



HPE Deployment Guide for Red Hat OpenShift Container Platform on HPE Synergy with HPE Nimble Storage

Implementing a resilient on-premises container solution

Contents

Overview	4
Solution design	5
Solution creation process	8
Sizing considerations	8
Red Hat OpenShift Container Platform sizing	8
Red Hat OpenShift Container Platform role sizing	8
Red Hat OpenShift Container Platform cluster sizing	9
Prerequisites	9
Software versions	9
Deployment environment	10
Ansible Engine	11
Physical environment configuration	12
Cabling the HPE Synergy 12000 Frame and HPE Virtual Connect 40Gb SE F8 Modules for HPE Synergy	14
Configuring the solution switching	16
HPE Synergy 480 Gen10 Compute Modules	19
HPE Synergy Composer	21
Solution storage	24
OpenShift inventory file	26
Ansible Vault	26
Ansible host file configuration	26
Compute Module configuration	27
Management nodes	27
Red Hat Virtualization Hosts	29
Red Hat OpenShift worker nodes	41
OpenShift deployment	54
Virtual machine deployment and configuration	55
OpenShift-Ansible	58
Validate OpenShift deployment	59
Command Line validation	59
Grant cluster role to user	59
Ansible OpenShift deployment removal	62
Uninstall OpenShift	62
Unregister OpenShift components and delete deployed VMs	62
Appendix A - Playbook variables	63
Appendix B - Utilizing Ansible	64
Prerequisites	64
Tasks	64



Array setup..... 64

Install NLT on the target hosts and configure the group 65

Appendix C - Deploying worker node functions to virtual machines..... 65

Appendix D – OpenShift Container Platform deployment using Ansible Tower 68

Prerequisites..... 68

Ansible Tower installation..... 68

Create new projects in Ansible Tower..... 68

Create new inventory in Ansible Tower..... 71

Set up credentials in Ansible Tower..... 73

Import Ansible plays as templates in Ansible Tower 76

Create workflow templates in Ansible Tower 78

Appendix E – Setting up Prometheus Cluster Monitoring 81

Appendix F – Aggregating container logs using EFK..... 84

Appendix G: Assessing the security posture of Red Hat OpenShift Container Platform using the automated CIS Kubernetes benchmark with the Kube-Bench utility..... 86

Change Tracker 90

Resources and additional links 91



Overview

This document describes the steps required to create a Red Hat® OpenShift Container Platform environment running on HPE Synergy and HPE Nimble Storage. It is meant to be used in conjunction with files and Ansible playbooks found at <https://github.com/HewlettPackard/hpe-solutions-openshift/tree/master/synergy/scalable/nimble>.

Hewlett Packard Enterprise 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/tree/master/synergy/scalable/nimble>. It is recommended that the installer review this document in its entirety and understands all prerequisites prior to beginning an installation. It is also recommended that the installer review the OpenShift 3.11 installation process as described by Red Hat at <https://docs.openshift.com/container-platform/3.11/install/> in its entirety and understands all prerequisites and default values prior to beginning the installation

Figure 1 displays the minimal configuration of the deployment on HPE Synergy. The OpenShift master, infrastructure, etcd, load balancer, and management pieces 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. The three (3) HPE Synergy compute modules running Red Hat Virtualization Host provide both high availability (HA) and resources to support initial workload deployments. As the workload and number of container pods grow, the user can consider moving some or all of these services from VMs to bare metal nodes for performance reasons. The Red Hat OpenShift worker nodes are deployed on HPE Synergy 480 Gen10 Compute Modules running Red Hat Enterprise Linux (RHEL) 7.6 to optimize performance. The initial Ansible deployment playbooks support a minimum of three (3) physical worker nodes. However, the design can scale higher as required and the creation of this document is based on a six (6) worker node configuration.

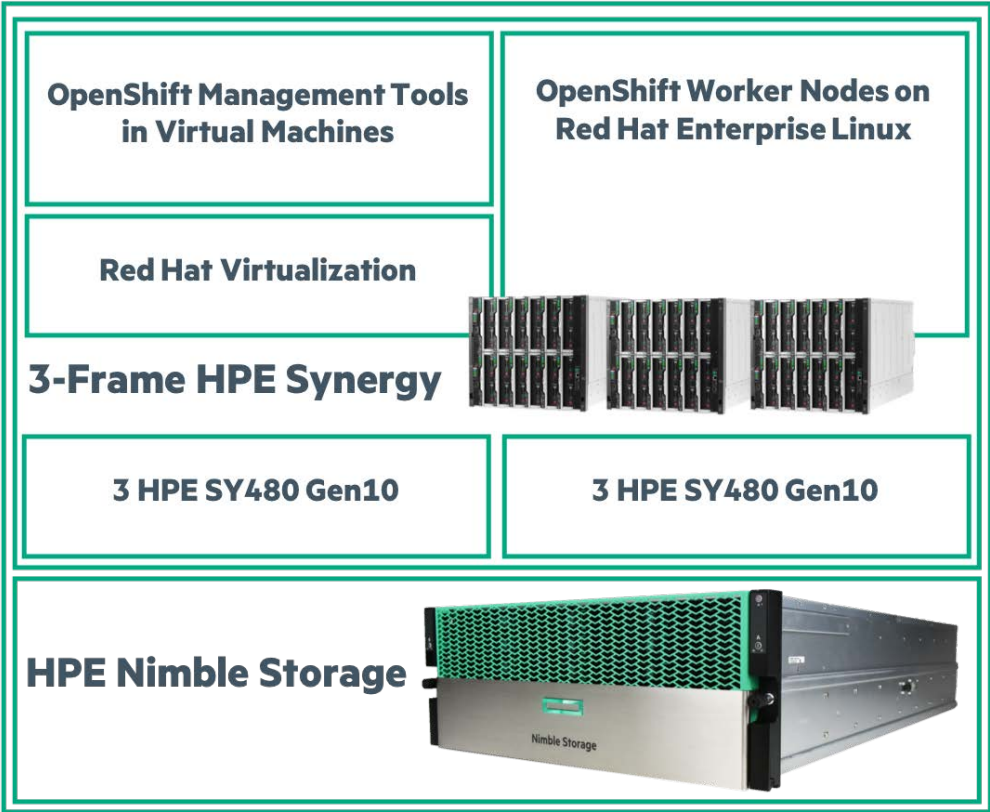


Figure 1. Solution layout



Note

The scripts described in this document and provided on the GitHub site are not supported by Hewlett Packard Enterprise or Red Hat. Scripts and files provided are examples of how to build out your infrastructure. It is expected that they will need to be adapted to work in the deployment environment.

Solution design

Figure 2 highlights the overview of the solution design from a layout and storage perspective. For detailed descriptions of the components used to create this solution consult the sections following this overview.

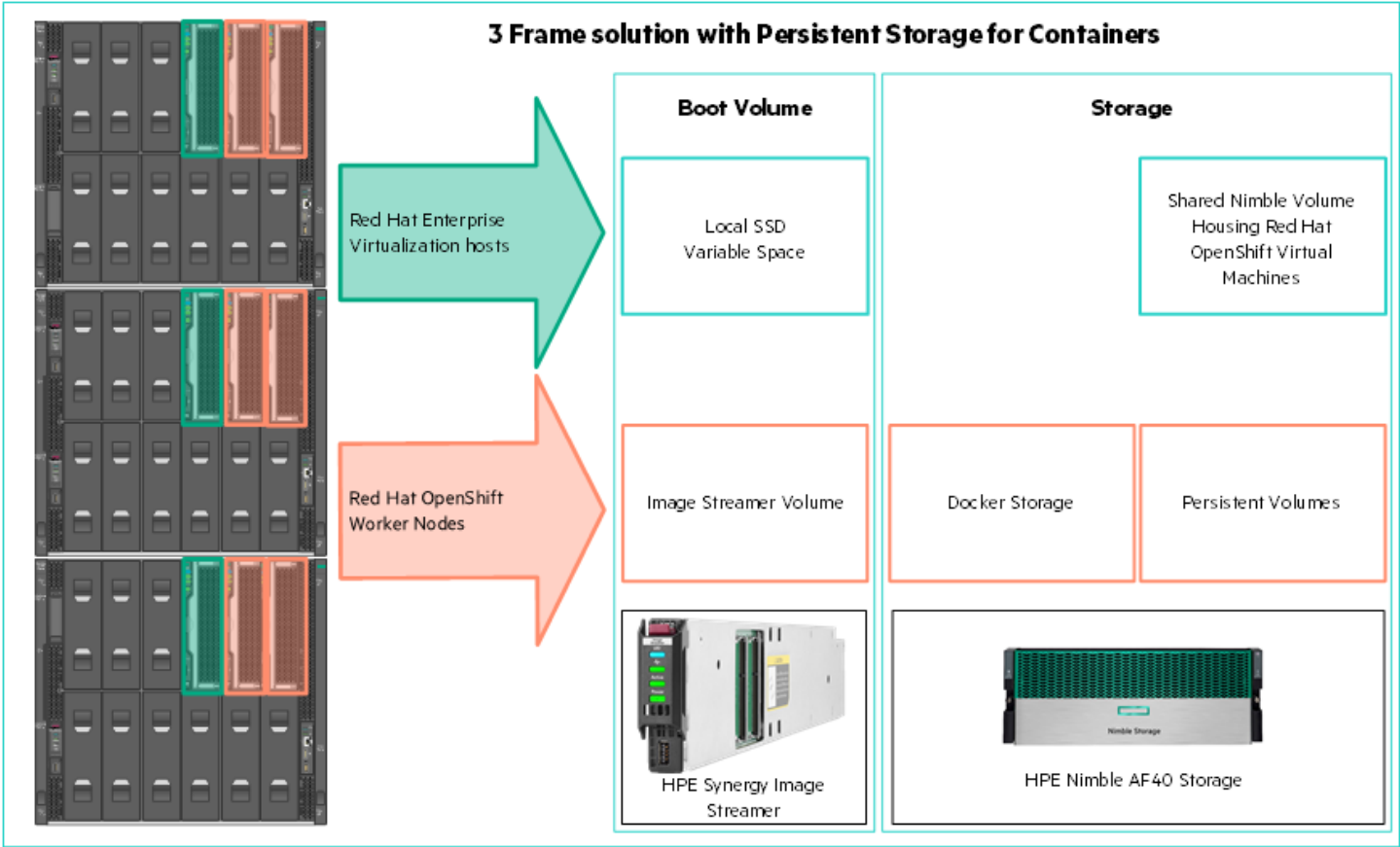


Figure 2. Solution design by function and storage type

Note

Containers and the images they are created from are stored in Docker's storage back end. This storage is ephemeral and separate from any persistent storage allocated to meet the needs of your applications. Docker Storage in Figure 2 refers to this ephemeral storage. For more information, visit https://docs.openshift.com/container-platform/3.11/install/host_preparation.html#configuring-docker-storage.



Figure 3 shows the logical design of the solution including volume attach points and virtual machine locations.

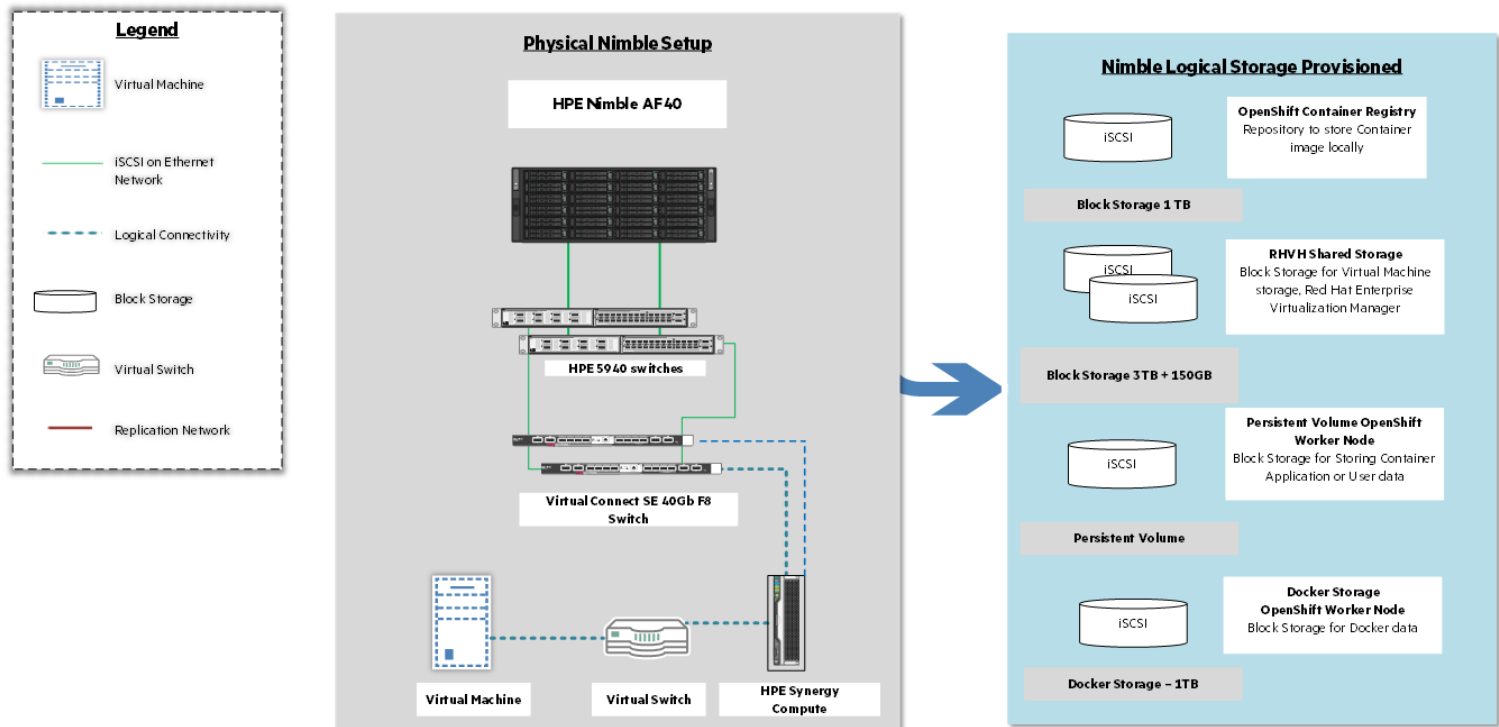


Figure 3. Logical layout of the solution stack

There are five networks defined in this solution:

1. PXE¹ - This network provides PXE boot instructions for the Red Hat Virtualization nodes.
2. Synergy Management Network - This network is specific to the requirements of HPE Synergy.
3. Management Network - This network facilitates the management of hardware and software interfaced by IT.
4. Data center Network - This network is a public access network used to connect end-users to applications.
5. iSCSI Network - This network provide iSCSI connectivity to HPE Nimble Storage Arrays. iSCSI Network B is a redundant link for iSCSI network.

¹ Preboot Execution Environment



Figure 4 describes the physical network layout within the environment and includes both the PXE deployment network for the Red Hat Enterprise Virtualization hosts and the network utilized by HPE Synergy Image Streamer to deploy worker nodes.

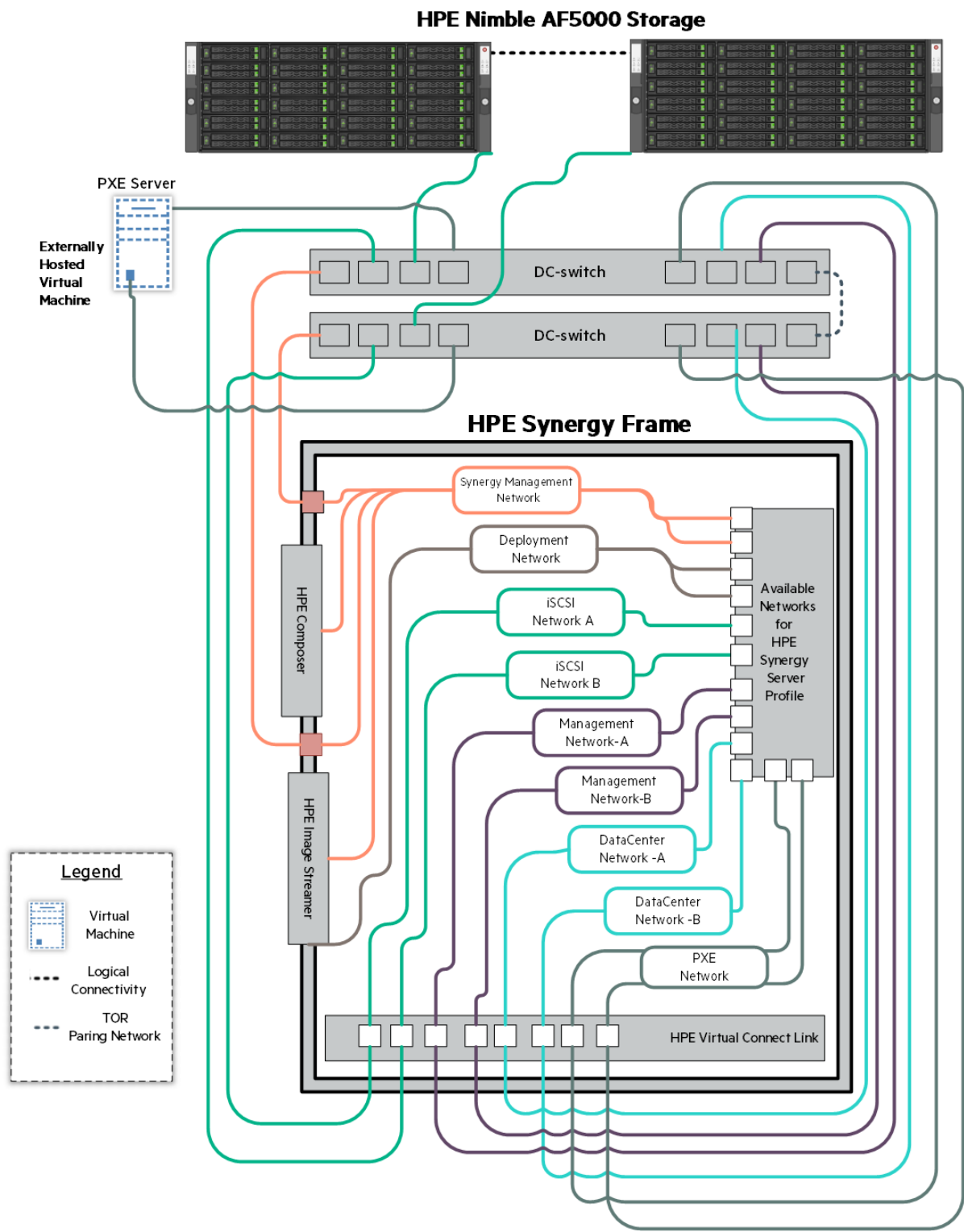


Figure 4. Physical network layout within the OpenShift solution



Solution creation process

Figure 5 shows the flow of the installation process and aligns with this document. For readability, a high-resolution copy of this image is located in the same folder as this document on GitHub. It is recommended that the installer downloads and reviews this image prior to proceeding.

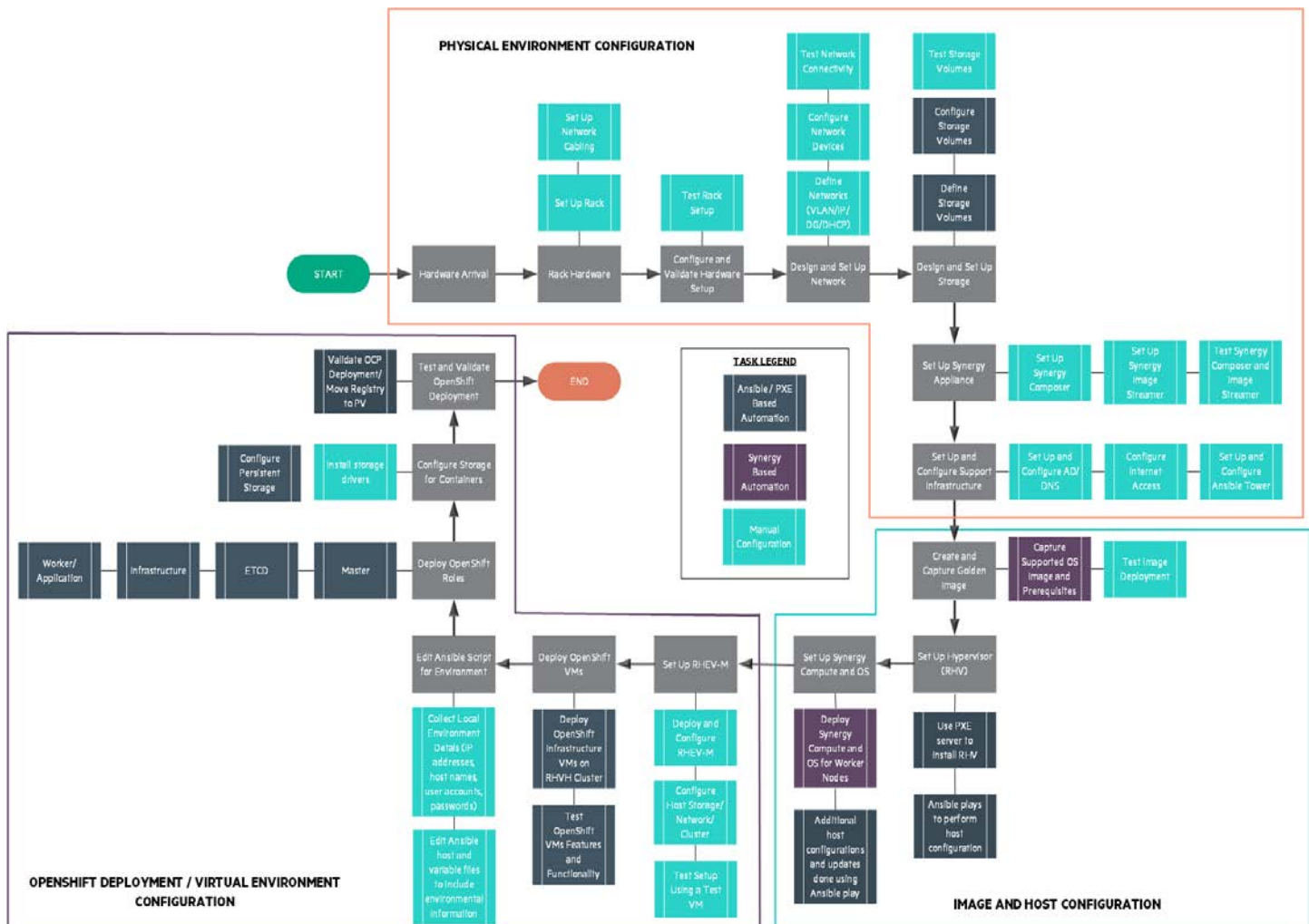


Figure 5. Solution creation flow diagram

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. This section discusses the sizing considerations for Red Hat 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 40 GB of 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 1000 pods.
- **Nodes** - Application nodes require a minimum of 1 vCPU and 8 GB RAM with a disk of 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 minimum 15 GB unallocated space per system running

containers for Docker's storage back end. 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, 24 GB RAM and 20 GB of storage for etcd data.
- Load balancer - Two (2) virtual machines were deployed in this solution serving the function of load balancing. The example in this document utilizes HA-Proxy, an open source solution. But, you should use commercially available application delivery controllers for enterprise deployments.
- Infra Nodes - When running EFK² stack on Infra nodes, 2 vCPU and 24 GB RAM is recommended for optimal performance.

Red Hat 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 2000 nodes per cluster
- Maximum of 150,000 pods per cluster
- Maximum of 250 pods per node
- Maximum pods per CPU core is the number of pods per node

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 a minimum of 20 application nodes with 250 pods per node ($5000 / 250 = 20$). In this environment with a default configuration of six (6) physical application nodes, the Red Hat OpenShift cluster should be expected to support 1,500 pods ($250 \text{ pods} \times 6 \text{ nodes} = 1,500 \text{ pods}$).

For more information about Red Hat OpenShift Container Platform sizing, refer to the Red Hat OpenShift Container Platform product documentation at: https://docs.openshift.com/container-platform/3.11/scaling_performance/index.html, and https://access.redhat.com/documentation/en-us/openshift_container_platform/3.11/html-single/scaling_and_performance_guide/index.

Prerequisites

Software versions

Table 1 describes the versions of important software utilized in the creation of this solution. The installer should ensure they have downloaded or have access to this software and that appropriate subscriptions and licensing are in place to use it within the planned timeframe.

Table 1. Major software versions used in solution creation.

Component	Version
Red Hat Enterprise Linux Server	7.6
Red Hat Virtualization, Red Hat Virtualization Manager	4.2
Red Hat OpenShift Container Platform	3.11
HPE Nimble Storage Linux Toolkit	2.4*
Nimble Kube Storage Controller	2.4*

* Latest sub-version should be installed

² Elasticsearch, Fluentd, and Kibana (EFK)



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 is 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, optionally on iSCSI networks.
DHCP	Required	Provides IP address leases on PXE, management and usually for data center networks. Optionally used to provide addresses on iSCSI networks.
TFTP/PXE	Required	Required to provide network boot capabilities to virtualized hosts that will install via a Kickstart file.
NTP	Required	Required to ensure consistent time across the solution stack
Active Directory/LDAP	Optional	May be used for authentication functions on various networks. This solution utilizes local authentication but instructions are provided to enable LDAP.

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.

DHCP

DHCP services must be in place for the PXE and management networks. DHCP services are generally in place on data center networks. As a convenience it may be useful to have them in place on iSCSI 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 management network of each physical server. This facilitates post-deployment configuration over SSH as well as a secure communication channel for running Ansible scripts. If DHCP services are present on the iSCSI networks reservations can simplify post-deployment configuration of the host on those networks.

TFTP/PXE

The virtualization hosts in this configuration were deployed via a combination of Kickstart files and manual configuration. In order to successfully complete the necessary portions of a Kickstart install, you need a host that is capable of providing 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 successfully boot and install hosts from the network, 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 URL and file locations specified in the text below will need to point to an available web server and the location of the `vmlinuz` and `initrd.img` files.

```
default menu.c32
prompt 0
timeout 300
ONTIMEOUT 1
menu title ### PXE Menu ###
label 1
menu label ^1) Install RHVH 4.2
kernel rhvh/vmlinuz
append rhvh/initrd.img inst.ks=http://<host or IP>/rhvh/ks.cfg inst.stage2=http://<host or IP>/rhvh
```

```
label 2
menu label ^3) Boot from local drive
localboot 0
```

Kickstart file options are covered under the [virtualization host configuration section](#) of this document.

NTP

A Network Time Protocol (NTP) 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 exists within the deployment environment and is accessible to the installer. Hewlett Packard Enterprise built this solution using Ansible version 2.7.9. The following repositories need to be enabled on the Ansible Engine host.

- rhel-7-server-extras-rpms
- rhel-7-server-rpms
- rhel-7-server-ose-3.11-rpms
- rhel-7-server-ansible-2.7-rpms
- rhel-7-server-rhv-4-mgmt-agent-rpms



Physical environment configuration

The configuration deployed for this solution is described in greater detail in this section.

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 from the HPE Information Library at <http://h17007.www1.hpe.com/us/en/enterprise/integrated-systems/info-library/index.aspx?cat=convergedsystems&subcat=cs750>. The user also has flexibility in customizing the HPE components throughout this stack per their unique IT and workload requirements or building with individual components. Figure 6 shows the physical configuration of the racks used in this solution.

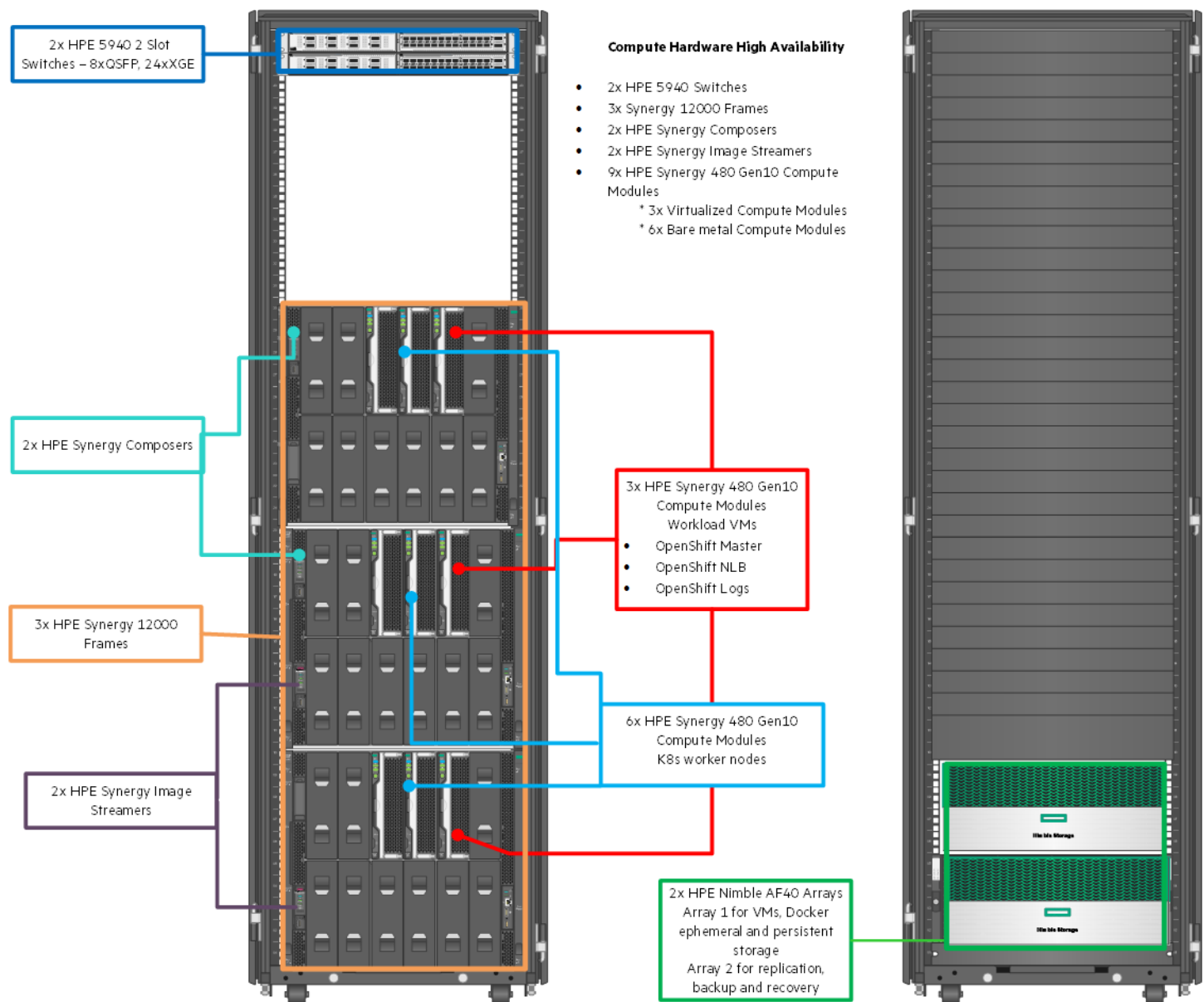


Figure 6. Physical layout of the compute within the solution

As noted in the overview, firmware recipes for the individual components adhere to HPE Converged Solution 750 specifications which can be found in the Firmware and Software Compatibility Matrix downloadable from <http://h17007.www1.hpe.com/us/en/enterprise/integrated-systems/info-library/index.aspx?cat=convergedsystems&subcat=cs750>. It is recommended that the installer utilize the latest available matrix.

Figure 6 also shows two HPE Nimble Storage AF40 arrays which were used in the deployment of this solution. The topic of container backup is covered in the backup and recovery guide for HPE Nimble Storage found at https://github.com/HewlettPackard/hpe-solutions-openshift/tree/master/synergy/scalable/backup_and_recovery.

Table 3 below highlights the individual components and their quantities as deployed within the solution.

Table 3. Components utilized in the creation of this solution.

Component	QTY	Description
HPE Synergy 12000 Frame	3	Three (3) HPE Synergy 12000 Frames house the infrastructure used for the solution
HPE Virtual Connect 40Gb SE F8 Module	2	A total of two (2) HPE Virtual Connect 40Gb SE F8 Modules provide network connectivity into and out of the frames
HPE Synergy 480 Gen10 Compute Module	9	Three (3) virtualized management hosts and six (6) bare metal or virtualized hosts for worker nodes
HPE FlexFabric 2-Slot Switch	2	Each switch contains one (1) each of the HPE 5940 modules listed below
HPE 5940 24p SFP+ and 2p QSFP+ Module	2	One module per HPE FlexFabric 2-Slot Switch
HPE 5940 8-port QSFP+ Module	2	One module per HPE FlexFabric 2-Slot Switch
HPE Nimble Storage AF40	1	One array for virtual machines, Docker storage and persistent volumes
HPE Nimble Storage AF40	1	One array for remote replication as outlined in the Backup and Recovery Guide
HPE Synergy Image Streamer	2	Provides OS volumes to OpenShift worker nodes
HPE Synergy Composer	2	Core configuration and lifecycle management for the Synergy components



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

This section shows the physical cabling between frames as well as the physical connectivity between 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 7 shows the cabling of the HPE Synergy Interconnects, HPE Synergy Frame Management, and Management and Intelligent Resilient Fabric (IRF) connections. These connections handle east-west network communication as well as management traffic within the solution.

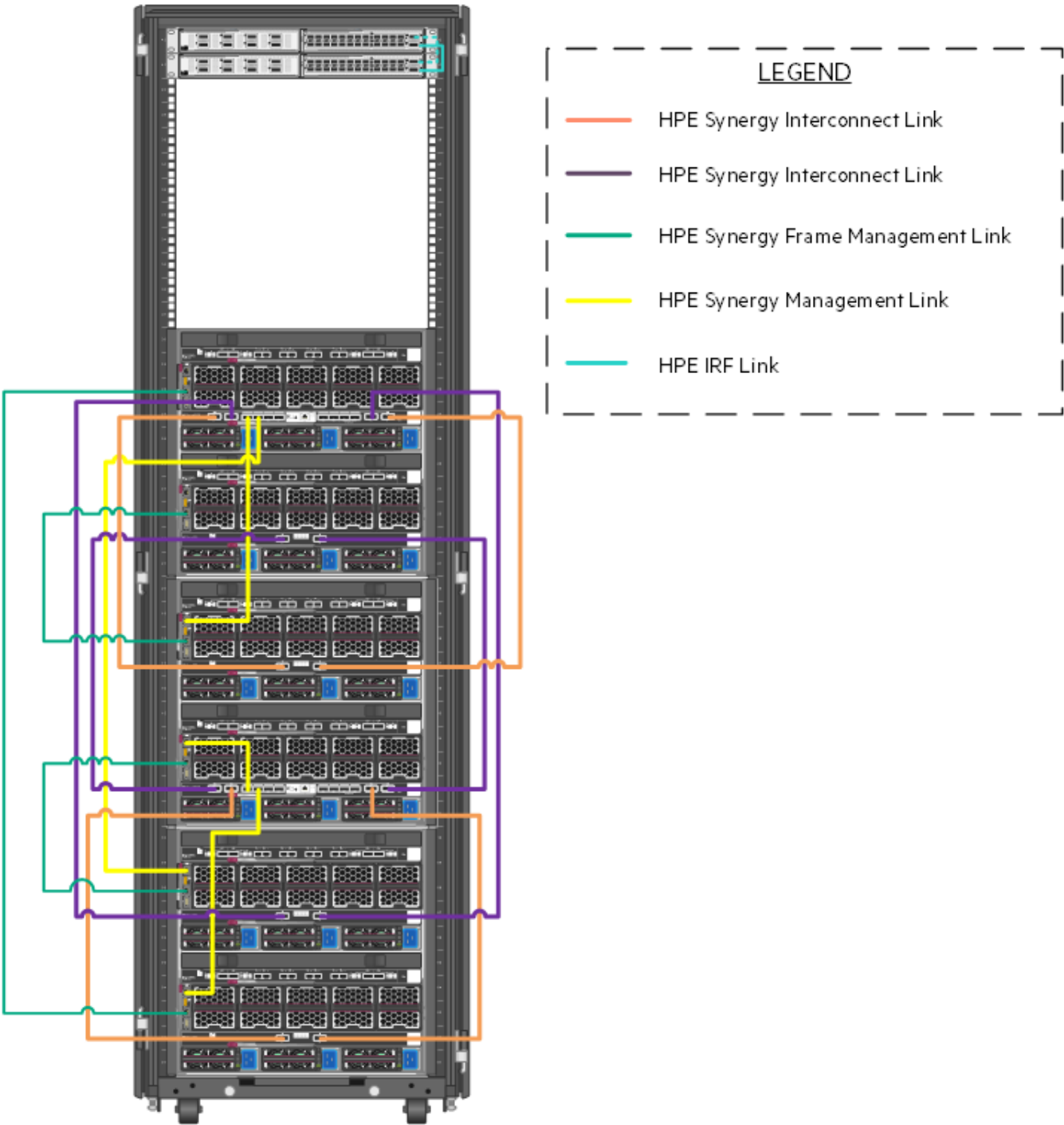


Figure 7. Cabling of the management and inter-frame communication links within the solution.



Figure 8 shows the cabling of HPE Synergy Frames to the network switches. The specific networks contained within the Bridge-Aggregation Groups (BAG) are described in more detail later in this section. At the lowest level, there are four (4) 40GbE connections dedicated to carrying redundant, production network traffic to the first layer switch where it is further distributed. iSCSI traffic is separated into two (2) VLANs and is carried to the first network switch pair over two (2) 40GbE links per VLAN. Unlike the Ethernet traffic which is distributed between the switches, each iSCSI VLAN is sent directly to one switch configured with a pair of access ports.

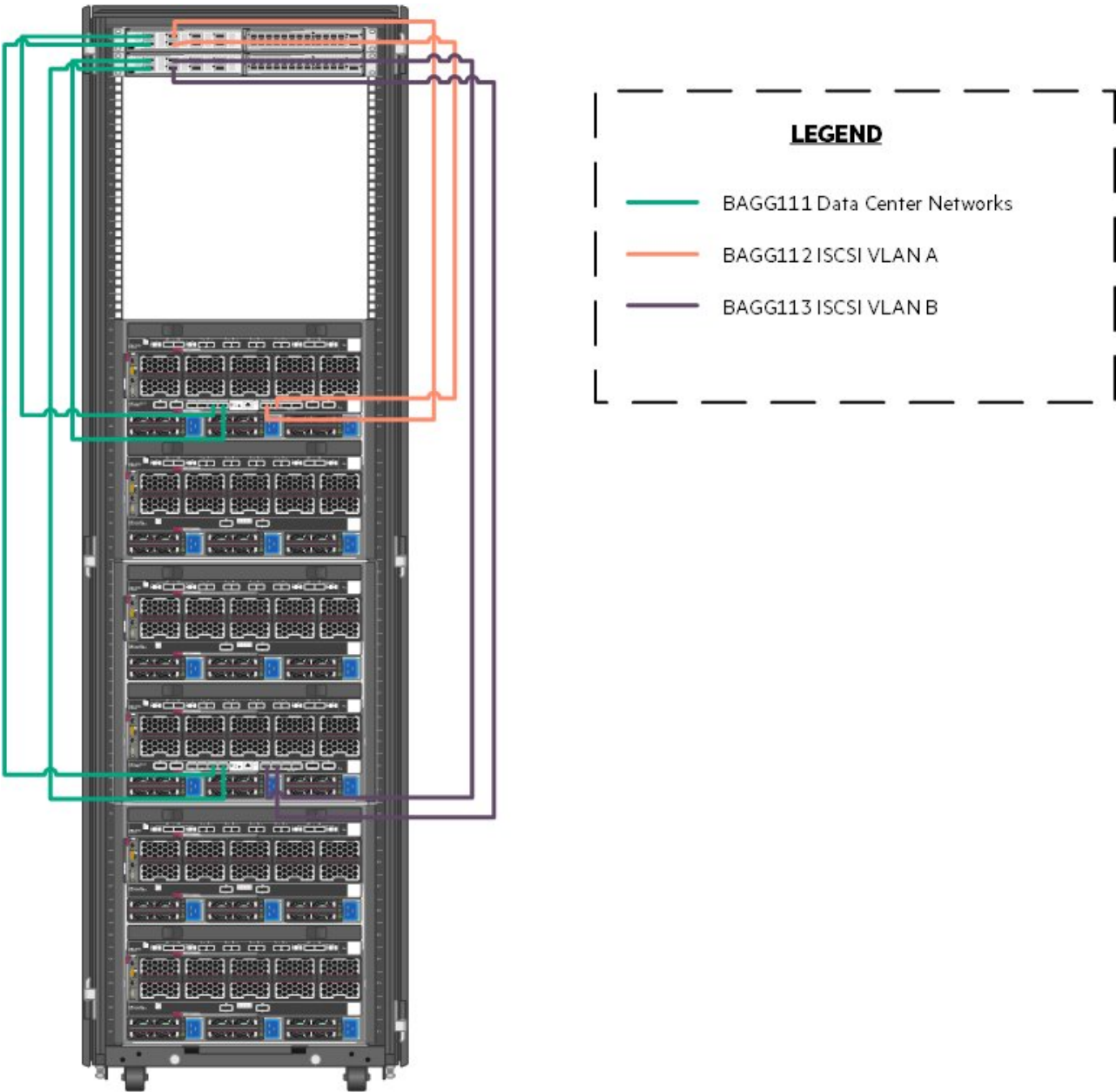


Figure 8. Cabling of the HPE Synergy 12000 Frames to the HPE FF 5940 switches.



Table 4 explains the cabling of the Virtual Connect interconnect modules to the HPE FF 5940 switching.

Table 4. Networks used in this solution

Uplink Set	Synergy Source	Switch Destination
Network	Enclosure 1 Port Q3	FortyGigE1/1/1
	Enclosure 1 Port Q4	FortyGigE2/1/1
	Enclosure 2 Port Q3	FortyGigE1/1/2
	Enclosure 2 Port Q4	FortyGigE2/1/2
iSCSI_SAN_A	Enclosure 1 Port Q5	FortyGigE1/1/5
	Enclosure 1 Port Q6	FortyGigE1/1/6
iSCSI_SAN_B	Enclosure 2 Port Q5	FortyGigE2/1/5
	Enclosure 2 Port Q6	FortyGigE2/1/6

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 IRF.³ To understand the process of configuring IRF, refer the HPE FlexFabric 5940 Switch Series Installation Guide at https://support.hpe.com/hpsc/doc/public/display?sp4ts.oid=null&docLocale=en_US&docId=emr_na-c05212026. This guide may also be used to understand the initial installation of switching as well as creation of user accounts and access methods. The remainder of this section is built with the assumption that the switch has been installed, configured for IRF, hardened, and is accessible over SSH.

The installer may choose to utilize end of row switching to reduce switch and port counts in the context of the solution. If end of row switching is the approach, then this section should be used as guidance for how to route network traffic outside of the HPE Synergy Frames.

³ Intelligent Resilient Framework



Physical cabling

Table 5 is a map of source ports to ports on the HPE FlexFabric 5940 switches.

Table 5. HPE FlexFabric 5940 port map

Source Port	Switch Port
Nimble Management Port Eth1	TenGigE1/2/17
Nimble Controller A TG1	TenGigE1/2/13
Nimble Controller A TG2	TenGigE2/2/13
Nimble Controller B TG1	TenGigE1/2/14
Nimble Controller B TG2	TenGigE2/2/14
Nimble Replication Port Eth2	TenGigE1/2/15
Virtual Connect Frame U30, Q3	FortyGigE1/1/1
Virtual Connect Frame U30, Q4	FortyGigE2/1/1
Virtual Connect Frame U30, Q5	FortyGigE1/1/5
Virtual Connect Frame U30, Q6	FortyGigE1/1/6
Virtual Connect Frame U40, Q3	FortyGigE1/1/2
Virtual Connect Frame U40, Q4	FortyGigE2/1/2
Virtual Connect Frame U40, Q5	FortyGigE2/1/5
Virtual Connect Frame U40, Q6	FortyGigE2/1/6
To Upstream Switching	Customer Choice

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

Network definitions

Table 6 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 the composer. This solution utilizes unique VLANs for the data center and solution management segments. Actual VLANs and network counts will be determined by the requirements of your production environment.

Table 6. Networks used in this solution

Network Function	VLAN Number	Bridge Aggregation Group
PXE	501	111
Solution_Management	1193	111
Data_Center	2193	111
iSCSI_A	3193	112
iSCSI_B	3194	113

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

```
# sys
# vlan 501 1193 2193 3193 3194
```

2. 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
# quit
```



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

3. 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 previous tables, run the following commands.
4. For the data center and management VLANs, run the following commands:

```
# interface Bridge-Aggregation111
# link-aggregation mode dynamic
# description <FrameNameU30>-ICM
# quit

# interface range name <FrameNameU30>-ICM interface Bridge-Aggregation111
# quit

# interface range FortyGigE 1/1/1 to FortyGigE 1/1/2 FortyGigE 2/1/1 to FortyGigE 2/1/2
# port link-aggregation group 111
# quit

# interface range name <FrameNameU30>-ICM
# port link-type trunk
# undo port trunk permit vlan 1
# port trunk permit vlan 501 1193 2193
# quit
```

5. For the VLANs that will carry iSCSI traffic, run the following commands:

```
# interface Bridge-Aggregation112
# link-aggregation mode dynamic
# description <FrameNameU30>-ICM 3
# quit

# interface range FortyGigE 1/1/5 to FortyGigE 1/1/6
# port link-aggregation group 112
# quit

# interface Bridge-Aggregation 112
# port link-type trunk
# undo port trunk permit vlan 1
# port trunk permit vlan 3193
# quit

# interface Bridge-Aggregation113
# link-aggregation mode dynamic
# description <FrameNameU30>-ICM 6
# quit

# interface range FortyGigE 2/1/5 to FortyGigE 2/1/6
# port link-aggregation group 113
# quit

# interface Bridge-Aggregation 113
# port link-type trunk
# undo port trunk permit vlan 1
# port trunk permit vlan 3194
# quit
```

6. 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
```



7. Assign the network ports that will connect the various HPE Nimble Storage interfaces with the switches:

```
# interface range TenGigE 1/2/13 to TenGigE 1/2/14
# port access vlan 3193
# quit

# interface range TenGigE 2/2/13 to TenGigE 2/2/14
# port access vlan 3194
# quit
```

8. Place the HPE Nimble Storage management interface into the management VLAN:

```
# interface TenGigE 1/2/17
# port access vlan 1193
# quit
```

9. Place the replication traffic for the VLAN into the default VLAN or a VLAN of your choice:

```
# interface TenGigE 1/2/15
# port access vlan 1
# quit
```

10. When you have completed configuration of the switches, ensure you save your state and apply it by typing “save”.

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 HPE Synergy 480 Gen10 Compute Modules, regardless of function, were all configured identically. Table 7 describes the individual components. Server configuration should be based on customer needs and the configuration used in the creation of this solution may not align with the requirements of any given production implementation.

Table 7. Host configuration

Component	Quantity
HPE Synergy 480/660 Gen10 Intel Xeon-Gold 6130 (2.1GHz/16-core/125W) FIO Processor Kit	2 per server
HPE 8GB (1x8GB) Single Rank x8 DDR4-2666 CAS-19-19 Registered Smart Memory Kit	20 per server
HPE 16GB (1x16GB) Single Rank x4 DDR4-2666 CAS-19-19 Registered Smart Memory Kit	4 per server
HPE Synergy 3820C 10/20Gb Converged Network Adapter	1 per server
HPE Smart Array P204i-c SR Gen10 12G SAS controller	1 per server
HPE 1.92TB SATA 6GB Mixed Use SFF (2.5in) 3yr Warranty Digitally Signed Firmware SSD	2 per management host



Red Hat Virtualization Hosts

The solution calls for the installation of Red Hat Virtualization Host 4.2 (RHVH) on three (3) HPE Synergy 480 Gen10 Compute Modules. These hosts house the required management software for the solution installed on virtual machines which consolidates the infrastructure within the solution stack to virtual machines as opposed to physical hosts. Figure 9 highlights the connectivity of these hosts to the primary HPE Nimble Storage array. Networks that are carried on the individual BAG are shown in Table 6 in this document.

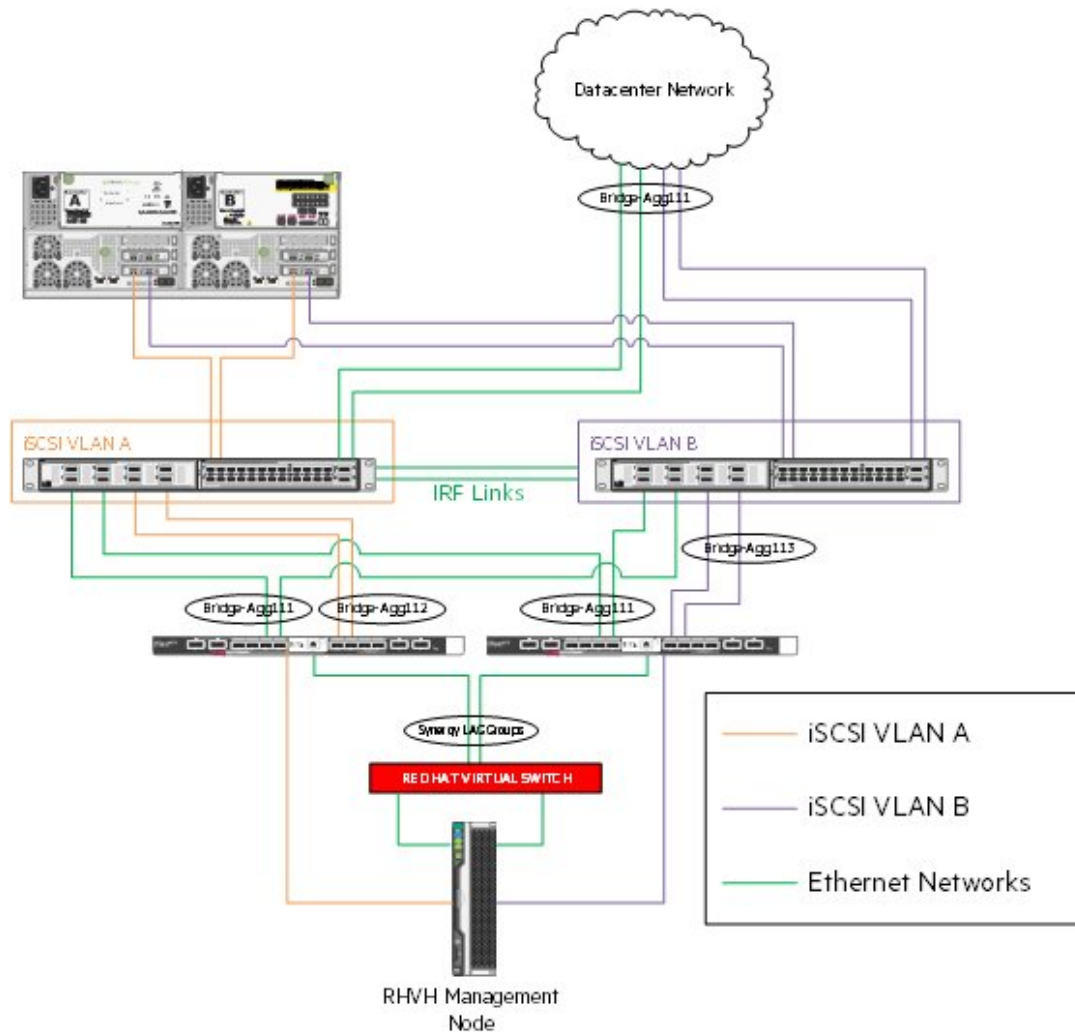


Figure 9. Red Hat Virtualization Host network connectivity

Each host is presented with access to a shared 3 TB storage volume that houses the virtual machines used in the solution. An additional shared 150 GB volume houses the Red Hat Virtualization Manager VM. Figure 3 and Figure 11 in this document explain this relationship.

Note

If virtual worker nodes are implemented as in [Appendix C](#) of this document, the configuration outlined above will be used for the virtual worker node hosts.

Red Hat Enterprise Linux hosts

Six (6) HPE Synergy 480 Gen10 Compute Modules are deployed as Red Hat OpenShift worker nodes and run Red Hat Enterprise Linux Server 7.6 as their operating system. Figure 10 highlights the connectivity of these hosts to the primary HPE Nimble Storage array. As with the virtualization hosts, refer to Table 6 in this document for an explanation of networks carried on each individual BAG.

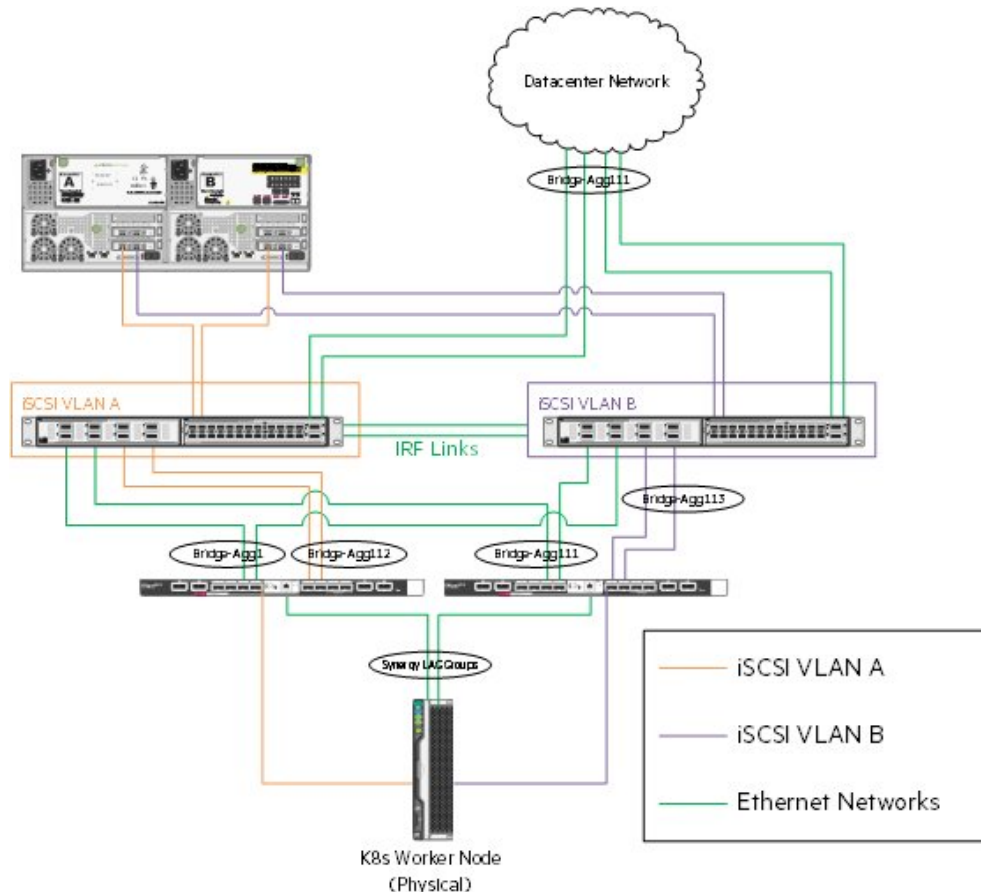


Figure 10. Red Hat OpenShift worker node network connectivity

HPE Synergy Composer

At the core of the management of the HPE Synergy environment is HPE Synergy Composer. A pair of HPE Synergy Composers are deployed across frames to provide redundant management of the environment for both initial deployment and changes over the lifecycle of the solution. 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 using a Virtual Network Computing (VNC) session

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 HPE 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, select **Connect** to start HPE OneView for Synergy.



5. Select **Hardware Setup** and enter the following information when prompted. Note that this solution places the HPE Synergy Composers on the management network. Pre-populating DNS with IP information is recommended.
 - **Appliance host name:** Enter a 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 **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 Frame Setup and Installation Guide available at www.hpe.com/info/synergy-docs offers suggestions to fix common issues.
7. Select the **OneView** menu at the top of the screen and select **Settings > Appliance**. Validate that both appliances are connected and show a green checkmark.

Configure appliance credentials

1. Log in to the **HPE OneView for Synergy appliance**.
2. At first login, you will be asked to define credentials for the administrator user. To do this, accept the end user license agreement (EULA) and in the **HPE OneView Support** box, ensure that **Authorized Service** is **Enabled**.
3. 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 at https://support.hpe.com/hpsc/doc/public/display?sp4ts.oid=null&docLocale=en_US&docId=emr_na-a00051226en_us. The solution used the firmware recipe from June of 2018.

1. Select the **OneView** menu and select **Settings**.
2. Under Appliance, select **Update Appliance** and update **Composer**.
3. Once the update process completes, validate that both composer modules are 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 Converged Solutions 750 recipe. Once the bundle starts uploading, 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 before proceeding further.



Configure Ethernet networks

As illustrated in [Network Configuration section](#) of the switch in 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 8 at a minimum. Your VLAN values will generally differ from those described below.

Table 8. Networks defined within HPE Synergy Composer for this solution

Network Name	Type	VLAN Number	Purpose	Requested Bandwidth (Gb)	Maximum Bandwidth (Gb)
Management	Ethernet	1193	Solution management	5	20
Data_Center	Ethernet	2193	Application, authentication and other user networks	5	20
ISCSI_VLAN_A	Ethernet	3193	ISCSI VLAN A	8	20
ISCSI_VLAN_B	Ethernet	3194	ISCSI VLAN B	8	20
PXE_Boot	Ethernet	501	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 HPE Synergy Composer, use the Guided Setup to create a Logical Interconnect Group (LIG) with three (3) uplink sets defined. For this solution, the uplink sets were named Network, **ISCSI_SAN_A**, and **ISCSI_SAN_B**. The iSCSI uplink sets carry the respective iSCSI VLANs. The uplink set "Network" carries all other networks defined for the solution.

Table 9 below defines the ports used to carry the uplink sets.

Table 9. Networks used in this solution

Uplink Set	Synergy Source
Network	Enclosure 1, Bay 3, Port Q3
	Enclosure 1, Bay 3, Port Q4
	Enclosure 2, Bay 6, Port Q3
	Enclosure 2, Bay 6, Port Q4
ISCSI_SAN_A	Enclosure 1, Bay 3, Port Q5
	Enclosure 1, Bay 3, Port Q6
ISCSI_SAN_B	Enclosure 2, Bay 6, Port Q5
	Enclosure 2, Bay 6, Port Q6

Create Enclosure Group

1. From the Guided Setup, select **Create enclosure group**.
2. Provide a name and enter the number of frames.
3. Select **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. Select **Create** when ready.

Create Logical Enclosure

Use the Guided Setup to create a logical enclosure making use of all three (3) enclosures. 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**. Click **Cancel** to exit.



Solution storage

An HPE Nimble Storage AF40 array provides shared and dedicated storage for a variety of purposes within this solution. Figure 11 shows the cabling of the HPE Nimble Storage AF40 to the HPE switching utilized in this solution. Note that this diagram shows the storage and switching in the same rack to provide clarity. As implemented for this solution, the switching resided in the HPE Synergy rack. The orange and purple wires in the figure represent the separate iSCSI VLANs. This figure represents two HPE Nimble Storage arrays which were implemented to provide replication. In a real-world implementation, these arrays will be in separate physical locations (two separate data centers, separate buildings, separate sections of the same data center) to maximize redundancy and minimize failure points.

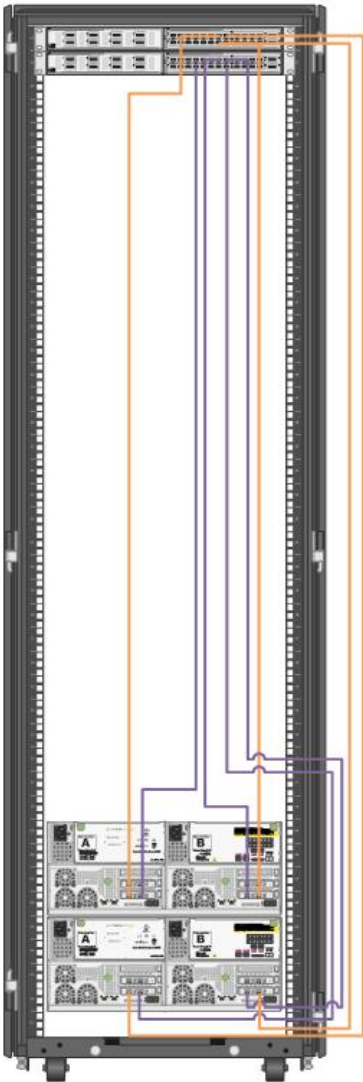


Figure 11. Cabling of the HPE Nimble Storage arrays to the HPE FF 5940 switches



Figure 12 describes the logical storage layout used in the solution. Local storage is used for the operating system installation on the virtualized hosts and the image streamer volume is used for the operating system installation on the bare metal worker nodes. The HPE Nimble Storage AF40 provides dedicated and shared volumes as outlined in the Figure 12.

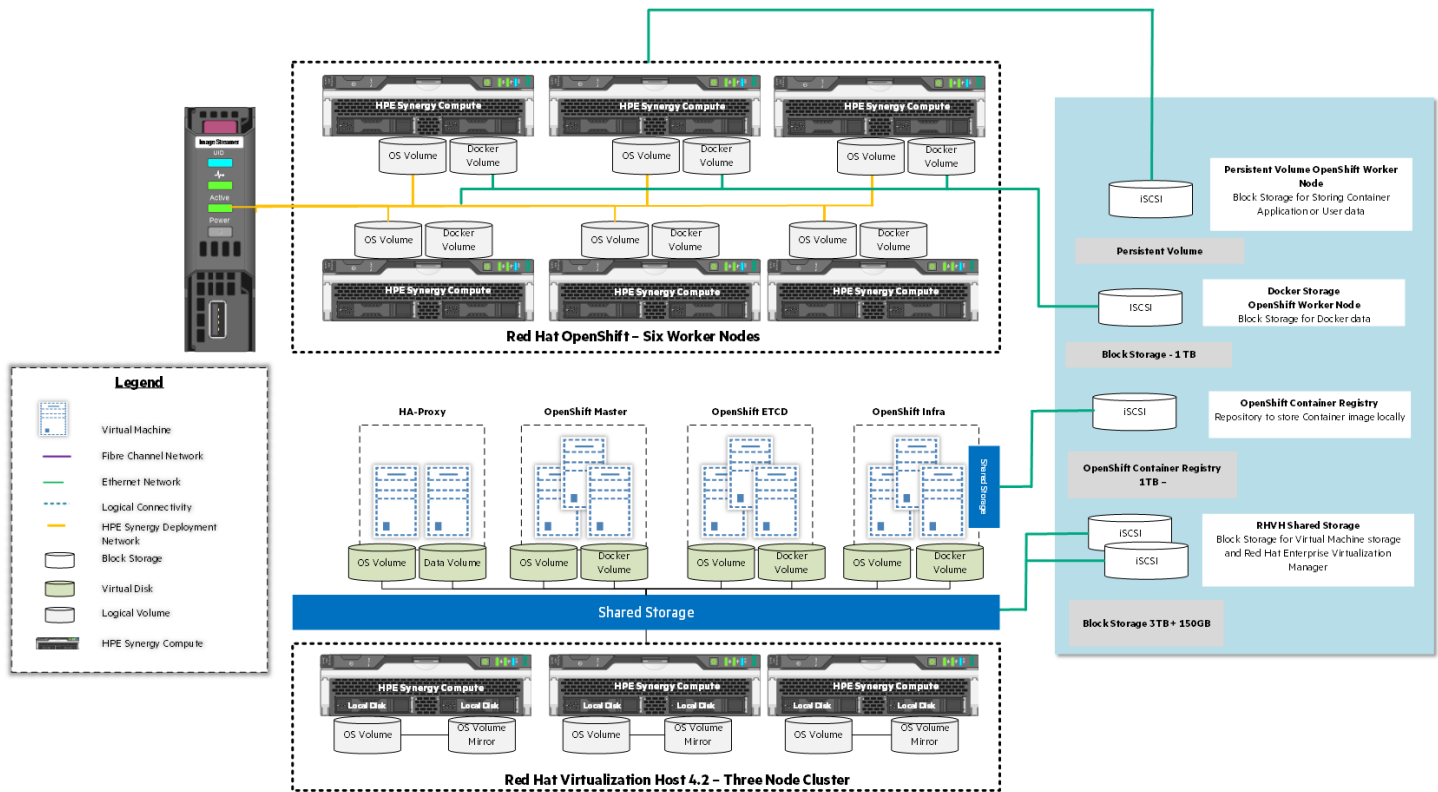


Figure 12. iSCSI physical layout

Each virtualized host is configured with two (2) HPE 1.92GB 6Gb SATA Mixed Use SSDs. These are configured into a logical disk with RAID1 protection. Depending on the unique requirements of each implementation larger or smaller capacity, hard drives may be selected.

The HPE Nimble Storage AF40 that is used as the primary storage target is mapped to the HPE FF 5940 switching as described in Table 10. The replication storage would be deployed in the same fashion, but generally on separate switching in a separate physical location such as a remote data center, a separate building on a campus, or in the same data center on unique power feeds in a separate rack. All switch ports are configured as access ports so the VLANs are untagged (see switch configuration section earlier in this document). The redundant array is implemented in a separate location with the same physical configuration.

Table 10. HPE Nimble Storage AF40 port mapping

HPE Nimble Storage AF40 Port	VLAN Number	Switch Port
Management Port Eth1, Controller A	1193	TenGigE1/2/17
Management Port Eth1, Controller B	1193	TenGigE2/2/17
Controller A TG1	3193	TenGigE1/2/13
Controller A TG2	3194	TenGigE2/2/13
Controller B TG1	3193	TenGigE1/2/14
Controller B TG2	3194	TenGigE2/2/14
Replication Port Eth2	Default	TenGigE1/2/15

Information about storage volumes/disks is described in Table 11. The installer may choose to manually create and present these volumes or use the Ansible resources specified after Table 11.

Table 11. Volumes and sources used in this solution

Volume/Disk Function	Qty	Size	Source	Hosts	Shared/Dedicated
Hypervisor	3	1.92TB	Local Disk	RHVH hosts	Dedicated
Operating System	6	29GB+	HPE Image Streamer	OpenShift worker nodes	Both
Virtual Machine Hosting	1	3TB	HPE Nimble	RHVH hosts	Shared
Persistent Application Data	N	App Specific	HPE Nimble	OpenShift worker nodes	Dedicated
Docker Local Storage	6	100GB	HPE Nimble	1 per OpenShift worker node	Dedicated
RHV-M Hosting	1	150GB	HPE Nimble	RHVH hosts	Shared
OpenShift Container Registry	1	1TB	HPE Nimble	Infrastructure nodes	Shared

Prior to defining these volumes, you must initialize and configure the array. Hewlett Packard Enterprise has provided resources to automate the initialization and configuration of the array. Refer [Appendix B](#) for information about utilizing Ansinimble to automate the configuration of array parameters.

Note

The volumes marked as “Dedicated” in Table 11 should be created and presented to a single host that will consume the volume. Volumes marked as “Shared” should be presented to the group of hosts identified in the “Hosts” column.

OpenShift inventory file

Prior to configuring the compute modules, the installer should retrieve the required Ansible plays and files from GitHub by running the following command from the Ansible Engine server:

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

The files that are cloned from the GitHub site include a sample inventory file. The installer should review this file carefully (located at `/etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/hosts`) and ensure that the information within the file accurately reflects the information in their environment.

Ansible Vault

A preconfigured Ansible Vault file is provided with the solution. Run the following command to edit the vault to match the installer's environment:

```
# ansible-vault edit /etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/vault_pass.yml
```

Note

The default password for the vault is ‘changeme’.

Ansible host file configuration

An Ansible host file is required for providing the host details and variables for the Ansible plays, and is separated into sections that disseminate universal configuration parameters, options for virtualized hosts, and options for bare metal hosts. 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.



Use an editor (vi or vim) to edit the Ansible host file located at `/etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/hosts`.

```
#vim /etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/hosts
```

Warning

During host deployment, ensure that adapter names and functions are accurately recorded for the installation environment as variations in install procedures may result in different adapter functions than what is represented in the following sections. This will result in the failure of the automated configuration steps.

Compute Module configuration

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

HPE Synergy Composable Infrastructure using HPE Virtual Connect provides the construct of a server profile. A server profile allows a set of configuration parameters, including firmware recipe, network and SAN connectivity, BIOS tuning, boot order configuration, local storage configuration, and more to be templated. These templates are the key to delivering the “infrastructure as code” capabilities of the HPE Synergy platform. For the purpose of this solution, two templates were created. One template was used to define virtualized management nodes and the other specified parameters for OpenShift worker nodes.


Management nodes

Create server profile template

1. For each management node in the solution, log in to the HPE Synergy Composer, select the **OneView** drop-down list, and then select **Server Profiles**. Click the **+ Create Profile** button.
2. Assign a logical name to the profile and select **None** from the **Server profile template** drop-down list.
3. Under Server hardware, select the system you are building the profile for.
4. Scroll to the Connections section of the profile. Select **Add connection** and create the following connections:
 - a. PXEBoot A, network is your PXE network, requested bandwidth is 1, boot is PXE primary.
 - b. PXEBoot B, network is your PXE network, requested bandwidth is 1, boot is PXE secondary.
 - c. Management A, network is your management network, requested bandwidth is 5Gb/s, the network is not bootable.
 - d. Management B, network is your management network, requested bandwidth is 5Gb/s, the network is not bootable.
 - e. Datacenter A, network is your data center network, requested bandwidth is 5Gb/s, the network is not bootable.
 - f. Datacenter B, network is your data center network, requested bandwidth is 5Gb/s, the network is not bootable.
 - g. iSCSI_A, network is your storage network, requested bandwidth is 8Gb/s, the network is not bootable.
 - h. iSCSI_B, network is your storage network, requested bandwidth is 8Gb/s, the network is not bootable.



Figure 13 describes the configuration of the network interfaces as part of the profile template. There are eight networks that are defined.

Connections  [Edit](#)

Expand all Collapse all

	ID	Name	Network	Port	Boot
▶	1	pxe	PXEBoot VLAN500	Mezzanine 3:1-a	PXE primary
▶	2	pxe02	PXEBoot VLAN500	Mezzanine 3:2-a	PXE secondary
▶	3	mgmt01	TenNet VLAN1193	Mezzanine 3:1-b	Not bootable
▶	4	mgmt02	TenNet VLAN1193	Mezzanine 3:2-b	Not bootable
▶	5	dcnet01	TwentyNet VLAN2193	Mezzanine 3:1-c	Not bootable
▶	6	dcnet02	TwentyNet VLAN2193	Mezzanine 3:2-c	Not bootable
▶	7	iscsi_a	iSCSI SAN A VLAN3193	Mezzanine 3:1-d	Not bootable
▶	8	iscsi_b	iSCSI SAN B VLAN3194	Mezzanine 3:2-d	Not bootable

Figure 13. Server connections as part of the profile template

5. In addition to defining the networks, configure the local storage.

Figure 14 shows the configuration of the local storage from within the profile template. Each server houses two (2) solid state disks. These are configured as a RAID 1 logical drive.

Local Storage

Integrated storage controller

Mode

Managed by OneView

●

Name

Type

Logical Drive ID

RAID Level

Number of Drives

Size GB

Drive Technology

Boot

loc01

Logical drive

1

RAID 1

2

n/a

SATA SSD

No

Figure 14. Local storage used for this solution

6. Under **Boot settings**, select **Manage boot mode** > **Legacy BIOS**. Choose to have hard disk and then PXE as the first two items in the boot order.
7. When complete, click **Create**.

Create server profile using server profile template

You can use the server profile template to create individual server profile for each of the three virtualized management nodes:

1. From the **OneView** drop-down list, select **Server Profiles**.
2. Click the **+ Create Profile** button.
3. Assign a logical name to the profile and choose the already created server profile template from the drop-down list.

The network connections and storage settings configured as part of the template are fetched for the profile.

Note

You may choose to select **None** from the **Server Profile Template** drop- down list. In this case, you will have to configure the network connections, local storage, and so on.



Figure 15 is an example that describes the iSCSI connections of an individual profile. This solution uses Ethernet which creates software iSCSI initiators rather than hardware-based initiators. The solution calls for bandwidth of at least 8 GB per adapter, but this is customizable based on solution requirements.

Note
The MAC addresses are defined and available even if a compute module has not become active.

▼	●	7	iscsi_a	iSCSI_SAN_A	VLAN3193	Mezzanine 3:1-d	Not bootable
			Interconnect	2S1721PK4K, interconnect 3			
			Type	Ethernet			
			MAC address	06:FB:29:B0:03:D2 (v)			
			Requested virtual functions	None			
			Requested bandwidth	8 Gb/s			
			Allocated bandwidth	8 Gb/s			
			Max bandwidth	20 Gb/s			
			Link aggregation group	None			
▼	●	8	iscsi_b	iSCSI_SAN_B	VLAN3194	Mezzanine 3:2-d	Not bootable
			Interconnect	MXQ73007JR, interconnect 6			
			Type	Ethernet			
			MAC address	06:FB:29:B0:03:D3 (v)			
			Requested virtual functions	None			
			Requested bandwidth	8 Gb/s			
			Allocated bandwidth	8 Gb/s			
			Max bandwidth	20 Gb/s			
			Link aggregation group	None			

Figure 15. iSCSI connections within the server profile

4. Once the server profile is created, power on the server.

Red Hat Virtualization Hosts

The Red Hat Virtualization Host installation and configuration overview appears below:

1. Install RHVH on each cluster node.
2. Access the Cockpit UI to deploy the RHV-M (hosted engine) on a single node.
3. Complete the wizard to deploy the hosted engine virtual machine.
4. Access the RHV Administration Portal on the hosted engine virtual machine.
5. Add additional hosts to the RHV cluster.
6. Ensure each host is configured to run hosted engine.
7. Configure host networking.
8. Configure storage.
9. Test failover of hosted engine.

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 Host installation

This solution utilizes a combination of Kickstart, Ansible playbooks, and manual processes to configure Red Hat Virtualization Host on three (3) HPE Synergy 480 Gen10 Compute Modules.

The Kickstart file used in the creation of this guide was hosted over HTTP. The file used appears below and can be created with a text editor such as Vim or Nano. The installer will need to enter the unique data for their environment including the location of their squashfs.img, unique hashed password, gateway, and DNS information.

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

autopart --type=thinp
zerombr
clearpart --all --initlabel
liveimg --url=http://192.168.3.3/rhvininstall/squashfs.img
rootpw --iscrypted L0nGCh@r@ct3RS+r!nG
ignoredisk --only-use=sda
timezone --utc America/Chicago
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 --noefroute --activate
network --device=ens3f7 --bootproto=static --noefroute --activate
network --device=bond1 --bondslaves=ens3f2,ens3f3 --bondopts=mode=802.3ad --bootproto=static --gateway=10.0.1.1 -
-nameserver=10.0.1.254,20.1.1.254 --activate
network --device=bond2 --bondslaves=ens3f4,ens3f5 --bondopts=mode=802.3ad --bootproto=static --noefroute --
activate
text
reboot

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

Note

Instructions on obtaining the squashfs.img may be found at https://www.tldp.org/HOWTO/html_single/SquashFS-HOWTO/

Editing and running the Ansible playbooks

The remaining host configuration is handled via Ansible playbooks. If you have not already done so, clone the repository from GitHub to your Ansible Engine host. The repository is located at <https://github.com/hewlettpackard/hpe-solutions-openshift/>. To clone it, run the following command:

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

Configuration of the virtualization hosts is handled by the Nimble RHVH (nrhvh) and Nimble RHVH hosts (nrhvhosts) roles within the repository. These roles and their accompanying plays consist of the following files:

Root directory

- hosts - this file contains the definition of the hosts that will be used within the solution as well as variables for networking and iSCSI iqn.
- nrhvh.yaml - this file is used to run the plays contained within the Nimble RHVH role.
- nrhvhosts.yaml - this file is used to run the plays contained within the Nimble RHVH hosts role.



Roles files directories (<root>/roles/<role name>/files)

- <root>/roles/nrhvhosts/files
 - multipath.conf - a default multipath.conf file to use with the virtualized hosts.

Roles tasks directories (<root>/roles/<role name>/tasks)

- <root>/roles/nrhvh/tasks
 - main.yaml - this file defines the order that the various plays are run in.
 - hostname.yaml - this file calls the hostname.j2 template to configure the hostname for each virtualization host and edits the contents of /etc/hosts.
 - hostreg.yaml - this file registers the host and attaches pools. It pulls user credentials from the vault file.
 - repos.yaml - this file disables all repositories and then enables the required repository.
 - yumtasks.yaml - this file updates the hosts.
 - mgbond.yaml - configures the management network bond for the host. This file uses variables defined in both the hosts and vars/main.yaml files.
- <root>/roles/nrhvhosts/tasks
 - dcbond.yaml - configures the data center bond for the host. This file uses variables defined in both the hosts and vars/main.yaml files.
 - detach.yaml - the initial operations that Ansible performs require it to connect using the PXE network. This final file disconnects the network interface on the PXE network and then reboots the hosts.
 - iqns.yaml - this file sets the IQN of the host and uses the templates/iqn.j2 template file.
 - iscsi.yaml - this file configures and activates the NICs used for iSCSI connectivity within the environment and pulls information from the hosts file.
 - main.yaml - this file defines the order that the various plays are run in.
 - services.yaml - this file starts and enables the Docker service.

Roles templates directories (<root>/roles/<role name>/templates)

- <root>/roles/nrhvh/templates
 - hostname.j2 - this file uses the hostname from within the hosts file to configure each host.
- <root>/roles/nrhvhosts/templates
 - iqns.j2 - this template file overwrites the /etc/iscsi/initiatorname.iscsi file and appends the host name to the IQN string.

Roles variables directories (<root>/roles/<role name>/vars)

- <root>/roles/nrhvh/vars
 - main.yaml - this file contains variables required to complete the tasks for the role.
- <root>/roles/nrhvhosts/vars
 - main.yaml - this file contains variables required to complete the tasks for the role.

The installer should configure variables within the hosts file and within <root>/roles/<role name>/vars/main.yaml files prior to running the /<root>/nrhvh.yaml play.



Note

To find Pool IDs for the Red Hat Virtualization Host, execute the following command:

```
# subscription-manager list --available --matches '*Virtualization*'
```

Once the files have been configured, run the following plays in order to finalize host configuration:

```
# ansible-playbook -i hosts playbooks/nrhvh.yaml --ask-vault-pass
# ansible-playbook -i hosts playbooks/nrhvhosts.yaml --ask-vault-pass
```

Verifying Red Hat Virtualization Host

This section outlines the steps to verify Red Hat Virtualization Host setup:

- 1. Using a browser, open the Cockpit web interface at <https://<ipaddress>:9090>. Use the IP of Bond1. Log on as the root user. Figure 16 displays the console that is presented upon successful login.

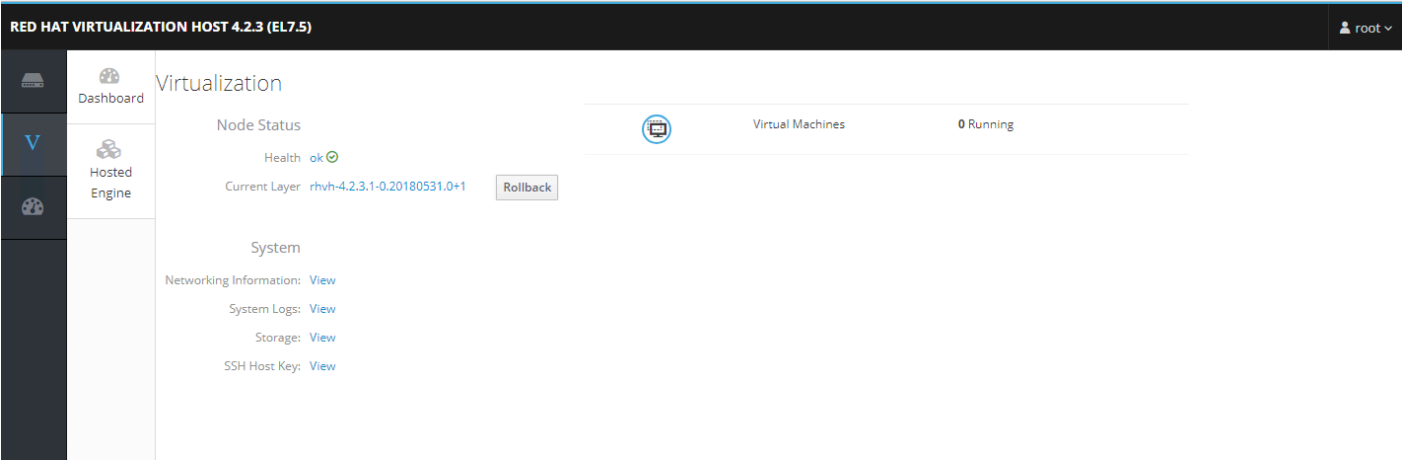


Figure 16. Individual host web interface

- 2. From the left navigation bar, select **Localhost > Networking**. Figure 17 shows the list of bond IPs. This should align to the settings that is configured in the Ansible playbooks.

Interfaces				Add Bond	Add Team	Add Bridge	Add VLAN
Name	IP Address	Sending	Receiving				
bond1	10.0.9.7/16	16.0 Kbps	7.51 Kbps				
bond2	20.0.9.7/8	0.001 bps	0.001 bps				
ens3f0		Inactive					
ens3f1		Inactive					
ens3f6	30.0.9.7/16	0.001 bps	0.001 bps				
ens3f7	40.0.9.7/16	0.001 bps	0.001 bps				

Figure 17. Networking screen for the virtualized host



Create a Nimble data access group

After running the Ansible play (`nrhvhhosts.yml`), RHVH IQN will be changed. Run the following command to fetch the IQN details of each RHVH host:

```
# cat /etc/iscsi/initiatorname.iscsi
```

1. Log in to the primary HPE Nimble Storage array using the admin account.
2. Create an initiator group by providing names of the hosts and their IQN details as in **Error! Reference source not found.**
 - a. Choose **Manage > Data Access**.
 - b. Select **+Add**.
 - c. Select **Create Initiator group** and fill in the details using the IQNs recorded for the RHVH hosts as in Figure 18.

CREATE INITIATOR GROUP

NAME *

SUBNETS

INITIATORS

NAME	IQN	IP ADDRESS
<input type="text" value="buranrhvh01"/>	<input type="text" value="iqn.1994-05.com.redhat:buranrhvh01"/>	<input type="text" value="*"/>
<input type="text" value="buranrhvh02"/>	<input type="text" value="iqn.1994-05.com.redhat:buranrhvh02"/>	<input type="text" value="*"/>
<input type="text" value="buranrhvh03"/>	<input type="text" value="iqn.1994-05.com.redhat:buranrhvh03"/>	<input type="text" value="*"/>

Figure 18. Create an initiator group

3. Create two volumes on the array, one for the hosted engine and the other for the virtual machines:
 - a. Select **Manage > Data Storage**.
 - b. Click the **+** icon.
 - c. Create the volume by providing the required details such as name, storage size, and so on.
 - d. Click **Create**.

Note

The values for storage volumes are specific to the individual environment.

4. Ensure that the service start request does not return an error. Run the following commands to connect to the HPE Nimble Storage:

```
# iscsiadm -m discovery -t sendtargets -p <Nimble storage discovery IP address - 1>
# iscsiadm -m node -T <initiator details obtained from above command> -p <Nimble storage discovery IP address - 1>--login
# iscsiadm -m discovery -t sendtargets -p <Nimble storage discovery IP address - 2>
# iscsiadm -m node -T <initiator details obtained from above command> -p <Nimble storage discovery IP address - 2>--login
```



Configuring the virtualization environment

Once the three hosts have been configured as above, the virtualization environment will need to be configured and a cluster created across the three RHVH servers using the shared storage presented from the HPE Nimble Storage AF40. 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 cockpit user interface. Figure 19 shows cockpit user interface.

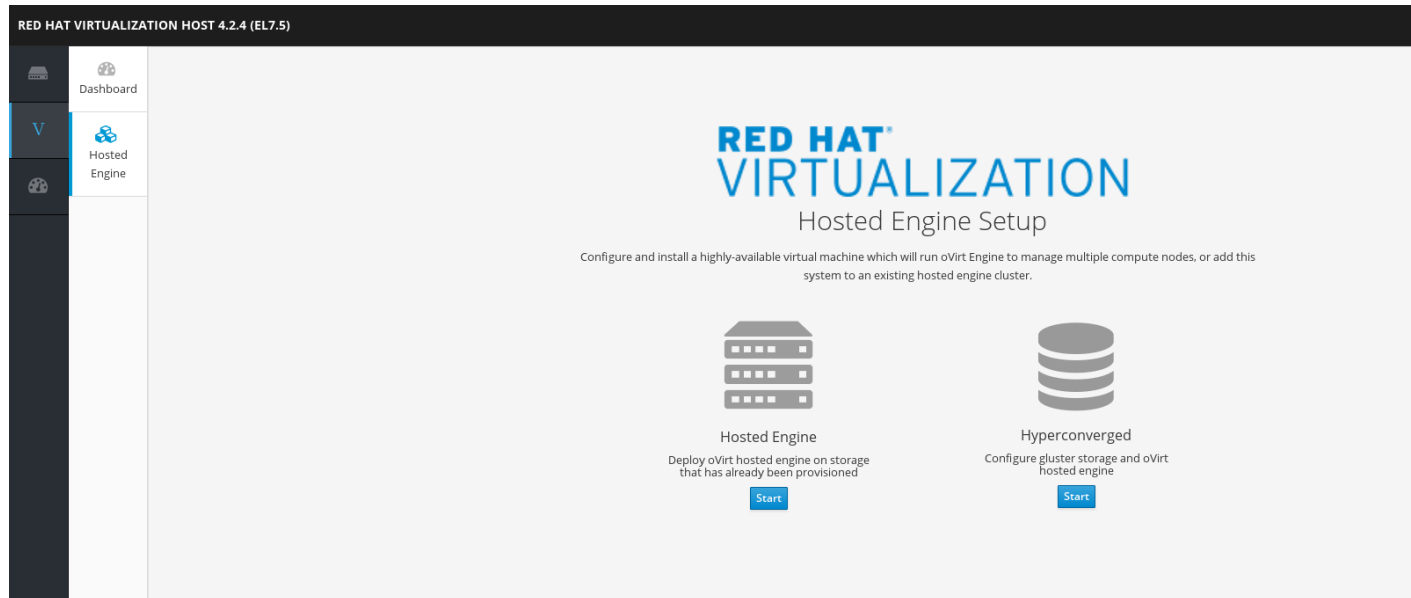


Figure 19. Hosted engine setup screen of the cockpit interface

1. If not logged on, reconnect to the cockpit user interface using the instructions in the prior section. Log into the cockpit user interface with the host's local account credentials.

Note

Installing Red Hat Virtualization Manager should be performed only on one Red Hat Virtualization Host.

2. Select the **Hosted Engine Start** icon to begin the Hosted Engine deployment. Hosted Engine deployment is divided into two sections:
 - a. Preparing the hosted engine virtual machine.
 - b. Preparing the hosted engine storage.



3. Complete the hosted engine deployment configuration items as in Figure 20. Once all values have been entered, click **Next**.

The screenshot shows the 'Hosted Engine Deployment' window with a progress bar at the top indicating five steps: VM (1), Engine (2), Prepare VM (3), Storage (4), and Finish (5). The 'VM' step is currently active. Below the progress bar, the 'VM Settings' section contains the following fields:

- Engine VM FQDN:
- MAC Address:
- Network Configuration:
- VM IP Address:
- Gateway Address:
- DNS Servers:
- Bridge Interface:
- Root Password:
- Root SSH Access:
- Number of Virtual CPUs:
- Memory Size (MiB): 254,716 MiB available

At the bottom right of the window are three buttons: 'Cancel', '< Back', and 'Next >'.

Figure 20. Hosted engine deployment configuration example

4. On the Engine screen, enter the engine credentials for your environment and click **Next**.
5. On the Prepare VM screen, review the summary and click **Prepare VM**.
6. 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 **Next**.
7. On the Storage screen, choose **iSCSI** as the storage type, and enter the management IP address of the HPE Nimble Storage along with the portal username and portal password. When complete, click **Retrieve Target List**.



8. Select a target from the paths shown. When the list of LUNs (volumes) is returned, select the volume you created to house the hosted engine as in Figure 21 and click **Next**.

Hosted Engine Deployment

VM

Engine

Prepare VM

Storage

Finish

1

2

3

4

5

Storage Type

iSCSI

Portal IP Address

10.0.41.95

Portal Port

3260

Portal Username

admin

Portal Password

.....

Retrieve Target List

The following targets have been found:

iqn.2007-11.com.nimblestorage:rhvhcluster-v228206f1c6a6b33f.00000004.e424b5ce, TPGT: 2460

30.0.0.10:3260

40.0.0.10:3260

iqn.2007-11.com.nimblestorage:rhvmvolume-v228206f1c6a6b33f.000000d3.e424b5ce, TPGT: 2460

30.0.0.10:3260

40.0.0.10:3260

The following luns have been found on the requested target:

ID: 2292bf3c9c1377c286c9ce900ceb524e4

Size (GiB): 150.00

Description: Nimble Server

Status: free

Number of Paths: 2

Cancel

< Back

Next >

Figure 21. Storage selection example

9. Review Summary and click **Finish Deployment**. The process will take some time. Do not interrupt or shut down the system.

A green rectangular box, likely a placeholder for a logo or additional information.

Configuring hosted engine

1. Once the hosted engine deployment has successfully completed, log in to the Red Hat Virtualization Administration Portal at <https://<hosted engine FQDN>>, using the credentials supplied during the hosted engine deployment in the previous section. Figure 22 shows the **Welcome** screen.

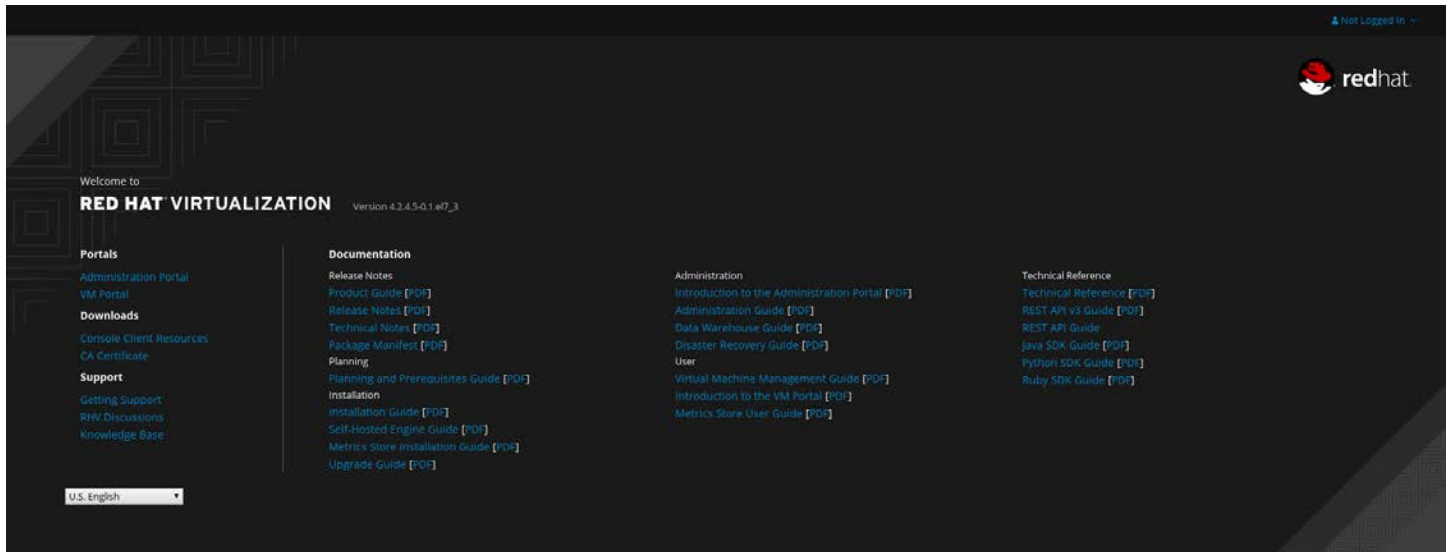


Figure 22. Red Hat virtualization welcome screen

2. Click **Administration Portal** under the Portals link. Log in to the portal using the username and password configured during the deployment.
3. Add Additional RHVH hosts to the hosted engine using the Red Hat Virtualization Administration Portal.

The following steps are required to configure the hosted engine and create the RHV cluster:

- a. Create an account on each iLO to facilitate power management.
- b. Add additional RHVH hosts to the cluster.
- c. Configure host networking.
- d. Configure cluster storage.

Create an account on each iLO to facilitate power management

RHV power management facilitates the powering down and powering up of the servers through the Red Hat Virtualization Administration Portal. This functionality is enabled via HPE iLO.

1. Log in to HPE OneView.
2. Choose the node on which the hosted engine is deployed.
3. Select **iLO Address**.
4. From the left navigation menu, select **Administration**.
5. Under **User Administration**, click **New**.



6. Create an account by providing the user credentials and user privileges. The privileges can be limited to Login and Virtual Power. Also, enable the **Service Account** checkbox and then select **Add User**. Figure 23 shows the account creation process.

The screenshot shows the HPE iLO 5 Administration interface. The left sidebar contains navigation links for Information, System Information, Firmware & OS Software, iLO Federation, Remote Console & Media, Power & Thermal, iLO Dedicated Network Port, Remote Support, Administration (selected), Security, Management, Synergy Frame, Intelligent Provisioning, and HPE OneView. The main content area is titled 'Administration - User Administration' and contains a sub-tab 'Add/Edit Local User'. The form includes fields for 'Login Name' (rhvadmin) and 'User Name' (rhvadmin). There is a checkbox for 'Change password' and a checked checkbox for 'Service Account'. Under 'User Privileges', there are checkboxes for 'select all', 'Login' (checked), 'Remote Console', 'Virtual Power and Reset' (checked), 'Virtual Media', 'Host BIOS', 'Configure iLO Settings', 'Administer User Accounts', 'Host NIC', 'Host Storage', and 'Recovery Set'. At the bottom, there is a field for 'IPMI/DCMI Privilege based on above settings:' with the value 'user'.

Figure 23. HPE iLO administration – create an user for power management

Adding additional RHVH hosts to the Hosted Engine using the Red Hat Virtualization Administration Portal

Prior to completing this section, the installer should create a user on the iLO of each virtualized host. For more information, see [Create an account on each iLO to facilitate power management](#). Note the IP address or FQDN of each iLO.

The following steps are required to configure the hosted engine and create the RHV cluster:

- Add additional RHVH hosts to the cluster. This solution will add two (2) additional hosts for a total of three (3).
- Configure host networking.
- Configure cluster storage.

Adding additional hosts

1. In the Red Hat Virtualization Administration Portal, select **Compute > Hosts > New**.
2. On the **General** tab, complete the following fields:
 - a. Host Cluster
 - b. Name
 - c. Hostname
 - d. Password
3. On the **Power Management** tab, complete the following:
 - a. Select **Enable Power Management**.



b. Select **+** under **Add Fence Agent** and configure the following parameters as in Figure 24.

- **Address** – IP the address of the iLO.
- **User Name** – the login name of the user that was created.
- **Password** – enter the password for the named user.
- **Type** – select ilo_ssh.

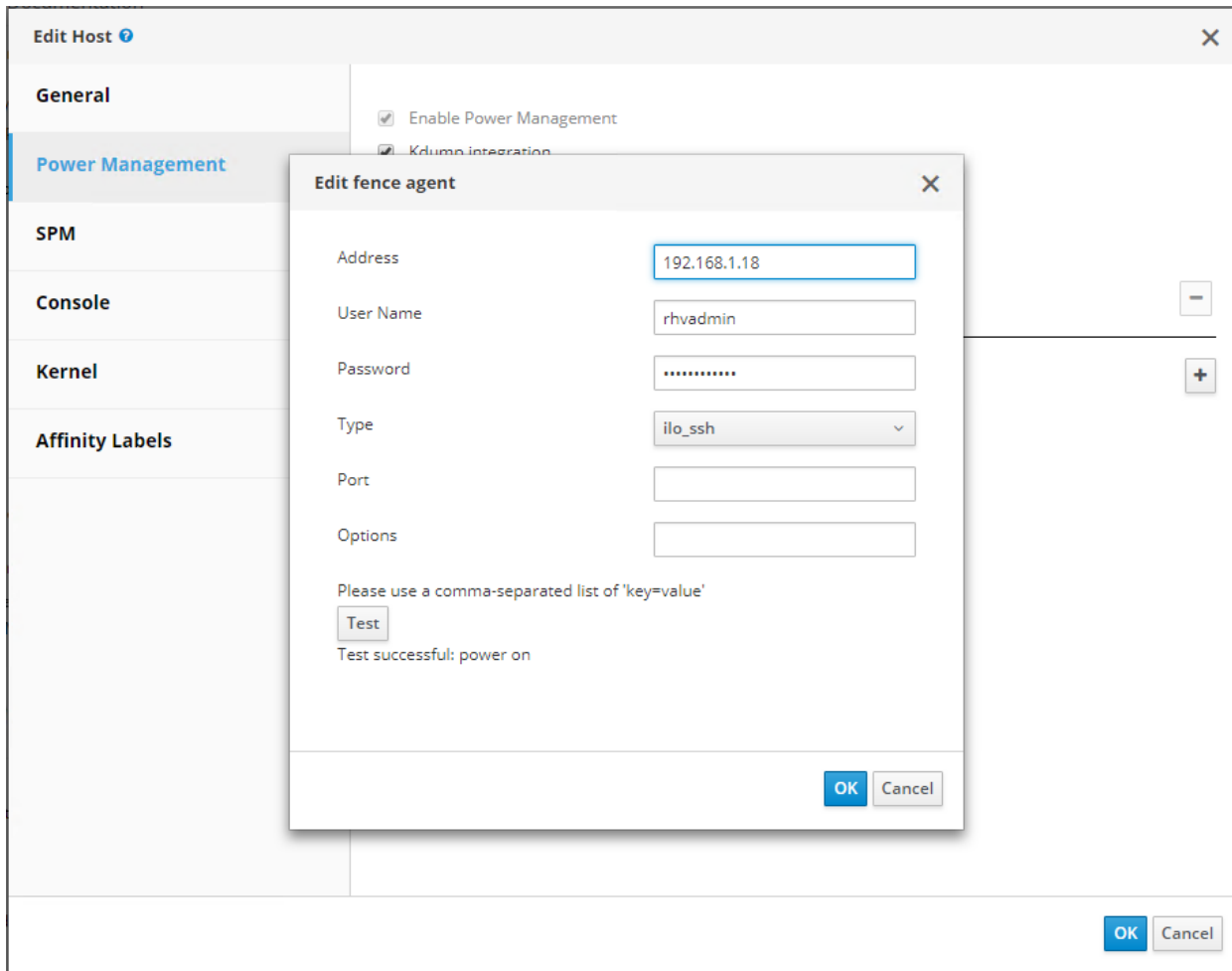


Figure 24. Red Hat Virtualization Host power management configuration

4. Click **Test**.

5. Once the test is successful, click **OK**.



6. On the **Hosted Engine** tab, from the **Choose hosted engine deployment action** drop-down list, select **Deploy**. This enables the hosted engine to run on the new host. Figure 25 illustrates the three RHVH hosts added to the default cluster. Note that the small crown icon in the second column indicates if a host is capable of running the hosted engine VM. A gold crown icon indicates the host that is currently running the hosted engine VM.

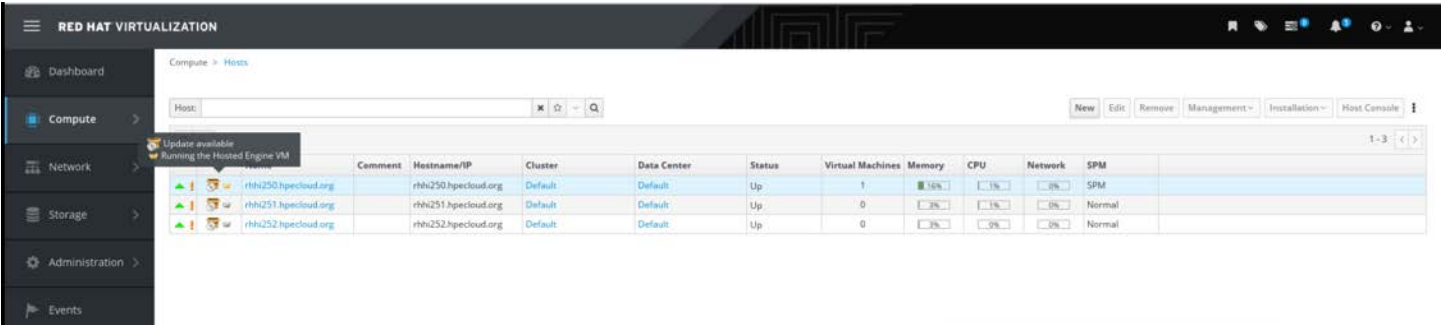


Figure 25. Virtualized hosts as added to the default cluster.

7. Once hosts are added, select **Network > Networks > New**.

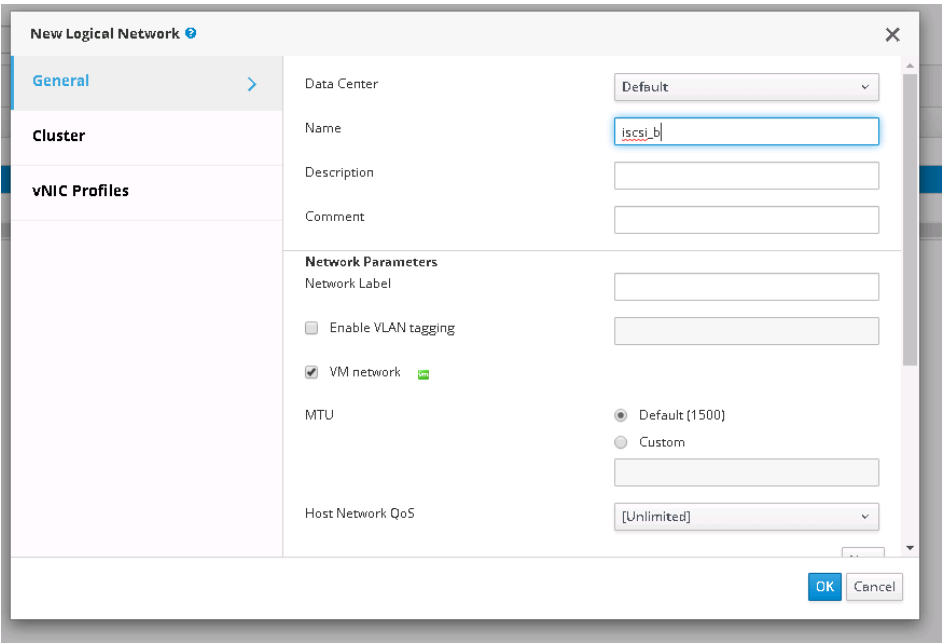


Figure 26. Adding a new logical network

8. On the **General** tab, one at a time, add your data center and iscsi networks ensuring to provide labels and descriptions as in Figure 26.
9. Select **Compute** and then select **Hosts**.



10. Select each host one at a time and for each host select **Network Interfaces** as in the Figure 27.

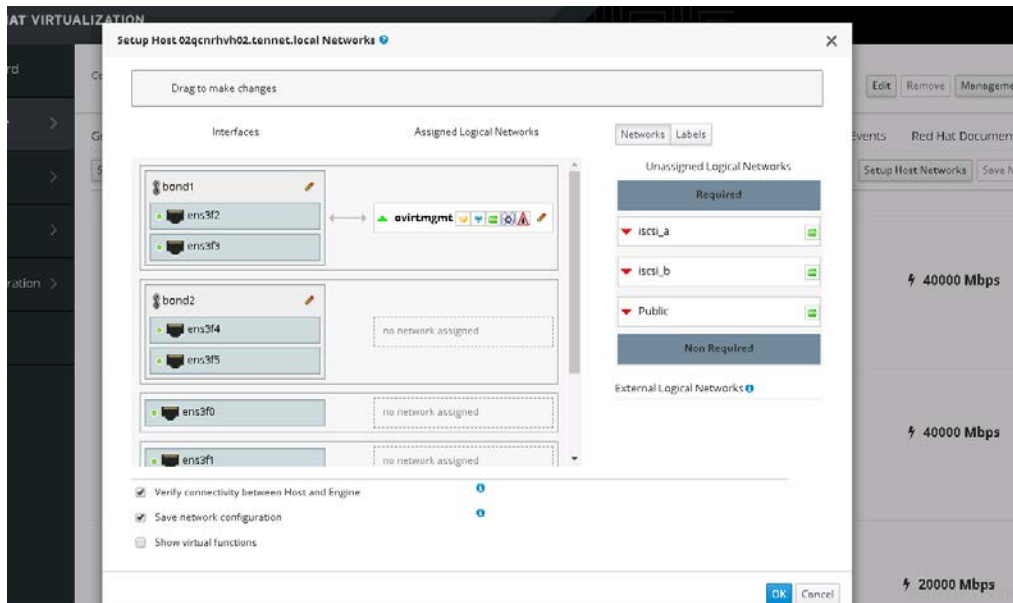


Figure 27. Selecting network interfaces

11. From the Network Interfaces screen, select **Setup Host Networks**.
12. Drag the network labels you created onto the appropriate interface on each host.
13. When complete, select **OK**.

Red Hat OpenShift worker nodes

OS installation and configuration with HPE Synergy Image Streamer

The worker nodes will be deployed and customized using HPE Synergy Image Streamer while ensuring that the required storage volumes are created and attached. This section outlines the steps required to install the host. At a high level, these steps can be described as:

1. Download the artifacts for HPE Image Streamer from the HPE GitHub site.
2. Add the artifact bundles to HPE Image Streamer.
3. Build a worker node image.
4. Capture a Golden Image.
5. Copy and edit the NIC teaming plan script.
6. Copy and edit the OS Build Plan.
7. Create a deployment plan.
8. Deploy the hosts.
9. Post-deployment Ansible configuration.

Download the artifacts for HPE Synergy Image Streamer

Red Hat Enterprise Linux bundles for HPE Image Streamer may be downloaded from <https://github.com/HewlettPackard/image-streamer-rhel/>. Sample foundation artifact bundles should be downloaded from <https://github.com/HewlettPackard/image-streamer-tools/tree/v4.2/foundation/artifact-bundles>. The artifacts utilized in the creation of this solution may be found at <https://github.com/HewlettPackard/image-streamer-rhel/tree/v4.2/artifact-bundles>.



Add the artifact bundles to HPE Synergy Image Streamer

1. From within the HPE Synergy Image Streamer interface, navigate to the **Artifact Bundles** page.
2. Select **Add artifact bundle**, and add the downloaded RHEL artifact bundle. If not already present, add the sample foundation bundle.
3. From the **Actions** menu, select **Extract** to extract the artifacts from each downloaded bundle.

Build a worker node image

In order to create a worker node image, follow the steps below:

1. Create a temporary server profile and attach it to a host. Adjust the following settings:
 - a. Create a 40 GB (40960 MB) Image Streamer volume that will be used to install RHEL by selecting **HPE-Create Empty volume** from the OS deployment plan drop-down list.
 - b. Configure the connections in the following order:
 - Deployment1 - 1GB primary boot
 - Deployment2 - 1GB secondary boot
 - Management_A - 2GB
 - Management_B - 2GB
 - Datacenter_A - 9GB
 - Datacenter_B - 9GB
 - ISCSI_A - 8GB
 - ISCSI_B - 8GB
 - c. Configure the boot settings:
 - UEFI Optimized for Boot mode
 - Secure boot disabled
 - Manage boot order
 - Primary boot device: Hard disk
 - d. Once complete, apply the profile to a single host and click **Create**.
2. From the **Actions** drop-down list found on the server profile's OneView page, select **Launch console**.
3. Attach the RHEL 7.6 image and select **Power Switch > Momentary Press**.
4. Check your media and then select “**e**” at the Install Red Hat Enterprise Linux screen.



5. Append the following to the Install kernel boot parameter: **rd.iscsi.ibft=1** as shown in Figure 28.

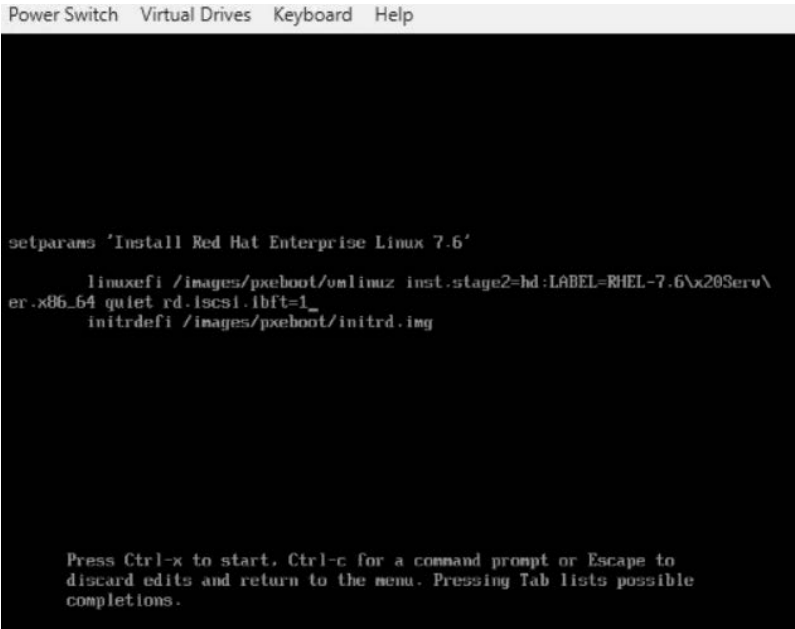


Figure 28. Editing the kernel boot parameter

- 6. Press **Ctrl-X** to boot the system.
- 7. Fill in the appropriate language and localization settings ensuring you set your time zone to align with your environment practices.
- 8. Click **Installation Destination** to choose the appropriate location to install the operating system.
- 9. On the **Installation Destination** screen, click **Add a disk** as in Figure 29.

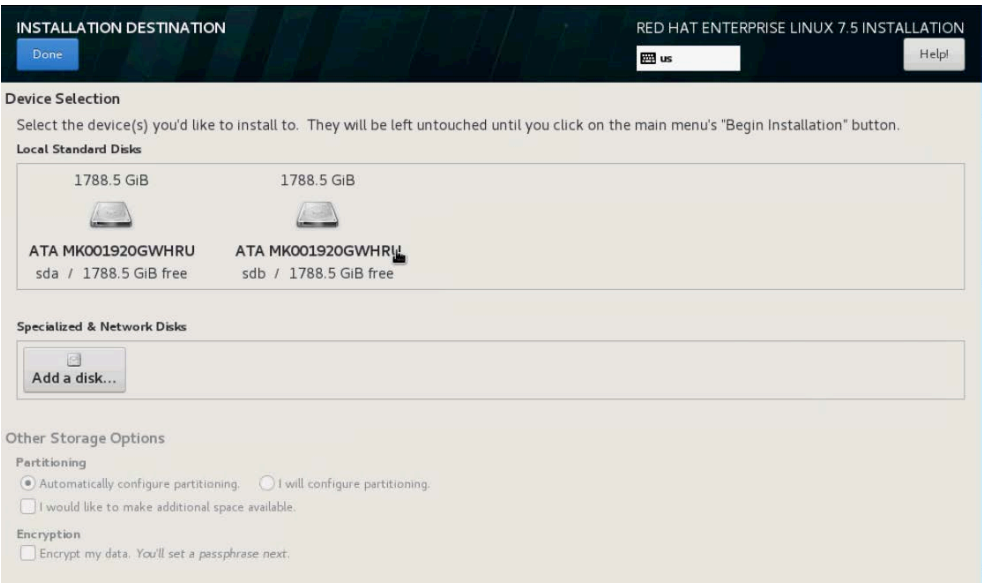


Figure 29. Installation Destination screen



10. Select the **Image Streamer disk** as in Figure 30 and then click **Done**.

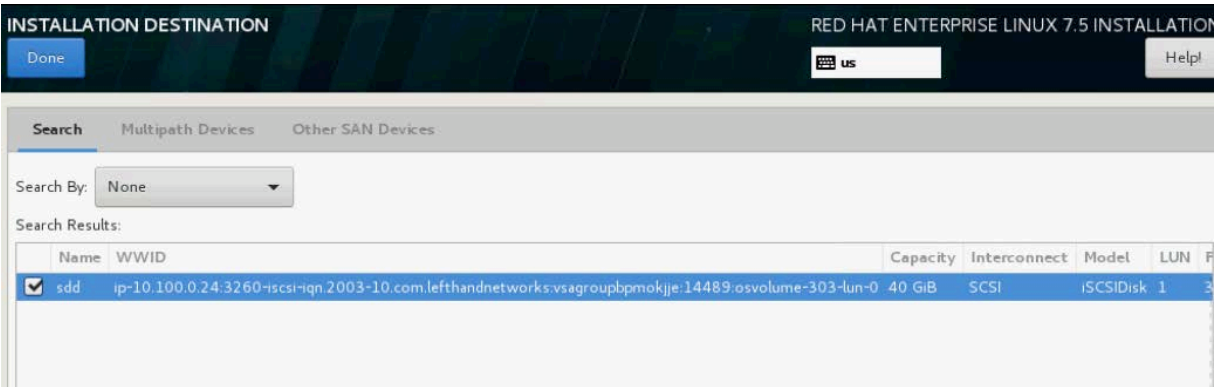


Figure 30. The Image Streamer volume as seen from within the Installation Destination screen

- 11. Once back on the Installation Destination screen, select **I will configure Partitioning** and then click **Done**.
- 12. On the Manual Partitioning screen, select **Click here to create them automatically** as shown in Figure 31.

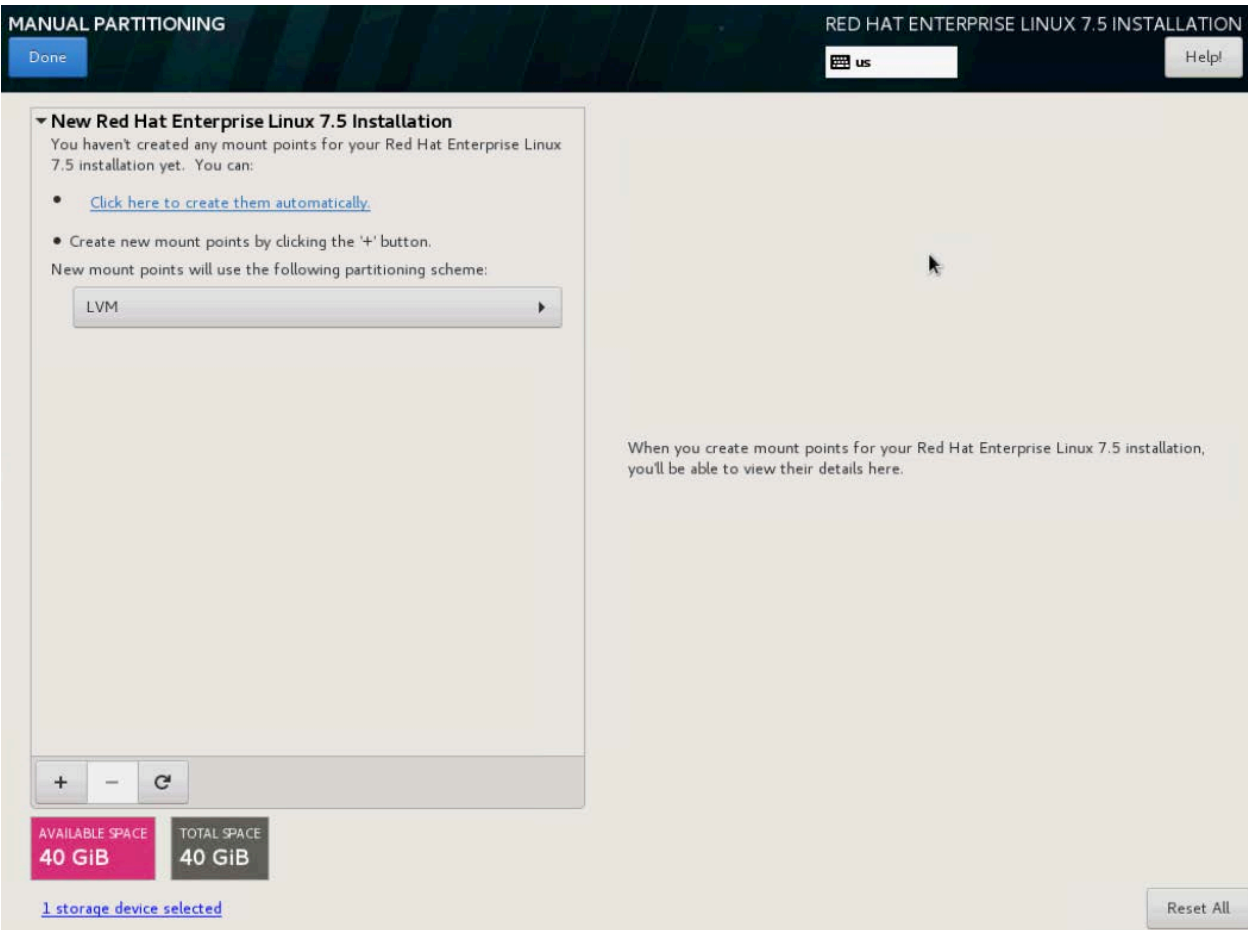


Figure 31. Manual partitioning screen



13. Create or alter the partitions as follows:
- a. /boot 1024MiB ext4
 - b. / 18.8 GiB ext4
 - c. Click the + icon and add the following:
 - /var - Leave capacity empty.
 - Change to ext4 once complete.

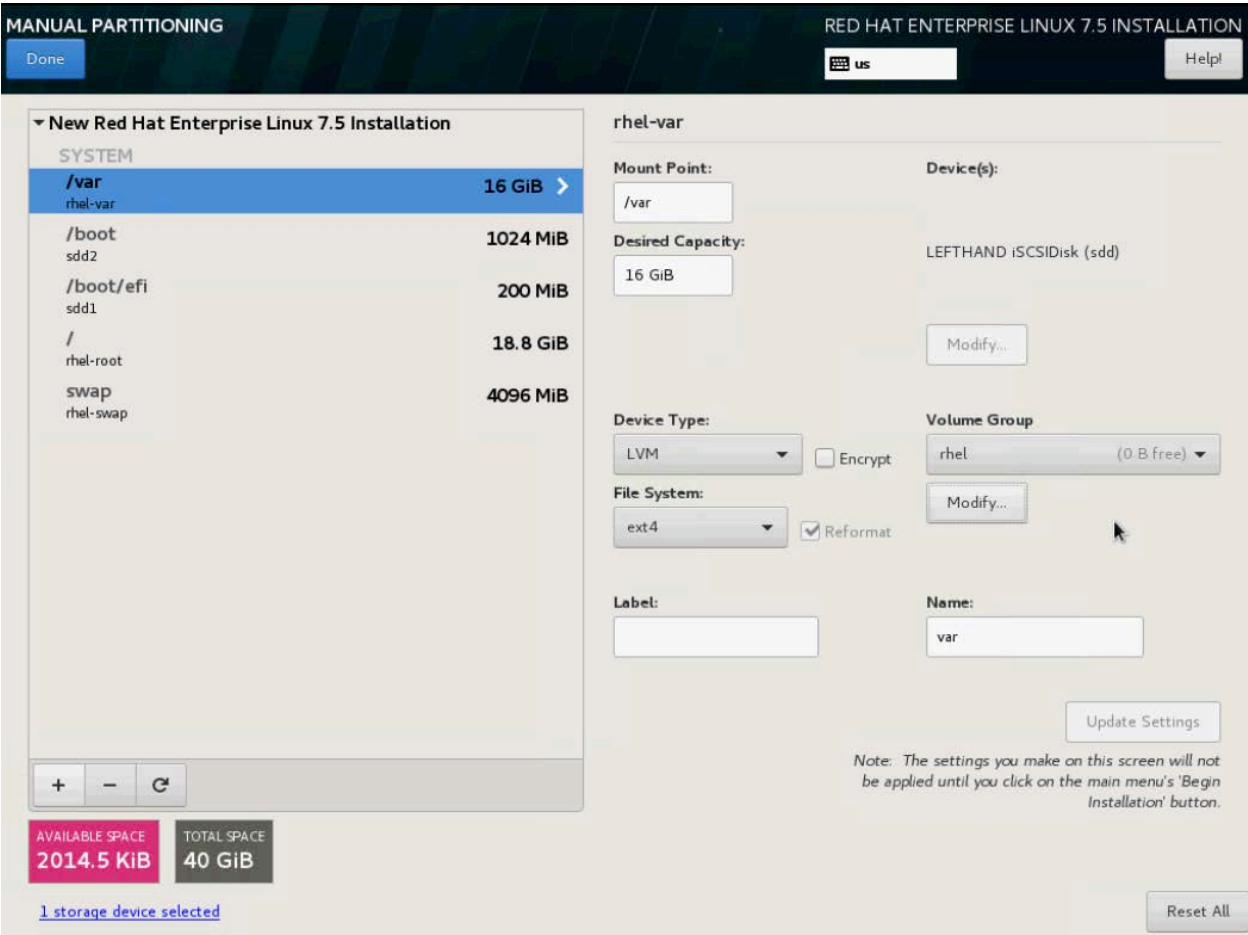


Figure 32. Partitions used during installation

14. Click **Done** and then click **Accept Changes**.
15. Select **Network & Hostname**.



16. Select **Ethernet (ens3f2)** and switch it to **ON**. Provide a temporary name for the host and click **Apply**. When complete, click **Done** as in Figure 33.

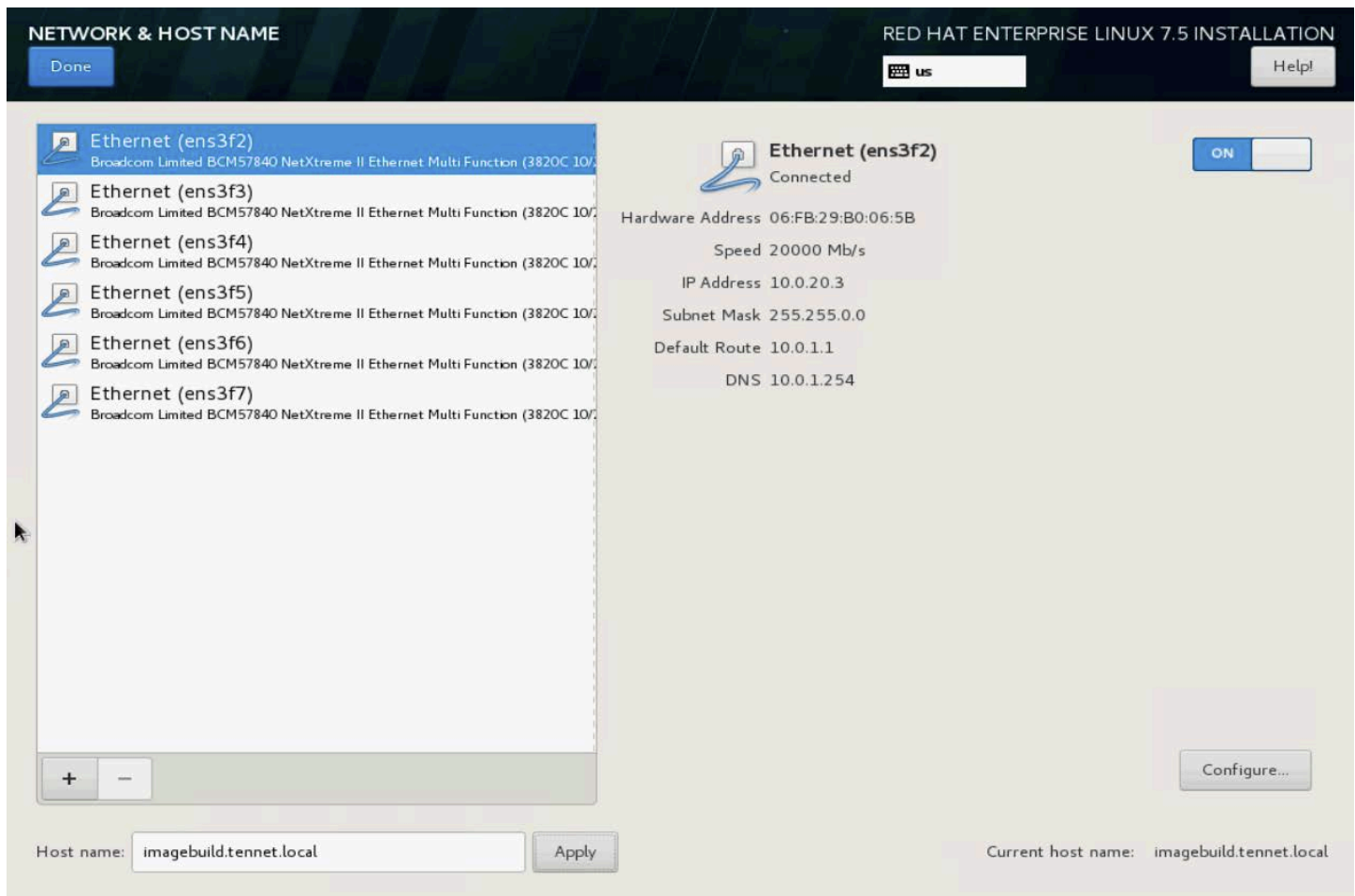


Figure 33. Network & Host Name screen.

17. Click **Begin Installation**. While waiting for the installation to complete, provide a root password and optionally create a user account (the Ansible plays will create an additional user account later).
18. Once the installation is complete, you can reboot the host by selecting **Reboot** when prompted.
19. Once the system is rebooted, log in as root and type in the following:

```
# subscription-manager register
```

20. Enter your RH username and password when prompted.
21. Attach the subscriptions for RHEL and Red Hat Cloud Infrastructure to enable the host to be updated. For each subscription, execute the following command:

```
# subscription-manager --attach-pool=<your-pool-id>
```

22. Disable all repositories and then enable the required repository by running the following commands:

```
# subscription-manager repos --disable=*
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-extras-rpms
# subscription-manager repos --enable=rhel-7-server-ose-3.11-rpms
```



```
# subscription-manager repos --enable=rhel-7-fast-datapath-rpms
# subscription-manager repos --enable=rhel-7-server-ansible-2.7-rpms
```

23. Install teamd by running the following command:

```
# yum install teamd -y
```

24. Update the host by running the following command:

```
# yum update -y
```

25. Reboot the host by typing the following:

```
# reboot
```

26. Once the host has restarted, log in as the root user and open the appropriate firewall ports by running the following commands:

```
# firewall-cmd --add-port=3260/tcp --permanent
# firewall-cmd --add-port=4789/udp --permanent
# firewall-cmd --add-port=8053/udp --permanent
# firewall-cmd --add-port=8053/tcp --permanent
# firewall-cmd --add-port=443/tcp --permanent
# firewall-cmd --add-port=8443/tcp --permanent
# firewall-cmd --add-port=10250/tcp --permanent
# firewall-cmd --add-port=53/tcp --permanent
# firewall-cmd --add-port=53/udp --permanent
# firewall-cmd --reload
```

27. Verify your settings by running the following command:

```
# firewall-cmd --list-all
```

28. Ensure you remove subscriptions from the host and unregister it by running the following command:

```
# subscription-manager unregister
```

29. Run the following command to shut down the host in preparation for image capture:

```
# shutdown -h now
```

Capture a golden image

You will utilize the build plan **RHEL-generalize-2018-06-27** to capture the golden image from the host you just created. To do this, follow these steps:

1. From within HPE OneView, highlight the server profile of the system just installed and scroll in the window until you see **OS volume**. Record the ID of the volume.
2. From the **OneView** drop-down list, select **OS Deployment Servers** and launch the **Image Streamer UI** by selecting the link as shown in Figure 34.

✓ **Image Streamer** | General ▾ | 🔍

General

State	Connected
Type	Image Streamer
Image Streamer UI	10.0.0.31
Description	Deployment

Figure 34. Image Streamer launch link from within HPE OneView



- 3. From within the Image Streamer UI, select **Golden Images > Create golden image**.
- 4. Create the golden image as in Figure 35:
 - a. **Name** – Provide a name for the golden image.
 - b. **Description** – Provide a short description about the golden image.
 - c. **OS volume** – Select the **OS volume**.
 - d. **Capture OS build plan** – Use the **RHEL-generalize-2018-06-27** OS build plan.
 - e. Click **Create**.

Create Golden Image?

Name

Nimble Worker Image

Description

Image for Nimble Worker Nodes

OS volume

OSVolume-336

x

Capture OS build plan

HPE-Capture-RHEL75-Image-2018-07-31

x

4

Changed: Description to "Image for Nimble Worker Nodes"

Create

Create +

Cancel

Figure 35. Golden image creation screen

- 5. Once the image is successfully captured, return to the OneView interface and delete the server profile from the host used to capture the image. Ignore the warning that the volume will be deleted. This deletes the original volume used for installing the operating system and does not remove the golden image.



Copy and edit the OS build plan

- 1. From within the Image Streamer UI, select **OS Build Plans** from the **Image Streamer** drop-down list.
- 2. Select the **RHEL-personalize-and-NIC-teamings-LVM-BP-2018-09-11** build plan.
- 3. From the **Actions** menu, select **Copy**.
- 4. Assign a name to the new plan. It is recommended to add the current date or another unique identifier that will help quickly identify the copy being used.
- 5. Under Custom attributes, select **Team1NIC1** and select the edit icon. Uncheck the box **Allow no network connection** and click **OK** as shown in Figure 36. Repeat this process for **Team1NIC2**.

Edit Custom Attribute Team1NIC1

TypeNIC

Description

The mac sub-attribute is a mandatory field in NIC type custom attributes.

optional

IPv4 configuration

☒ Allow static

☒ Allow DHCP

☐ Allow no network connection

Interface configuration parameters

Team1NIC1.dns1

Team1NIC1.dns2

Team1NIC1.domain

Team1NIC1.gateway

Team1NIC1.ipaddress

Team1NIC1.mac

Team1NIC1.netmask

Team1NIC1.vlanid

OK

Cancel

Figure 36. Edit custom attribute for OS Build Plan screen

- 6. Edit TotalNICTeamings under Custom Attributes and change the **Default value** to 2.
- 7. Click **OK** when complete.



Create a deployment plan

The following steps should be used to create a deployment plan that will be used to deploy the worker nodes:

- 1. From the Image Streamer interface, select **Image Streamer > Deployment Plans > Create Deployment Plan**.
- 2. On the **Create Deployment Plan** screen, complete the following using Figure 37 as an example:
 - a. **Name** – Enter a name for the deployment plan.
 - b. **Description** – Provide a short description for the deployment plan.
 - c. **OS build plan** – Select the OS build plan that you created earlier.

Create Deployment Plan ?

General

Name

Nimble Worker Deployment 07262019

Description

Plan Attributes

OS build plan

Nimble Worker Build Plan 07262019

Custom attributes

Name	Type	Constraint	Visible on deployment	Value
DiskName	String	options:/dev/sda	<input checked="" type="checkbox"/>	/dev/sda
DomainName	FQDN	options:	<input checked="" type="checkbox"/>	
FirstPartitionSize	Number	options:10	<input checked="" type="checkbox"/>	10
HostName	Hostname	options:	<input checked="" type="checkbox"/>	
LogicalVolumeGroupName	String	options:new_vol_group	<input checked="" type="checkbox"/>	new_vol_group

Changed: OS build plan to "Nimble Worker Build Plan 07262019"

Create

Create +

Cancel

Figure 37. Create deployment plan screen from within image streamer

- 3. From the **Golden Image** drop-down list, select the golden image that you already created. For more information on creating the golden image, see [Capture a golden image](#).
- 4. Click **Create** to finish creating the deployment plan.



Create the server profile template

- 1. From the **OneView** drop-down list, select **Server Profile Templates > Create server profile template**.
- 2. Assign a name and description to the profile and then select the appropriate **Server hardware type** and **Enclosure group**.
- 3. Select the **OS Deployment plan** you created from the drop-down list.
- 4. Scroll to **Connections** and add the following connections in order (you will see two deployment network connections in place prior to adding these networks). Click **Add+** until the final network and then click **Add**.
 - a. Management_a, ethernet, select your management network, 2Gb.
 - b. Management_b, ethernet, select your management network, 2Gb.
 - c. Datacenter_a, ethernet, select your dc net, 9Gb.
 - d. Datacenter_b, ethernet, select your dc net, 9Gb.
 - e. Iscsi_a, ethernet, iSCSI_SAN_A for the network, 8Gb.
 - f. Iscsi_b, ethernet, iSCSI_SAN_B, 8Gb.
- 5. Edit the **deployment connections** and set the **bandwidth** to either 1Gb or 2Gb as shown in Figure 38.

Edit Connection

General

Name

Management_a

Function type

Ethernet

Network

TenNet

x

🔍

Port

Auto

x

🔍

Link aggregation group

None

x

🔍

Requested bandwidth (Gb/s)

2

Requested virtual functions

☒ None

☐ Custom

☐ Auto

Boot

Not bootable

Figure 38. Server profile template screen



- 6. Return to the deployment settings and customize the following settings.
 - a. Enter a **NewRootPassword** and confirm it.
 - b. Create a new, **non-root user** and **password**.
 - c. For **Team0NIC1** select **Management_a** and select the radio button **User-specified**. Fill in the network information requested.
 - d. For **Team0NIC2** select **Management_b** and select the radio button **User-specified**.
 - e. For **Team1NIC1** select **Datacenter_a** and select the radio button **User-specified**. Fill in the Datacenter network details.
 - f. For **Team0NIC2** select **Datacenter_b**. and select the radio button **User-specified**.
- 7. Leave **HostName** blank.
- 8. Fill in any RBSU customizations and then select **Create**.

Deploying the worker nodes

- 1. To deploy a new server profile from template to the worker node, navigate to **Server Profile** from OneView and select **Create profile**. Ensure that the deployment plan that you created is selected from the drop-down list in the **OS Deployment** portion of the server profile.
- 2. Fill in the remaining settings as in Figure 39, inserting appropriate values. For **IPv4 address**, you should enter management network and Datacenter network IPs. Enter the hostname. When complete, click **Create**.

Create Server Profile

OS Deployment

?

OS Deployment

Team0NIC1

Management_a

IPv4 configuration

☐ DHCP

☒ User-specified

IPv4 address

10.0.9.3

Netmask

255.255.0.0

Gateway

10.0.11

DNS 1

10.0.1.254

DNS 2

20.0.1.254

Domain

tennet.local

MAC address

pending assignment

Team0NIC2

Management_b

IPv4 configuration

☐ DHCP

☒ User-specified

MAC address

pending assignment

Team1NIC1

Datacenter_a

IPv4 configuration

☐ DHCP

☒ User-specified

IPv4 address

20.0.9.3

Netmask

255.0.0.0

Gateway

20.1.1.1

Create

Create +

Cancel

Figure 39. Create server profile

- 3. Repeat the steps for all worker nodes and ensure that each node is powered on once the profile is deployed.



Post-deployment Ansible configuration

The remaining host configuration is handled via Ansible playbooks. The plays for this section is included in the earlier clone from GitHub. This section describes the plays within the context of the directory they are found.

Configuration of the worker nodes is handled by the “nworkers” (Nimble worker nodes) and “nworkerconf” (Nimble worker configuration roles) within the repository. These roles and the accompanying plays consist of the following files.

Root directory

- `hosts` - this file contains the definition of the hosts that will be used within the solution as well as variables for networking and iSCSI `iqn`. This is the same file that exists for the hypervisor hosts.
- `nworkers.yaml` - this file is used to run the plays that register and update the host as well as create the user that will be responsible for installing OpenShift Container Platform.
- `nworkerconf.yaml` - this file is used to run the plays that configure the Docker environment, start services, configure the network connections for the host, alter the IQN, and bring up the networks.

Roles tasks directories (<root>/roles/<role name>/tasks)

- `nworkers/` - houses the required files used by the `nworkers` role:
 - `main.yaml` - this file defines the order that the various plays are run in.
 - `usertasks.yaml` - creates a non-root user to install OCP, assigns them to the required group, and configures `/etc/sudoers` file.
 - `hostreg.yaml` - this file registers the host and attaches pools. It pulls user credentials from the vault file.
 - `repos.yaml` - this file disables all repositories and then enables the required repository.
 - `yumtasks.yaml` - this file updates the host.
- `nworkerconf/` - houses the required files used by the `nworkerconf` role:
 - `main.yaml` - this file defines the order that the various plays are run in.
 - `files.yaml` - copies the `daemon.json` file to `/etc/docker` and removes the default Docker storage configuration.
 - `iscsi.yaml` - this file configures and activates the NICs used for iSCSI connectivity within the environment and pulls information from the `hosts` file.
 - `iqns.yaml` - this file sets the IQN of the host and uses the `templates/iqn.j2` template file.
 - `bringup.yaml` - This file brings up the network connections to facilitate a new connection point from the Ansible host.
- Roles templates directory - only the `nworkerconf` role has templates (<root>/roles/nworkerconf/templates):
 - `iqns.j2` - this template file overwrites the `/etc/iscsi/initiatorname.iscsi` file and appends the host name to the IQN string.
- Roles variables directories (<root>/roles/<role name>/vars)
 - `main.yaml` - this file contains variables specific to each role.

The installer should configure variables within the `hosts` file and within the individual variable files (`roles/nworkers/vars/main.yaml` and `roles/nworkerconf/vars/main.yaml`) on a per role basis prior to running the play.

Running the roles

1. From the Ansible Engine host, run the following commands to finalize the deployment of the worker nodes:

```
# cd /etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/
# ansible-playbook -i hosts playbooks/nworkers.yaml --ask-vault-pass
```



2. Once the play is complete and the hosts are restarted, SSH into one of the worker nodes and run the following command:

```
# subscription-manager status
```

You should see a return that tells you the status is current.

3. From within the Ansible Engine host, run the following command:

```
# ansible-playbook -i hosts playbooks/nworkerconf.yaml -ask-vault-pass
```

4. Once the play is complete and the hosts are restarted, run the following commands from the Ansible Engine host:

```
# ping <iscsi A address>

# ping <iscsi B address>

# ping <data center bond IP>
```

5. After the remote hosts reboot, run the following command from the Ansible Engine host:

```
# ping <management bond IP>
```

OpenShift deployment

This section describes the process to automatically deploy Red Hat OpenShift Container Platform 3.11. This section is built with the assumption that the required repository is already cloned and is available in the Ansible Engine. For more information about cloning the repositories, see [Editing and running the Ansible playbooks](#).

Prerequisites

In order to utilize the scripts and procedures documented in this deployment section, the following prerequisites 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.
- Storage and networking are configured within hosted engine.
- DNS entries should exist for all hosts.
- A user should be created in Active Directory (AD) for authentication and the user AD values should be known.

Note

In case htpasswd is used for authentication, an htpasswd file should be created. htpasswd file can be created using the tool available at <http://www.htaccess tools.com/htpasswd-generator/>. Save the file to `/etc/oshift-hash-pass.htpasswd` and refer this file in the identity provider section under OCP variables in the host file.

```
# openshift_master_identity_providers=[{'name': 'htpasswd_auth', 'login': 'true', 'challenge': 'true', 'kind':
'HTPasswdPasswordIdentityProvider',}]
```

1. On the Ansible Engine host, run the following command to generate a key:

```
# ssh-keygen -t rsa
```

This creates a key file at `~/.ssh/id_rsa.pub`.



2. Copy this SSH public key to `/var` and change the permission using the below commands:

```
# cp /root/.ssh/id_rsa.pub /var/id_rsa.pub
# chmod 666 /var/id_rsa.pub
```

3. If the `ca.pem` certificate file does not exist at `/etc/pki/ovirt-engine` in the hosted engine, download the certificate on to the Ansible 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"
```

4. 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 3.11 deployment are listed below:

- Create the virtual machine template.
- Deploy the virtual machines from the virtual machine template.
- Install prerequisites for the OpenShift installation on the virtual machines.

Create the virtual machine template

The first play is to create a virtual machine template in the hosted engine. This template will later be used to clone and provision the Red Hat OpenShift Infrastructure services VMs (master, infrastructure, etcd and load balancer) for OpenShift deployment.

To create a virtual machine template, perform the following steps:

1. On the Ansible Engine host, locate the variable yml file 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 contain information about the RHVM engine URL, image download location, template name, and size.

2. Edit the variable file – Change the values for qcow url and disk storage domain name (provide any name and this needs to be mapped to the LUN created on HPE Nimble).

In the variable file, `qcow_url` is the URL for the Red Hat Enterprise Linux 7.6 KVM guest image. The image can be downloaded from <https://access.redhat.com/downloads>. This will be unique to each subscriber.

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

```
# ansible-playbook -i hosts playbooks/deployTemplate.yml --ask-vault-pass
```



When completed, a template appear in the Red Hat Virtualization Administration Portal as in Figure 40.

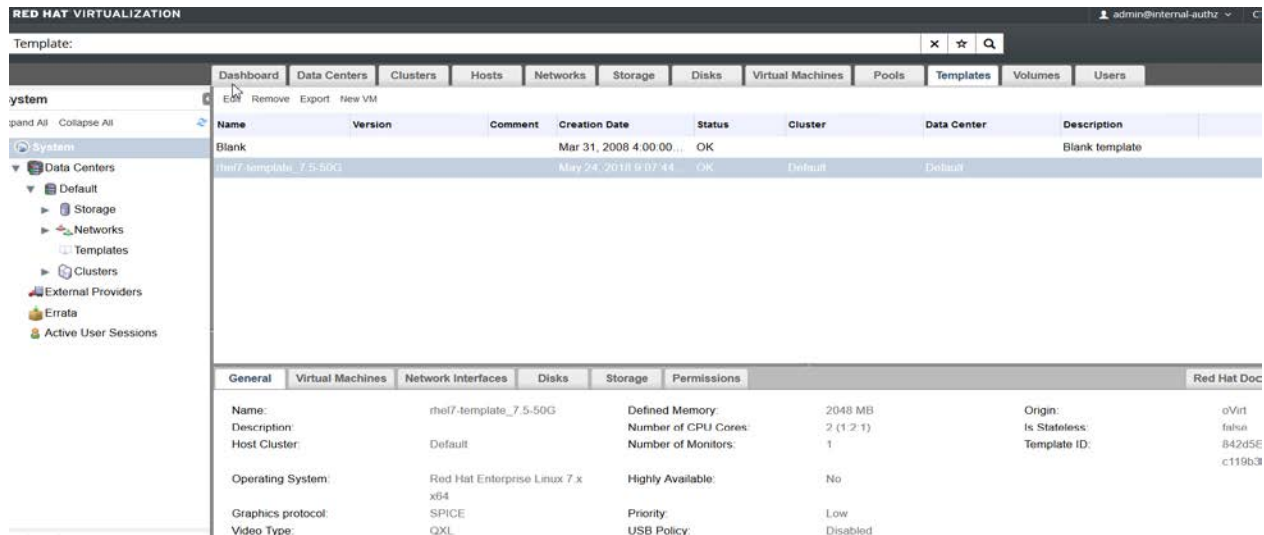


Figure 40. VM template view from the Red Hat Virtualization Administration Portal

Deploy virtual machines from the template

The `deployVM.yml` playbook creates the following virtual machines:

- nmaster01
- nmaster02
- nmaster03
- ninfra01
- ninfra02
- ninfra03
- netcd01
- netcd02
- netcd03
- nlb01
- nlb02

To deploy virtual machines from the virtual machine template, perform the following steps:

1. Using an editor such as `vi` or `nano`, edit the variable file (`main.yml`) at `/etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/roles/deploy-vm/vars/main.yml` file.

The file should look like the example provided in `/etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/roles/deploy-vm/vars/main.yml` of the git repository for this solution and contain information about the VMs, hosted engine, hostnames, IPs, memory, and CPU.

2. When you have completed editing the variables file, run the following command to deploy all of the VMs required for the OpenShift Container Platform:

```
# ansible-playbook -i hosts playbooks/deployVM.yml --ask-vault-pass -e@vault_pass.yml
```



All the VMs are populated on the hosted engine with the domain name, IP, and so on.

3. Select **Compute > Virtual Machine** to see all the VMs deployed from the `deployVm` play.

Note

To find Pool IDs for the management and worker VMs, execute the following command and look for `System Type: Virtual`

```
# subscription-manager list --available --matches '*OpenShift*'

```

Install prerequisites for OpenShift installation on the virtual machines and worker nodes

Next play prepares the hosts for OpenShift installation. Two Ansible roles (`roles/virtual-host-prepare/` and `roles/physical-host-prepare/`) are available for preparing the management virtual machines and physical worker nodes.

Containers and the images they are created from are stored in Docker's storage back end. This storage is ephemeral and separate from any persistent storage allocated to meet the needs of applications. The default storage back end for Docker on RHEL 7 is a thin pool on loopback devices which is not supported in production environments. To work around this, we use an additional block device to create a thin pool device for Docker's local storage.

1. Edit the files `roles/virtual-host-prepare/vars/main.yml` and `roles/physical-host-prepare/vars/main.yml` in a text editor such as `vi` or `nano` and enter the path to the second disk.

For virtual machines created by running the `deployVM.yml` play, the default location of the second disk is `/dev/vdb` and is already updated in the variable file available at `roles/virtual-host-prepare/vars/main.yml`:

```
second_disk_vms: /dev/vdb

```

For physical worker nodes created as per the [Red Hat OpenShift Worker Nodes](#) section, the default location of the second disk is `/dev/mapper/mpatha` and is already updated in the variable file available at `roles/physical-host-prepare/vars/main.yml`:

```
second_disk_physical: /dev/mapper/mpatha

```

2. The host prepare play accomplishes the following:

- Disables the firewall for the OpenShift installation. This will be re-enabled post-install.
- Creates a user group with password-less sudo rights.
- Creates a sudo user and adds the user to the password-less sudo group.
- Uploads the public SSH key to allow secure access without credentials.
- Registers the host using Subscription Manager.
- Enables the required repositories.
- Installs the basic utilities.
- Performs a yum update to ensure the latest patches and updates are applied.
- Installs Red Hat OpenShift related packages.
- Installs the latest version of Docker which should be at 1.13-94 or above.
- Configures Docker local storage.

3. To prepare virtual machines, execute the following command on the Ansible Engine host.

```
# ansible-playbook -i hosts playbooks/virtual-hostprepare.yml --ask-vault-pass

```

4. To prepare physical worker nodes, execute the following command on the Ansible Engine host.

```
# ansible-playbook -i hosts playbooks/physical-hostprepare.yml --ask-vault-pass

```



OpenShift-Ansible

The following Ansible playbooks deploy Red Hat OpenShift Container Platform on the machines that have been created and configured by the previous Ansible playbooks. In order to get the OpenShift-Ansible playbooks from the 'Red Hat OpenShift Container Platform 3.11' repository, run the following command:

```
# yum install openshift-ansible
```

The variables for the OpenShift deployment are maintained in the Ansible inventory file, for example, `/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 the `/usr/share/ansible/openshift-ansible/playbooks/prerequisites.yml` playbook:

```
# ansible-playbook -i /etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/hosts  
/usr/share/ansible/openshift-ansible/playbooks/prerequisites.yml
```

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

```
# ansible-playbook -i /etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/hosts  
/usr/share/ansible/openshift-ansible/playbooks/deploy_cluster.yml
```

3. When the deployment is complete, the installer may access the OpenShift webpage, shown in Figure 41, using the credentials provided in the `htpasswd` file or the Active Directory. The URL for the webpage is <https://<load balancer>:8443>.

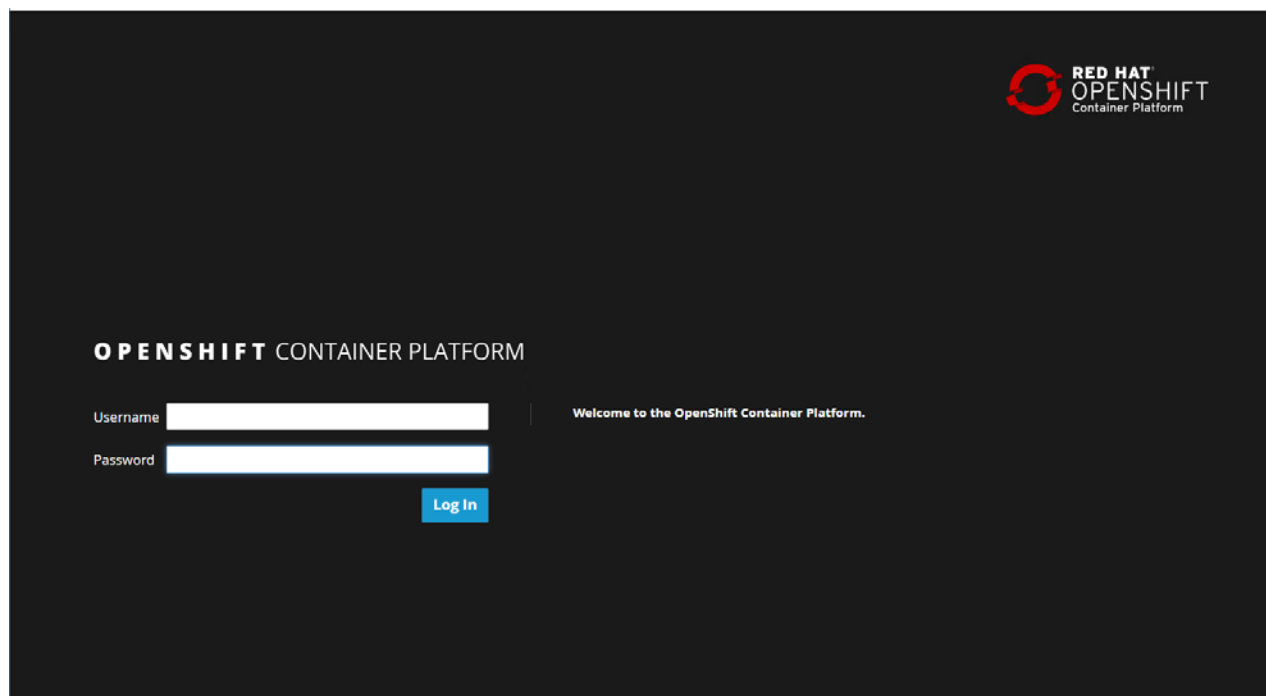


Figure 41. OpenShift user interface



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
nworker01.tennet.local	Ready	compute	18h	v1.10.0+b81c8f8
nworker02.tennet.local	Ready	compute	18h	v1.10.0+b81c8f8
nworker03.tennet.local	Ready	compute	18h	v1.10.0+b81c8f8
ninfra01.tennet.local	Ready	infra	18h	v1.10.0+b81c8f8
ninfra02.tennet.local	Ready	infra	18h	v1.10.0+b81c8f8
ninfra03.tennet.local	Ready	infra	18h	v1.10.0+b81c8f8
nmaster01.tennet.local	Ready	master	19h	v1.10.0+b81c8f8
nmaster02.tennet.local	Ready	master	19h	v1.10.0+b81c8f8
nmaster03.tennet.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

Grant cluster role to user

1. From the nmaster01 node, log in as the default system admin account as shown below:

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

2. Once logged in, the system displays 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
```

3. While logged in as the system administrator, 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.



Figure 42 shows the final VM layout once OpenShift has been deployed. Validate that your environment aligns to this design.

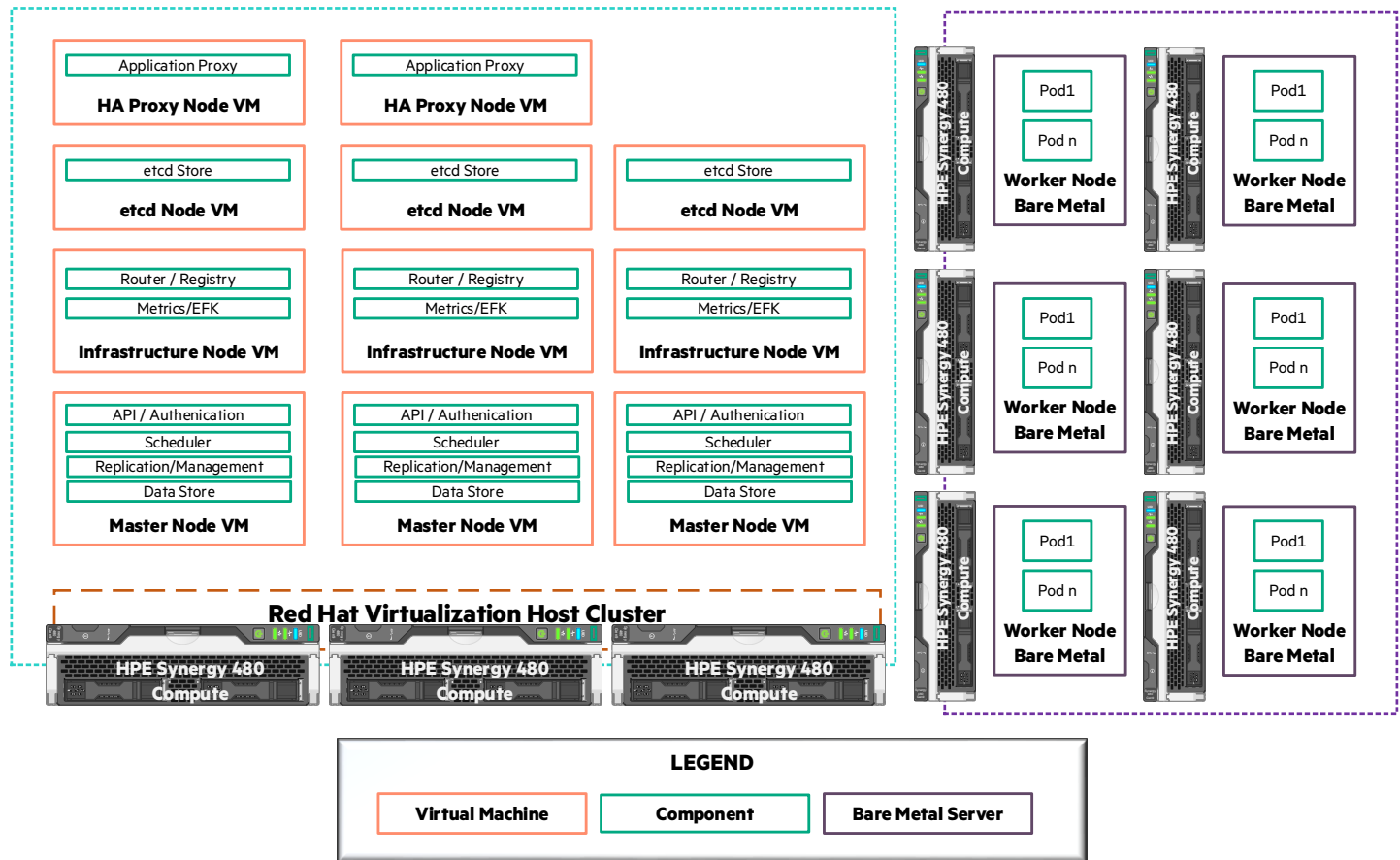


Figure 42. Final OpenShift layout, post deployment

Install the Nimble Linux Toolkit on the worker nodes

The latest version of Nimble Linux Toolkit is 3.x. However, the installer should use the NLT version 2.4.x.x to implement this solution. If Ansible playbooks has been used to automate the installation of the array(s) in the environment, skip this section. Otherwise, perform the following tasks to install Nimble Toolkit on the worker nodes:

1. Refer Appendix B to use Ansible to install the NLT on the worker nodes. If the installer is not using Ansible, ensure that the Nimble Linux Toolkit has been copied to each worker, infrastructure and master node.
2. Log in via SSH or at the iLO console of each node and run the following commands:

```
# cd <location_where_nlt_installer_was_copied>
# chmod +x nlt_installer_2.4.x.x
# ./nlt_installer_2.4.x.x --silent-mode --docker --accept-eula --flexvolume
```

3. Run the following commands from nltadm where the IP address is the IP of the management adapter on the HPE Nimble Storage group:

```
# nltadm --group --add --ip-address #.#.#.# --username admin --password 'Password'
# nltadm --group --verify --ip-address #.#.#.#
```

4. To discover the FlexVolume driver, restart the OpenShift node service after installing NLT using the following command:

```
# systemctl restart atomic-openshift-node
```

Note

Uninstall NLT before re-installing OpenShift nodes, in order to avoid any certificate corruption in HPE Nimble Storage. multipathd should be re-installed after any upgrade/re-install of OpenShift to ensure the correct multipath.conf file is being used. Uninstall NLT using the following commands:

```
# cd /tmp
# nlt_uninstall
```

Install the HPE Nimble Kube Storage Controller

1. On a master node, ensure to login as a user with the system:admin role. verify the same by running the following command:

```
# oc whoami -c
default/masternode.example.domain:8443/system:admin
```

2. Clone the example specification files from the Nimble Storage GitHub by running the following command:

```
# git clone https://github.com/nimblestorage/container-examples
```

3. Run the following commands to deploy the HPE Nimble Kube Storage Controller:

```
# cd container-examples/NLT/OpenShift/ocp310
# oc create -f dep-kube-storage-controller.yaml
```

4. To validate the installation of the HPE Nimble Kube Storage Controller run the following:

```
# oc get deploy --namespace kube-system
NAME                                DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
kube-storage-controller-dorcyd 1          1          1          1          3s
```

Deploy Docker Registry with persistent storage

An integrated Docker Registry is created along with the OpenShift installation. This Registry pod will be running in the namespace 'default'. Running the `validate-deployment` play will check if the Docker registry is running and, if it is not, will create an integrated Docker registry. If the Docker registry is running with an ephemeral volume, this play will create a PVC and attach the persistent volume to the Registry pod.

```
# ansible-playbook -i /etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/hosts /etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/playbooks/validate-deployment.yaml
```

The following message confirms the success of the deployment:

```
"msg": "Successfully deployed Docker Registry with Persistent Storage, Registry pod name < Name of the Registry Pod>
```

Validate deployment

There are a couple of YAML files in the repository used to create a StorageClass, Persistent Volume Claims (PVC) and an example deployment. Pay attention to the different API versions and annotations needed for the different versions of OpenShift.

1. Create an application optimized StorageClass by running the following command:

```
# oc create -f sc-transactionaldb.yaml
storageclass "transactionaldb" created
```

2. Create a PVC from the StorageClass by running the following command:

```
# oc create -f pvc-mariadb.yaml
persistentvolumeclaim "mariadb-claim" created
```



3. Create a deployment with a PVC reference by running the following command:

```
# oc create -f dep-mariadb.yaml
secret "mariadb" created
deployment "mariadb" created
service "mariadb" created
```

4. Create a default StorageClass for "classless" PVCs as below:

```
# oc create -f sc-default.yaml
storageclass "general" created
```

5. Create a PVC without a StorageClass as follows:

```
# oc create -f pvc-default.yaml
persistentvolumeclaim "default-claim" created
```

6. To observe the created resources, run the following command:

```
# oc get deploy,storageclass,pvc,pv
```

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deploy/mariadb	1	1	1	1	45s

NAME	TYPE
general [default]	hpe.com/nimble
transactionaldb	hpe.com/nimble

NAME	STATUS	VOLUME	CAPACITY	ACCESSMODES	AGE
pvc/default-claim	Bound	general	32Gi	RWO	12s
pvc/mariadb-claim	Bound	transactionaldb	16Gi	RWO	1m

NAME	CAPACITY	ACCESS	RECLAIMPOLICY	STATUS	CLAIM	AGE
pv/general	32Gi	RWO	Delete	Bound	default-claim	9s
pv/transactionaldb	16Gi	RWO	Delete	Bound	mariadb-claim	1m

Ansible OpenShift deployment removal

This Ansible play removes the Red Hat OpenShift 3.11 deployment. It is a two-step process that requires uninstalling the application followed by unregistering Red Hat OpenShift components and deleting any deployed VMs.

Uninstall OpenShift

To uninstall OpenShift 3.11 from all nodes run the following command

```
# ansible-playbook -i /etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/hosts
/usr/share/ansible/openshift-ansible/playbooks/adhoc/uninstall.yml
```

Unregister OpenShift components and delete deployed VMs

To unregister the nodes from Red Hat Subscription and to delete the VMs from Red Hat Virtualization Manager run the following command

```
# ansible-playbook -i /etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/hosts /etc/ansible/hpe-
solutions-openshift/synergy/scalable/nimble/playbooks/delete-vm.yml --ask-vault-pass
```



Appendix A - Playbook variables

Table A1 describes the variables used with the VM template.

Table A1. Variables used with the VM template

Variable	Scope	Description
engine_url	RHV-M	RHV-M REST API web UI link
engine_cafile	RHV-M	Location of CA file in RHV-M for authentication
datacenter	RHV-M	Data center within RHV-M which will be used for template deployment
cluster	RHV-M	Cluster within RHV-M which will be used for template deployment
template_disk_storage	RHV-M	RHV-M Storage domain
qcow_url	RHV-M	Red Hat Enterprise Linux 7.6 KVM Guest Image download path
image_path	RHV-M	Image download location on RHV-M /var/images with file name rhel-server-7.6-50G
template_name	RHV-M	Template name in RHV-M
vm_name	RHV-M	Temporary VM before cloning to template
template_memory	RHV-M	Template memory
template_cpu	RHV-M	Template vcpu count
template_disk_interface	RHV-M	Network interface type
template_disk_size	RHV-M	OS disk
template_disk_size_2	RHV-M	Second disk for Docker local storage
image_cache_download	RHV-M	Create template if the Image is already downloaded at /var/images directory
template_nics name: nic1 profile_name interface	RHV-M	Adapter name RHV-M network name Interface name

Table A2 describes the variables used in the deployment of the VMs.

Table A2. Variables used in the deployment of the VMs

Variable	Scope	Description
engine_url	RHV-M	RHV-M REST API web UI link
engine_cafile	RHV-M	Location of CA file in RHV-M for authentication
datacenter	RHV-M	Data center within RHV-M which will be used for template deployment
cluster	RHV-M	Cluster within RHV-M which will be used for template deployment
network	RHV-M	RHV-M Storage domain
vmtemplate	RHV-M	Red Hat Enterprise Linux 7.6 KVM Guest Image download path
mgmt_gateway: mgmt_mask: mgmt_dns: <vmname>_ ip:	RHV-M	Details of the OpenShift VM network



Table A3 describes the variables used during host preparation.

Table A3. Variables used during host preparation

Variable	Scope	Description
second_disk_physical	Worker node	Path to the second disk
second_disk_vms:	Management VMs	Path to the second disk

Appendix B - Utilizing Ansinimble

This section provides a high-level overview of utilizing Ansinimble. Detailed examples and documentation about Ansinimble are available at <https://github.com/NimbleStorage/ansinimble>.

Prerequisites

- Make sure your Ansible server has Python 3 installed and the Ansible Engine running is using Python 3. If otherwise, refer the blog from Red Hat on installing Python 3 at <https://developers.redhat.com/blog/2018/08/13/install-python3-rhel/>.

To check the python version used by Ansible engine, run the following command:

```
# ansible --version |grep 'python version'
# python version = 3.6.3 (default, Jan 9 2018, 10:19:07) [GCC 4.8.5 20150623 (Red Hat 4.8.5-11)]
```

- JMESPath ("james path") needs to be installed on the Ansible host. Refer [https:// http://jmespath.org/](https://http://jmespath.org/) for more information.
- All target hosts (worker nodes) should be added to the known_hosts file.

Tasks

Download the Ansinimble role

Run the following command on the Ansible Engine host to download the Ansinimble Role:

```
# ansible-galaxy install NimbleStorage.Ansinimble
```

This role is deployed to the location `/root/.ansible/roles/NimbleStorage.Ansinimble`.

Create an inventory file

Run the following command on the Ansible Engine host to create an inventory file:

```
# cd /root/.ansible/roles/NimbleStorage.Ansinimble
# touch hosts
```

The following example host file consists of the worker nodes which will be connected to the Nimble Arrays as well as the Ansible Engine host defined under `ansible_host`:

```
#[ansible_host]
localhost      ansible_ssh_user=root ansible_ssh_pass=<Your password> *

#[worker nodes]
nworker01.tennet.local  ansible_ssh_user=root ansible_ssh_pass=<Your password> *
nworker02.tennet.local  ansible_ssh_user=root ansible_ssh_pass=<Your password> *
nworker03.tennet.local  ansible_ssh_user=root ansible_ssh_pass=<Your password> *
```

Edit the variable file

`defaults/main.yml` is the variable file for the entire Ansinimble role. The installer should update the file to meet the needs of the environment.

Array setup

1. Download the Nimble Windows Toolkit (NLT) from HPE InfoSight and install it on the installer laptop.
2. With the laptop connected to the same switch as the Nimble array, use the tool to discover the array.
3. Record the serial number for the array.



4. The Ansible playbook for Nimble array setup uses variables defined in `default/main.yml` under the section 'nimble_array_config'. Copy this file outside of the role structure and create a new variable file for your project with what you need.
5. Run the following command to configure the array from an initial state over the network:

```
# ansible-playbook -i hosts -e nimble_array_serial=<Nimble Array Serial Number> sample_array_setup.yml
```

This play needs to be run against both arrays in a redundant configuration. In such a scenario, the `nimble_array_config` value for both arrays should be edited.

Install NLT on the target hosts and configure the group

Target hosts (worker nodes) should have the Nimble Linux Tool (NLT) kit installed and configured against the upstream array.

1. Download the NLT from <https://infosight.hpe.com> and copy it to the `/tmp` directory on the Ansible Engine host.
2. Make the following changes to the `default/main.yml` file if NLT needs to be installed:

```
nimble_group_options:
  ip-address: 10.0.2.95
  username: admin
# Store the password with Ansible Vault.
nimble_group_password: admin
```

3. Run the following command to install NLT on the worker nodes:

```
# ansible-playbook -I hosts -e nimble_linux_toolkit_bundle=/tmp/nlt_installer_2.4.1.13 -e
nimble_linux_toolkit_protocol=iscsi sample_install.yml
```

Appendix C - Deploying worker node functions to virtual machines

Some customers may have a preference for deploying the worker nodes as virtual machines. This appendix highlights the changes required to variable files and playbooks to facilitate the creation of worker node VMs that can be used during OpenShift Container Platform deployment. It is presented on a role and file basis with alterations listed for each file. Changes are not required to the inventory file. The installer should ensure that the names of the worker node VMs appear in the correct location just as if they were physical.

Role: deploy-template

No changes are required to this role. However, the installer should validate variable values under `/etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/roles/deploy-template/vars/main.yml` to ensure that they are aligned to their environment.

Role: deploy-vm

In addition to the VMs outlined in the [Deploy Virtual Machines from the Template](#) section of this document, the installer should also deploy the OpenShift worker node VMs. To do this, the installer should perform the following steps:

1. Edit the file: `/etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/roles/deploy-vm/vars/main.yml`
2. Add worker node fqdn to the section that outlines FQDN details of VMs as shown in the following example:

```
fqdnworker1: <fqdn of hostname for node 1>
fqdnworker2: <fqdn of hostname for node 2>
fqdnworker3: <fqdn of hostname for node 3>
```

3. For each worker node VM, add the IP details to the section under 'IP details of the Worker Nodes' as shown in the following example:

```
nworker01_ip: <ip address>
nworker02_ip2: <ip address>
nworker03_ip3: <ip address>
```

In this case, `ip` is the main communication network, `ip2` is the iscsia network, and `ip3` is the iscsib network.

No changes are needed to the `deploy-vm` play.



Role: physical-host-prepare

If no physical worker nodes are being used, you can ignore the 'playbooks/physical-hostprepare.yml' play.

Role: virtual-host-prepare

You need to run the play 'playbooks/virtual-hostprepare.yml', but no changes are required to the play.

Ansible hosts file

1. Ensure that the worker nodes are listed under the [virtual-nodes] section.
2. Under the [physical-nodes] section, comment out any hosts that are present in the host file available at /etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/hosts.
3. Create a new section as follows substituting your own worker node host names:

```
[virtual-workers]
workername01.domain.local
workername02.domain.local
workername03.domain.local
```

Role: virtual_workers

Add a new role in /etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble by doing the following:

1. Create a new folder structure for the roles:

```
# cd /etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble
# mkdir roles virtual-workers
# mkdir roles virtual-workers/templates
# mkdir roles virtual-workers/tasks
```

2. Create a new file to run the playbook when complete:

```
# vi playbooks/vworkers.yaml
---
- name: worker-nodes role
  hosts:
    - worker-nodes
  roles:
    - ./roles/virtual-workers
```

3. Copy the template file roles/nworkers/templates/iqn.j2 into the virtual-workers template folder:

```
# cp roles/nworkers/templates/iqns.j2 roles/virtual-workers/templates/.
```



4. Create and edit the file main.yaml in the tasks folder of your virtual-workers role:

```
# vi roles/virtual-workers/tasks/main.yaml
###
## Copyright [2018] Hewlett Packard Enterprise Development LP
##
## Licensed under the Apache License, Version 2.0 [the "License"];
## You may not use this file except in compliance with the License.
## You may obtain a copy of the License at
##
## http://www.apache.org/licenses/LICENSE-2.0
##
## Unless required by applicable law or agreed to in writing, software
## distributed under the license is distributed on an "AS IS" BASIS,
## WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
## See the License for the specific language governing permissions and
## limitations under the License.
####
---
```

- name: Install extra software required on the host
 yum:
 name: "{{ item }}"
 state: latest
 with_items:
 - iscsi-initiator-utils
 - device-mapper-multipath
 - firewalld
- name: Create a new IQN on each host
 template:
 src: iqns.j2
 dest: /etc/iscsi/initiatorname.iscsi
 owner: root
 group: root
 mode: 0755
- name: Start and enable the iscsi service
 service:
 name: iscsi
 state: started
 enabled: yes
- name: Start and enable the firewalld service
 service:
 name: firewalld
 state: started
 enabled: yes
- name: open port 3260/tcp for iscsi traffic
 firewalld:
 port: 3260/tcp
 permanent: yes
 state: enabled
 immediate: yes

5. Once these steps are complete and all VMs are up, run the final plays prior to installing OpenShift Container Platform:

```
# ansible-playbook -i hosts playbooks/vworkers.yaml --ask-vault-pass -e@vault_pass.yml
```

The remaining plays including the connection of storage resources will behave the same with the virtual worker nodes as with physical worker nodes.



Appendix D – OpenShift Container Platform deployment using Ansible Tower

Ansible Tower is a web-based solution for managing Ansible Engine. It features role-based access control, job scheduling, workflows, and graphical inventory management with a simple user interface. The interface provides a dashboard with state summaries of all the hosts, allows quick deployments, and monitors all configurations. Tower allows you to share SSH credentials without exposing them, logs all jobs, manages inventories graphically, and syncs them with a wide variety of cloud providers.

In Ansible Tower, organizations are created for better manageability user access control. Projects are created within organizations which contain the playbooks repositories. Project contains templates, which are Ansible playbooks, and details required to run the playbooks. Each playbook runs against an inventory that consists of the host details. Workflows configure the job templates in sequence.

Prerequisites

- Ansible Engine should be installed and configured and capable of communicating with the hosts within this solution.
- Red Hat Virtualization Host (RHVH) is installed on at least three HPE Synergy 480 Compute Modules.
- RHVHs are configured as a Red Hat Hyperconverged Infrastructure (RHHI) cluster.
- Make sure that both storage and networking are configured within the hosted engine.
- DNS entries should exist for all hosts.

Ansible Tower installation

1. Open the URL <https://releases.ansible.com/ansible-tower/setup/> to see the available Ansible Tower versions.
2. Download the zip file **ansible-tower-setup-3.4.3-1.tar.gz** and copy it to your Ansible Tower under `/var` and extract the file.
3. Install Ansible Tower as per the steps mentioned in the readme file available at `/var/ansible-tower-setup-3.4.3-1/README.md`.
4. After installing Ansible Tower, login to the Ansible Tower using the URL `https://<ansible tower fqdn>` with the credentials provided during installation.
5. Copy the public SSH key from `/root/.ssh/id_rsa.pub` to `/var/id_rsa.pub`.
6. Change the permission for the SSH key using the following command:

```
# chmod 666 /var/id_rsa.pub
```

Create new projects in Ansible Tower

1. Clone the GitHub repository to the location `/var/lib/awx/projects` in Ansible Tower using the following command:

```
# git clone https://github.com/HewlettPackard/hpe-solutions-openshift.git
```

2. From the navigation bar, select **Organizations**.



3. Click the + icon to create a new organization as shown in Figure D1.

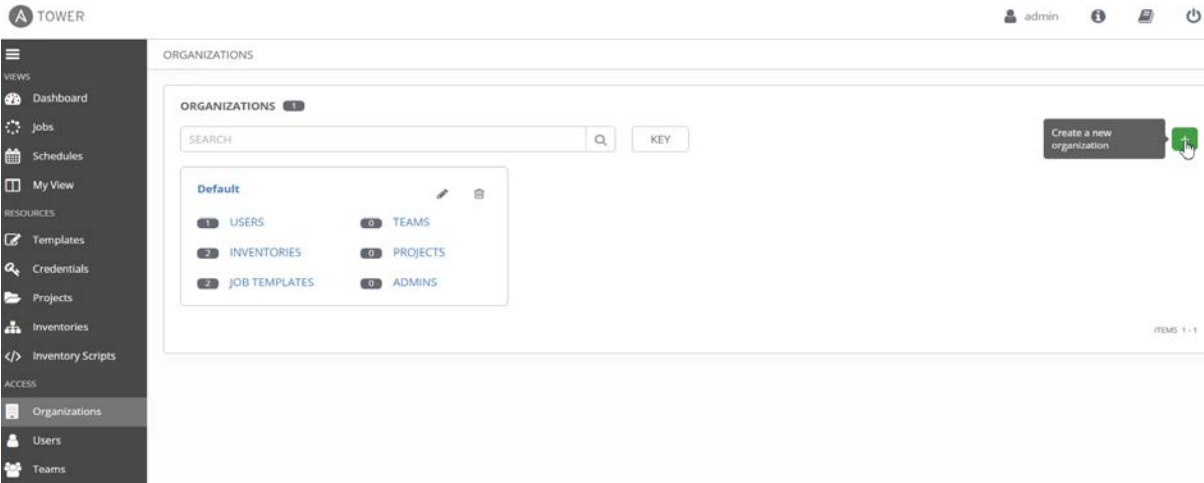


Figure D1. Ansible Tower organization view

A screen to create a new organization appears.

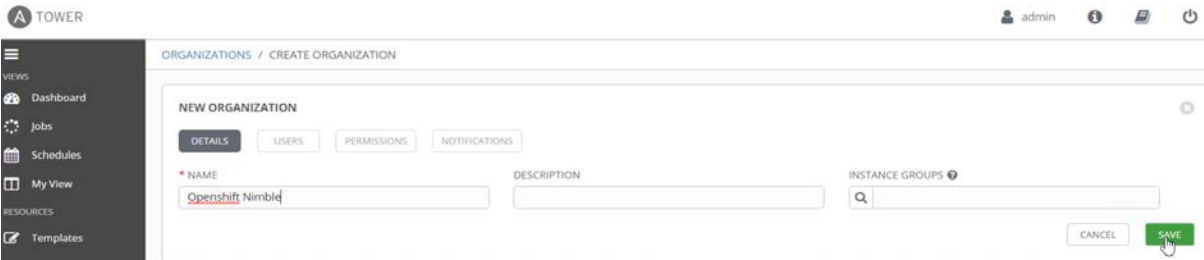


Figure D2. Create organization

- 4. Provide a name for the organization as 'OpenShift Nimble' and click **Save** as in Figure D2.
- 5. From the navigation bar, select **Projects**.



6. Click the + icon to create a new project as in Figure D3.

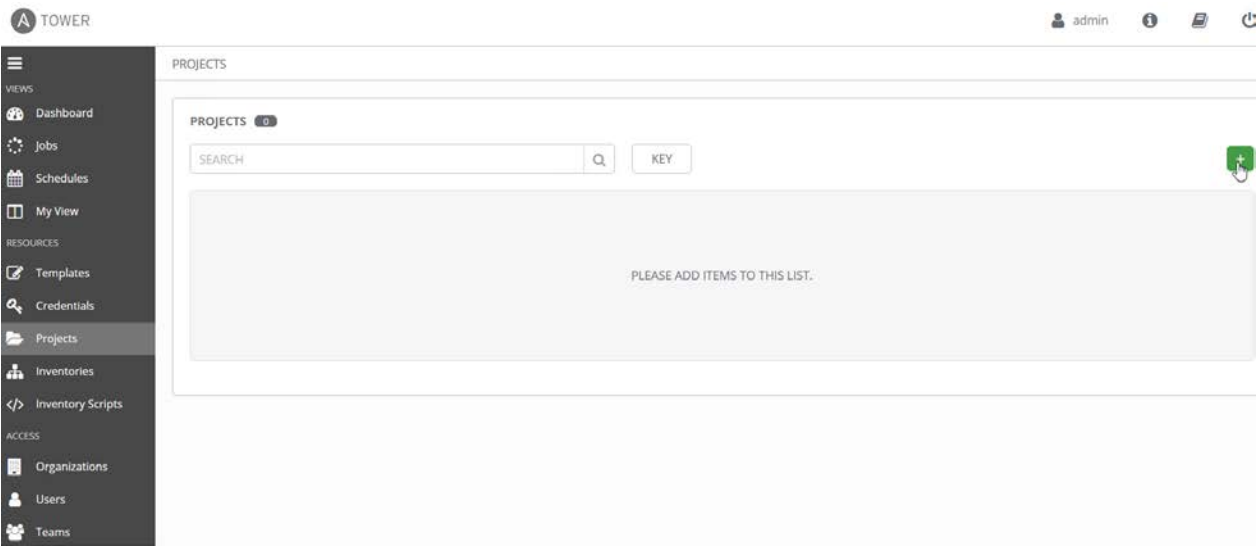


Figure D3. Create project

7. Provide an appropriate name for the project and select the organization that was created earlier. Select the SCM type as manual. For the playbook directory, provide the folder name (hpe-solutions-openshift) of the GitHub repository that you cloned earlier, and then click **Save** as shown in Figure D4.

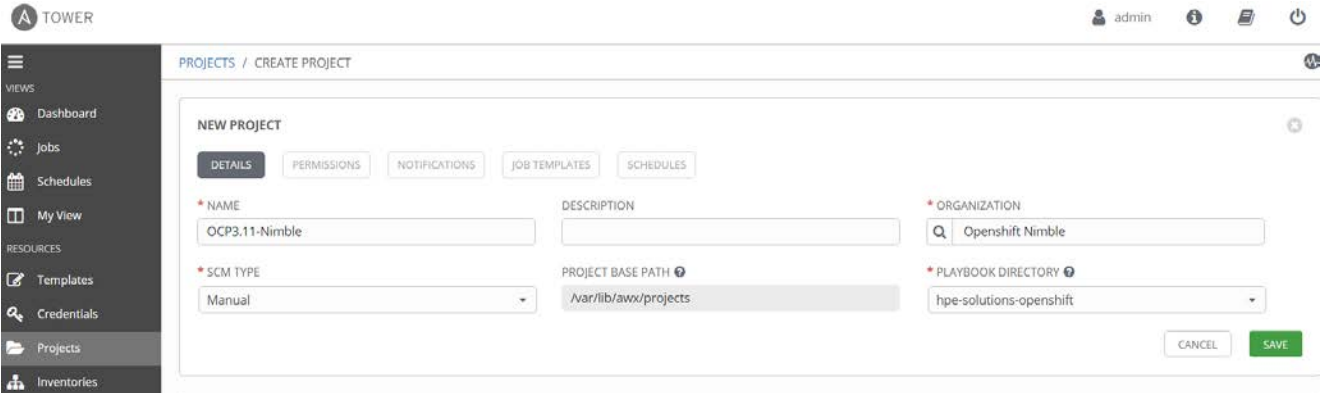


Figure D4. Details of a project



8. Similarly, create another project for the Openshift-Ansible repository. Select SCM Type as Git, and provide SCM URL as <https://github.com/openshift/openshift-ansible.git>. Provide SCM Branch as 'release-3.11', and then click **Save** as shown in Figure D5.

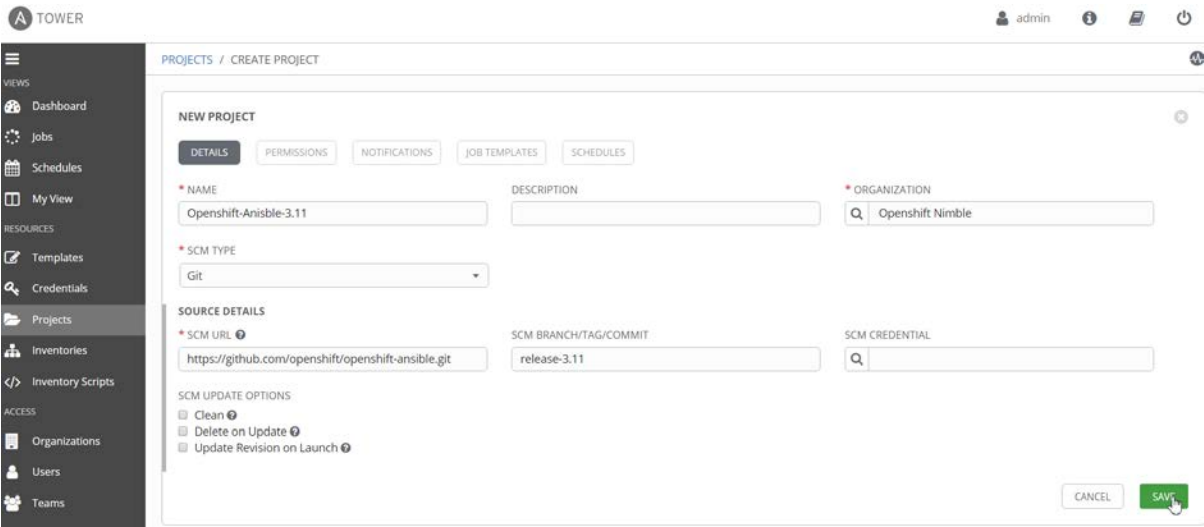


Figure D5. Create project

Create new inventory in Ansible Tower

1. Make the required changes to reflect the installer's environment details in the host file, vault file, and variables files located at `/var/lib/awx/projects/hpe-solutions-openshift/synergy/scalable/nimble/`.
2. Create a blank inventory resource by selecting **Inventories** from the navigation bar. Click the **+** icon and select **Inventory** as shown in Figure D6.

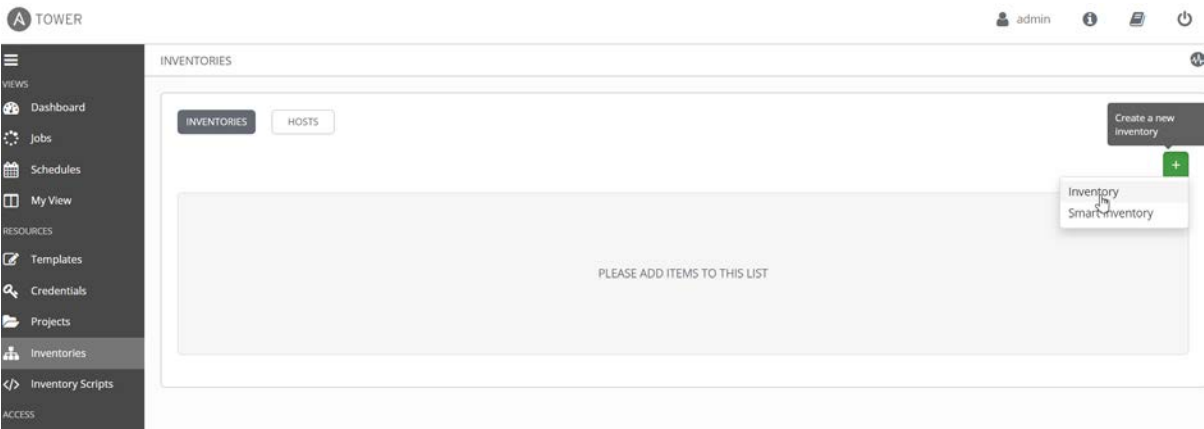


Figure D6. Create new inventory



3. Provide a name and description for the inventory. Provide the name of the organization that you created earlier and click **Save** as shown in Figure D7.

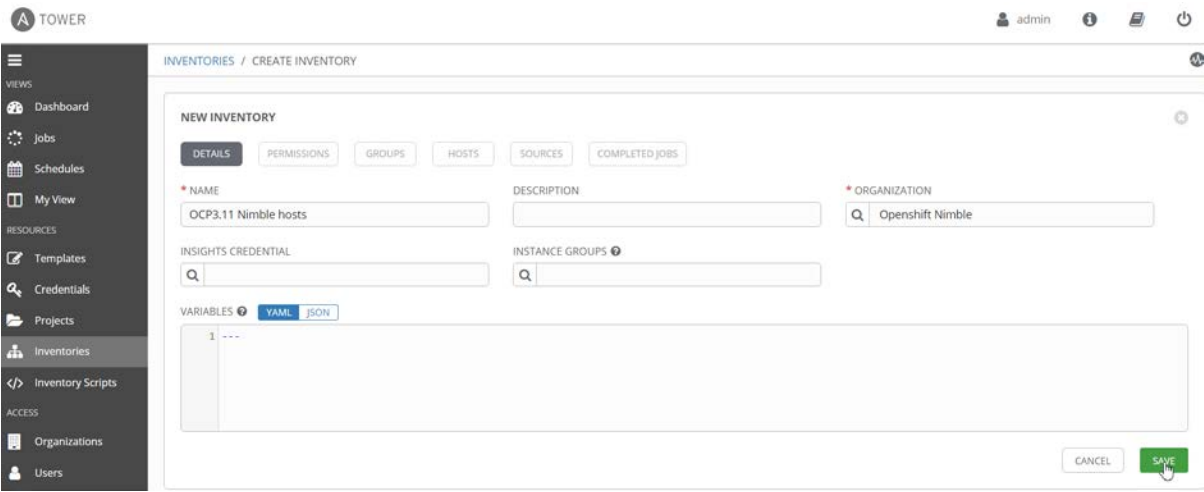


Figure D7. Details of an inventory

4. Run the following command to import the inventory variables, groups, and hosts to the newly created inventory resource:

```
# awx-manage inventory_import --inventory-name 'OCP3.11 Nimble hosts' --source /var/lib/awx/projects/hpe-solutions-openshift/synergy/scalable/nimble/hosts
```

5. After running the above command successfully, the inventory shows a green status as shown in Figure D8.

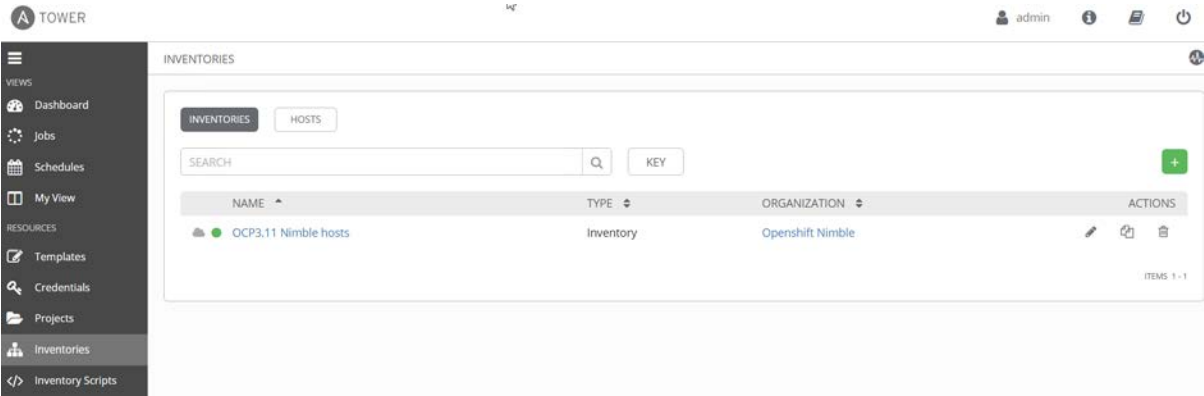


Figure D8. Status of the inventory



Set up credentials in Ansible Tower

To set up credentials for Ansible Tower, perform the following steps:

1. From the navigation bar, select **Credential Types**.
2. Click the **+** icon as shown in Figure D9.

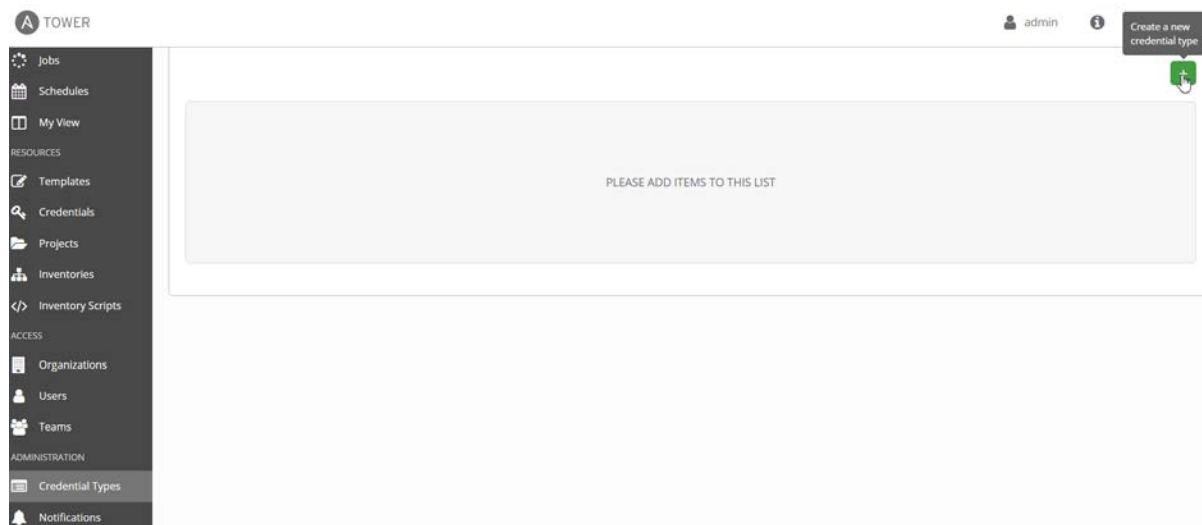


Figure D9. Add a new credential type

3. Provide a name for the credential type as 'Ansible SSH root user', and under **Input Configuration**, add the following parameters as shown in Figure D10:

```
fields:
- type: string
  id: username
  label: Ansible SSH user name [root]
- secret: true
  type: string
  id: password
  label: Ansible SSH root password
```



4. Under **Injector Configuration**, add the following parameters and click **Save** as shown in Figure D10.

```
extra_vars:
  ansible_ssh_pass: '{{ password }}'
  ansible_ssh_user: '{{ username }}'
```

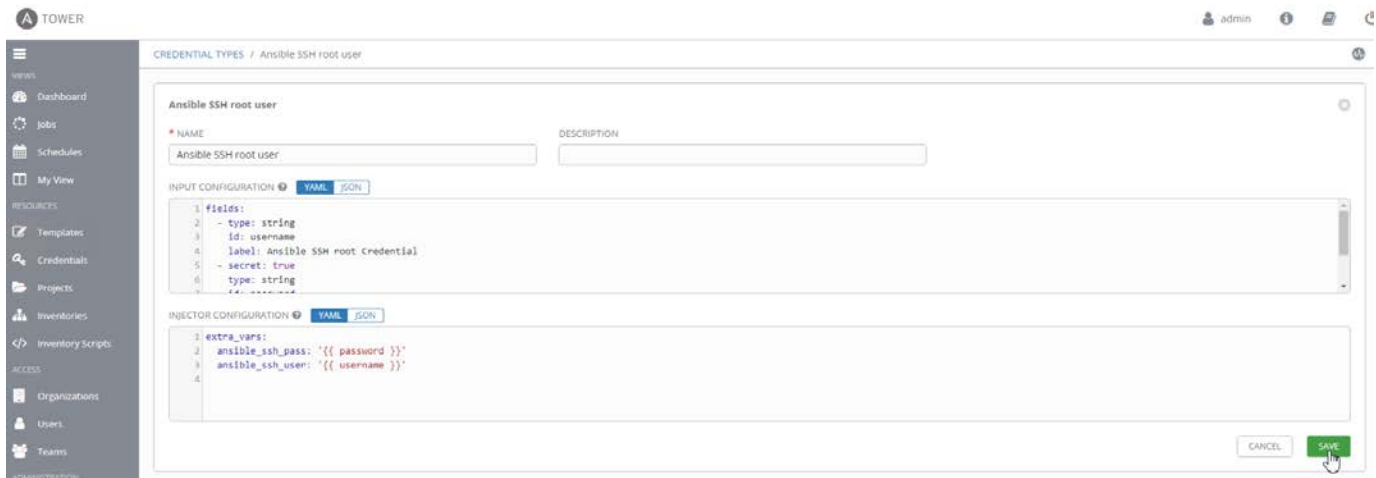


Figure D10. Configure credential type

5. Similarly, create another credential type. Provide the name of the credential type as 'Red Hat Subscription'.
6. Under **Input Configuration**, provide the following parameters as shown in Figure D11:

```
fields:
  - type: string
    id: username
    label: Redhat Subscription username
  - secret: true
    type: string
    id: password
    label: Redhat Subscription password
```



7. Under **Injector Configuration**, add the following parameters and click **Save** as shown in Figure D11.

```
extra_vars:
  vault_rhsub_pass: '{{ password }}'
  vault_rhsub_user: '{{ username }}'
```

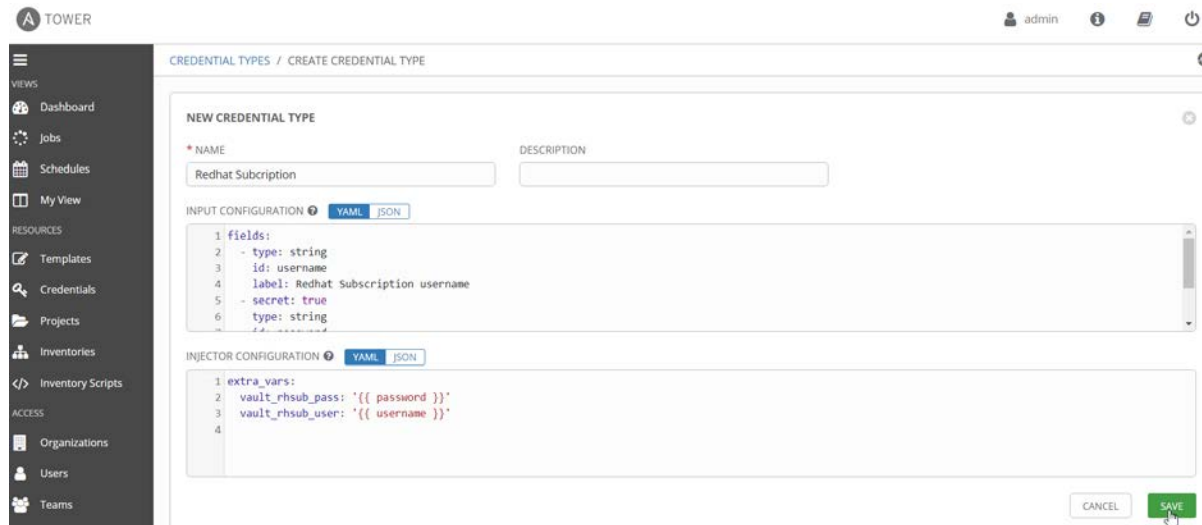


Figure D11. Configure credential type

8. Create four credentials for installing OpenShift Container Platform:
- **Ansible SSH User** - created already as part of credential type creation
 - **Red Hat Subscription** - created already as part of credential type creation
 - **Red Hat Virtualization Manager** – credential to log in to Red Hat Virtualization Manager
 - **Vault credentials** – vault password
9. To add these credentials to Ansible Tower, perform the following steps:
- a. From the navigation bar, select **Credentials**.
 - b. Click the **+** icon as shown in Figure D12.

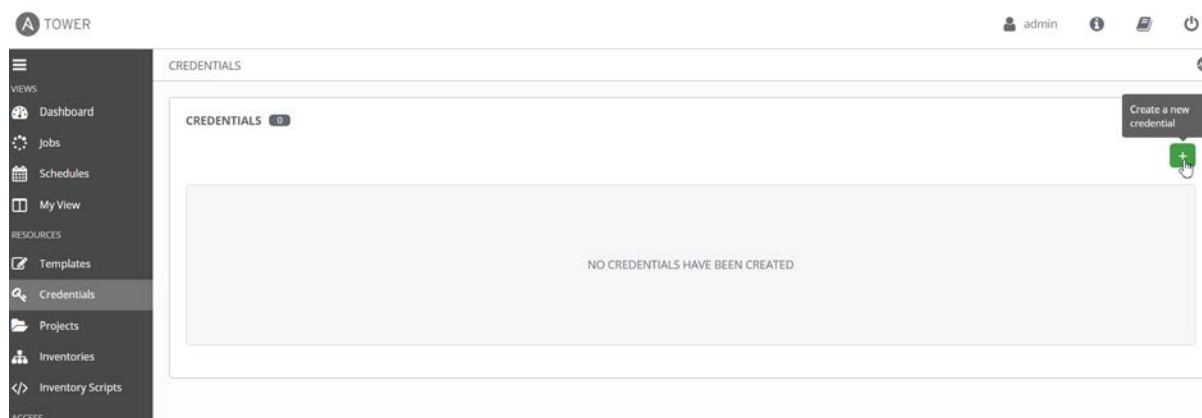


Figure D12. Create a new credential



- c. Provide a name for the credential as 'Ansible SSH User', and select the organization created earlier.
- d. Select the credential type as 'Ansible SSH root user' that you created earlier and click **Select** as shown in Figure D13.

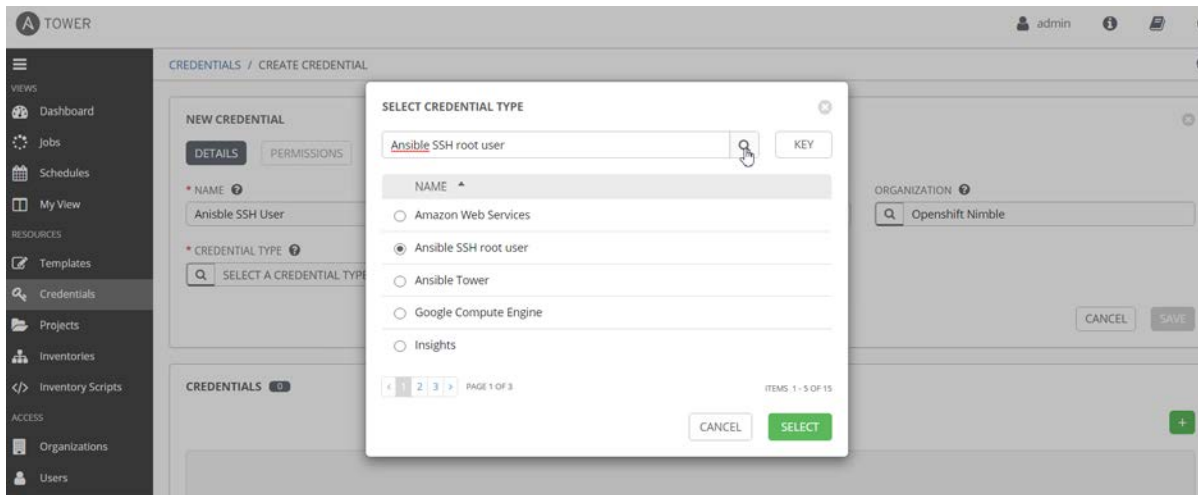


Figure D13. Configure credential from credential type

- e. Enter the username as 'root' and the root password for OCP hosts, and then click **Save** as shown in Figure D14.

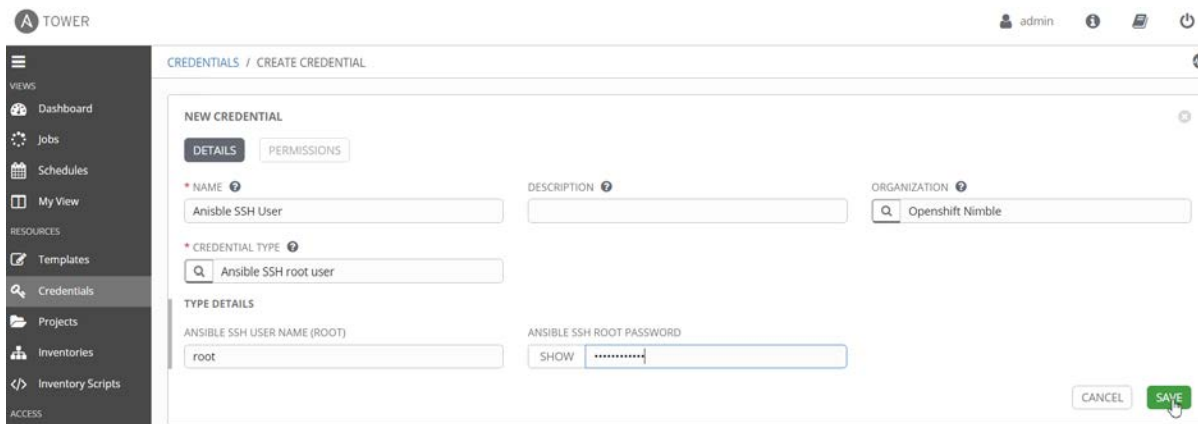


Figure D14. Configure credential from credential type

10. Similarly, create credentials for Red Hat Subscription, Red Hat Virtualization Manager, and Vault credentials.

Import Ansible plays as templates in Ansible Tower

The following Ansible plays should be imported as templates to Ansible Tower:

- All plays listed under `/var/lib/awx/projects/hpe-solutions-openshift/synergy/scalable/nimble/playbooks`
- `openshift-ansible/playbooks/prerequisites.yml`
- `openshift-ansible/playbooks/deploy_cluster.yml`



Follow these steps to import a play as template:

- 1. From the navigation bar, select **Template**. Click the + icon and select **Job Template** as shown in Figure D15.

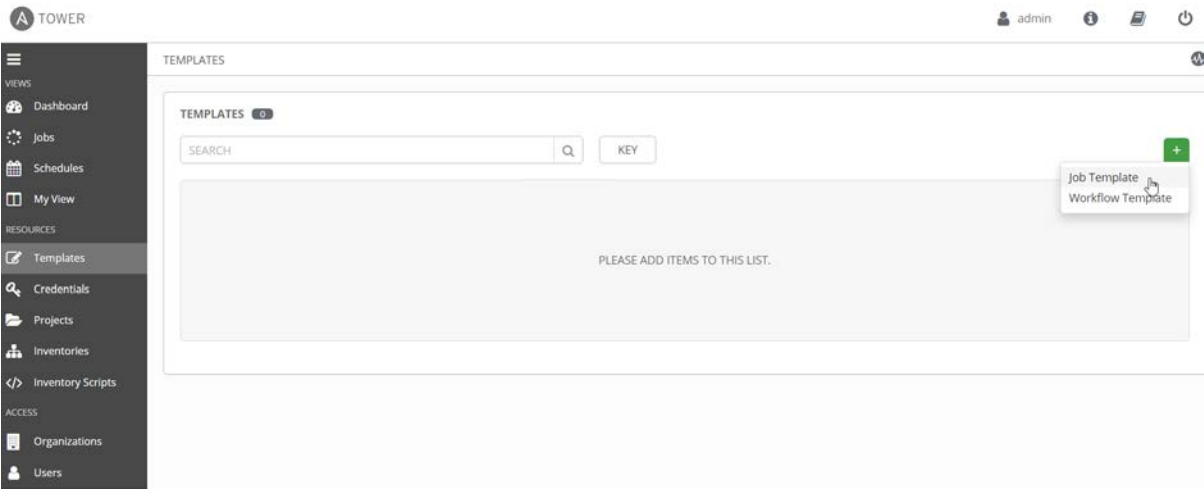


Figure D15. Select job template

- 2. Provide the following details:
 - a. Provide a name for the template (can be same as the Ansible play name).
 - b. Select the job type as **Run**.
 - c. Select the newly created inventory with name 'OCP3.11-Nimble host'.
 - d. Select the project created earlier.
 - e. Add the credentials '**Vault**' and '**RHVM**'.
 - f. Select the playbook `scalable/nimble/playbooks/deployVM.yml` and save the template as shown in Figure D16.

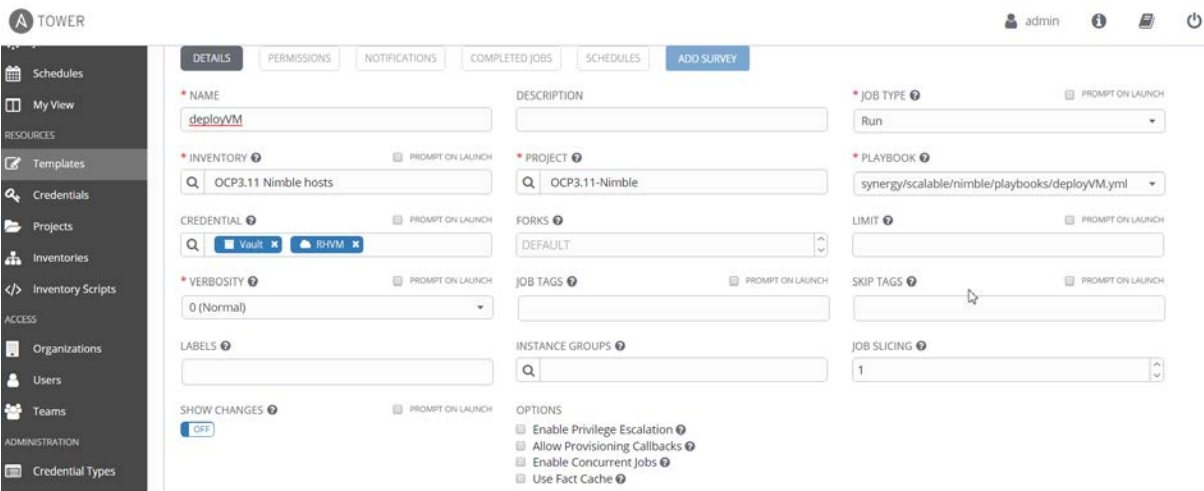


Figure D16. Import a play as template

- 3. Similarly import all the plays under `/var/lib/awx/projects/hpe-solutions-openshift/synergy/scalable/nimble/playbooks` as templates.



4. Import the following plays under the project 'Openshift-Ansible-3.11'. Select the credentials "Redhat Subscription" and "Ansible SSH User" and save the template as shown in Figure D17.

- playbooks/prerequisites.yml
- playbooks/deploy_cluster.yml

The screenshot shows the 'deployVM' template configuration page in Ansible Tower. The left sidebar contains navigation links for Views (Dashboard, Jobs, Schedules, My View) and Resources (Templates, Credentials, Projects, Inventories, Inventory Scripts, Access, Organizations, Users, Teams). The main content area is titled 'TEMPLATES / deployVM' and includes tabs for DETAILS, PERMISSIONS, NOTIFICATIONS, COMPLETED JOBS, SCHEDULES, and ADD SURVEY. The configuration fields are as follows:

- NAME:** deploy_cluster
- DESCRIPTION:** (empty)
- JOB TYPE:** Run
- INVENTORY:** OCP3.11 Nimble hosts
- PROJECT:** Openshift-Ansible-3.11
- PLAYBOOK:** playbooks/deploy_cluster.yml
- CREDENTIAL:** Redhat Subscription, Ansible SSH User
- FORKS:** DEFAULT
- LIMIT:** (empty)
- VERBOSITY:** 0 (Normal)
- JOB TAGS:** (empty)
- SKIP TAGS:** (empty)
- LABELS:** (empty)
- INSTANCE GROUPS:** (empty)
- JOB SLICING:** 1

Figure D17. Import the plays

Create workflow templates in Ansible Tower

To create a workflow for the OCP deployment, perform the following steps:

1. From the navigation bar, select **Templates**.
2. Click the + icon and select **Workflow Template** as shown in Figure D18.

The screenshot shows the 'TEMPLATES' page in Ansible Tower. The left sidebar is the same as in Figure D17. The main content area is titled 'TEMPLATES' and includes a search bar and a 'KEY' button. A list of templates is displayed:

- deploy_cluster** (Job Template):
 - INVENTORY: OCP3.11 Nimble hosts
 - PROJECT: Openshift-Ansible-3.11
 - CREDENTIALS: Ansible SSH User, Redhat Subscription
 - LAST MODIFIED: 6/12/2019 2:11:48 AM by admin
- deployTemplate** (Job Template):
 - ACTIVITY: (progress bar)
 - INVENTORY: OCP3.11 Nimble hosts
 - PROJECT: OCP3.11-Nimble
 - CREDENTIALS: RHVM, Vault
 - LAST MODIFIED: 6/13/2019 2:37:21 AM by admin
 - LAST RAN: 6/13/2019 2:37:21 AM

A dropdown menu is open in the top right corner, showing 'Job Template' and 'Workflow Template' options.

Figure D18. Create workflow template



3. Provide a name for the template, select the organization and inventory, and then click **Save** as shown in Figure D19.

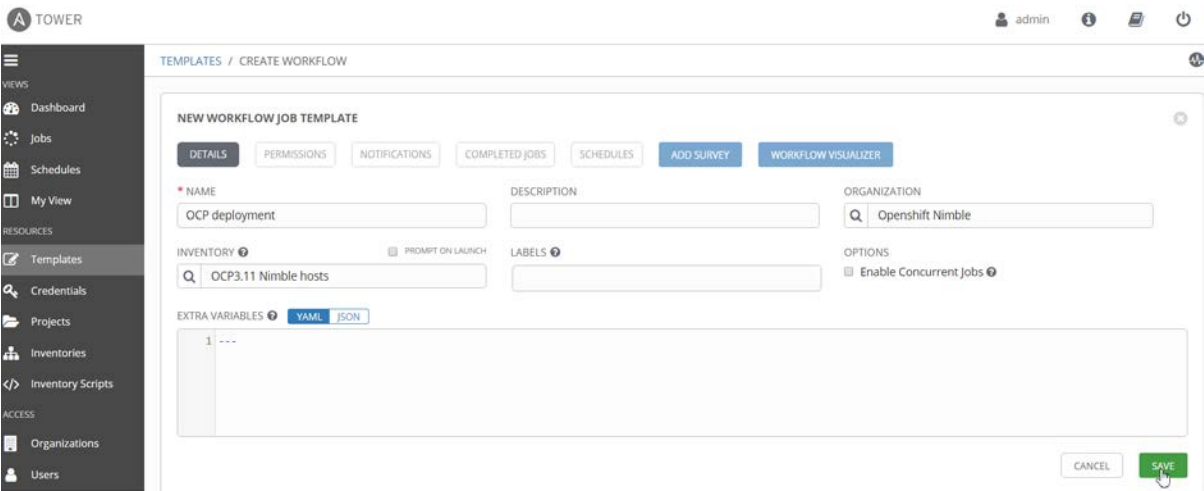


Figure D19. Configure workflow template

The workflow visualizer appears.

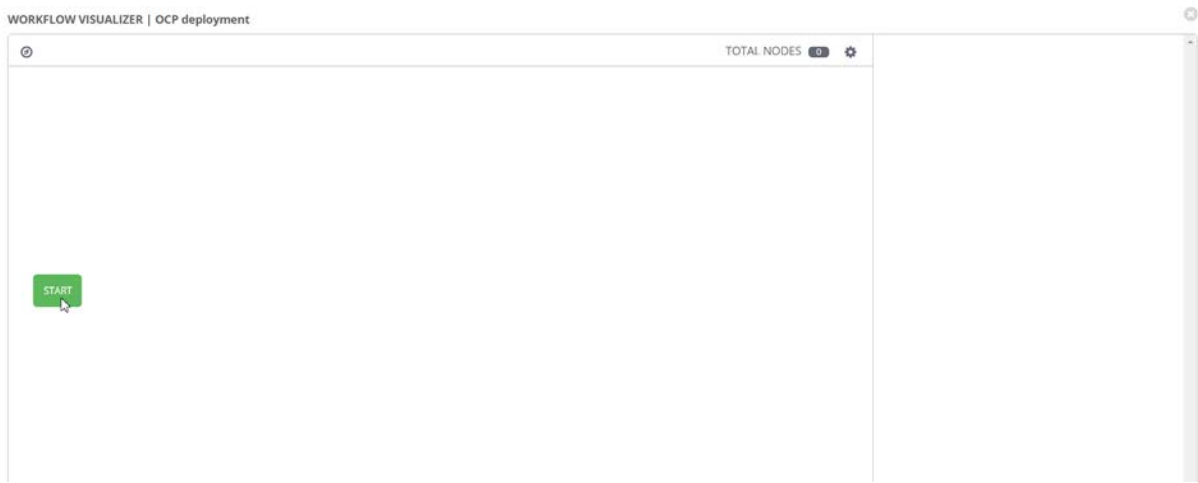


Figure D20. Workflow visualizer

4. Click **Start** to add templates to the workflow as shown in Figure D20.



5. From the right drop-down list, select the “nworkers” template as shown in Figure D21.

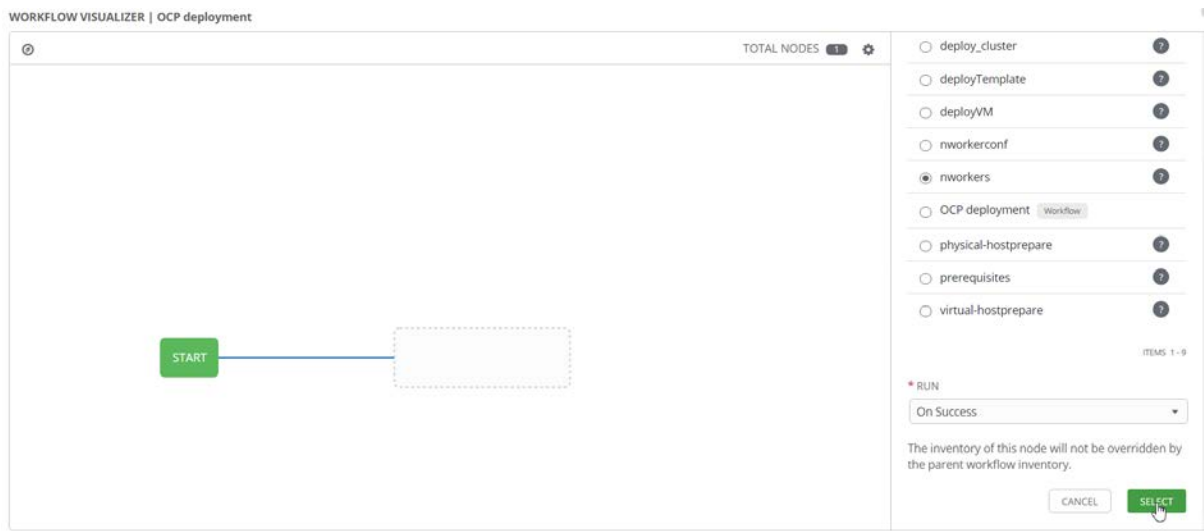


Figure D21. Adding templates to workflow visualizer

6. Click the + icon on the template box to add next play/template. Make sure “Run” is set to “On Success” as shown in Figure D22.

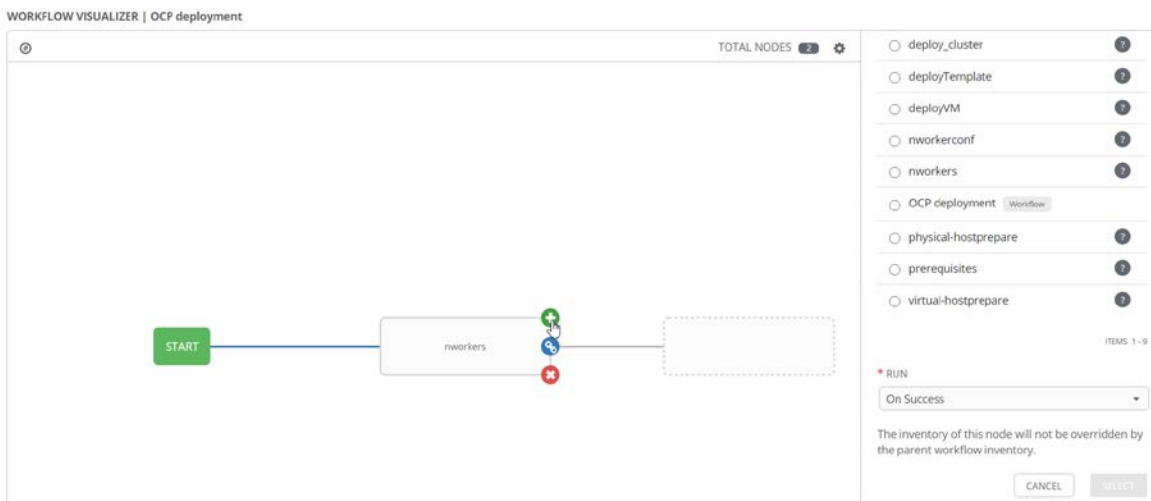


Figure D22. Ansible Tower adding workflow visualizer run status

7. Similarly, add the “nworker.conf”, “deploy_Template”, “deployVM”, “HostPrepare”, “prerequisite”, and “deploy_cluster” templates in order as shown in Figure D23 to the workflow and click **Save**.



Figure D23. Ansible Tower workflow with all templates for OCP deployment



8. To deploy OCP from the workflow template, click the **“Start a job”** icon as in Figure D24.

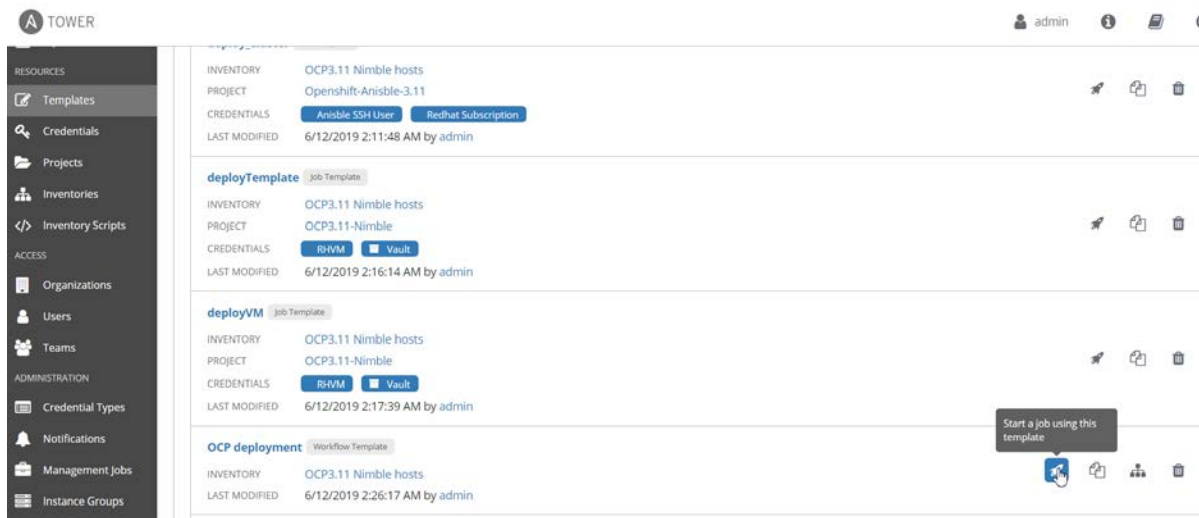


Figure D24. Execute a job

Appendix E – Setting up Prometheus Cluster Monitoring

OpenShift Container Platform (OCP) ships with a pre-configured and self-updating monitoring stack that is based on the Prometheus open source project and its wider eco-system. The stack provides monitoring of cluster components and ships with a set of alerts to immediately notify the cluster administrator about any occurring problems and a set of Grafana dashboards.

In this solution, cluster monitoring uses persistent storage to store metrics to a persistent volume and can survive a pod being restarted or recreated.

By default, Prometheus cluster is created when OCP is installed by running the **deploy_cluster** play. However, Prometheus Monitoring is deployed without persistent storage for metrics. Hence to setup Prometheus cluster with persistent storage, follow the steps:

1. Add the following parameters to the host file under **[OSEv3:vars]** section:

```
openshift_cluster_monitoring_operator_install=true
openshift_cluster_monitoring_operator_node_selector={"node-role.kubernetes.io/infra": "true"}
openshift_cluster_monitoring_operator_prometheus_storage_enabled=true
openshift_cluster_monitoring_operator_alertmanager_storage_enabled=true
openshift_cluster_monitoring_operator_prometheus_storage_capacity=100Gi
openshift_cluster_monitoring_operator_alertmanager_storage_capacity=20Gi
openshift_cluster_monitoring_operator_prometheus_storage_enabled=true
openshift_cluster_monitoring_operator_alertmanager_storage_enabled=true
```

2. Run the following play to install Prometheus Monitoring:

```
# ansible-playbook -i hosts /usr/share/ansible/openshift-ansible/playbooks/openshift-monitoring/config.yml --ask-vault-pass -e@vault_pass.yml
```



3. After successful installation of Prometheus Monitoring, log in to the Grafana web console using the route which generates a URL for Grafana as shown in Figure E1.

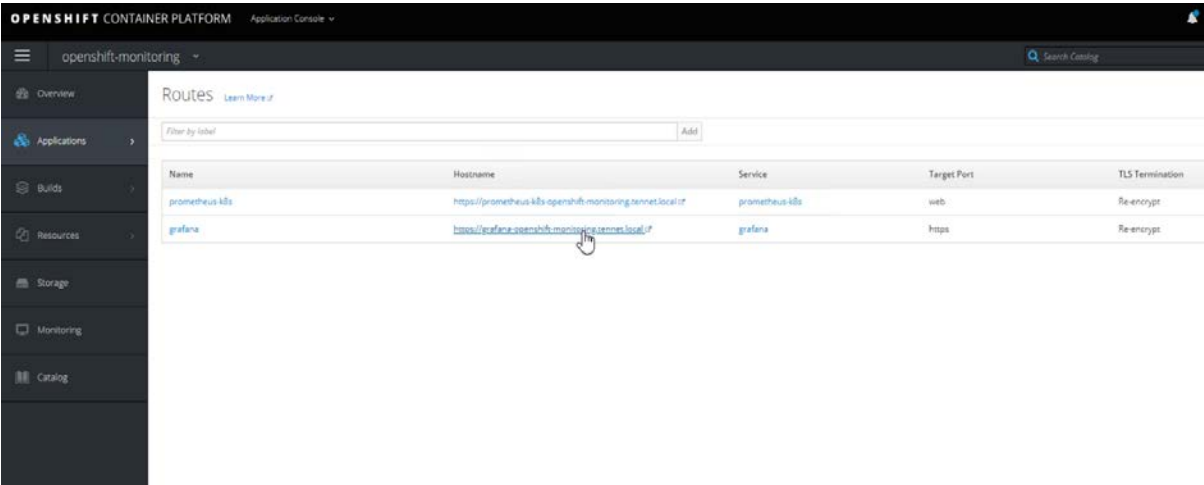


Figure E1. Grafana route

- 4. Click **Log in with OpenShift** to log in to the console using the OCP administrator login credentials created while installing OCP.
- 5. Once logged in to Grafana, a Home Dashboard as shown in Figure E2 is displayed.

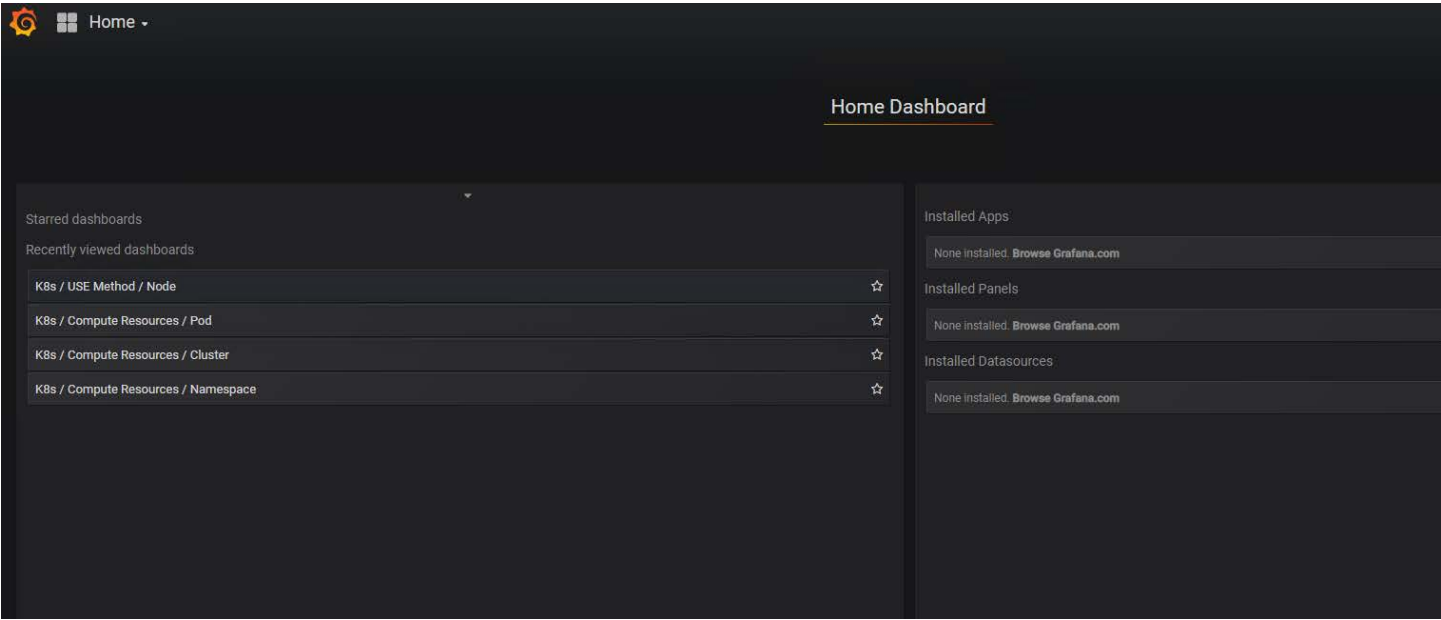


Figure E2. Grafana home page



6. From the navigation bar, select **Manage** which shows the available general monitoring parameters as shown in Figure E3.

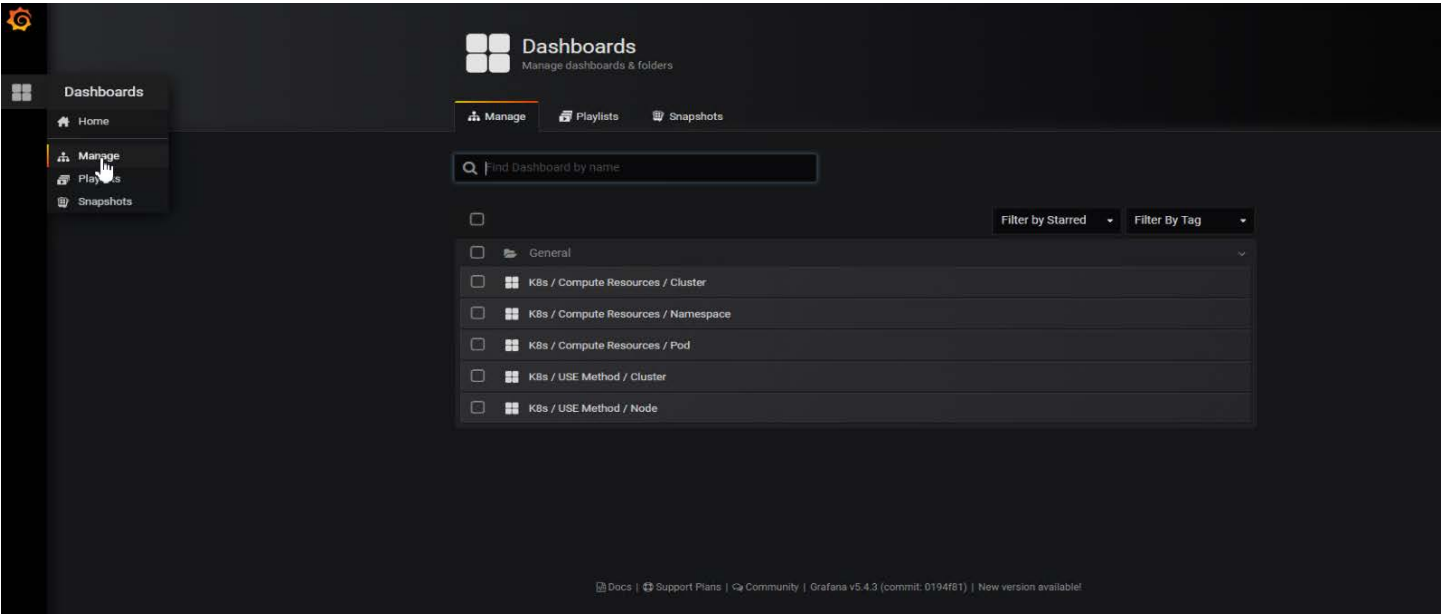


Figure E3. Prometheus data source

7. Select **K8s/Compute Resources/Cluster** under **Manage** as shown in Figure E3.

The **K8s/Compute Resources/Cluster** dashboard displays the usage of compute resources as shown in Figure E4.



Figure E4. Prometheus data source



8. Once Prometheus Cluster Monitoring is installed, the resources usage can be monitored from the **Monitoring** navigation bar as in Figure E5.

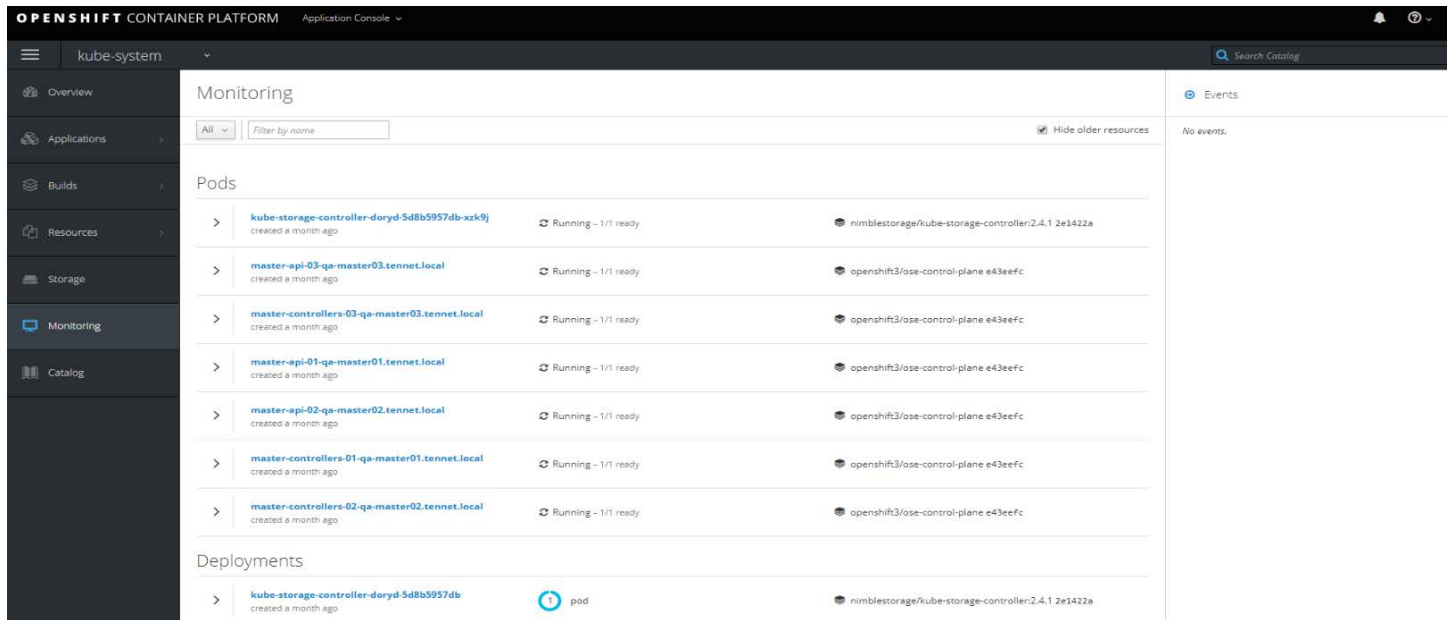


Figure E5. Monitoring the resource usage from OCP navigation bar

Appendix F – Aggregating container logs using EFK

Installer can deploy Elasticsearch + Fluentd + Kibana (EFK) stack to aggregate logs from hosts and applications for a range of OpenShift Container Platform services.

To aggregate container logs using EFK with persistent Elasticsearch storage, follow the steps:

1. Add the following parameters to the host file under '[OSEv3:vars]' section:

```
openshift_logging_install_logging=true
openshift_logging_es_pvc_dynamic=true
openshift_logging_es_pvc_size=200Gi
openshift_logging_elasticsearch_storage_type=pvc
openshift_logging_es_pvc_prefix=oc-efk-log
openshift_logging_es_cluster_size=3
openshift_logging_es_node_selector={"node-role.kubernetes.io/infra": "true"}
openshift_logging_kibana_node_selector={"node-role.kubernetes.io/infra": "true"}
openshift_logging_curator_node_selector={"node-role.kubernetes.io/infra": "true"}
openshift_logging_fluentd_node_selector={"node-role.kubernetes.io/infra": "true"}
openshift_logging_es_number_of_replicas=3
openshift_logging_es_allow_external=true
openshift_logging_es_hostname=es.router.tennet.local
openshift_logging_kibana_hostname=kibana.router.tennet.local
```

2. Run the following play to install EFK stack:

```
# ansible-playbook -i hosts /usr/share/ansible/openshift-ansible/playbooks/openshift-logging/config.yml --ask-
vault-pass -e@vault_pass.yml
```



Note

You can also deploy EFK stack, along with the OCP deployment, using the `deploy_cluster.yml` play. However, the PVC for the persistent Elasticsearch storage will not be created until the NLT is installed and storage class is created. Hence, it is recommended to run the EKF stack play separately after installing OCP.

- 3. After successful installation of EFK stack, log in to the Kibana web console using the Kibana hostname provided in the host file, accessible at `https://<kibana hostname>`.
- 4. Click **Log in with OpenShift** as shown in Figure F1 to log in to the console using the OCP administrator login credentials created while installing OCP.

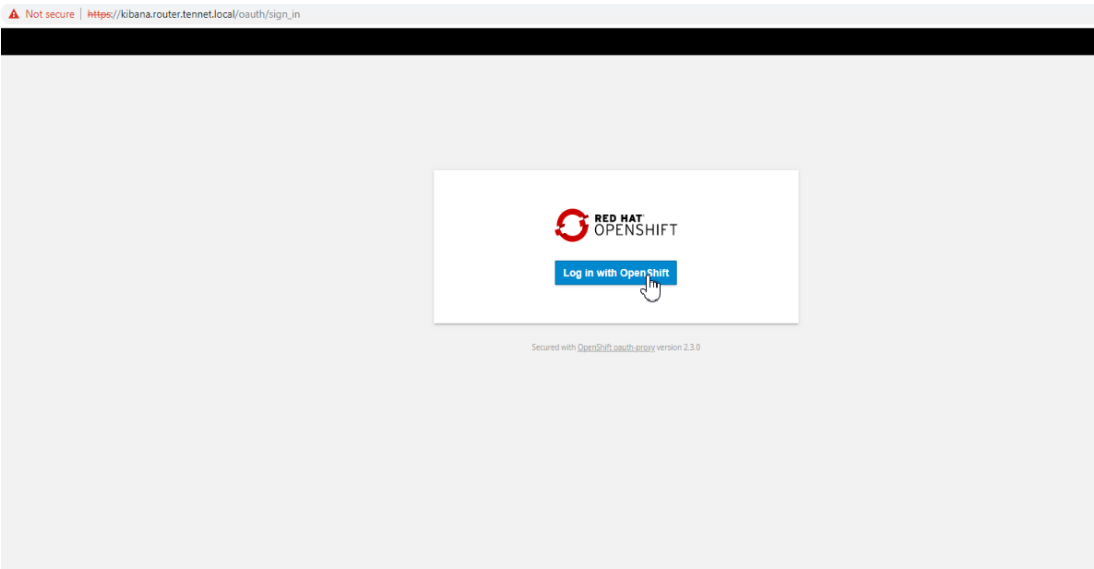


Figure F1. Kibana login

- 5. You can view the OCP logs from Kibana dashboard as shown in Figure F2.

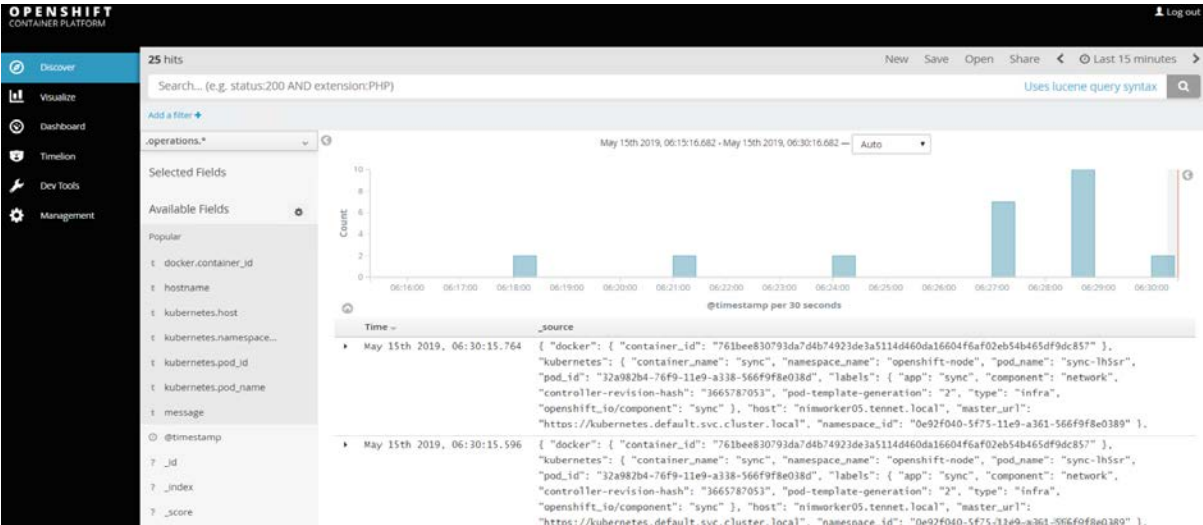


Figure F2. Kibana dashboard



Appendix G: Assessing the security posture of Red Hat OpenShift Container Platform using the automated CIS Kubernetes benchmark with the Kube-Bench utility

Application container technologies are a form of operating system virtualization combined with application software packaging. Containers provide a portable, reusable, and automatable way to package and run applications. Understanding the container technology architecture and lifecycle leads to the following security challenges:

- The entire container infrastructure is at a scale that makes it difficult to ensure no vulnerable images are being deployed in production.
- Scale can make it challenging for operations to assess the compliance posture of containers and Kubernetes environments.
- There is a lack of visibility into container infrastructure and security incidents at runtime.
- Inspecting containers after they are gone is not possible.
- Too many images in the registry and identifying which images have critical vulnerabilities that require a fix is important.
- Keeping track of secrets and credentials exposed by an image among thousands of images is complicated and time consuming.
- Identifying if an image is exposing any blacklisted ports is important to stop backdoor entry for the hacker.
- Tracking licenses and their types used by an image is critical.
- Performing compliance checks on each container to identify any compliance violations must be done.
- Performing regular health check on the containers is a core requirement.

To address these container security challenges for Red Hat OpenShift Container Platform, this document proposes a solution that uses the kube-bench utility to secure and monitor Red Hat OpenShift Container Platform, an enterprise-ready container platform installed and configured on HPE Synergy Composable Infrastructure.



Figure G1 shows the solution diagram for assessing the security posture of Red Hat OpenShift Container Platform on HPE Synergy using kube-bench. It begins with the user configuring the Red Hat OpenShift Container Platform on HPE Synergy Composable Infrastructure to enable access to the OpenShift cluster for kube-bench. Kube-bench is a Go application that checks whether Kubernetes, which is at the core of OpenShift, is deployed securely by running the checks documented in the CIS Kubernetes Benchmark.

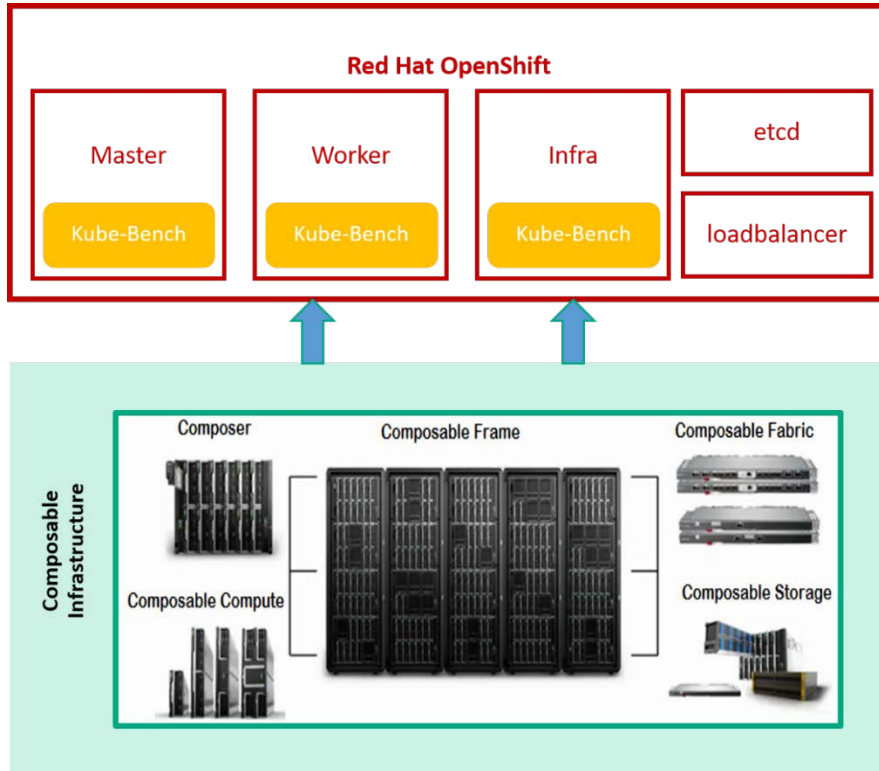


Figure G1. Solution Overview

To install kube-bench, use the repository at <https://github.com/HewlettPackard/hpe-solutions-openshift/tree/master/synergy/scalable/security>.

About

This repo contains Ansible plays and scripts to automate the installation of kube-bench on the Red Hat OpenShift Container Platform 3.11 nodes.

Contents of the repo

- **playbooks:** This folder contains the playbook required for kube-bench installation.
- **roles:** This folder contains a role called "kube-bench-deploy-ocp" which is responsible for performing the actions required for kube-bench integration.
- **hosts:** This is the host file which will be used by Ansible Engine to reference hosts during kube-bench deployment. Update the OCP master nodes, worker nodes, and infra nodes complete host name in this file.
- **site.yaml:** This file will import the playbook "kube-bench-deployment.yaml" that defines the workflow for kube-bench integration.

Prerequisites

- Red Hat OpenShift Container Platform 3.11 is up and running.
- All of the nodes in the Red Hat OpenShift Container Platform 3.11 deployment are virtual nodes running Red Hat Enterprise Linux 7.6.
- The user has access to the internet to clone the public GitHub repository for kube-bench and go on each node.



Custom attributes\variable files and plays

Each playbook has a role associated with it. Each role has a set of tasks under the "task" folder. This file contains the automated steps for golang, kube-bench, running kube-bench, and storing results back to Ansible Engine node.

```
security-kube-bench/roles/kube-bench-deploy-ocp/tasks/main.yml
```

How to use

1. Clone the repository to the Ansible Engine host using the following command:

```
# git clone https://github.com/hewlettpackard/hpe-solutions-openshift/
```

2. From the Ansible Engine command prompt, browse the kube-bench playbooks in the clone directory:

```
# cd hpe-solutions-openshift\synergy\scalable\security-kube-bench
```

3. Open and edit the "hosts" file in an editor by typing the following command. Provide the IP or fully qualified domain name of the master, infrastructure, and worker nodes as shown in Figure G2:

```
# vi hosts
```

```
[masters]
secnmaster01.tennet.local
secnmaster02.tennet.local

[workers]
secnworker01.tennet.local
secnworker02.tennet.local

[infra]
secninfra01.tennet.local
secninfra03.tennet.local
```

Figure G2. Host file for kube-bench integration

4. Once the host file has been edited, run the play using the following command:

```
# ansible-playbook -i hosts site.yml
```

5. Once the playbook has completed the execution on the Ansible Engine, browse the "/tmp/" directory and check all the log files generated from each of the masters, workers, and infra nodes specified in hosts file are available in the directory as shown in Figure G3.

```
# cd /tmp
```

```
[root@zsec-ansibleeng tmp]# ls
kube-bench-secninfra01.tennet.local  kube-bench-secnworker01.tennet.local
kube-bench-secninfra03.tennet.local  kube-bench-secnworker02.tennet.local
kube-bench-secnmaster01.tennet.local
kube-bench-secnmaster02.tennet.local
[root@zsec-ansibleeng tmp]#
[root@zsec-ansibleeng tmp]#
```

Figure G3. Log files on the Ansible Engine

6. Depending on the number of nodes specified by the user in the hosts file, same number of log files should be generated.



7. Each log file lists out the details of the CIS Benchmark rules against the node tested as in Figure G4.

```
[INFO] 1 Securing the OpenShift Master
[INFO] 1 Protecting the API Server
[INFO] 1.1 Maintain default behavior for anonymous access
[PASS] 1.2 Verify that the basic-auth-file method is not enabled
[INFO] 1.3 Insecure Tokens
[PASS] 1.4 Secure communications between the API server and master nodes
[PASS] 1.5 Prevent insecure bindings
[PASS] 1.6 Prevent insecure port access
[PASS] 1.7 Use Secure Ports for API Server Traffic
[INFO] 1.8 Do not expose API server profiling data
[PASS] 1.9 Verify repair-malformed-updates argument for API compatibility
[PASS] 1.10 Verify that the AlwaysAdmit admission controller is disabled
[FAIL] 1.11 Manage the AlwaysPullImages admission controller
```

Figure G4. Log files for an OpenShift master node

8. For each failed test, a set of remediation steps will be shown in the logs as in Figure G5.

```
== Remediations ==
1.11 Edit the kubernetes master config file /etc/origin/master/master-config.yaml
and add the the entry below.

admissionConfig:
  pluginConfig:
    AlwaysPullImages:
      configuration:
        kind: DefaultAdmissionConfig
        apiVersion: v1
        disable: false

1.15 Edit the Openshift master config file /etc/origin/master/master-config.yaml, update the followi
ng entry and restart the API server.

auditConfig:
  auditFilePath: "/etc/origin/master/audit-ocp.log"
  enabled: true
  maximumFileRetentionDays: 30
  maximumFileSizeMegabytes: 10
  maximumRetainedFiles: 10
```

Figure G5. Remediation steps

9. The end of log file shows a summary of results.



Change Tracker

Version	Release Date	Changes
3.0	07/03/2019	Initial release
3.0.1	07/15/2019	Updated document, fixed formatting issues, minor grammar fixes, addition of multiple appendices covering Ansible Tower, Prometheus, log aggregation with EFK and Kube-bench.
3.0.2	08/06/2019	Revised golden image creation, updates to Table 11 and Figure 20 to enhance clarity and readability, revised the section on importing artifact bundles into HPE Synergy Image Streamer, substantial updates to worker node processes and multiple URL fixes.
3.0.3	08/06/2019	Substantial revision of the appendix focused on kube-bench.
3.0.4	08/30/2019	Updates to numerous instructions for enhanced clarity.



Resources and additional links

Red Hat, <https://www.redhat.com>

Red Hat OpenShift Container Platform 3.11 Documentation, <https://docs.openshift.com/container-platform/3.11/welcome/index.html>

HPE Synergy, <https://www.hpe.com/info/synergy>

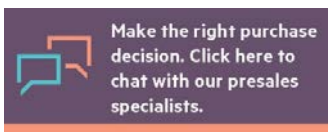
HPE Nimble Storage, <https://www.hpe.com/us/en/storage/nimble.html>

HPE Solutions for OpenShift GitHub, <https://github.com/hewlettpackard/hpe-solutions-openshift>

HPE FlexFabric 5940 switching, <https://www.hpe.com/us/en/product-catalog/networking/networking-switches/pip.hpe-flexfabric-5940-switch-series.1009148840.html>

HPE Workload Aware Security for Linux, <https://h20392.www2.hpe.com/portal/swdepot/displayProductInfo.do?productNumber=WASL>

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



Sign up for updates

© Copyright 2019 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Red Hat® is a registered trademark of Red Hat, Inc. in the United States and other countries. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries. Intel is a trademark of Intel Corporation in the U.S. and other countries.

All other third-party trademark(s) is/are the property of their respective owner(s).

OCP3801, September 2019, Version 3.0.4

