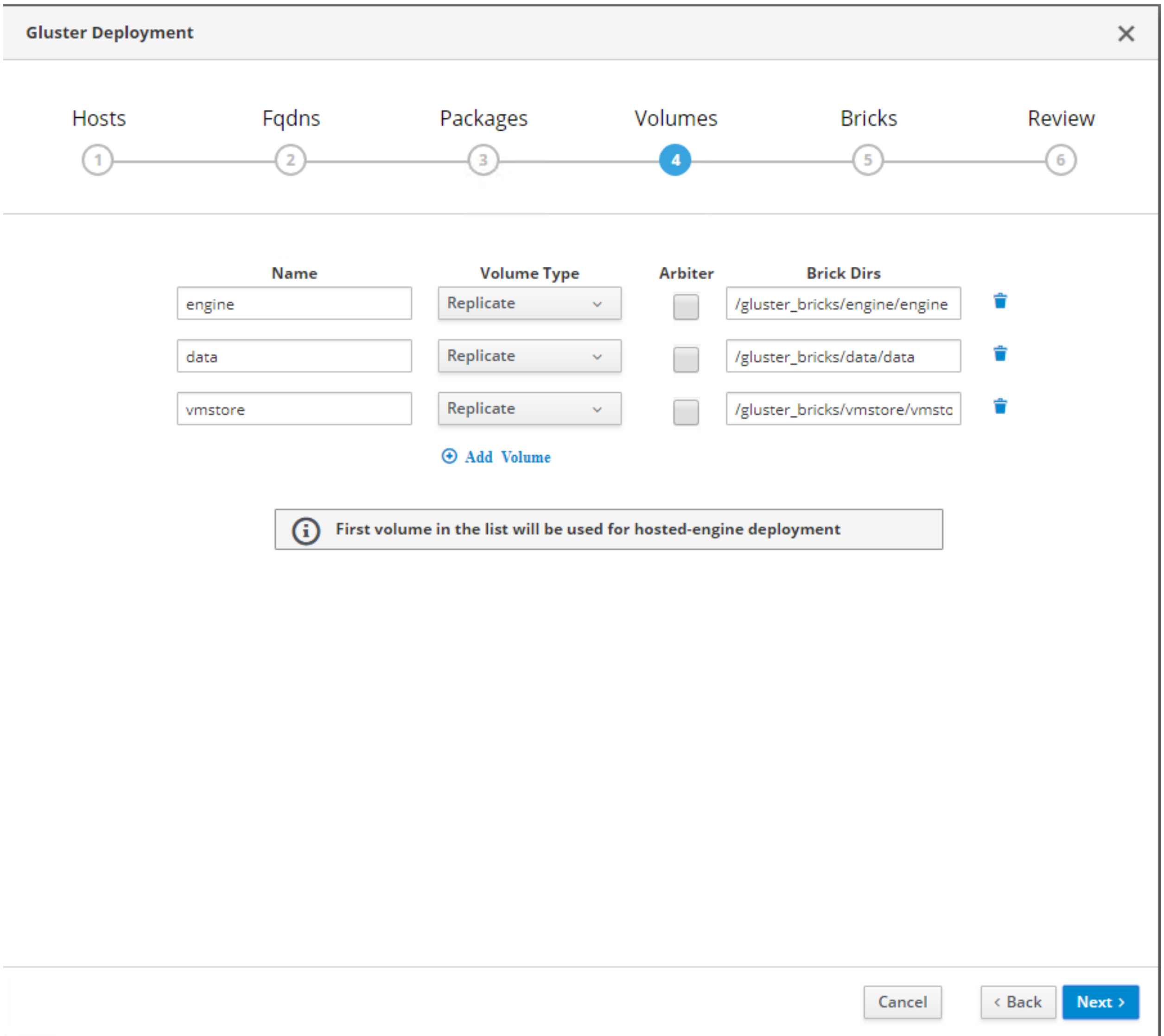

Figure 16: Gluster Deployment Volumes

6. The Bricks form provides input fields for the physical disk configuration that will host the Gluster volumes on the RHV hosts. In this solution, each of the three RHHI hosts are configured with a RAID 6 array that is identified as /dev/sdc. This device will be used for the Device Name in the Gluster Brick Configuration. Select **Next** to continue to the Review form.

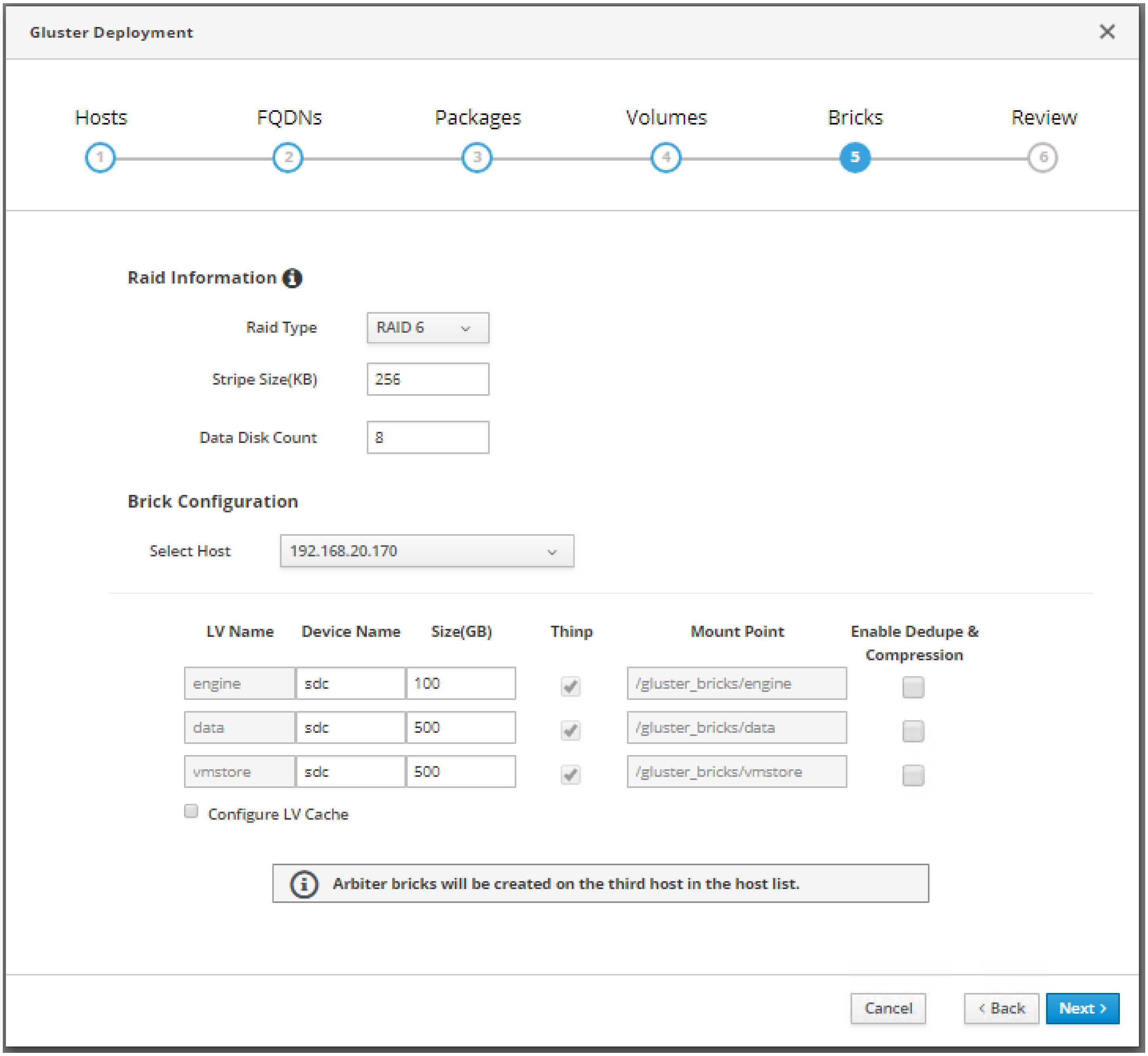

Figure 17: Gluster Deployment Bricks

7. The Review form allows the installer to review the selections in the Gluster deployment forms prior to deploying Gluster. Review the selections and click **Deploy** to deploy Gluster on the RHHI hosts.

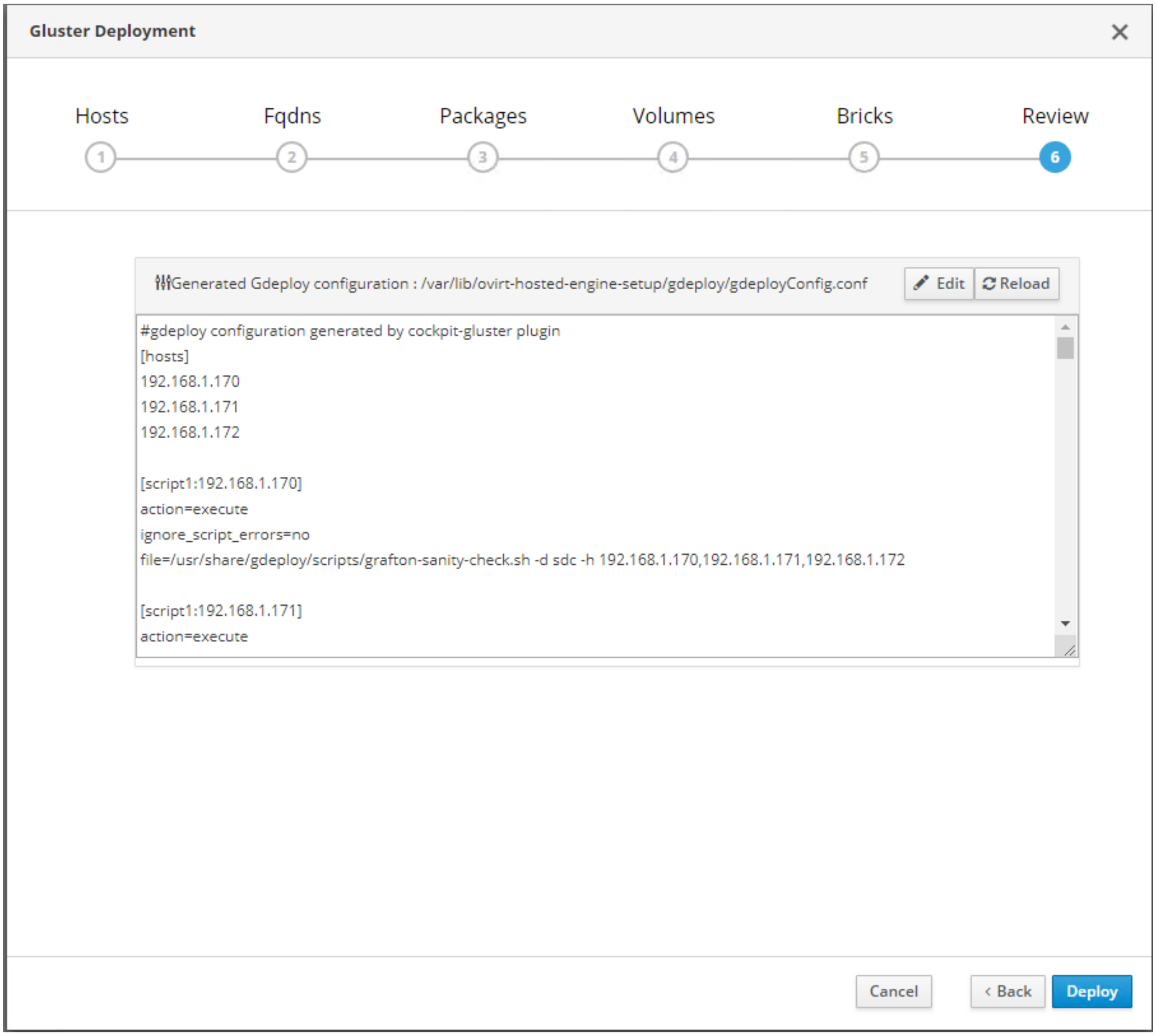

Figure 18: Gluster Deployment Review Selections

8. Once the Gluster deployment is completed successfully, the installer will be prompted to continue onto the Hosted Engine Deployment as shown in the figure below. Select **Continue to Hosted Engine Deployment** and continue to the next step.

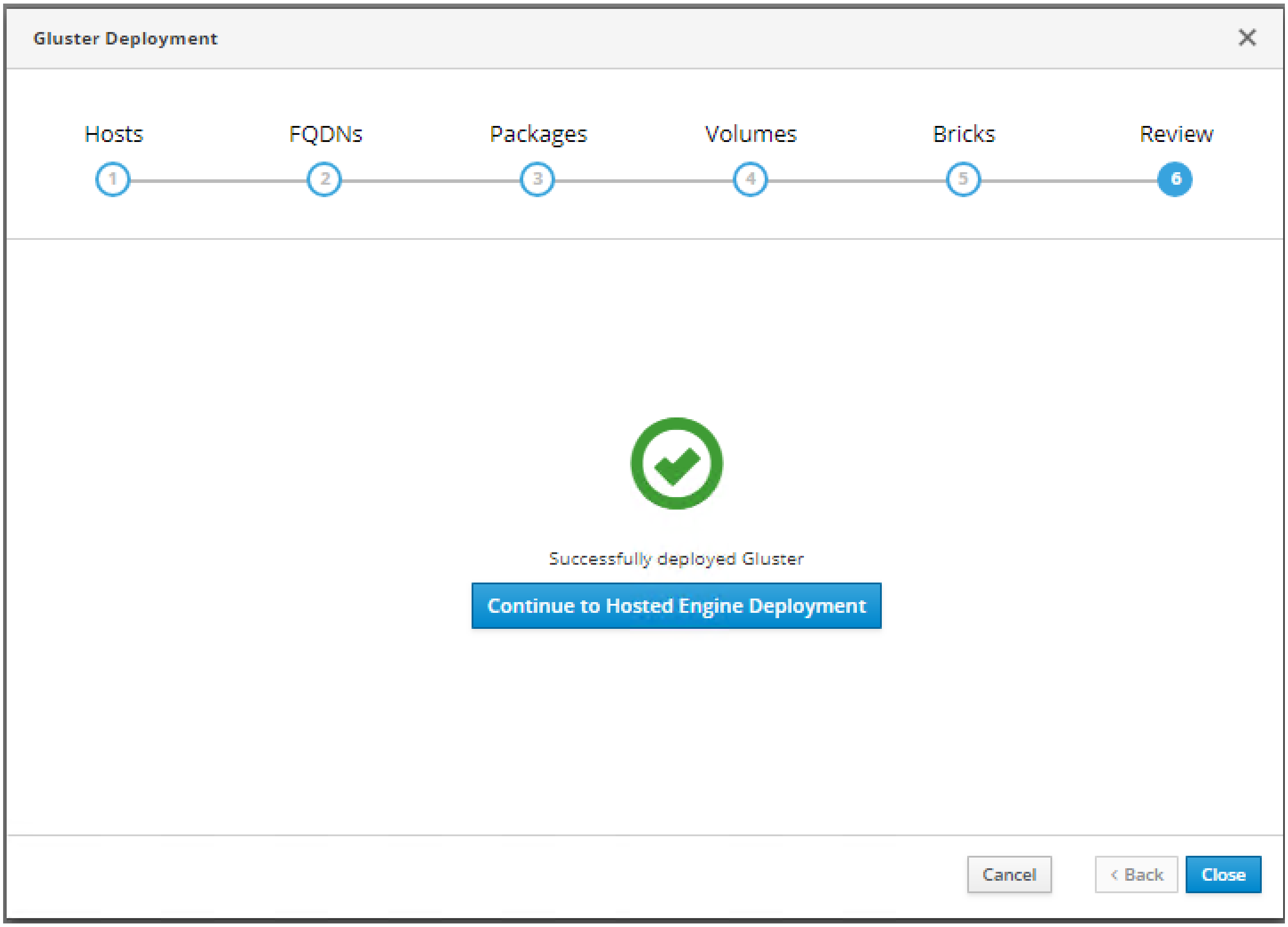

Figure 19: Gluster Deployment Success Notification

9. Complete the Hosted Engine deployment configuration items as in Figure 22. Ensure the Engine VM FQDN is resolvable in DNS. Provide a static IP address for the Engine Network Configuration and complete the configuration fields. In this solution, Bond0 is the management network. Select **Validate** to validate the settings. Select **Next** to continue.

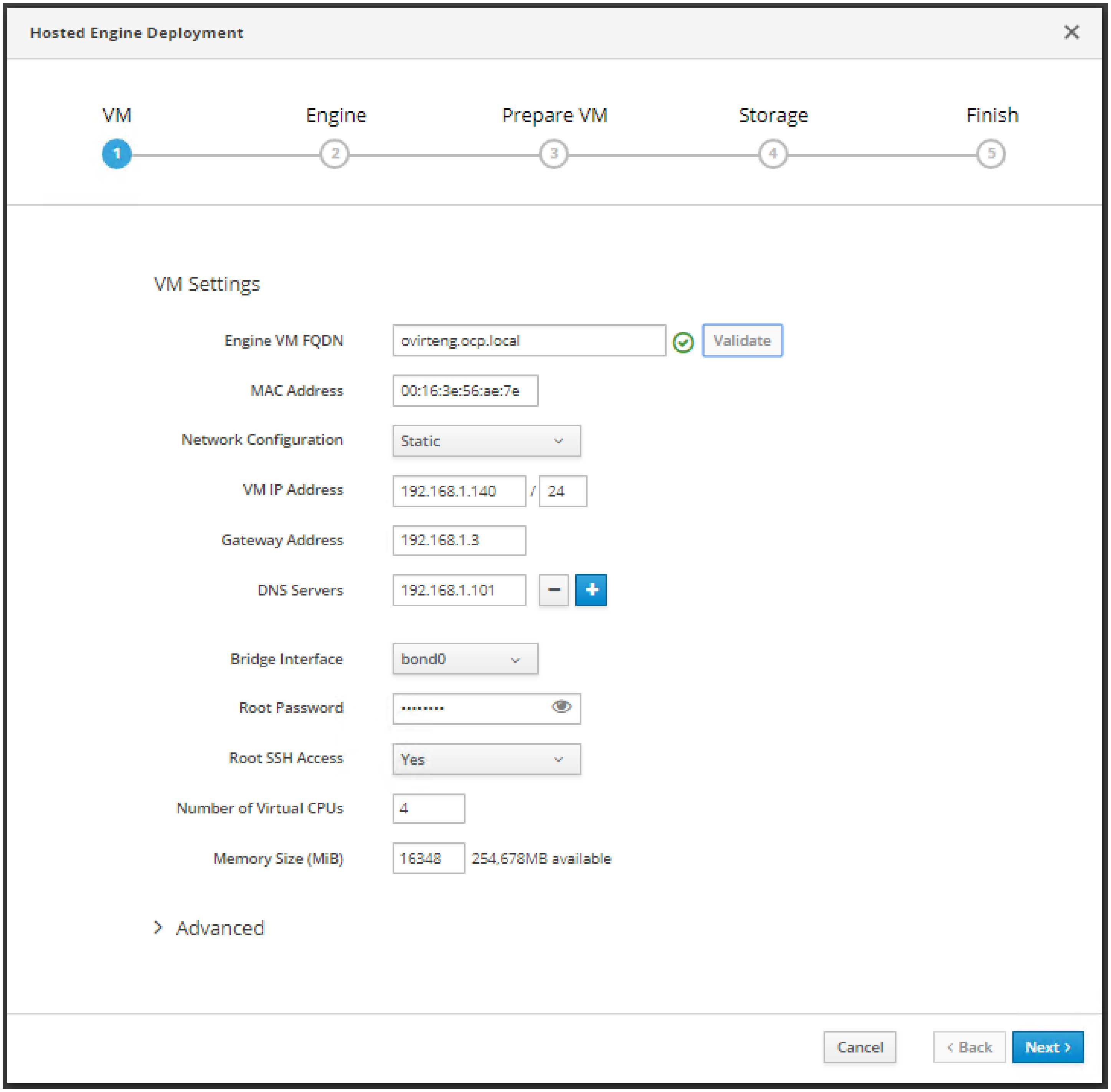

Figure 20: Hosted Engine Deployment Virtual Machine Settings

10. Fill in the Engine Credentials and Notification Settings for your environment and select **Next** to continue.

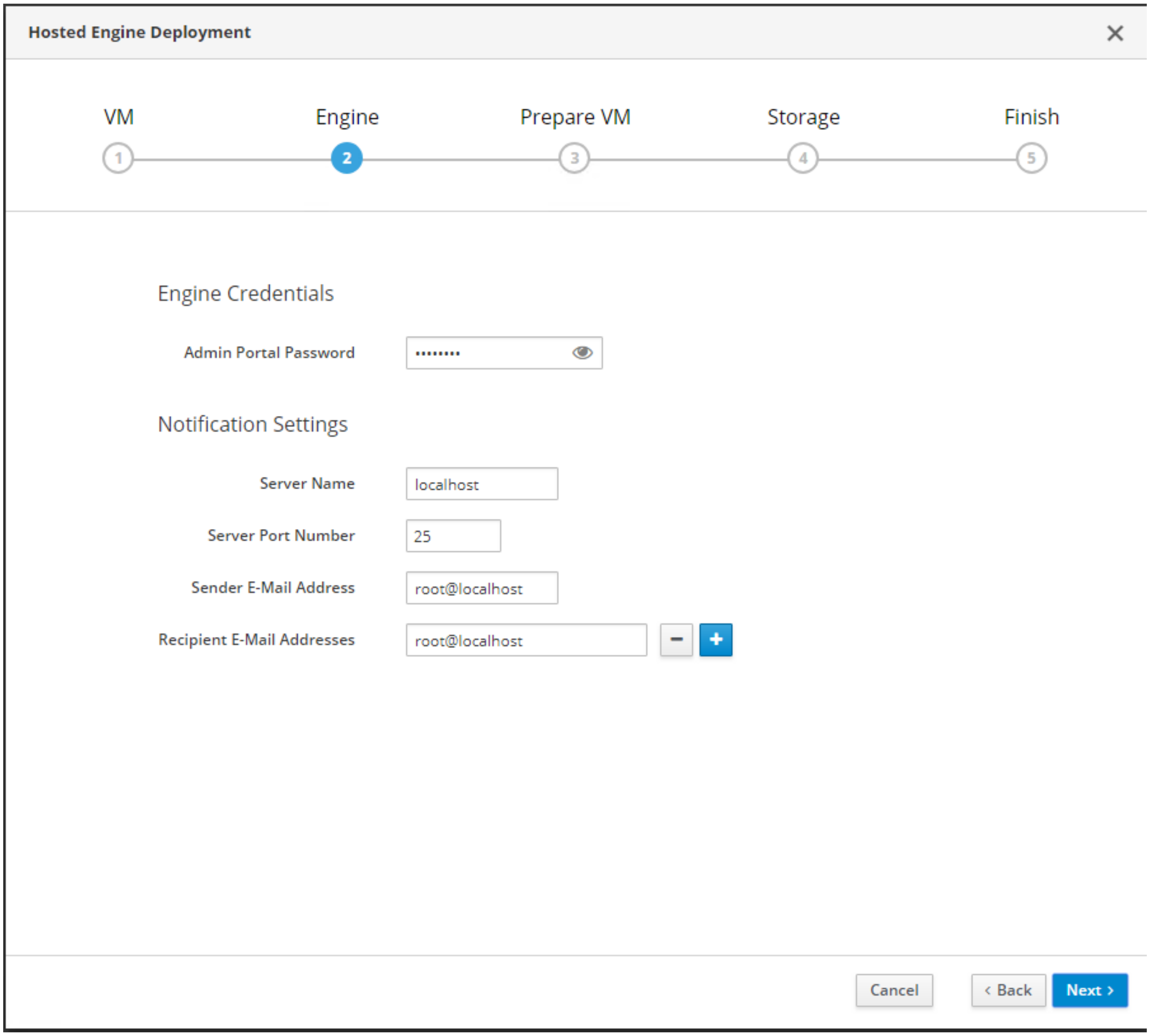

Figure 21: Hosted Engine Deployment Engine Credential and Notification Settings

11. Review the summary on the Prepare VM screen and select **Prepare VM**.

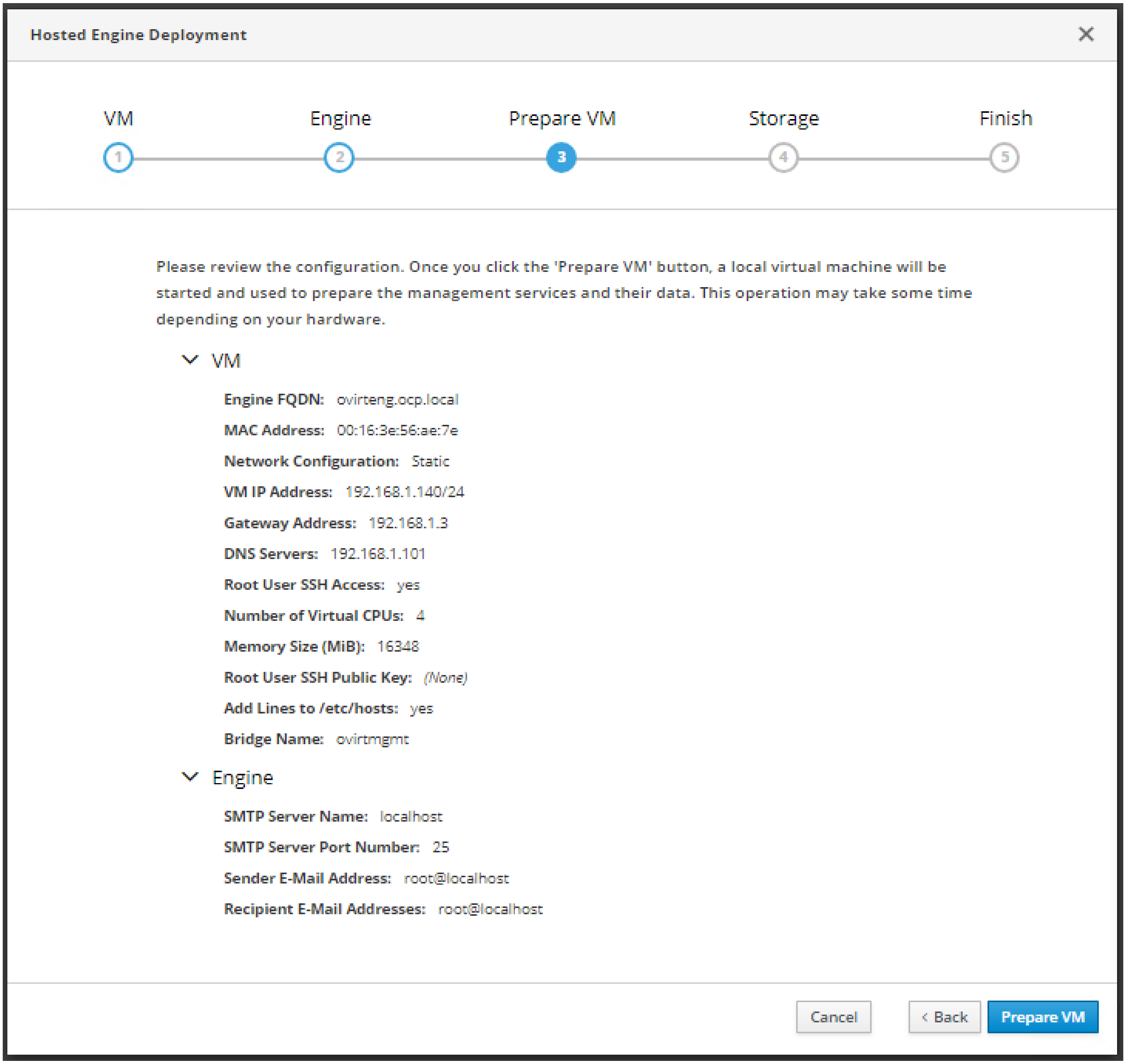

Figure 22: Hosted Engine Deployment Review Selections Prepare VM

12. Wait for the Hosted Engine virtual machine deployment to complete successfully and continue with Hosted Engine storage configuration. This can take some time. When the process completes, click on **Next** to continue to the Storage setup of the Hosted Engine Deployment.

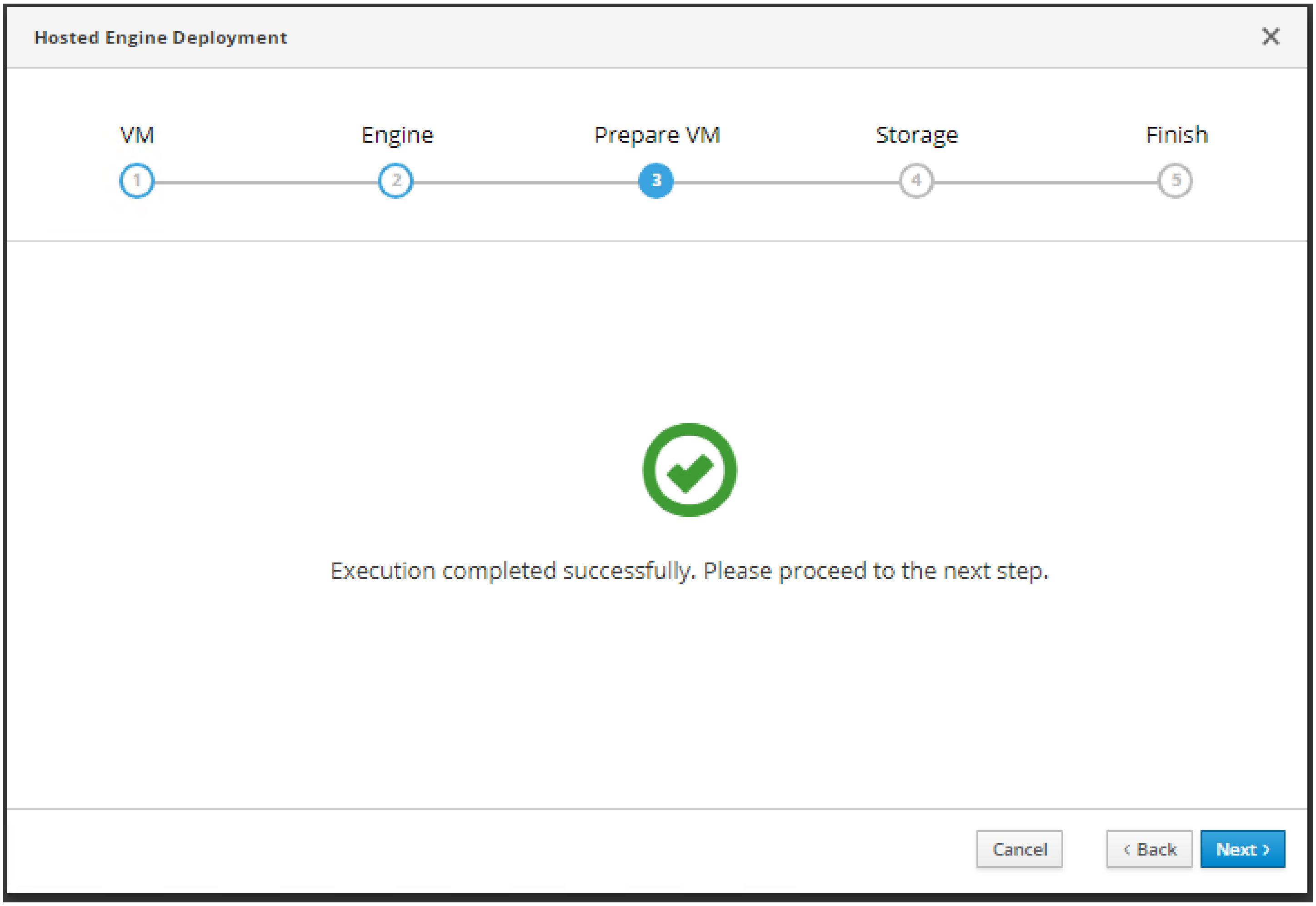

Figure 23: Hosted Engine Deployment Success Notification

13. In the Storage screen, under **Storage Settings**, the settings for this form will be preconfigured with the correct values for the environment. **Gluster** will be selected as the Storage Type. The Storage Connection is the IP address of the storage network interface for the first RHHI host. In this example the Storage network is 192.168.20.0 / 24, this network is assigned to bond2, the storage network bond. The Mount Options will be pre-populated with the storage network IP addresses for the additional two RHHI servers. Verify the information as shown in Figure 24 and click **Next** to continue.

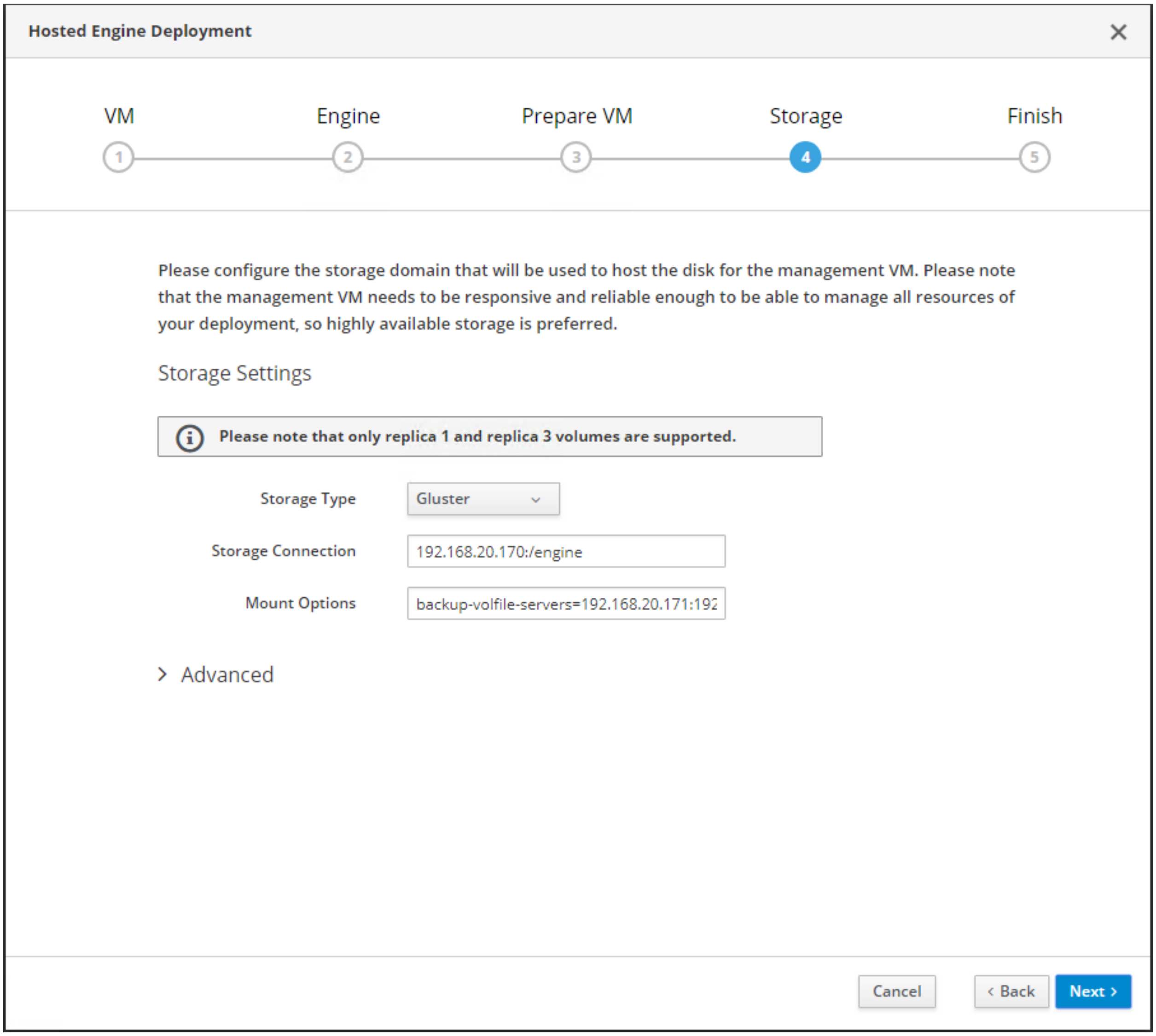

Figure 24: Hosted Engine Deployment Storage Settings

14. Review the storage configuration and select **Finish Deployment**. The process can take some time.

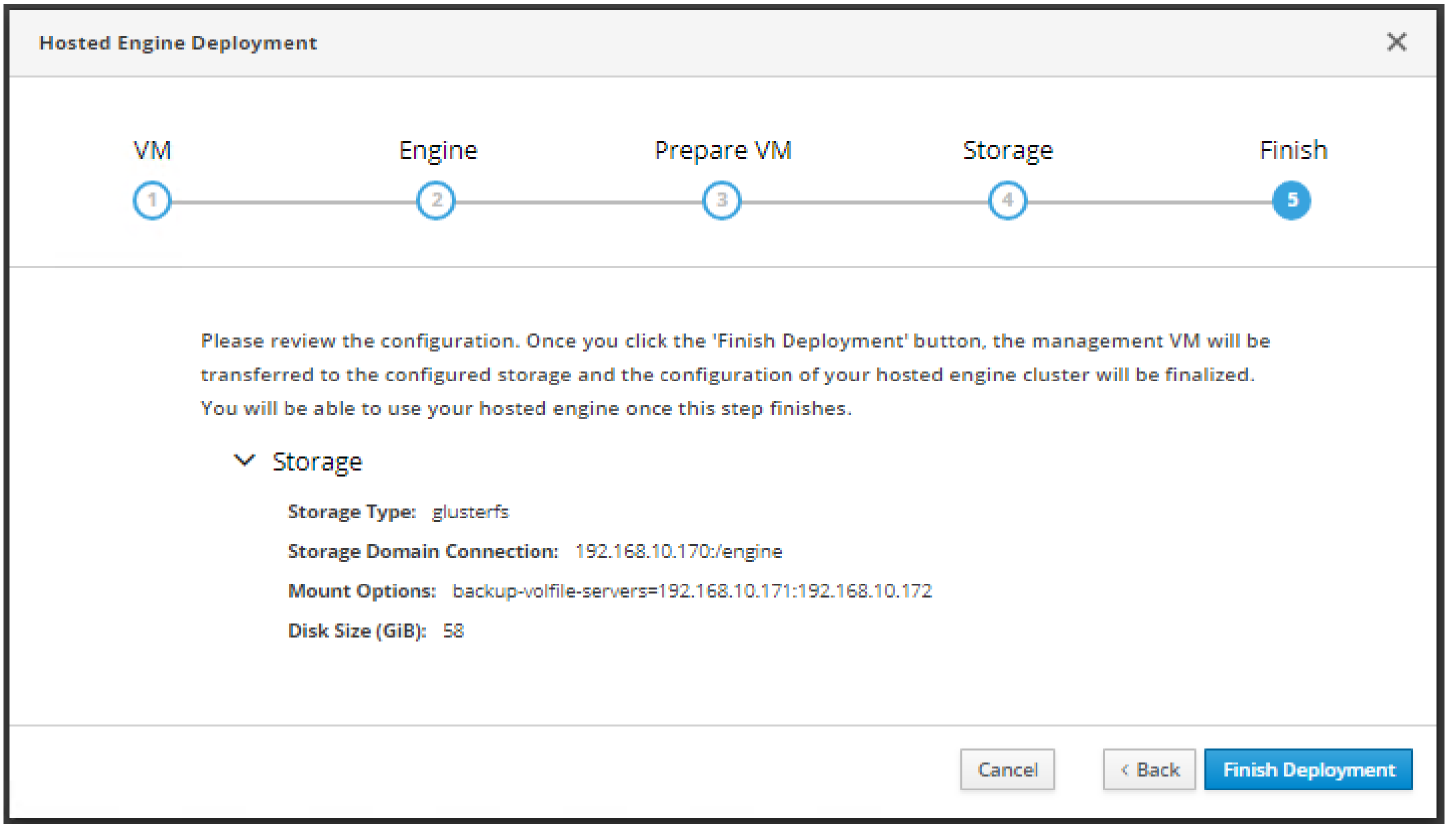

Figure 25: Hosted Engine Deployment Review Storage Settings Finish Deployment

15. When the Hosted Engine Deployment completes successfully click **Close** to exit the application deployment as shown below in Figure 26.

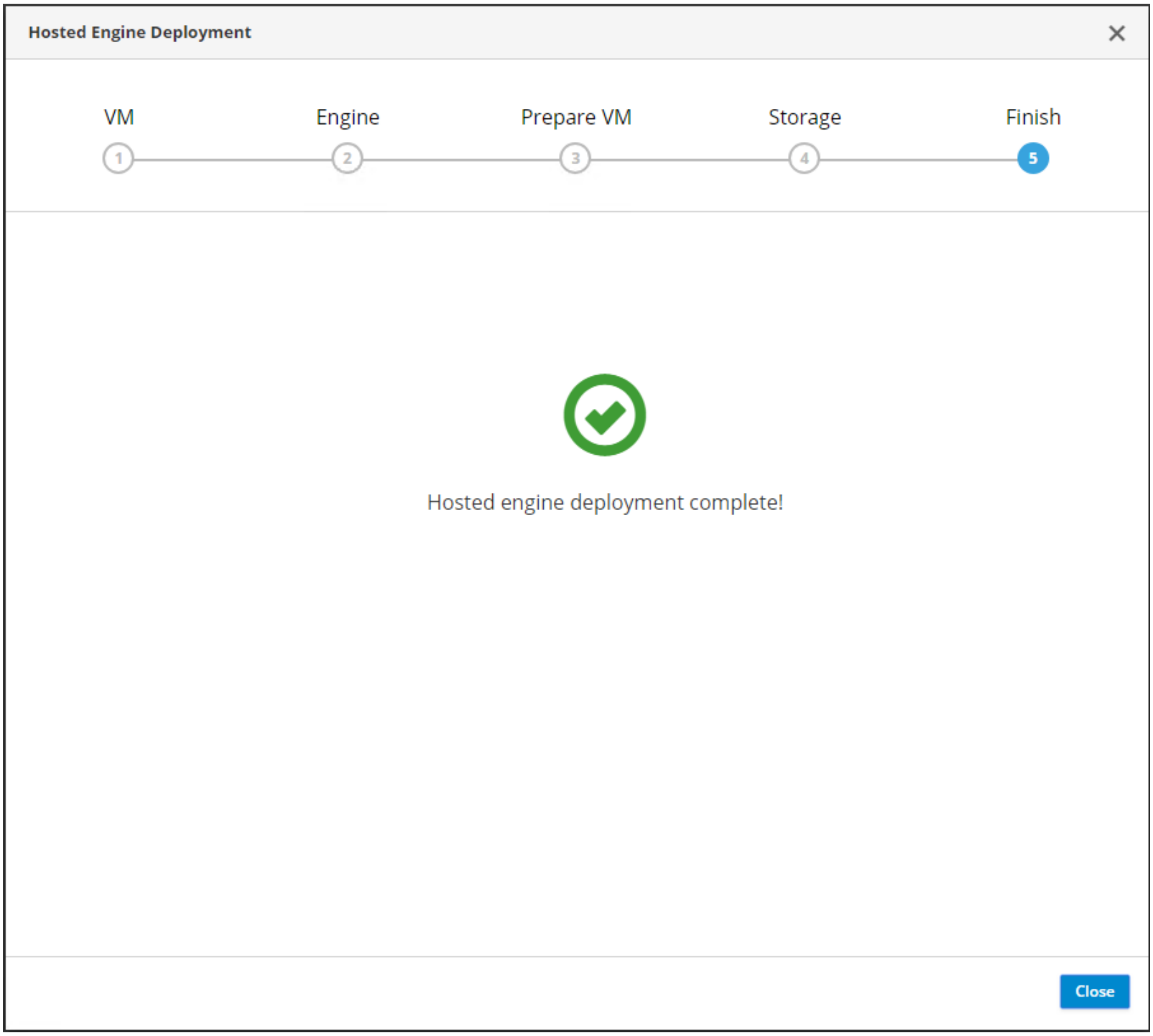

Figure 26: Hosted Engine Deployment Success Notification

Configuring Hosted Engine

The Hosted Engine configuration process must be completed manually using the Red Hat Virtualization Administration Portal.

1. Once the Hosted Engine deployment has successfully completed, log on to the Red Hat Virtualization Administration Portal at https://<hosted engine FQDN>. Log in to the Red Hat Virtualization Administration Portal using the credentials supplied during the Hosted Engine deployment in the previous section.

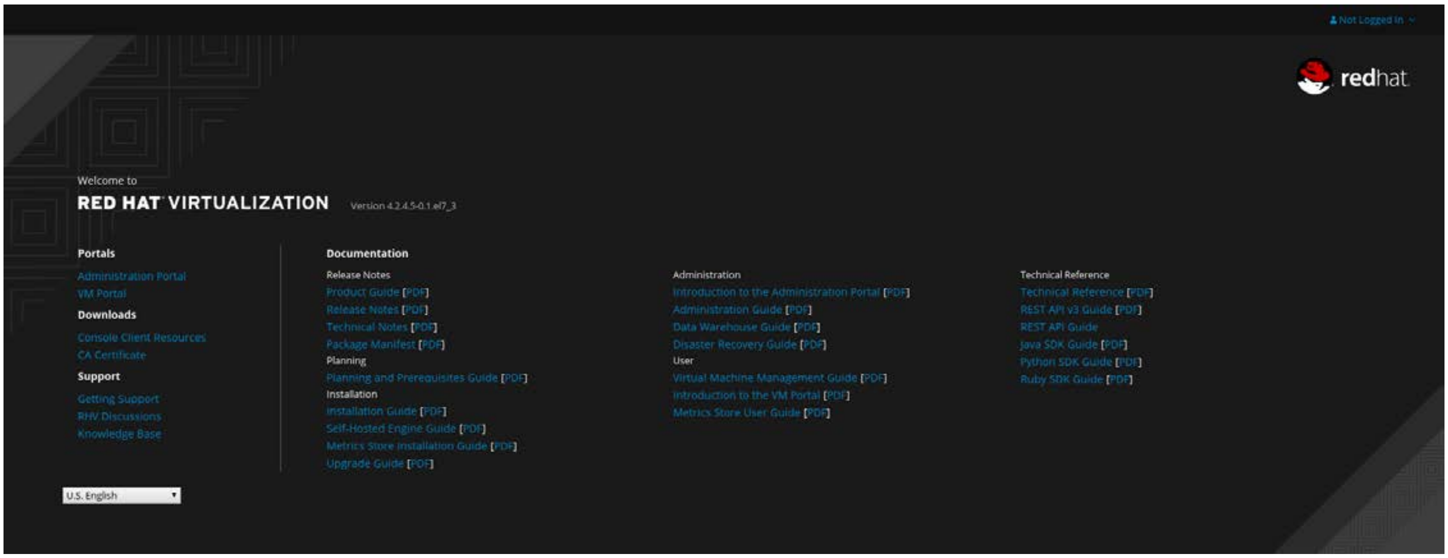

Figure 27: Red Hat Virtualization Welcome UI

2. Click on **Administration Portal** under the Portals link and log on using the Username and Password you configured during the deployment. Additional configuration is required using the Red Hat Virtualization Administration Portal.

Configure Hosted Engine networking

The hosts will require two additional networks, a Public network for application access and a network for Gluster storage. In this configuration bond1 will be used for the Public network and bond2 will be used for the Gluster network.

1. Click on the Network tab and select Networks. You will see the ovirtmgmt network has already been defined.
2. Click on New and create a Public network as in Figure 28. Customize this to your environment. In the Cluster section of the Public network deselect the Required check box. Once you have completed the form click on **OK**. Repeat the New Network process to create a new Gluster network. The Public network will be assigned to bond1 and the Gluster network will be assigned to bond2. On the Gluster network deselect the VM network check box in the General section and the Required check box in the Cluster section.

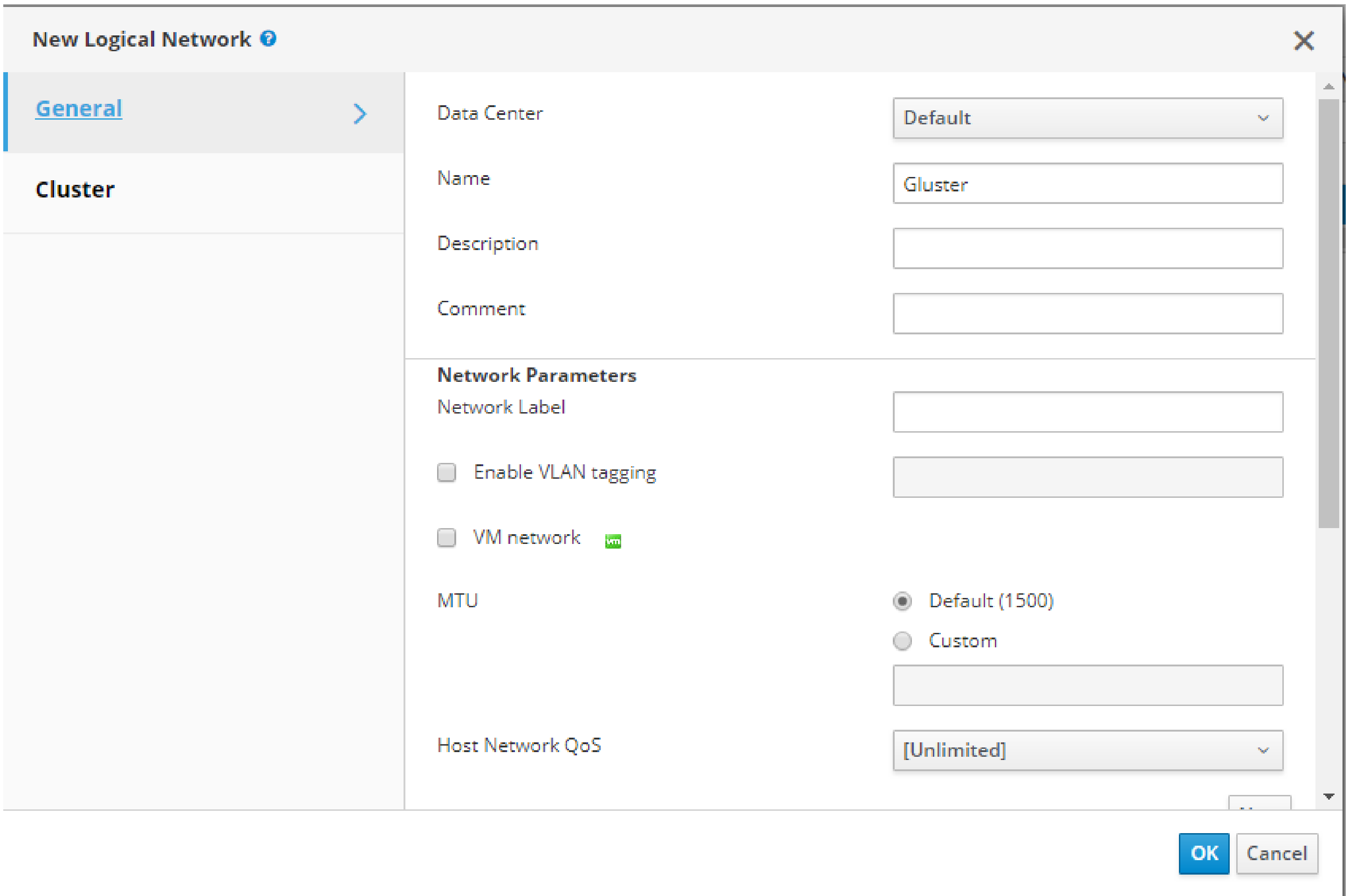

Figure 28: New Logical Network Form

3. Configure storage network as the Migration and Gluster Networks by selecting Network → Networks → Gluster → Clusters → Manage Network and select the Migration Network and Gluster Network check boxes.

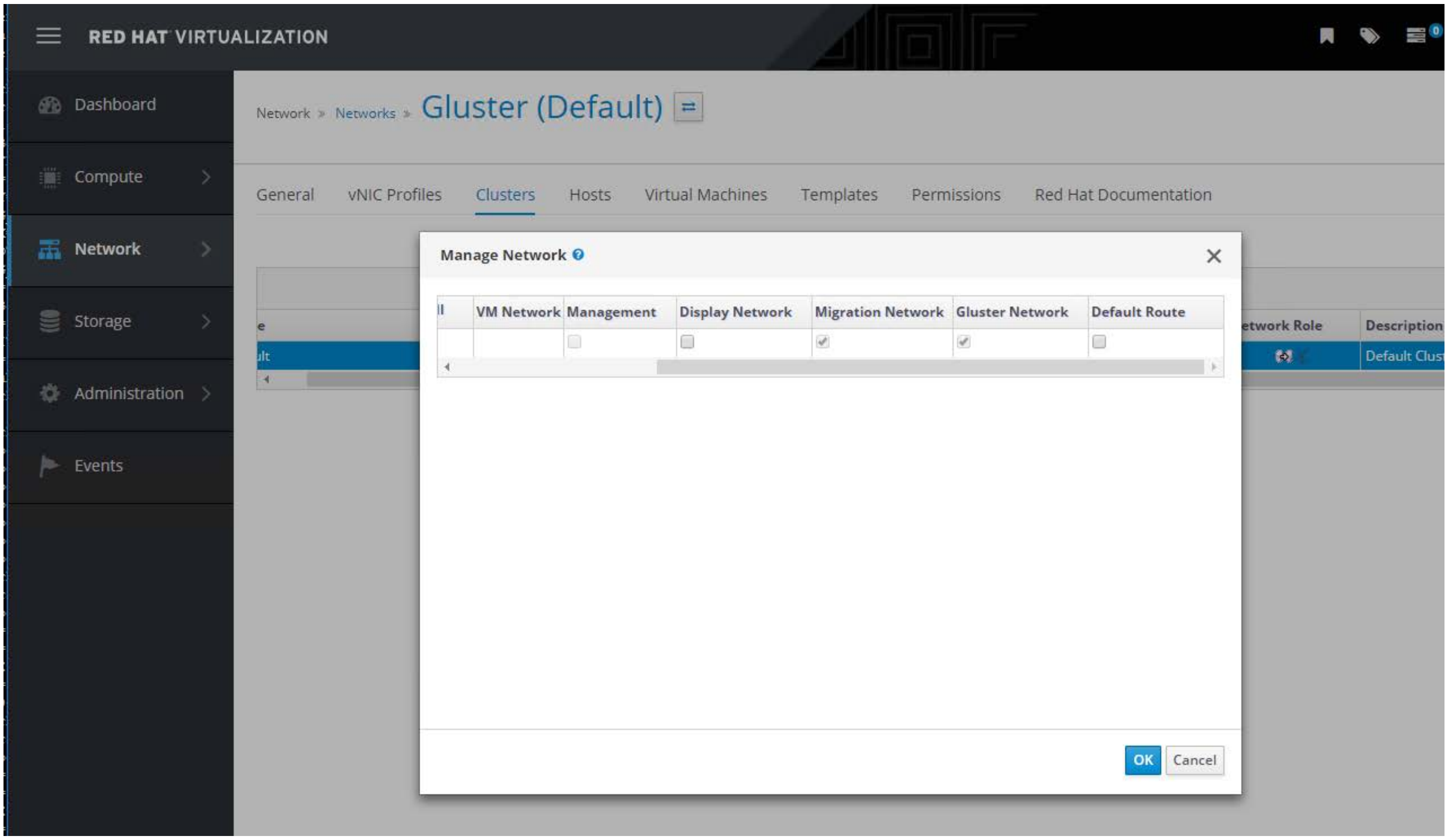

Figure 29: Gluster Storage Network Configuration

The final step requires you to click on the **Compute** tab then select **Hosts** and choose a host. For each host in the cluster do the following.

4. Click on the **Compute** tab and select **Hosts**. Click on an individual host within the tab and click **Setup Host Networks** from the host screen as in Figure 30.

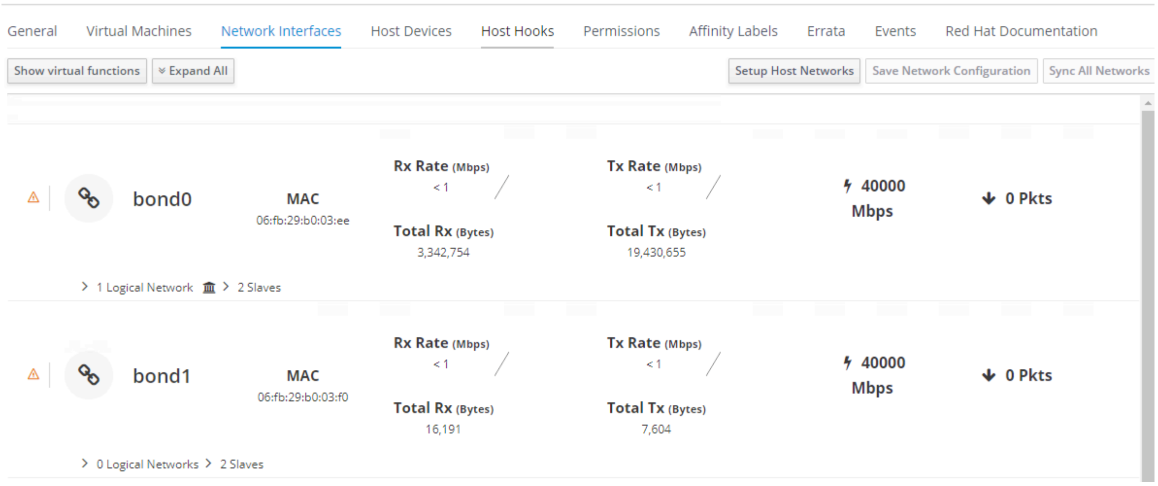

Figure 30: Setup Host Networks

5. Drag the Gluster and Public networks from **Unassigned Logical Networks** to the **Assigned Logical Networks** section of bond1 for Public and bond2 for the Gluster network as illustrated in Figure 31.

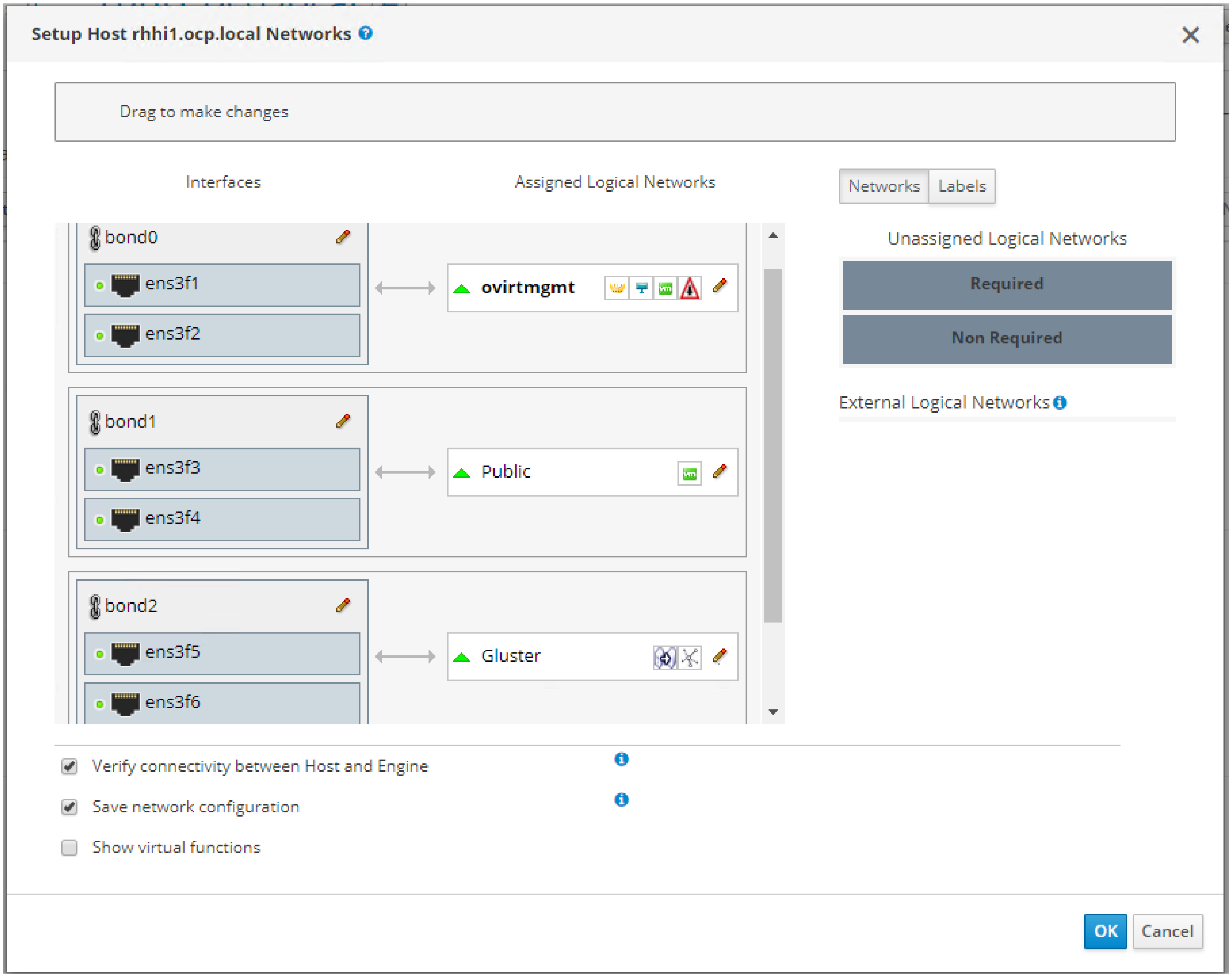

Figure 31: Host Networking Assign Public and Gluster Network to bonded interfaces

Click on **OK** to commit the changes.

Repeat the previous step for the other two RHHI hosts to add the Gluster and Public networks to the rhhi2 and rhhi3.

Configure RHV power management

RHV power management facilitates the powering down and powering up the RHHI servers through the Red Hat Virtualization admin portal.

1. The first step in configuring power management is to create a user account in each of the SY480 Compute Module's iLO Administration. The account privileges can be limited to Login and Virtual Power. The iLO console can be accessed through OneView → Server Hardware → Server Name → Overview → iLO Address.

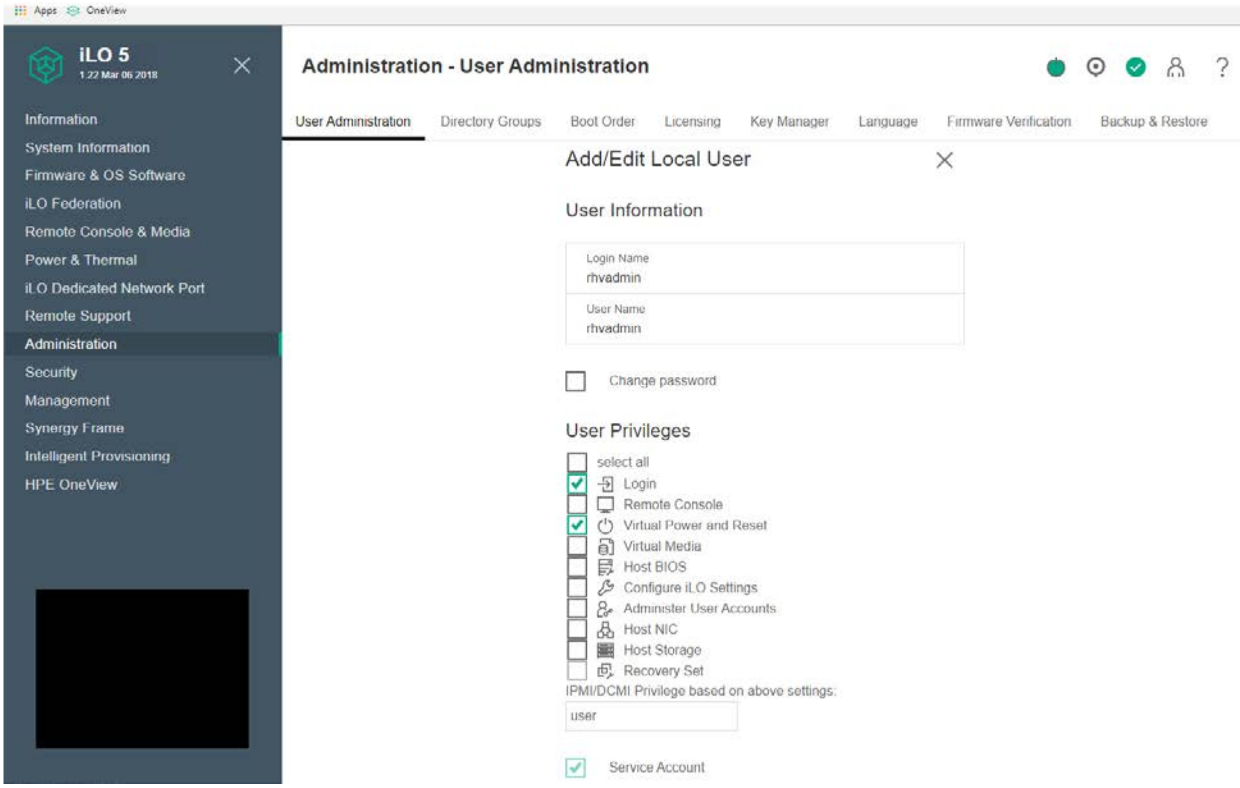

Figure 32: HPE iLO Administration – User Administration Create User for Power Management

2. Once the user account has been created, Power Management can be configured in the RHV Administration Portal. Select Compute → Hosts → Edit Host → Power Management → Edit Fence Agent. Complete the form as illustrated in the figure below.

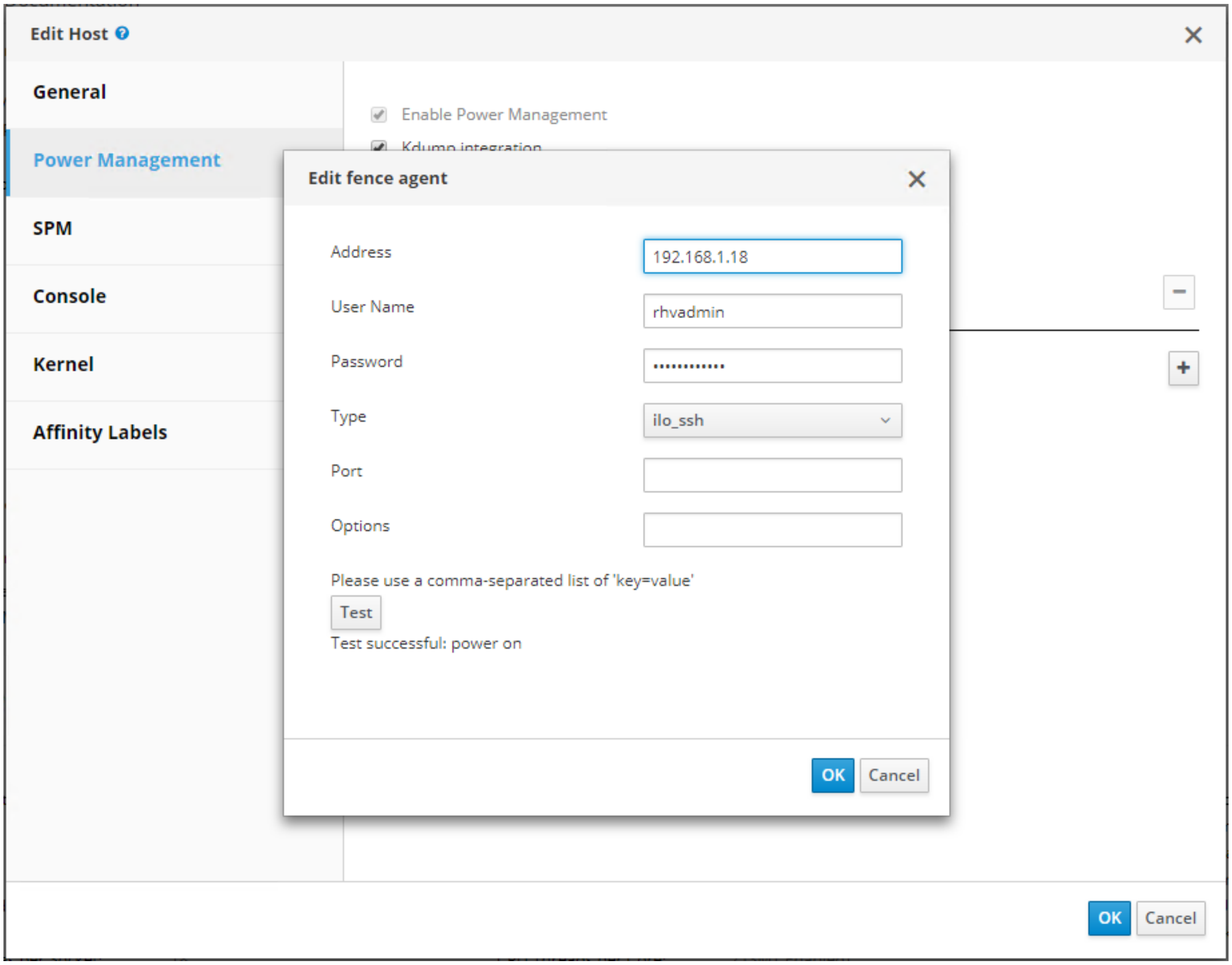

Figure 33: Red Hat Virtualization Host Power Management Configuration

3. Select Test to verify a successful power on.

OpenShift deployment

This section describes the process to automatically deploy Red Hat OpenShift 3.10.

To begin we will need to insure you have internet access and are able to reach the HPE GitHub site at https://github.com/HewlettPackard/hpe-solutions-openshift/tree/master/synergy. To validate, run the following command on the Ansible Engine host.

```
# curl https://github.com/HewlettPackard
```

From the Ansible Engine host run the following commands.

```
# cd /etc/ansible
# git clone http://github.com/HewlettPackard/hpe-solutions-openshift
```

Prerequisites

In order to utilize the scripts and procedures documented in this deployment section, the following perquisites must be met.

- Ansible Engine should be installed and configured and capable of communicating with the hosts within this solution.
- Red Hat Virtualization Host is installed on at least three HPE Synergy 480 Compute Modules.
- RHV hosts have been configured as an RHHI cluster.
- Make sure that both storage and networking are configured within Hosted Engine.
- DNS entries should exist for all hosts.

1. Create the htpasswd file:

   On the Ansible Engine host you will need to generate an htpasswd file. For this solution, HPE leveraged the tool at http://www.htaccesstools.com/htpasswd-generator/ in order to generate the hashed htpasswd. This file is saved to /etc/oshift-hash-pass.htpasswd

   An example is shown below.

   admin :$ <hashed password>

   user :$ <hashed password>

2. Generate a key:

   On the Ansible Engine host run the following command.

   # ssh-keygen -t rsa

   This will create a key file at ~/.ssh/id_rsa.pub.

3. Download the certificate from Hosted Engine:

   On the Ansible Engine host download the certificate from the Hosted Engine by running the following command.

   # curl --output ca.pem 'http://<rhvm-url>/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA'

   Provide the CA file to oVirt Ansible with the following variable:

   engine_cafile: /etc/pki/ovirt-engine/ca.pem

Virtual Machine deployment and configuration

The steps involved in setting up the virtual machines for the Red Hat OpenShift 3.10 deployment are listed below:

- Create the virtual machine template.
- Deploy the virtual machines from the virtual machine template.
- Install perquisites for the OpenShift installation on the virtual machines.
- Configure host device pass through for OpenShift Container Storage.

Create the Virtual Machine Template

1. The first play is to create a VM template in Hosted Engine. This template will later be used to clone and provision the Red Hat OpenShift Infrastructure services VMs (master, infra, loadbalancer, and application) for OpenShift deployment. On the Ansible Engine host the variable yml file is located at roles/deploy-template/vars/main.yml. The variable file should look like the example provided in roles/deploy-template/vars/main.yml of the git repository and contains information on the RHVM engine url, image download location, template name and size and more.

   In the above script sample, qcow_url is the URL for the Red Hat Enterprise Linux 7.5 KVM Guest image. The image can be downloaded from https://access.redhat.com/downloads.

The line template_disk_size_2 represents the size of the second disk which will be connected as Docker storage.

ansible-playbook -e@vault_pass.yml playbooks/deployTemplate.yml

Once complete you should see a template appear in the Red Hat Virtualization Administration Portal as in Figure 34.

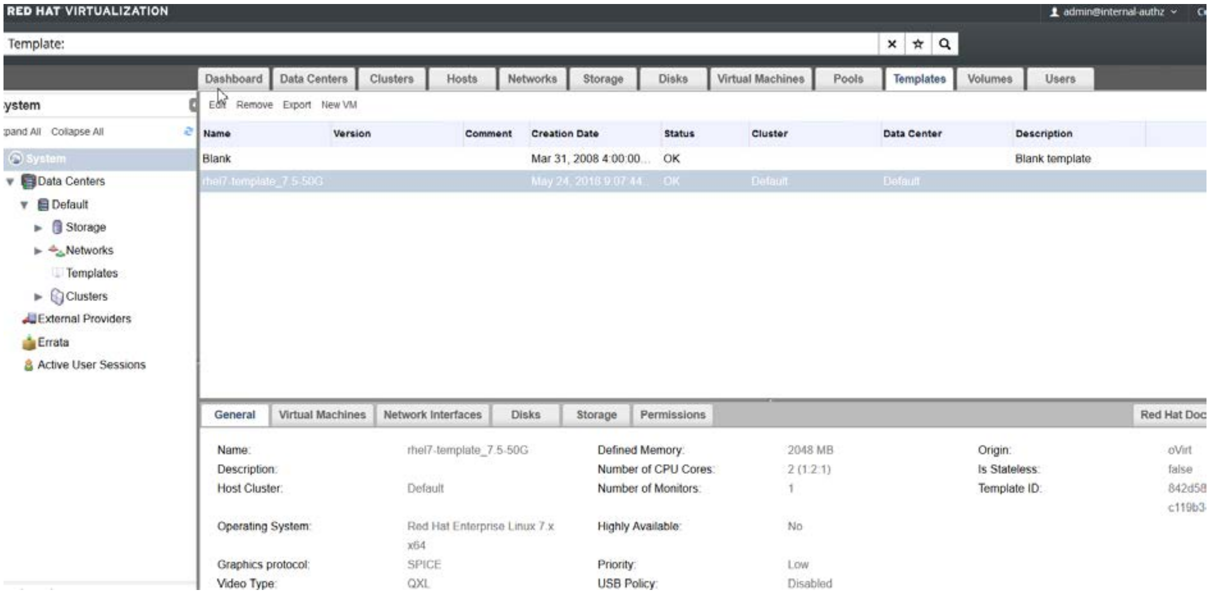

Figure 34. VM Template view from the Red Hat Virtualization Administration Portal

Deploy Virtual Machines from the template

The deployVM.yml playbook creates the following virtual machines:

- master0
- master1
- master2
- infra0
- infra1
- infra2
- app0
- app1
- app2
- lb0

1. In order to clone virtual machines from the template and create VMs for the OpenShift deployment, the installer will need to edit variables in the roles/deploy-vm/vars/main.yml file. Using an editor such as vi or nano, open the file roles/deploy-vm/vars/main.yml. The file should look like the example provided in roles/deploy-vm/vars/main.yml of the git repository for this solution and will contain information about the VMs, Hosted Engine, hostnames, IPs, memory, and CPU.
2. When the installer has completed editing the variables file, run the following command to deploy all of the VMs required for the OpenShift deployment.

   ansible-playbook -e@vault_pass.yml playbooks/deployVM.yml

Install prerequisites for OpenShift installation on the Virtual Machines

1. The next step is to edit the file roles/host-prepare/vars/main.yml in a text editor such as vi or nano. An example file is provided in roles/host-prepare/vars/main.yml in the GitHub repository.

   The first line of the file is the path to the disk where the Docker storage will be configured. The final line is the pool ID for the virtual nodes.

   The play will accomplish the following.

   - Disables the firewall for the OpenShift installation. This will be re-enabled post-install.
   - Creates a user group with passwordless sudo rights.
   - Creates a Sudo user and adds the user to the passwordless sudo group.
   - Uploads the public SSH key to allow secure access without credentials.
   - Registers the host using Subscription Manager.
   - Enables the required repos.
   - Installs the basic utilities.
   - Performs a yum update to insure the latest patches and updates are applied.
   - Installs Red Hat OpenShift related packages.
   - Installs the latest version of Docker.
   - Configures Docker local storage.
2. To run the play, execute the following command on the Ansible Engine host.

   ```
   ansible-playbook -e@vault_pass.yml playbooks/hostPrepare.yml
   ```

Configure Host Device pass through for OpenShift Container Storage

OpenShift Container Storage is deployed on the infrastructure nodes to provide persistent storage for the registry, logging, and metric services. A second instance of OpenShift Container Storage will be deployed on the application nodes to provide persistent storage for deployed applications. OpenShift Container Storage requires an available block device on each OpenShift Container Storage node. Each OpenShift Container Storage node will use a physical disk that is passed from the RHV host to the virtual machines. The virtual machine will then be "pinned" to the RHV host that is providing the physical disk. This requires six devices to be passed to the virtual machines, one for each of the three infrastructure nodes and one for each of the application nodes. The first step is to identify the disks to be used for OpenShift Container Storage. In this example the operating system is installed on /dev/sdb and Gluster is installed on /dev/sdc. The device that will be used for the OCS persistent storage for the infrastructure nodes is /dev/sdd and the device that will be used for OCS persistent storage for applications is /dev/sde. The output of sg_map -x is shown below. The host devices in this solution are 1101 and 1102 for /dev/sdd and /dev/sdd respectively.

1. Use sg_map -x to identify the physical disk that will be passed through to the virtual machine on each hypervisor. Go to the command line of the RHV host and execute sg_map -x. On each RHV host identify and take note of the host device (disk) that will be passed through to infrastructure virtual machine and the application virtual machine.

   ```
   sg_map -x
   /dev/sg7  0 0 0 0  0  /dev/sda
   /dev/sg10  1 1 0 3  0  /dev/sdb
   /dev/sg11  1 1 0 0  0  /dev/sdc
   /dev/sg12  1 1 0 1  0  /dev/sdd
   /dev/sg13  1 1 0 2  0  /dev/sde
   ```

Table 12, shown below, illustrates the Host Device mapping and SCSI IDs for the example solution used in this document. Verify the devices in your environment, using sg_map -x, as explained previously in this section.

Table 12. Host Device Mapping

Virtual Machine	Pinned Host	Device
Infra0	rhhi1	1101 /dev/sdd
Infra1	rhhi2	1101 /dev/sdd
Infra2	rhhi3	1101 /dev/sdd
app0	rhhi1	1102 /dev/sde
app1	rhhi2	1102 /dev/sde
app2	rhhi3	1102 /dev/sde

2. The next step is to shut down the infrastructure nodes infra0, infra1, and infra2 and the application nodes app0, app1, and app2. The virtual machines must be powered down to attach the host device. Using the Red Hat Virtualization Administration Portal, select Compute → Virtual Machines. Shut down all the infrastructure (infra0, infra1, infra2) and application virtual machines (app0, app1, app2).
3. Go to each virtual machine and select Host Devices → Add Device.
4. Choose the Pinned Host for each virtual machine as described in Table 12 and set the Capability scsi.
5. Select the device from the Available Host Devices, use the dropdown arrow to add the device to the "Host Devices to be attached" section.
6. Select OK and repeat for each of the Infrastructure and Application virtual machines.

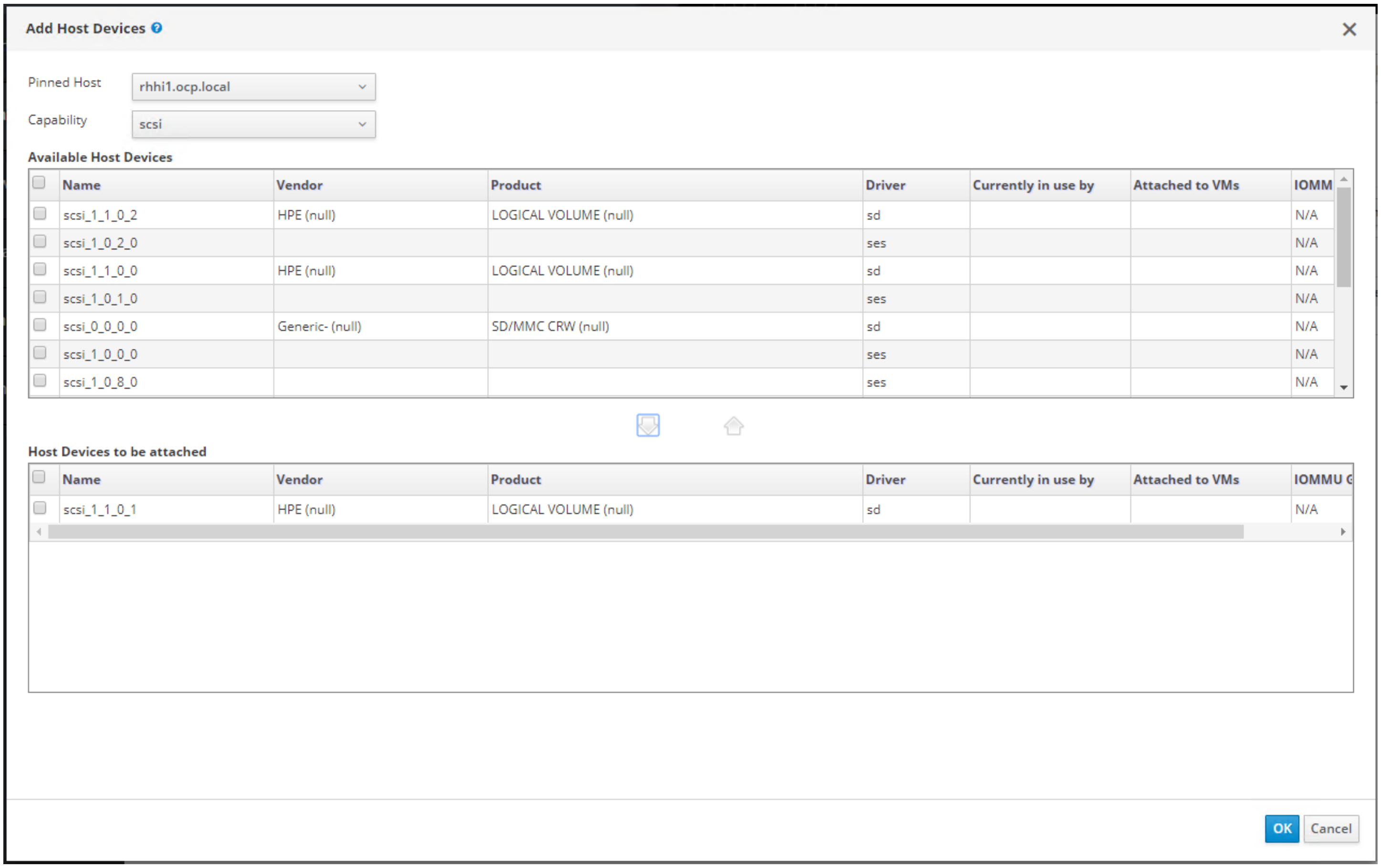

Figure 35: Add Host Device for OCS disks

7. Once the Host Device mapping is complete for all the Infrastructure and Application virtual machines, return to Compute → Virtual Machines and power on all the Infrastructure (infra0, infra1, infra2) and Application virtual machines (app0, app1, app2). These virtual machines now have a physical device available for the installation of OpenShift Container Storage, which will be installed automatically during the OpenShift deployment. There will be two OpenShift Container Storage clusters, one running on the Infrastructure virtual machines and one running on the Application virtual machines.

OpenShift-Ansible

The following Ansible playbooks will deploy Red Hat OpenShift Container Platform on the virtual machines that have been created by the previous Ansible playbooks. The variables for the OpenShift deployment are maintained in the Ansible inventory file, in this example that file is /etc/ansible/hosts. Review the sample hosts file provided in the GitHub repository for this solution located at https://github.com/HewlettPackard/hpe-solutions-openshift/tree/master/synergy

Install OpenShift

From the Ansible host, run the prerequisites.yml and deploycluster.yml playbooks that are located in /usr/ansible/openshift-ansible/playbooks/ on the Ansible host.

1. Run /usr/share/ansible/openshift-ansible/playbooks/prerequsites.yml playbook

   ansible-playbook -i /etc/ansible/hosts -e@vault_pass.yml /usr/share/ansible/openshift-ansible/playbooks/prerequisites.yml

2. Run /usr/share/ansible/openshift-ansible/playbooks/deploy_cluster.yml playbook

   ansible-playbook -i /etc/ansible/hosts -e@vault_pass.yml /usr/share/ansible/openshift-ansible/playbooks/deploy_cluster.yml

3. When the deployment has finished the installer may access the OpenShift webpage, shown in Figure 36 below, using the credentials provided in the htpasswd file. The URL will be https://openshift-master:8443.

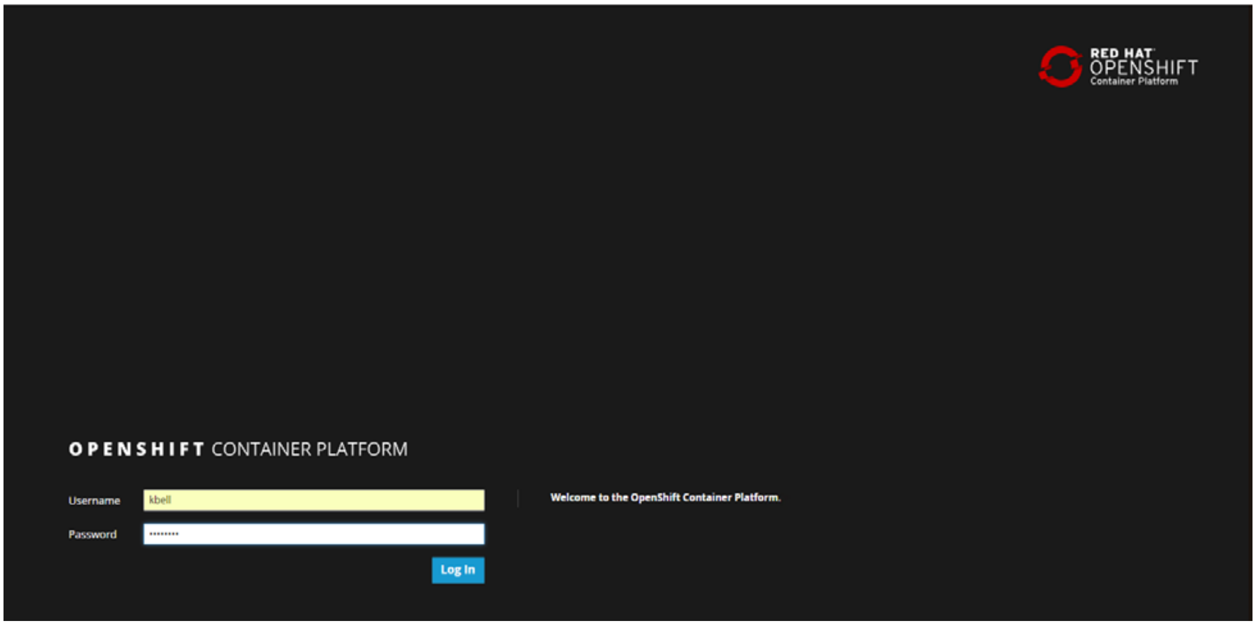

Figure 36: OpenShift User Interface

Note
The htpasswd for use with the OpenShift login is located at /etc/oshift-hash-pass.htpasswd

Validate OpenShift deployment

The final step in the process is to validate that the deployment succeeded. To accomplish this, we will demonstrate how to check the OpenShift nodes' status and log in using the default system account. Additionally, a sample application will be deployed.

Command Line validation

1. Log into the console or ssh into master0 virtual machine and run the **oc get nodes** command to ensure all nodes have a status of **Ready**.

```
oc get nodes
NAME               STATUS   ROLES     AGE   VERSION
app0.ocp.local     Ready    compute   18h   v1.10.0+b81c8f8
app1.ocp.local     Ready    compute   18h   v1.10.0+b81c8f8
app2.ocp.local     Ready    compute   18h   v1.10.0+b81c8f8
infra0.ocp.local   Ready    infra     18h   v1.10.0+b81c8f8
infra1.ocp.local   Ready    infra     18h   v1.10.0+b81c8f8
infra2.ocp.local   Ready    infra     18h   v1.10.0+b81c8f8
master0.ocp.local  Ready    master    19h   v1.10.0+b81c8f8
master1.ocp.local  Ready    master    19h   v1.10.0+b81c8f8
master2.ocp.local  Ready    master    19h   v1.10.0+b81c8f8
```

2. Run the **oc get pod** command to view the running pods. This command will display the running pods in the default project.

```
oc get pod

NAME                       READY   STATUS    RESTARTS   AGE
docker-registry-1-2z8q5    1/1     Running   0          22h
registry-console-1-mqdcl   1/1     Running   0          22h
router-1-7zx4m             1/1     Running   0          22h
router-1-gd6jw             1/1     Running   0          22h
router-1-gmkg2             1/1     Running   0          22h
```

Administrator Login and Cluster Admin Role

1. From the master0 node, log in as the default system admin account as shown below.

```
oc login -u system:admin
```

2. Once logged in, the system will display the projects that you have access to.

```
You have access to the following projects and can switch between them with 'oc project <projectname>':

    app-storage
  * default
    kube-public
    kube-system
    management-infra
    openshift
    openshift-infra
    openshift-logging
    openshift-node
    openshift-sdn
    openshift-web-console
    test2
Using project "default".
```

Grant Cluster Role to User

1. While logged in as the system admin, you can assign the cluster admin role to a user as shown below.

```
oc adm policy add-cluster-role-to-user cluster-admin <username>
```

Assigning the cluster-admin role is not required to deploy applications described in the next section of this document.

Sample application deployment

In this section a sample application will be deployed using persistent storage presented by OpenShift Container Storage running on the application nodes. The sample application selected for this example is Jenkins with persistent storage. The figure below illustrates the default catalog that is installed with OpenShift. Under My Projects notice the Development Project. When you deploy your first application in OpenShift, you will be prompted to create a project. Additional applications created by this user account will be created under that project.

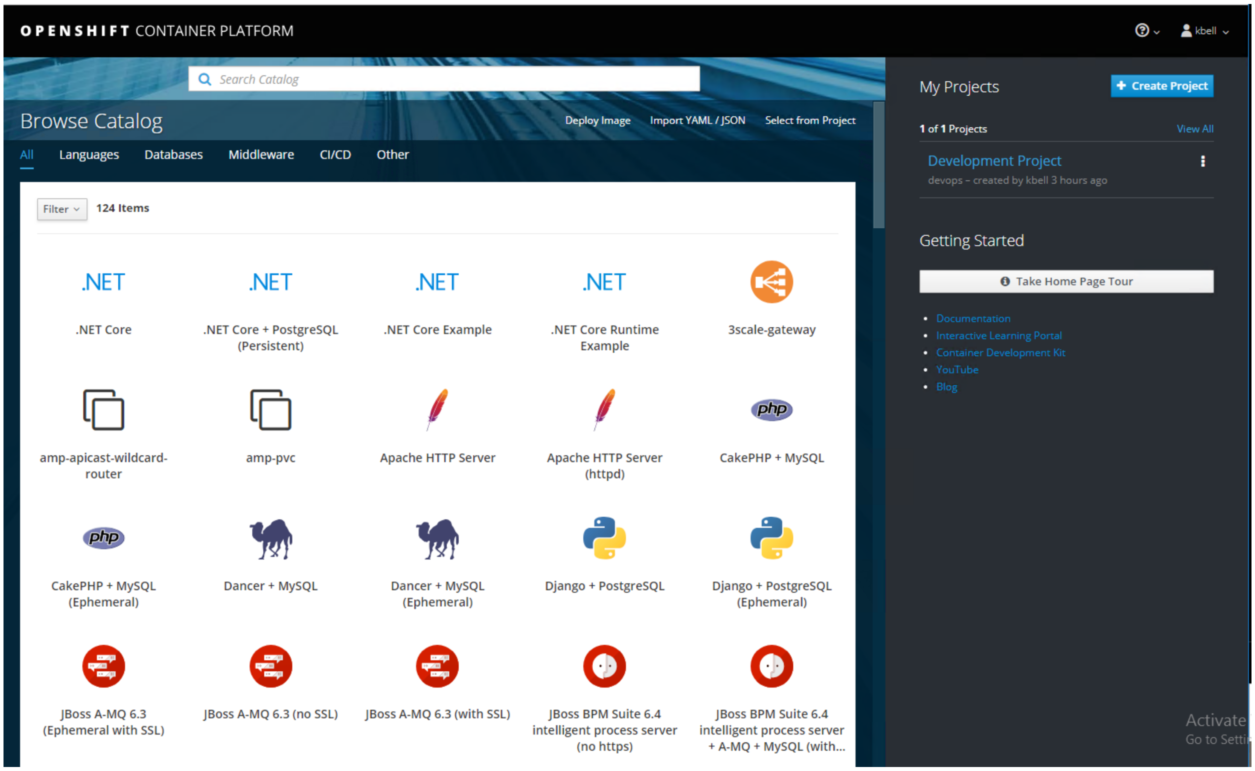

Figure 37: OpenShift Catalog

1. Before deploying the Jenkins with persistent storage application, we must create a persistent volume claim. Click on the project you created and select Storage. In this example it is Development Project → Storage.
2. In the Storage screen, select Create Storage.

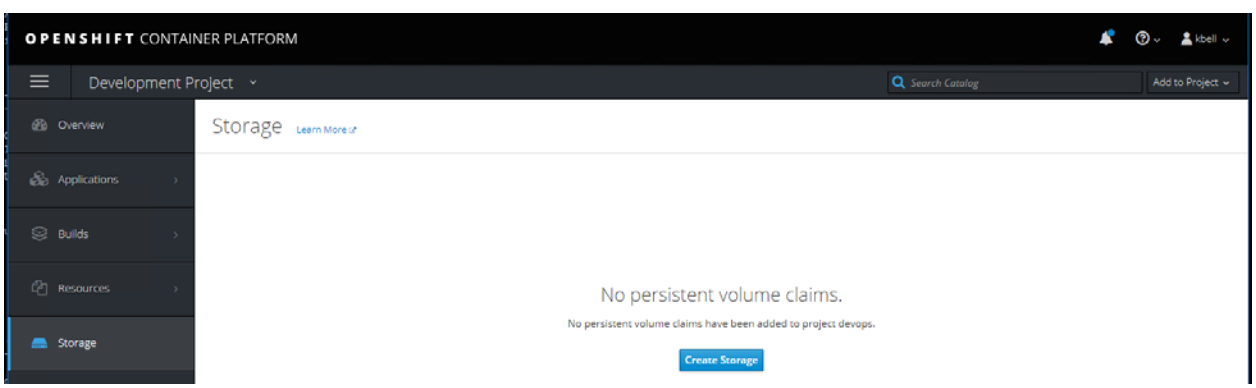

Figure 38: OpenShift Persistent Volumes

3. In the Create Storage screen select the Storage Class glusterfs-storage. This is the storage class for applications and is being provided by the OpenShift Container Storage that is running on the three application nodes (app0, app1, app2).

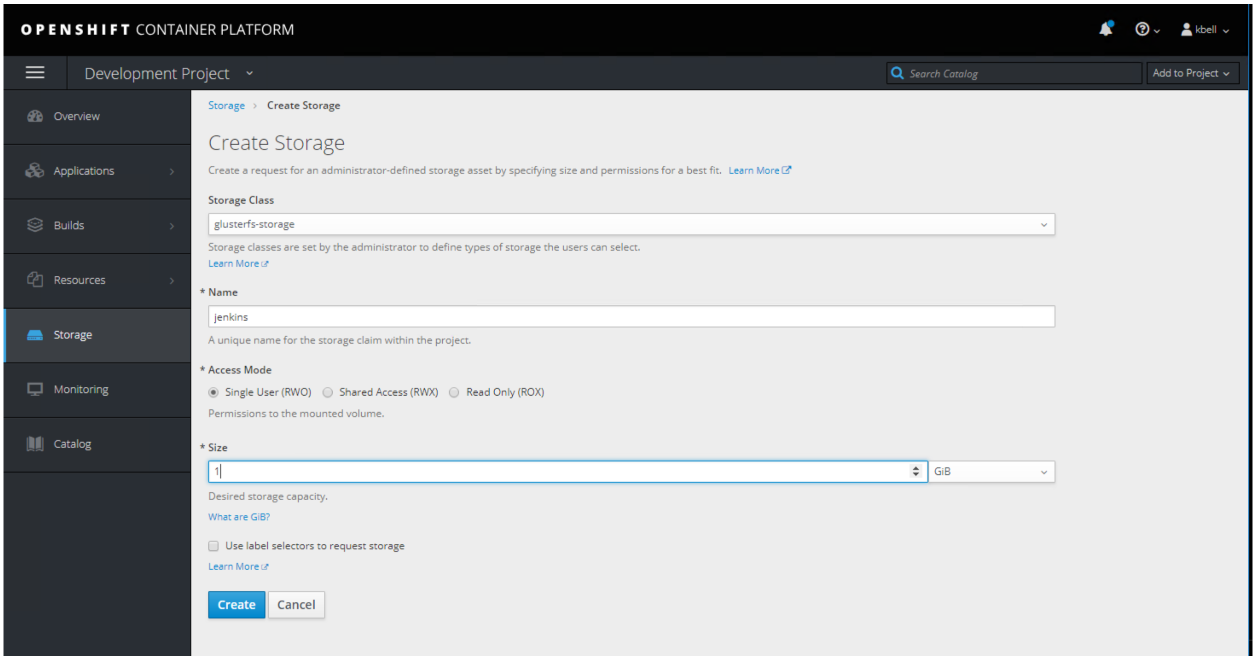

Figure 39: OpenShift Create Persistent Volume for Jenkins

4. In the Name enter jenkins and provide a size for the storage claim. Once the storage claim is complete, the status will show that it is bound to a volume.

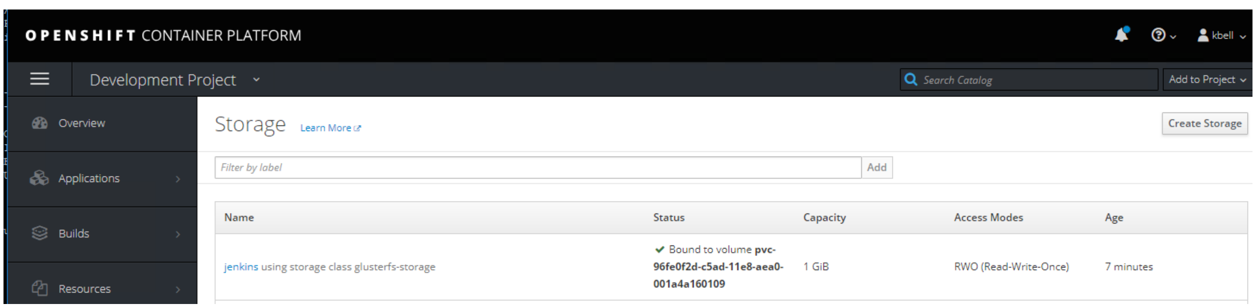

Figure 40: OpenShift Persistent Volume Claim

5. Go the Catalog and select the Jenkins application icon.

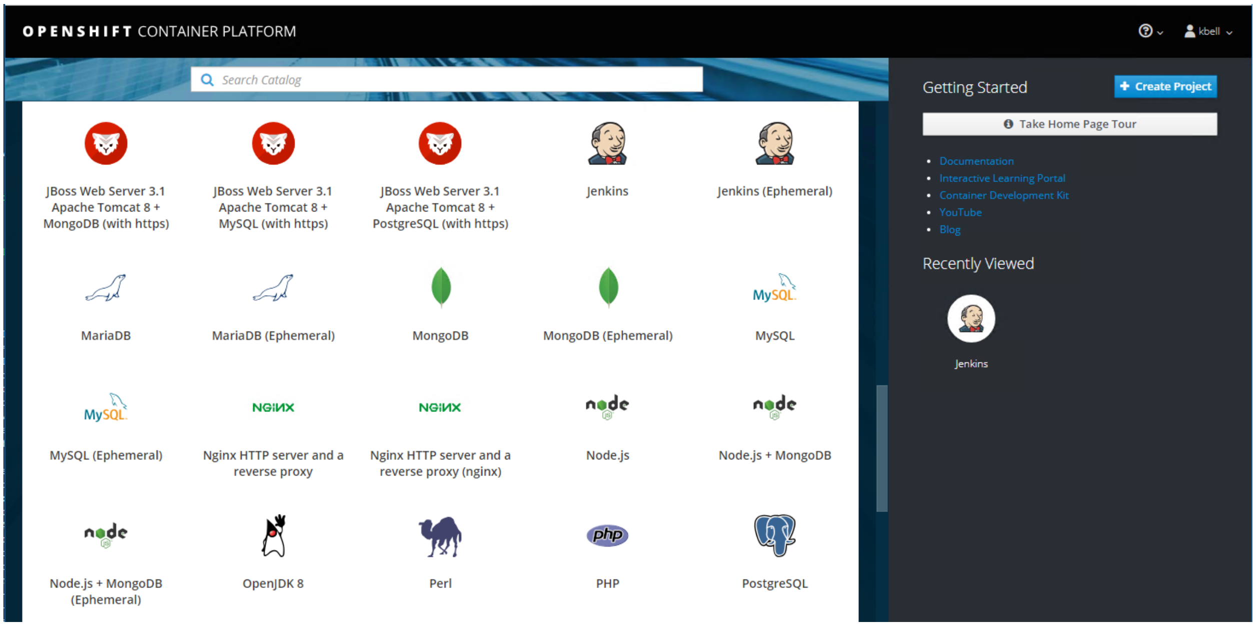

Figure 41: OpenShift Catalog Jenkins

6. Launch the Jenkins application, notice the Jenkins service with persistent storage. Jenkins can also be deployed as a test case using the Jenkins (ephemeral) icon.

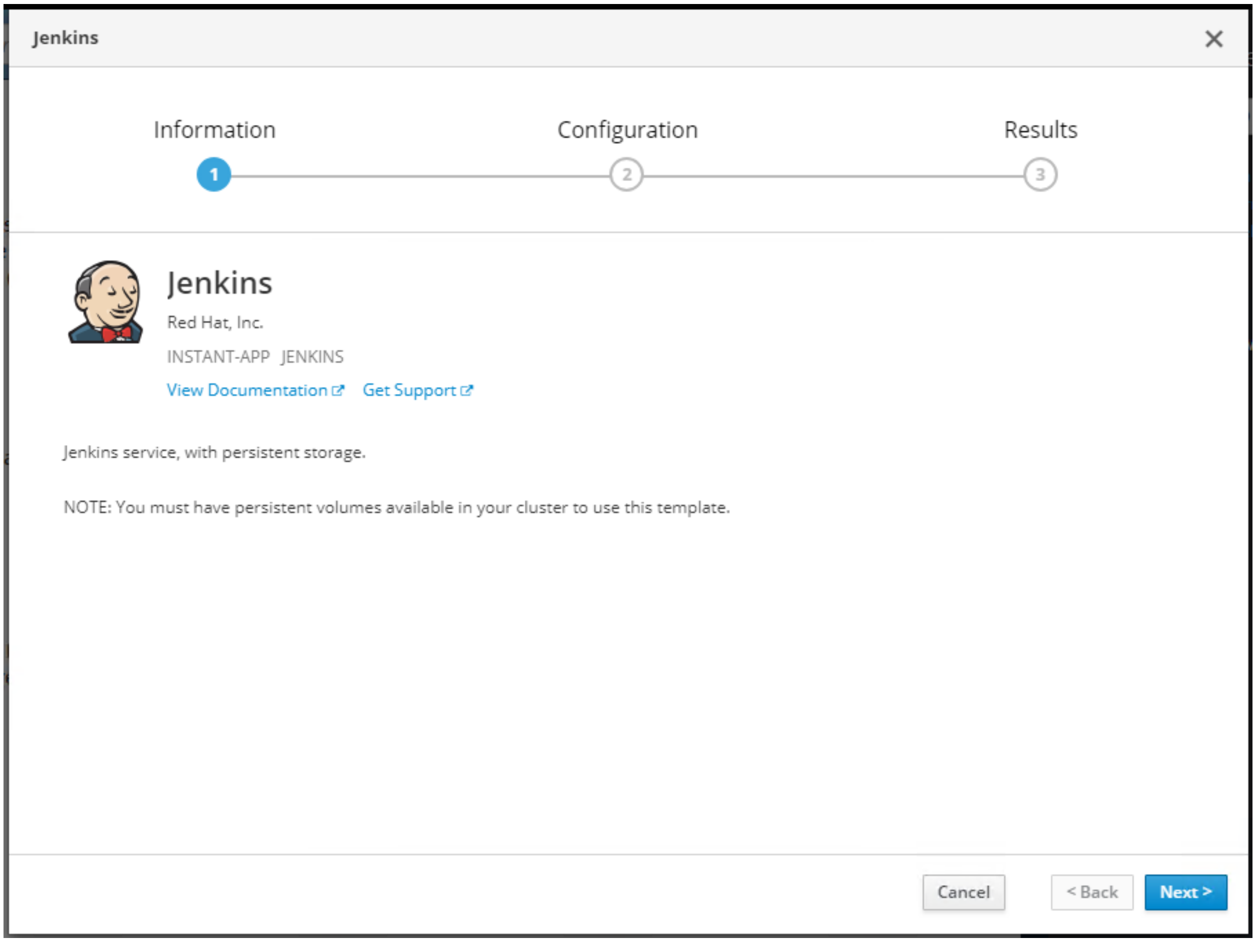

Figure 42: OpenShift Jenkins Deployment

7. Complete the Jenkins form, in this example the defaults were selected as shown in the figure below.

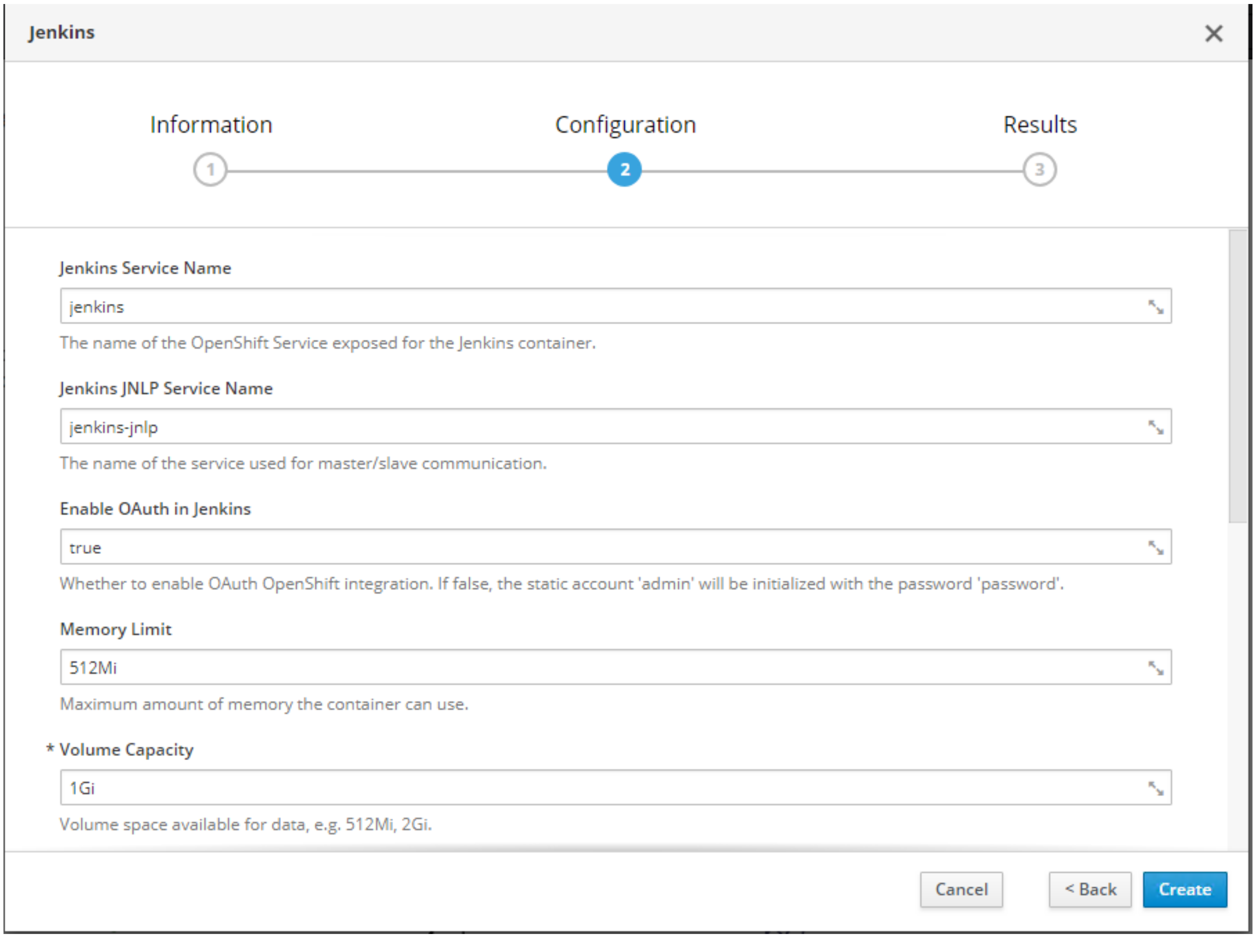

Figure 43: OpenShift Deployment Jenkins

8. Select Create to deploy the application.

   Details about the application can be viewed in the Project. Under Applications you can view the Deployments, Pods, Services, and Routes.

9. Select Application to view the status of the application deployment.

   The Deployment will show the applications that have been deployed. In this case there is only one application, Jenkins.

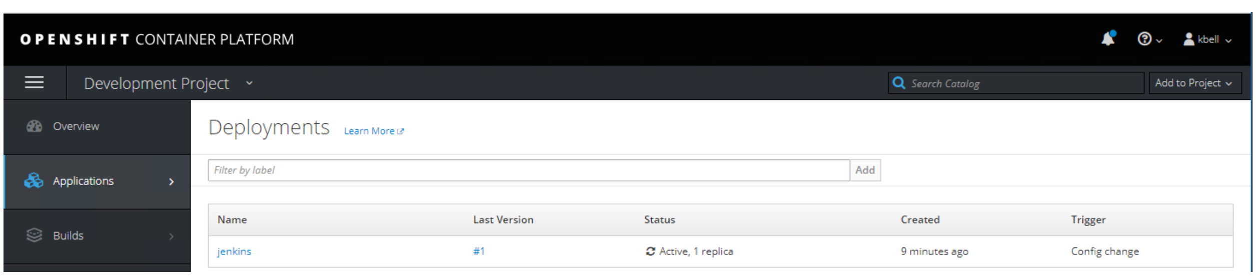

Figure 44: OpenShift Applications Deployments

10. Select Pods to view the pods that are running in the Development Project. Clicking on the Jenkins pod will provide detailed information about the pod.

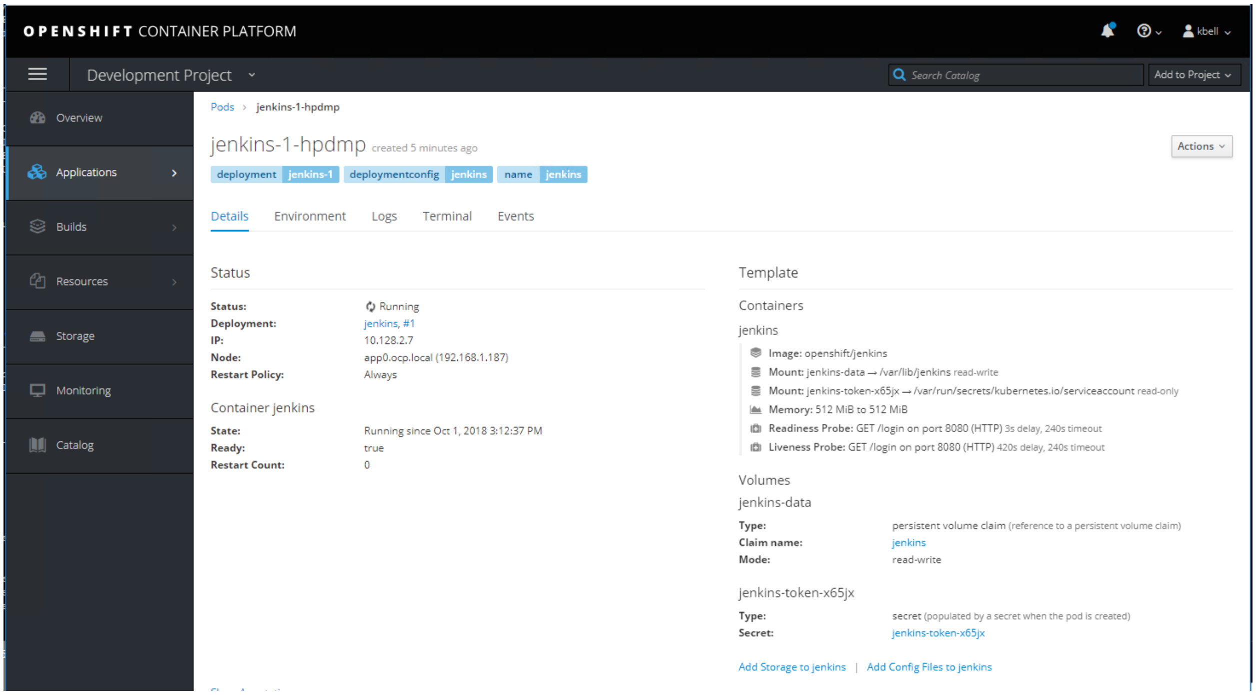

Figure 45: OpenShift Applications Pods – Jenkins

11. Selecting Applications → Services will show the services that are associated with the Pods.

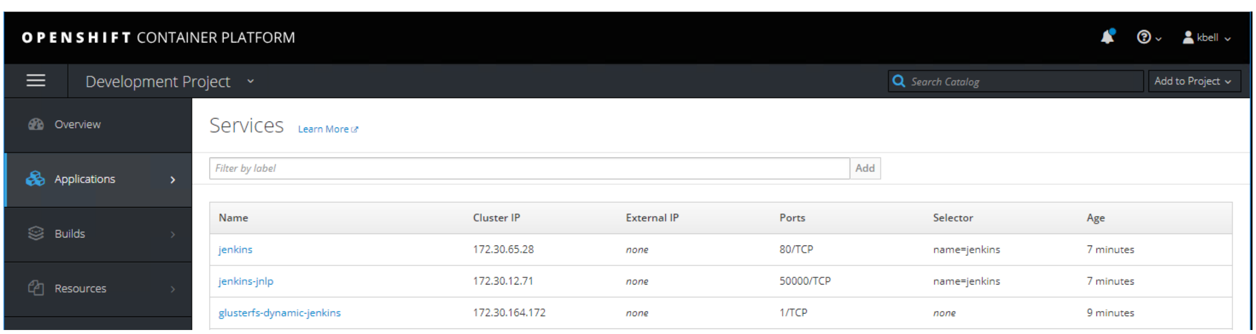

Figure 46: OpenShift Applications Services

12. Select Application → Routes to view the route to the application. In this sample deployment, the route is a DNS wildcard subdomain called app.ocp.local. The url to the Jenkins application is https://jenkins-devops.apps.ocp.local

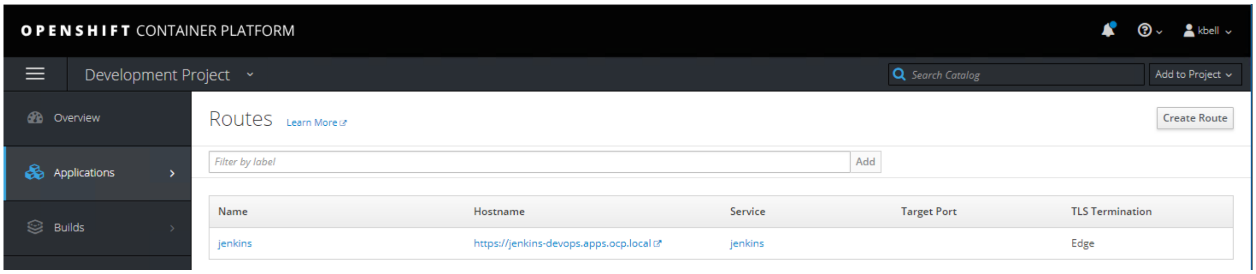

Figure 47: OpenShift Applications Routes

13. Click on the url to access Jenkins.

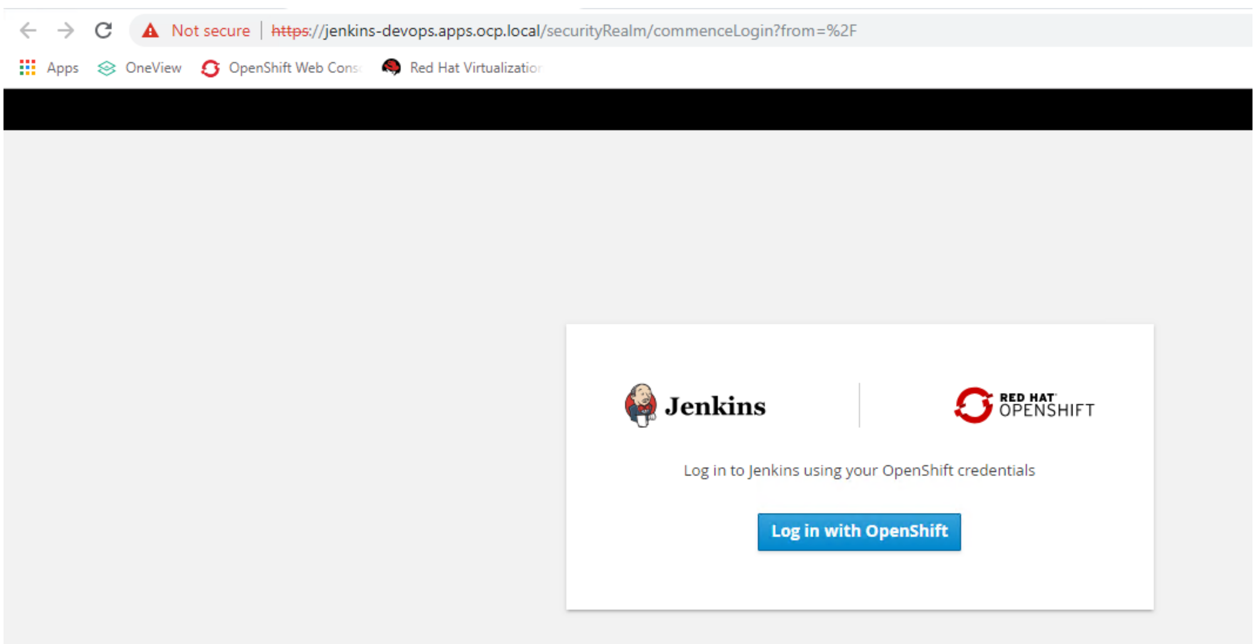

Figure 48: Jenkins Login

14. The Jenkins application will prompt you to log in with your OpenShift credentials. These are the same credential used to log into the OpenShift UI.

15. Once logged into Jenkins you will be presented with the Jenkins user interface.

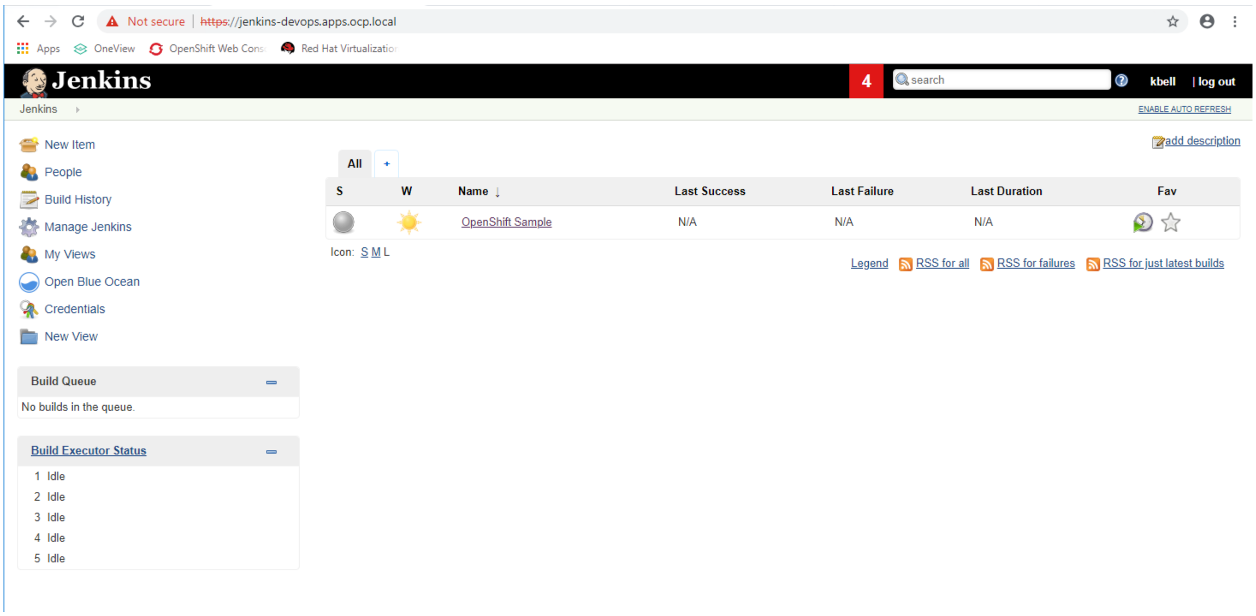

Figure 49: Jenkins User Interface

Refer to the OpenShift Developers Guide for more information on creating OpenShift CI / CD pipelines with Jenkins.

https://docs.openshift.com/container-platform/3.10/dev_guide/dev_tutorials/openshift_pipeline.html

https://access.redhat.com/documentation/en-us/reference_architectures/2017/html-single/application_cicd_on_openshift_container_platform_with_jenkins/index

Appendix A: BOM

Table 13. Bill of Materials

Rack components	Qty	Part Number
HPE 42U 600x1200mm Enterprise Shock Rack	1	BW908A
HPE Intelligent 8.6kVA/L15-30P/NA/J PDU	4	AF522A
HPE Integration Center Routg Service FIO	1	HC790A
HPE 600mm Rack Stabilizer Kit	1	BW932A
HPE 42U 1200mm Side Panel Kit	1	BW909A

HPE FlexFabric 5940 Switch Components	Qty	Part Number
HPE FF 5940 2-Slot Switch	2	JH397A
HPE 5930 24p SFP+ and 2p QSFP+ Module	2	JH180A
HPE 5930 8-port QSFP+ Module	2	JH183A
HPE X712 Back (Power Side) to Front (Port Side) Airflow High Volume Fan Tray	4	JG553A
HPE 58x0AF 650W AC Power Supply	4	JC680A
INCLUDED: Jumper Cable - NA/JP/TW	4	JC680A B2B
HPE X240 40G QSFP+ QSFP+ 1m DAC Cable	2	JG326A
HPE X240 40G QSFP+ QSFP+ 3m DAC Cable	4	JG327A

HPE Synergy 12000 Frame components	Qty	Part Number
HPE Synergy 12000 CTO Frame 1xFLM 10x Fan	1	797740-B21
HPE Synergy 12000F 6x 2650W AC Ti FIO PS	1	798096-B21
HPE Synergy Composer	1	804353-B21
HPE Synergy Frame Link Module	1	804942-B21
HPE Synergy 12000 Frame Rack Rail Option	1	804938-B21
HPE Synergy 12000 Frame 4x Lift Handle	1	804943-B21
HPE 2.0m 250V 16A C19-C20 Sgl IPD Jpr Crd	6	TK738A

HPE Synergy 480 Gen10 Compute Module components	Qty	Part Number
HPE SY 480 Gen10 CTO Premium Cmpt Mdl	3	871942-B21
HPE Synergy 480 Gen10 6142 Kit	3	872138-B21
HPE Synergy 480 Gen10 6142 Kit	3	872138-L21
HPE 32GB (1 x 32GB) dual rank x4 DDR4-2666 CAS-19-19-19 registered memory kit	24	815100-B21
HPE Smart Array P416ie-m SR Gen10 (8 Int 8 Ext Lanes/2GB Cache) 12G SAS Mezzanine Controller	3	804428-B21
HPE Smart Array P416ie-m SAS Cable Kit	3	871573-B21
HPE Synergy 3820C 10/20Gb CNA	3	777430-B21
HPE 96W Smart Stor Battery 260mm Cbl Kit	3	875242-B21
HPE 300GB SAS 10K SFF	6	872475-B21

HPE Synergy Fabric components	Qty	Part Number
HPE VC SE 40Gb F8 Module	2	794502-B23
HPE Synergy 12Gb SAS Connection Module	2	755985-B21

HPE Synergy Composable Storage components	Qty	Part Number
HPE Synergy D3940 CTO Storage Module	3	835386-B21
HPE Synergy D3940 IO Adapter	3	757323-B21
HPE 300GB SAS 10K SFF	24	872475-B21
HPE 400GB SSD Mixed Use	6	872374-B21

Cables and Transceivers	Qty	Part Number
HPE BLc 40G QSFP+ QSFP+ 3m DAC Cable	2	720199-B21
HPE BLc QSFP+ to SFP+ Adapter	8	720193-B21
HPE BLc 10G SFP+ SR Transceiver	8	455883-B21
HPE 15m Multi-mode OM3 LC/LC FC Cable	8	AJ837A
HPE CAT6A 4ft Cbl	9	861412-B21
HPE Synergy Dual 10GBASE-T QSFP+ 30m RJ45 Transceiver	2	838327-B21

Appendix B: PXE configuration

PXE configuration installing Red Hat Virtualization

The RHVH servers in this solution are installed using an external server that provides PXE boot, HTTP, and DHCP services. This server hosts the RHVH image on a httpd web server. DHCP is configured to assign specific IP addresses to the RHV server nodes. This is accomplished by setting a reservation of the ethernet MAC address of the network interface that is configured on the PXE network in the dhcp configuration file. The MAC address can be found in the OneView Server Profile for the respective RHV server. This section provides example configuration settings for creating a network boot server. For detailed setup and configuration information, refer to the Red Hat documentation at the following links:

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/installation_guide/chap-installation-server-setup

https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.2/html/installation_guide/advanced_rhvh_install#Automating_RHVH_Deployment

PXE server environment

1. On the PXE server enable to following repositories:
   a. subscription-manager repos --enable="rhel-7-server-rpms" --enable="rhel-7-server-extras-rpms" --enable="rhel-7-fast-datapath-rpms"
2. yum install syslinux tftp-server httpd dhcp xinetd
3. Configure the tftp server
   a. cp -r /usr/share/syslinux/* /var/lib/tftpboot/
   b. systemctl start tftp
   c. systemctl enable tftp
   d. mkdir /var/lib/tftpboot/pxelinux.cfg
   e. touch /var/lib/tftpboot/pxelinux.cfg/default
   f. vi /var/lib/tftpboot/pxelinux.cfg/default

```
default menu.c32
prompt 0
timeout 300
ONTIMEOUT local
menu title ###########PXE Boot Menu ###############
LABEL rhhi
MENU LABEL Install Red Hat Virtualization^Host
KERNEL vmlinuz
APPEND initrd=initrd.img inst.ks=http://192.168.10.101/rhvinstall/ks.cfg inst.stage2=http://192.168.10.101/rhvinstall
```

4. Mount the rhvh distribution image and copy the boot images
5. mount -o loop /dev/cdrom /mnt
   a. cd /mnt/
   b. cp images/pxeboot/vmlinuz /var/lib/tftpboot/
   c. cp images/pxeboot/initrd.img /var/lib/tftpboot/
6. Configure the firewall to pass ftp and http
   a. firewall-cmd --add-service=tftp --permanent
   b. firewall-cmd --add-service=http --permanent

7. Configure httpd
   a. mkdir /var/www/html/rhvinstall
   b. cp -rf /mnt/* /var/www/html/rhvinstall
   c. cp /mnt/Packages/redhat-virtualization-host-image-update-<image version>.noarch.rpm /tmp/
   d. cd /tmp
   e. rpm2cpio redhat-virtualization-host-image-update-<image version>.noarch.rpm | cpio -idmv
   f. cp /tmp/usr/share/redhat-virtualization-host/image/redhat-virtualization-host-4.1-20180126.0.el7_4.squashfs.img /var/www/html/rhvinstall/LiveOS
   g. cd /var/www/html/rhvinstall/LiveOS
   h. mv redhat-virtualization-host-4.1-20180126.0.el7_4.squashfs.img squashfs.img
8. Edit the kickstart file
   a. vi /var/www/html/rhvinstall/ks.cfg

   %pre --log=/tmp/pre.log

   %end

   autopart --type=thinp

   zerombr

   clearpart --all --initlabel

   liveimg --url=http://192.168.10.101/sqimage/squashfs.img

   rootpw --plaintext changeme

   ignoredisk --only-use=sdb

   timezone --utc America/Chicago

   network --device=ens3f1 --bootproto=static

   network --device=ens3f2 --bootproto=static

   network --device=ens3f3 --bootproto=static

   network --device=ens3f4 --bootproto=static

   network --device=ens3f5 --bootproto=static

   network --device=ens3f6 --bootproto=static

   network --device=bond0 --bondslaves=ens3f1,ens3f2 --bondopts=mode=802.3ad --bootproto=static --gateway=192.168.1.3 --nameserver=192.168.1.101 --activate

   network --device=bond1 --bondslaves=ens3f3,ens3f4 --bondopts=mode=802.3ad --bootproto=static --nodefroute --activate

   network --device=bond2 --bondslaves=ens3f5,ens3f6 --bondopts=mode=802.3ad --bootproto=static --nodefroute --activate

   text

   reboot

   %post --erroronfail

   imgbase layout --init

   nodectl init

   %end

9. Enable and start httpd
    a. systemctl enable httpd
    b. systemctl start httpd
10. Configure DHCP
    a. vi /etc dhcpd.conf

```
/etc/dhcpd.conf
# DHCP Server Configuration file.
option domain-name "ocp.local";
option domain-name-servers 192.168.1.101;
Allow booting;
Allow bootp;
default-lease-time 600;
max-lease-time 7200;
log-facility local7;
subnet 192.168.10.0 netmask 255.255.255.0 {
  range dynamic-bootp 192.168.10.150 192.168.10.190;
  option broadcast-address 192.168.10.255;
  filename "pxelinux.0";
  option domain-name-servers 192.168.10.101;
  option domain-search "ocp.local";
}
host rhhi1 {
hardware ethernet 62:6E:A4:80:00:39;
fixed-address 192.168.10.170;
}
host rhhi2 {
hardware ethernet 62:6E:A4:80:00:28;
fixed-address 192.168.10.171;
}
host rhhi3 {
hardware ethernet 62:6E:A4:80:00:2C;
fixed-address 192.168.10.172;
}
```

    b. systemctl enable dhcpd
    c. systemctl start dhcpd

Appendix C: Configuring the Ansible Host

Configuring the Ansible Host

Register Ansible host

Subscription-manager register

Enable required repositories

Required Repositories – Ansible

- rhel-7-server-ansible-2.4-rpms
- rhel-7-server-extras-rpms
- rhel-7-server-rhv-4-mgmt-agent-rpms
- rhel-7-server-rpms
- rhel-7-server-ose-3.10-rpms

Install ansible

    yum install ansible

Install openshift-ansible

    yum install openshift-ansible

Create the Ansible vault file

Ansible vault is used to store and encrypt sensitive information that will be used as variables in playbooks and tasks throughout this solution. The following variables are stored in the Ansible vault file:

ansible_ssh_user: <username>
ansible_ssh_password: <password>
ssh_pass: <password>
rhsm_user: <rhsm_user>
rhsm_pass: <rhsm_password>
username: admin@internal
password: changeme
engine_url: https://ovirteng.ocp.local/ovirt-engine/api
vault_engine_url: https://ovirteng.ocp.local/ovirt-engine/api
vault_engine_user: admin@internal
vault_engine_password: changeme
vault_root_ssh_key: /root/.ssh/id_rsa
vault_rhsub_user: <rhsm_user>
vault_rhsub_password: <rhsm_password>
vault_rhsub_pass: <rhsm_password>
vault_rhsub_pool: <rhsm_subscription_poolID
vault_rhsub_server:
vault_root_pw: changeme
ansible_user: <user>
ansible_ssh_pass: <password>

A sample Ansible vault file called vault_pass.yaml has been provided. This file is encrypted and the default password to edit the vault_pass.yaml file is changeme. In this solution an Ansible vault file is stored as /etc/ansible/vault_pass.yml.

#ansible-vault edit /etc/ansible/vault_pass.yml

You will need to edit the file with your own values. Use ansible-vault rekey to change the default password for vault_pass.yml

Ansible configuration file
Copy the Ansible configuration file from /usr/share/ansible/openshift-ansible/ansible.cfg to /etc/ansible/ansible.cfg.

Ansible Inventory (hosts) file
A sample inventory file is provided in the git repository for this solution. Edit this file to reflect settings specific to your environment.

Appendix D: Troubleshooting

Playbooks

Increase verbosity – when running playbooks, verbosity can be increased by adding the -v switch to the command line. You can increase verbosity by adding multiple "v" to the switch -vvvv

Idempotent – Ansible is idempotent, playbooks can be run multiple times and not change the state of a previously successful run. If a playbook fails due to latency of some other interruption the installer can rerun the playbook.

Latency – high latency may cause a failure during the OpenShift installation image availability check process. There are several options available to solve this problem. The installer can:

- Rerun the playbook
- Disable the image checks in the inventory file "openshift_disable_check=docker_image_availability"
- Preinstall the images before installing OpenShift with the playbook provided named image_pull.yaml

Routing

Ensure DNS resolution is correct for the application subdomain, *.apps.<domainname> points to the OpenShift routers.

Re-Install

When attempting a reinstall, it may be necessary to clear the physical devices that Gluster is installed on. This can be done with sgdisk -Z /dev/<device> or wipefs /dev/<device>.

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

Red Hat OpenShift Container Platform, openshift.com

Red Hat OpenShift Container Storage, redhat.com/en/technologies/cloud-computing/openshift-container-storage

Red Hat Virtualization, redhat.com/en/technologies/virtualization/enterprise-virtualization

Red Hat Hyperconverged Infrastructure for Virtualization,
redhat.com/en/resources/hyperconverged-infrastructure-for-virtualization-datasheet

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

November 2018