



Red Hat OpenShift Container Platform on HPE Synergy with HPE Nimble Storage

Implementing a resilient on-premises container solution

Contents

Overview.....	3
Solution design.....	4
Solution creation process.....	7
Sizing considerations.....	7
Red Hat OpenShift Container Platform sizing.....	7
Prerequisites.....	8
Software versions.....	8
Deployment environment.....	8
Ansible Engine.....	10
Physical environment configuration.....	10
Cabling the HPE Synergy 12000 Frame and HPE Virtual Connect 40Gb SE F8 Modules for Synergy.....	13
Configuring the solution switching.....	15
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 OpenShift worker nodes.....	41
OpenShift deployment.....	54
Virtual Machine deployment and configuration.....	54
OpenShift-Ansible.....	57
Validate OpenShift deployment.....	58
Command Line validation.....	58
Ansible OpenShift deployment removal.....	62
Appendix A - Playbook variables.....	63
Appendix B - Utilizing Ansinimble.....	65
Array Setup.....	65
Appendix C - Deploying worker node functions to virtual machines.....	68
Change Tracker.....	72
Resources and additional links.....	73



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>.

HPE plans to update this document over time with enhancements to deployment methodologies as well as new software versions, features and functions. Check for the latest document at <https://github.com/HewlettPackard/hpe-solutions-openshift/tree/master/synergy/scalable/nimble>. It is recommended that the installer review this document in its entirety and understand all prerequisites prior to beginning an installation.

Figure 1 displays the minimal configuration of the deployment on 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 provide both 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.

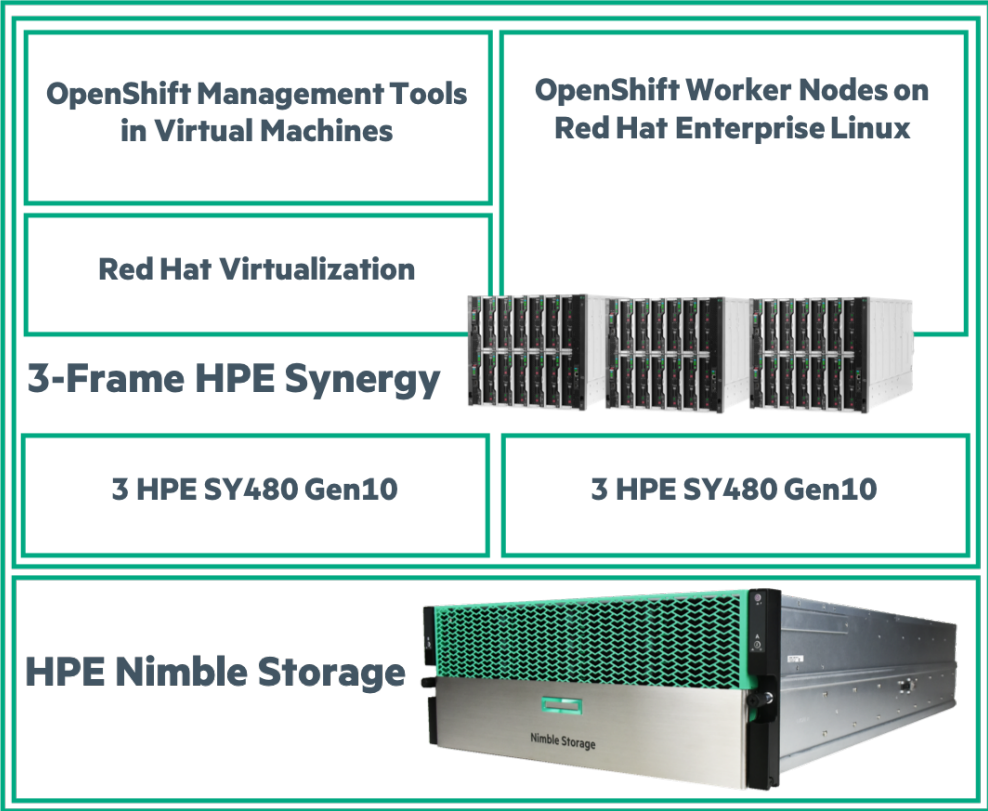


Figure 1. Solution layout

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 below 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.

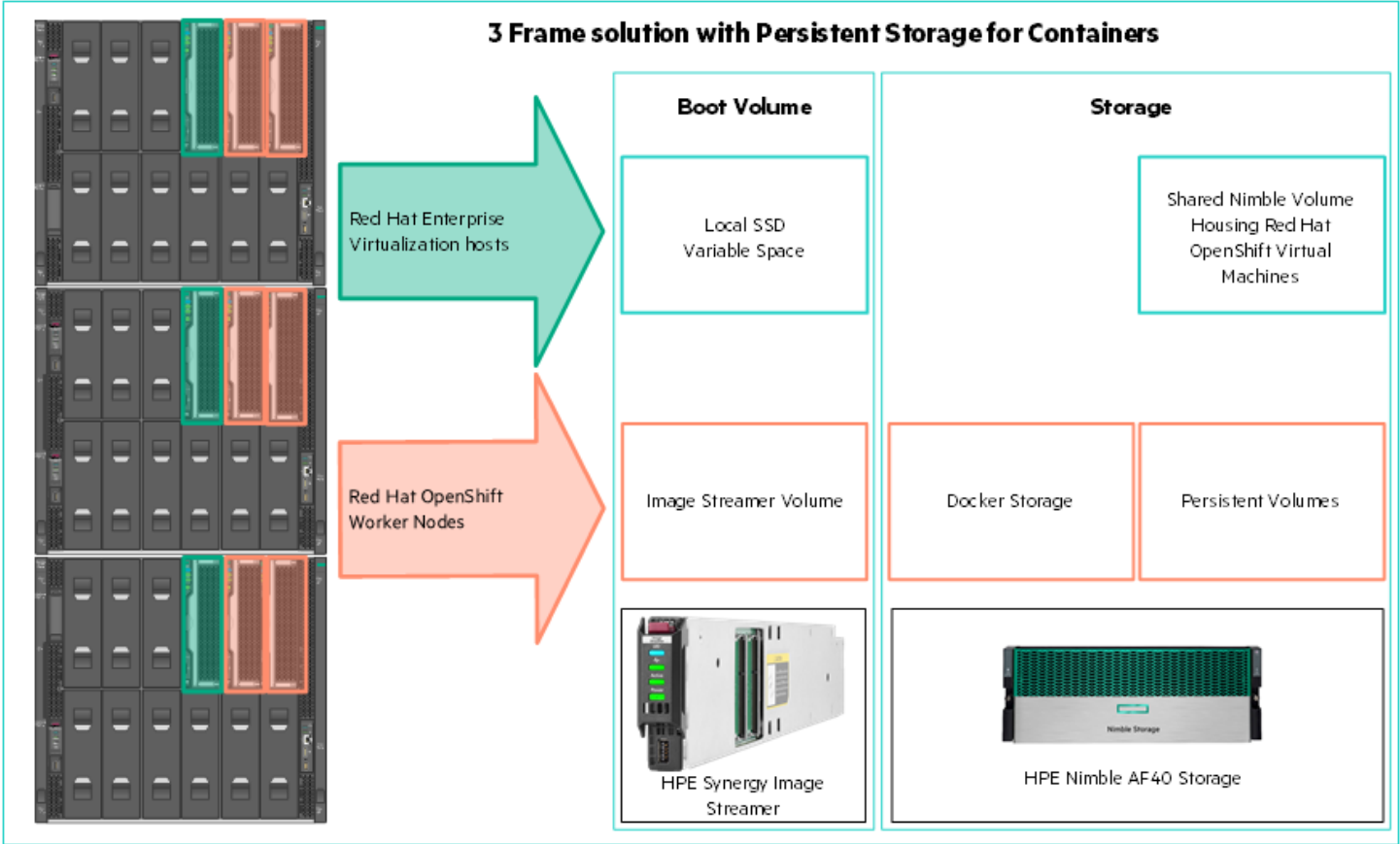


Figure 2. Solution design by function and storage type

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.

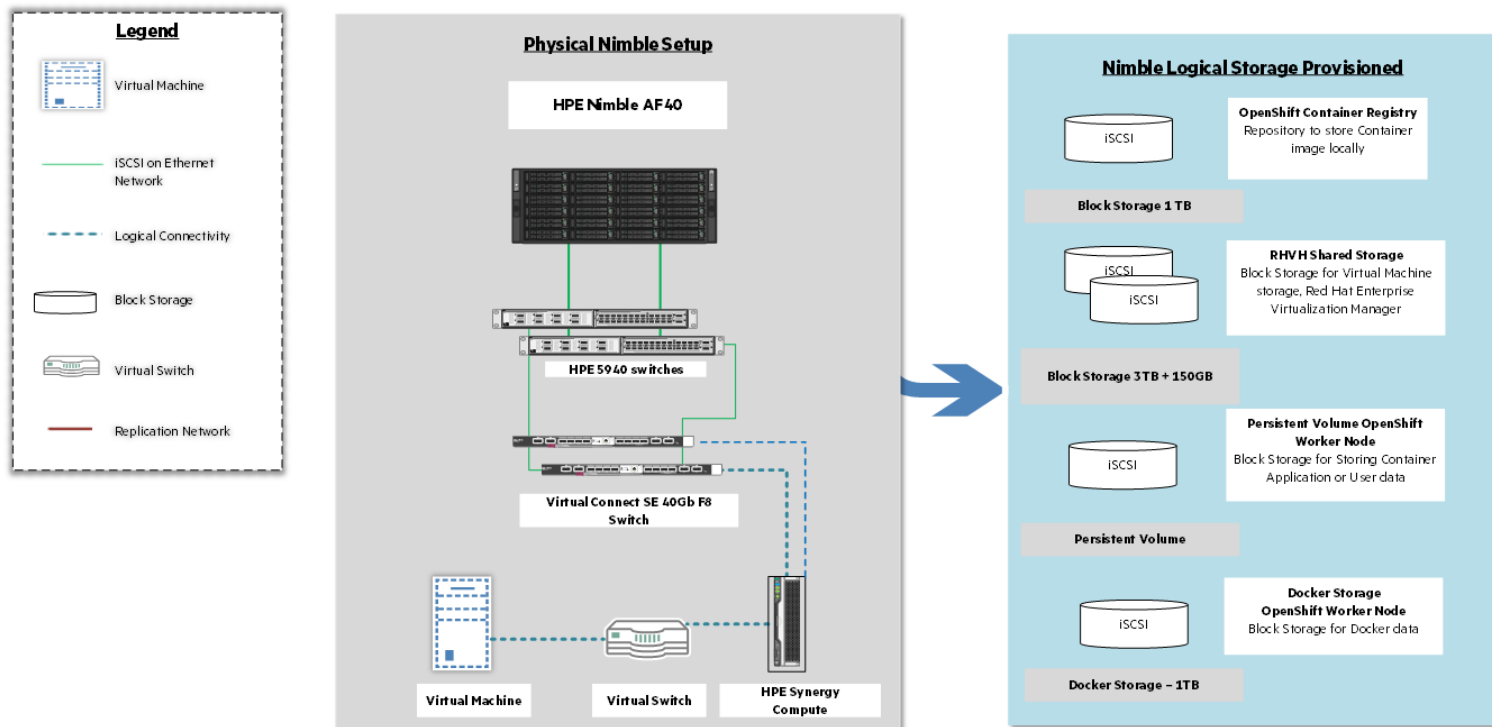


Figure 3. Logical layout of the solution stack.

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. There are five networks defined in this solution.

1. PXE - This network provides PXE boot instructions for the Red Hat Virtualization nodes.
2. Synergy Management Network which is specific to the requirements of HPE Synergy.
3. Management Network - This network facilitates the management of hardware and software interfaces by IT.
4. Datacenter 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

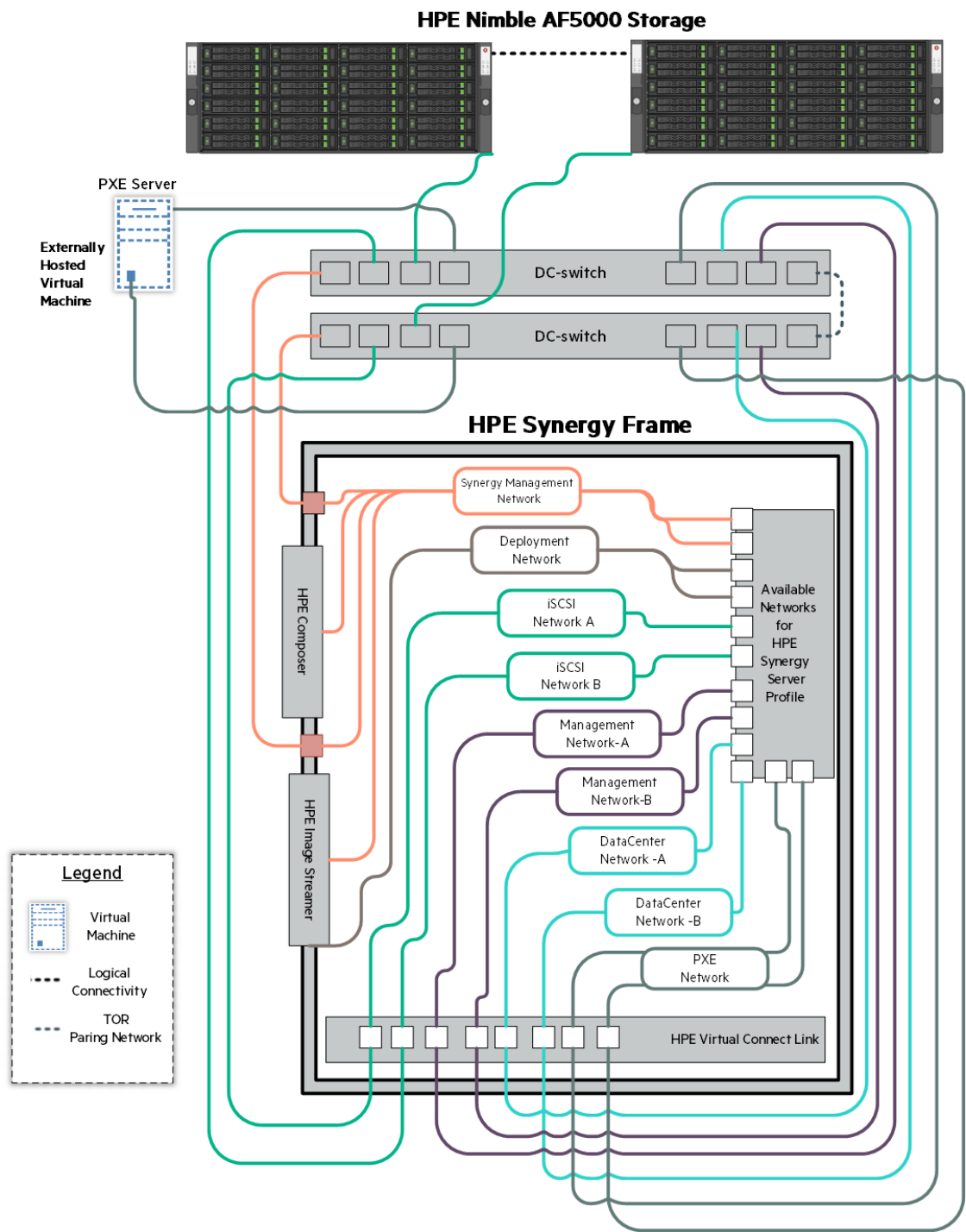


Figure 4. Physical network layout within the OpenShift solution



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 download and review this image prior to proceeding.

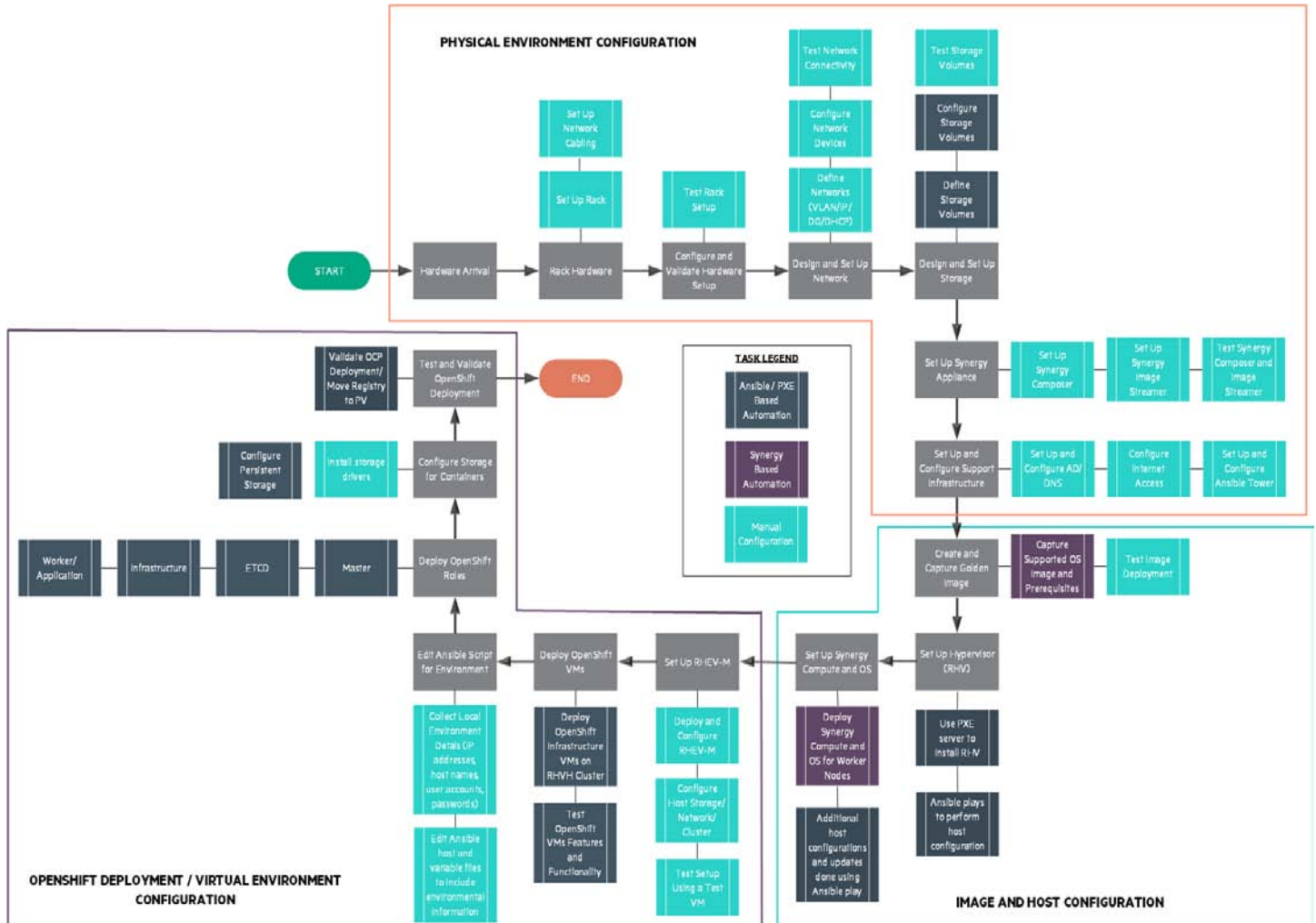


Figure 5. Solution creation flow diagram

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. In this section we will discuss 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 and Infrastructure nodes require a minimum of 1 vCPU and 8 GB RAM with a disk with at least 15 GB of space for /var/, 1 GB disk space for /usr/local/bin, and 1 GB disk space for the system's temporary directory, 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 commercially available application delivery controllers should be used for enterprise deployments.

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 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

Deployment environment

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



Services

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

Table 2. Services used in the creation of this solution.

Service	Required/Optional	Description/Notes
DNS	Required	Provides name resolution on management and data center networks, 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 will 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 `mlinuz` 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. HPE 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. Figure 5 below describes the various components involved in the solution.



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.

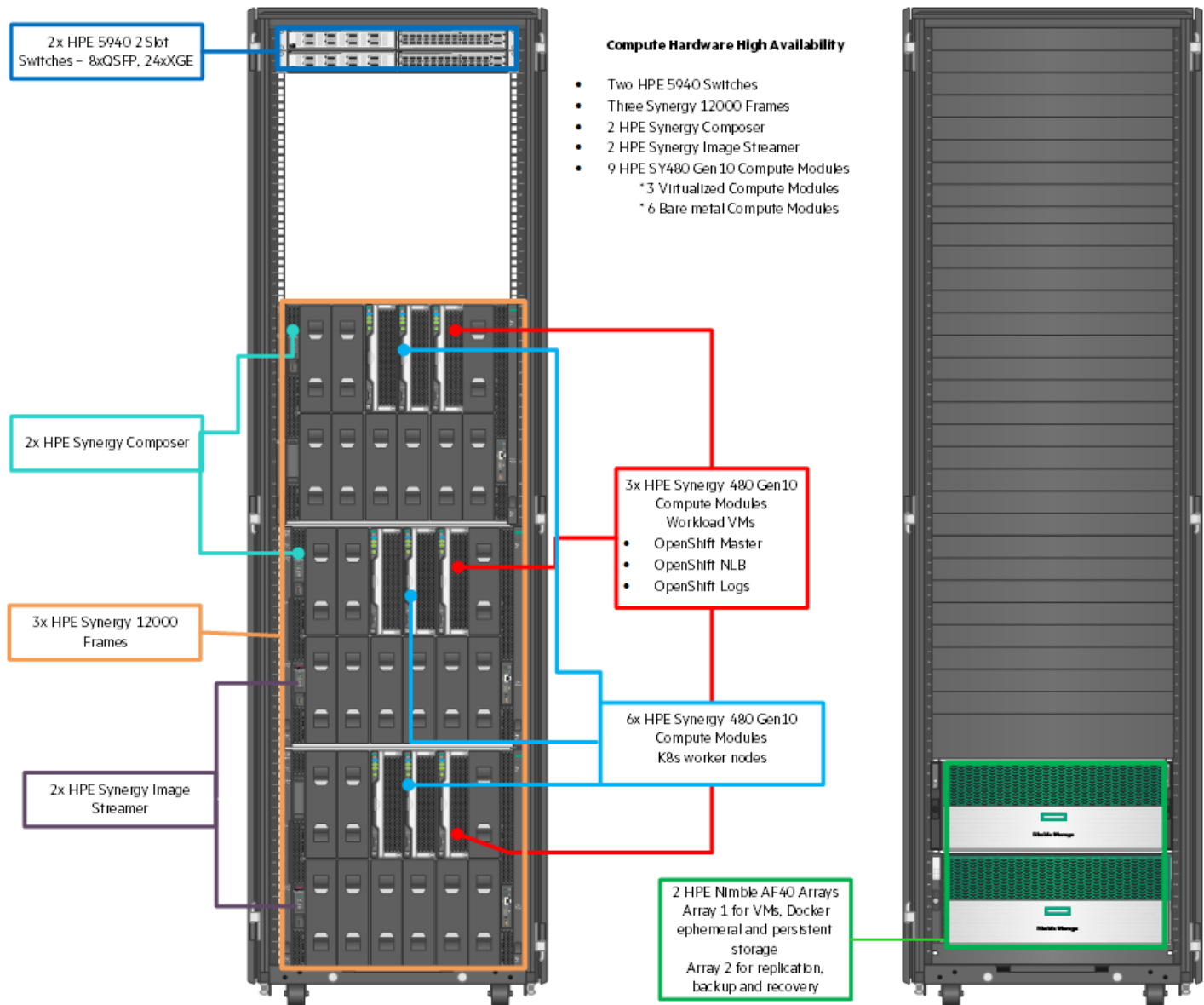


Figure 6.Physical layout of the compute within the solution

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](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>,

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 5930 modules listed below
HPE 5930 24p SFP+ and 2p QSFP+ Module	2	One module per HPE FlexFabric 2-Slot Switch
HPE 5930 8-port QSFP+ Module	2	One module per HPE FlexFabric 2-Slot Switch
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 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, Management and Intelligent Resilient Fabric (IRF) connections. These connections handle east-west network communication as well as management traffic within the solution.

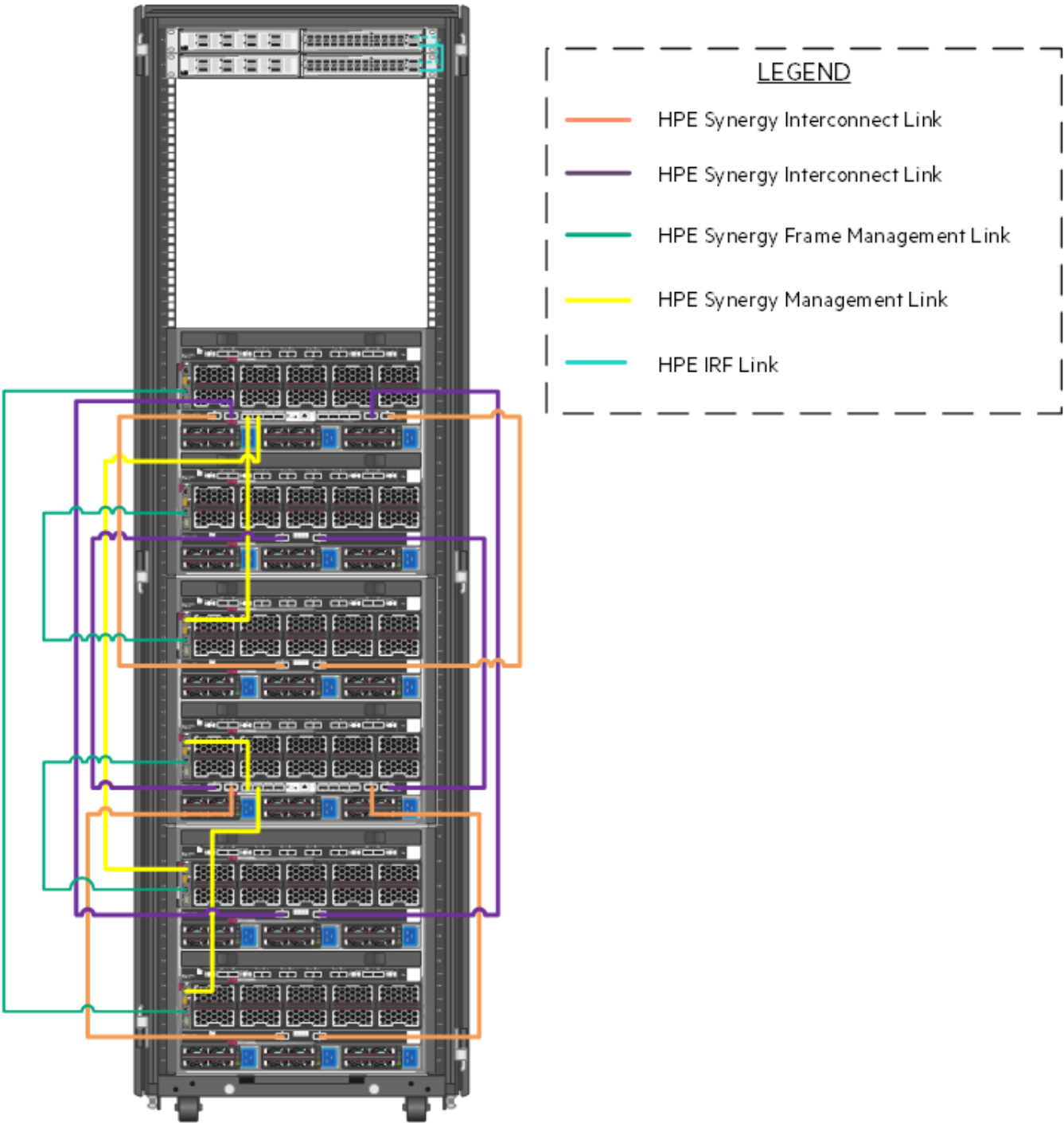


Figure 7. Cabling of the management and inter-frame communication links within the solution.



Figure 8 shows the cabling of the Synergy frames to the network switches. The specific networks contained within the Bridge-Aggregation Groups are described in more detail later in this section. At the lowest level there are 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.

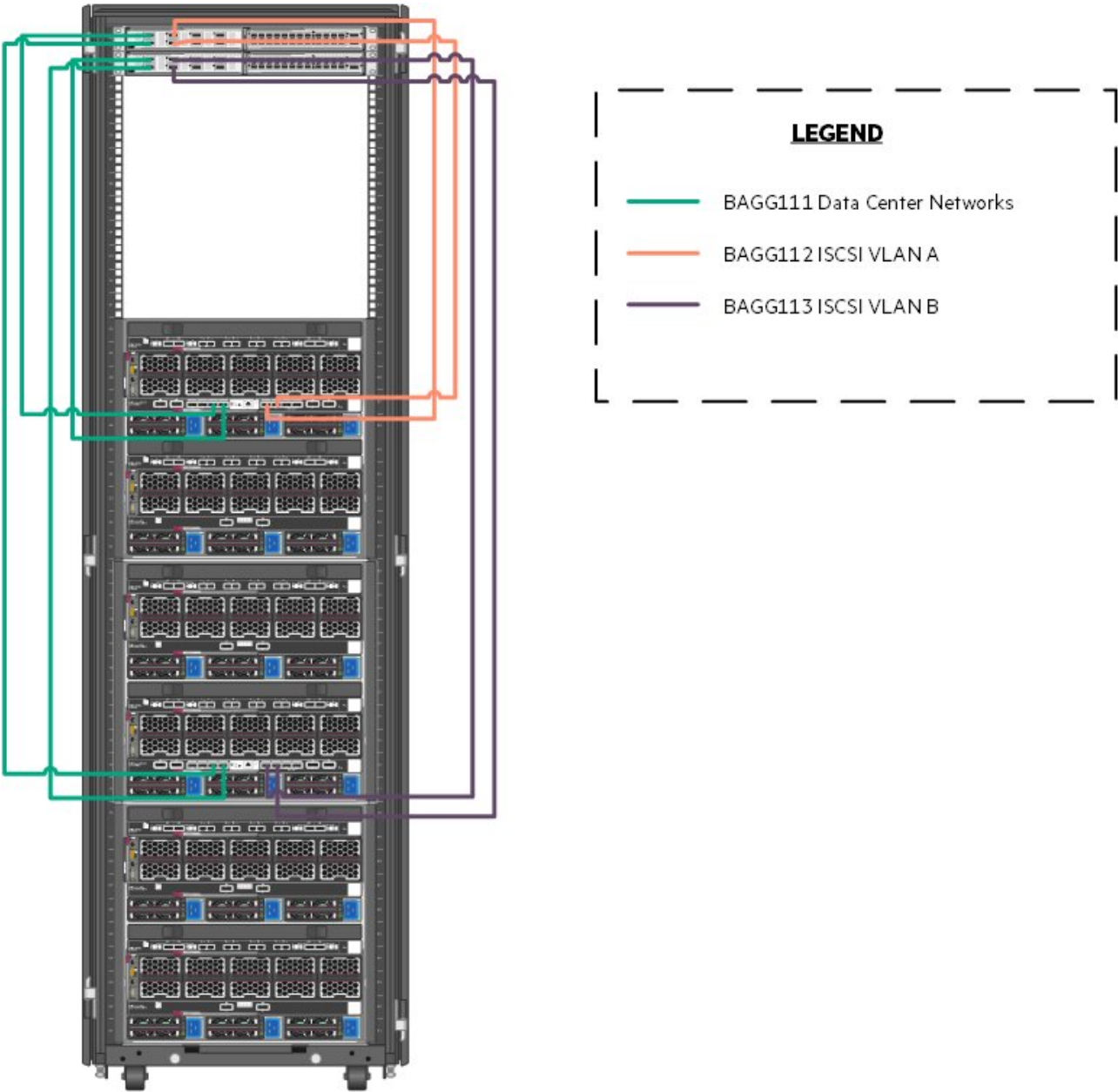


Figure 8. Cabling of the HPE Synergy 12000 Frames to the HPE FF 5940 switches.



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. To understand the process of configuring IRF consult the HPE FlexFabric 5940 Switch Series Installation Guide at https://support.hpe.com/hpsc/doc/public/display?sp4.ts.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 assumes 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.



Physical cabling

Table 5 below 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 TG1	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 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

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
```

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.

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

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

```
# interface Bridge-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
```



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
```

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
```

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
```

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

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

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
```

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



HPE Synergy 480 Gen10 Compute Modules

This section describes the connectivity of the HPE Synergy 480 Gen10 Compute Modules used in the creation of this solution. The 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 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 bridge aggregation groups are shown in Table 6 in this document.

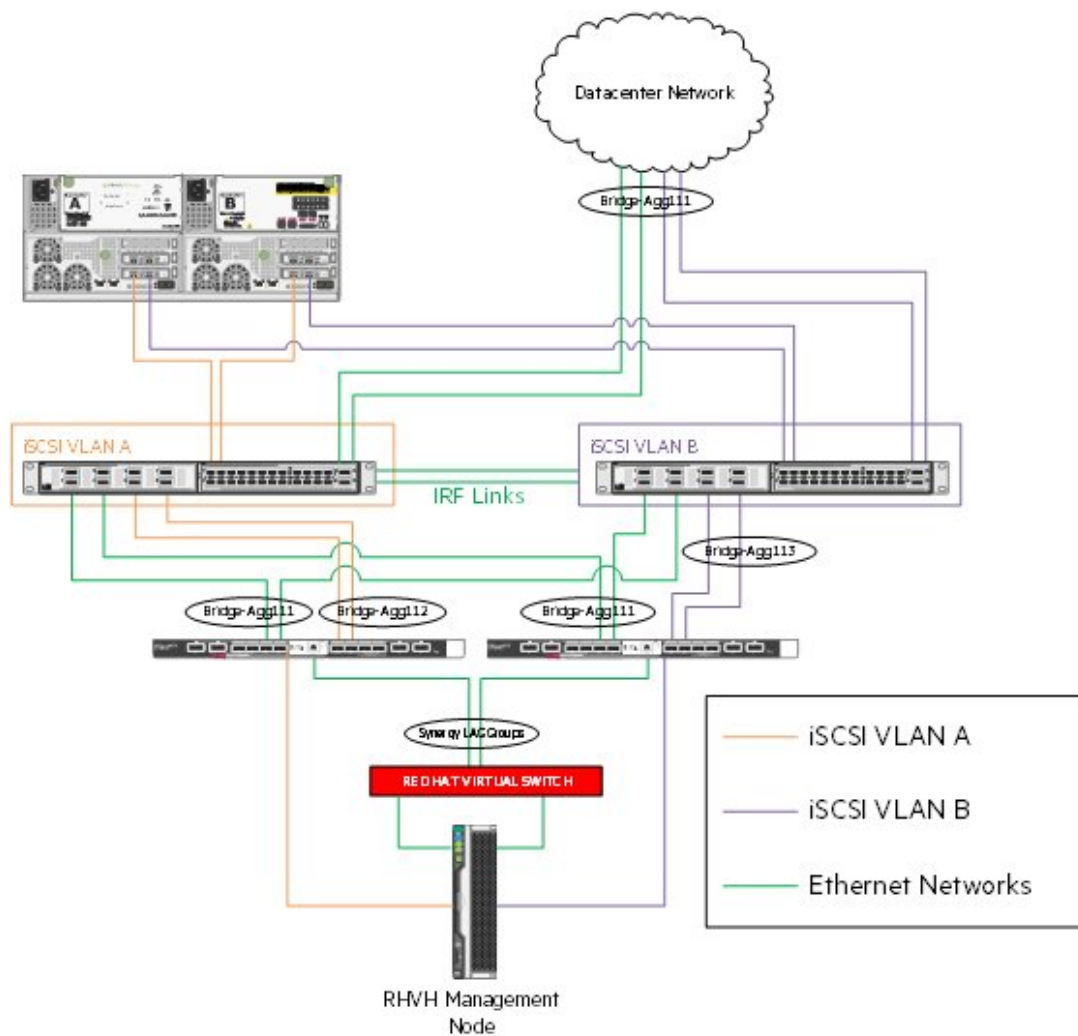


Figure 9. Red Hat Virtualization Host network connectivity

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 diagram 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 bridge aggregation group.

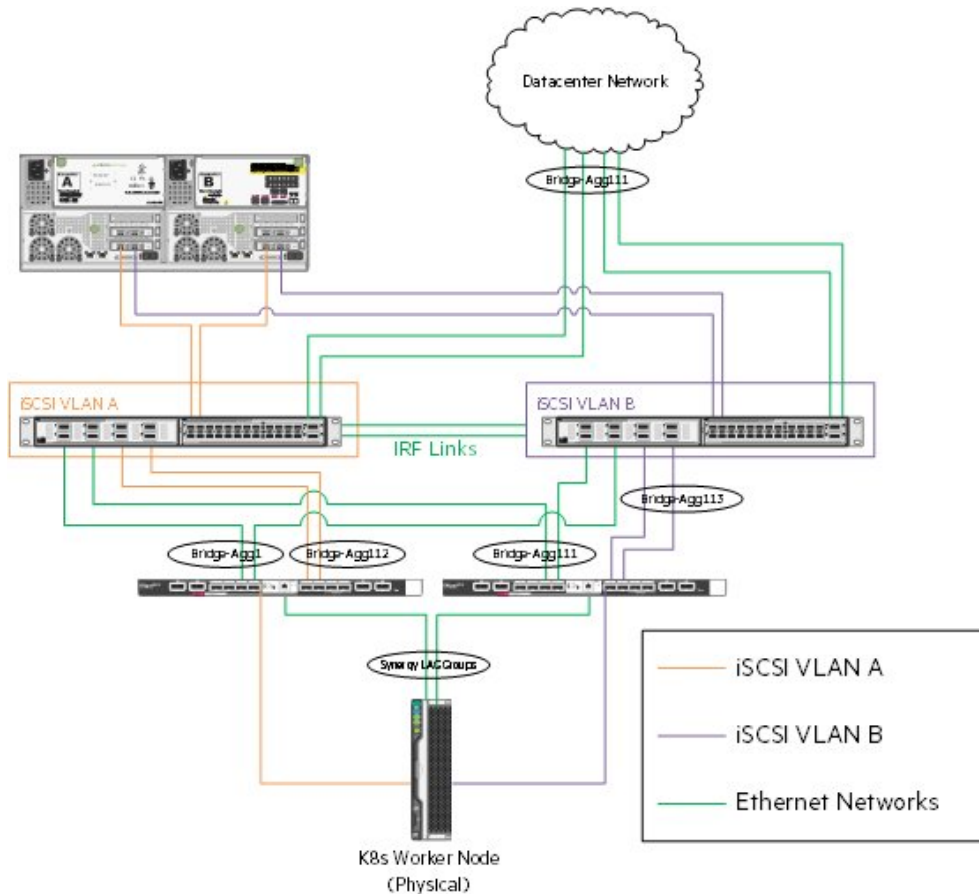


Figure 10. Red Hat OpenShift worker node network connectivity

HPE Synergy Composer

At the core of the management of the 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 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 on **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: **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 select **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.

Select the **OneView** menu at the top of the screen and select **Settings** and then on **Appliance**. Validate that both appliances are connected and show a green checkmark.

Configure appliance credentials

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

Configure solution firmware

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

1. Select the OneView menu and 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 aforementioned Converged Solutions 750 recipe. Once the bundle starts uploading you can proceed to additional steps without disrupting the upload.

Create an IP Pool on the management network

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



Configure ethernet networks

As explained in the Network Configuration portion of the switch configuration section of this document, the solution utilizes at least four (4) network segments. Use the **Create networks** section of the **Guided Setup** wizard to define the networks shown in Table 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 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, choose **Create enclosure group**.
2. Provide a name and enter the number of frames.
3. Choose **Use address pool** and utilize the management pool defined earlier.
4. Use the Logical Interconnect Group from the prior step in the creation of the Enclosure Group.

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**. Select **Cancel** to exit.

Solution storage

A 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.

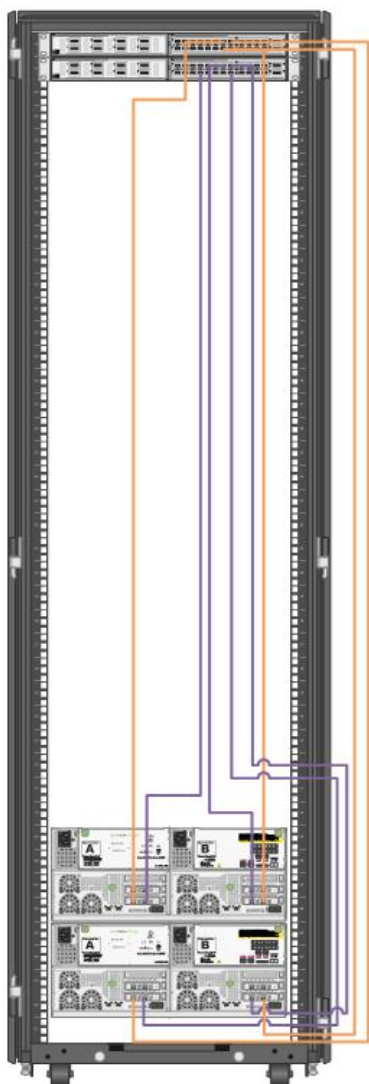


Figure 11. Cabling of the HPE Nimble Storage arrays to the HPE FF 5940 switches.



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

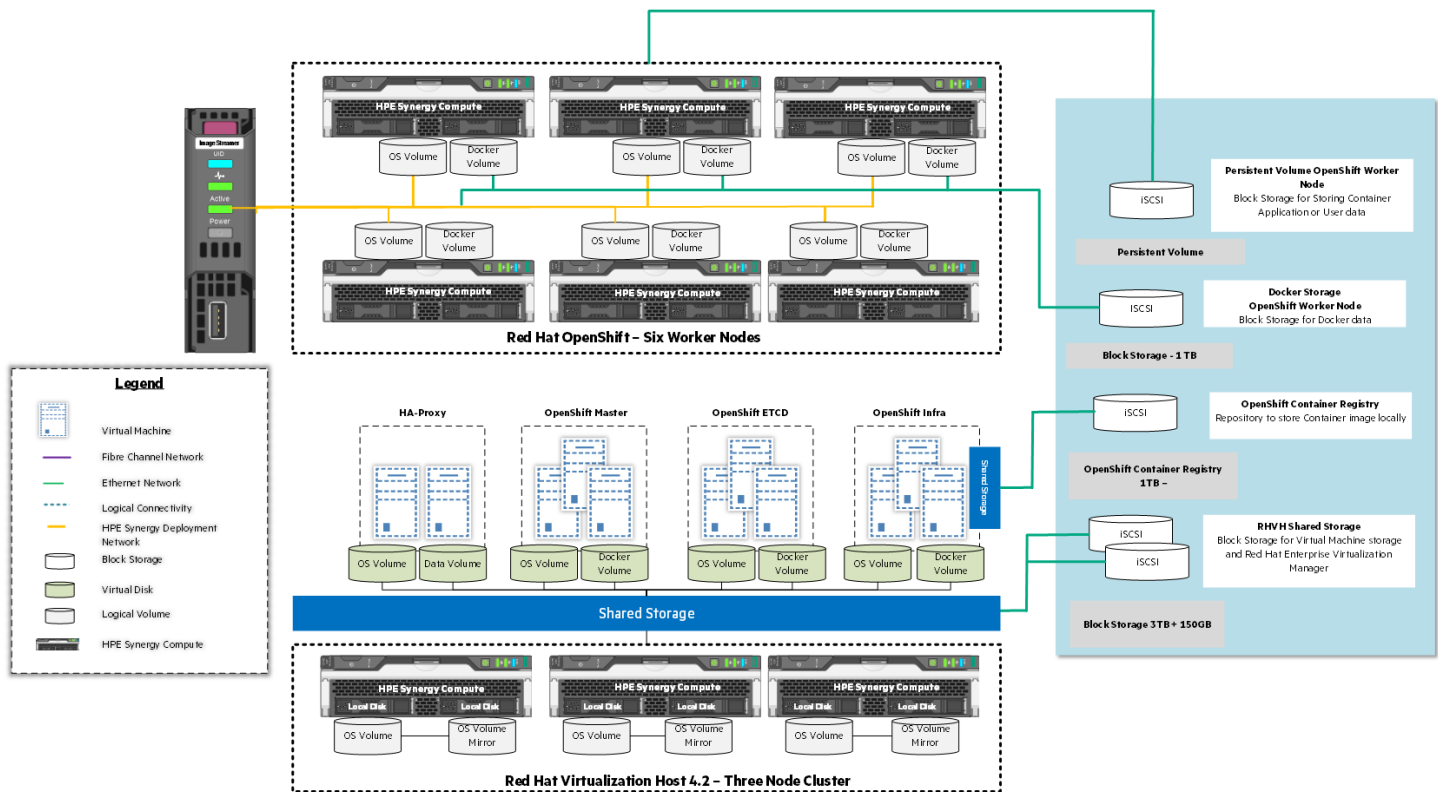


Figure 12. iSCSI physical layout

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 below. 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 below. The installer may choose to manually create and present these volumes or they may 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	1TB	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. HPE has provided resources to automate the initialization and configuration of the array. Please consult [Appendix B](#) for information about utilizing Ansinimble to automate the configuration of array parameters.

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 HPE GitHub site include a sample inventory file. The installer should review this file carefully (located at `/etc/ansible/Openshift-Synergy-RA/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 automated configuration steps failing.

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. From the OneView drop down, 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. Scroll to **Connections** and select **Add Connection**. Add the network connections. Select **Add+** until the final network and then select **Add**.

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.

- 4. 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		Managed by OneView						
Mode								
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.

- 5. Under **Boot settings**, select **Manage boot mode** and choose **Legacy BIOS**. Choose to have Hard disk and then PXE as the first two items in the boot order.
- 6. 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, select Server Profiles.
- 2. Select the **+ Create Profile** button.
- 3. Assign a logical name to the profile and choose the already created server profile template from the drop down.

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. 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.

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 on three (3) compute modules. Prior to installation you should create a hashed password. To generate a hashed password, type the following at a command prompt or in the console of the Linux PXE host.

openssl passwd -1 yourpass

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
nodectl init
%end
```



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 “nrhvh” (Nimble RHVH) and “nrvhhosts” (Nimble RHVH hosts) 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.
- nrhvhhosts.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/nrhvhhosts/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 repos 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/nrhvhhosts/tasks
 - dcbond.yaml - configures the datacenter 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/nrhvhhosts/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.

Once done, 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 setup:

1. Using a browser, open the Cockpit web interface at <https://<ipaddress>:9090>. Use the IP of Bond0. You should log on as the root user. Figure 16 displays the console that is presented upon successful login.

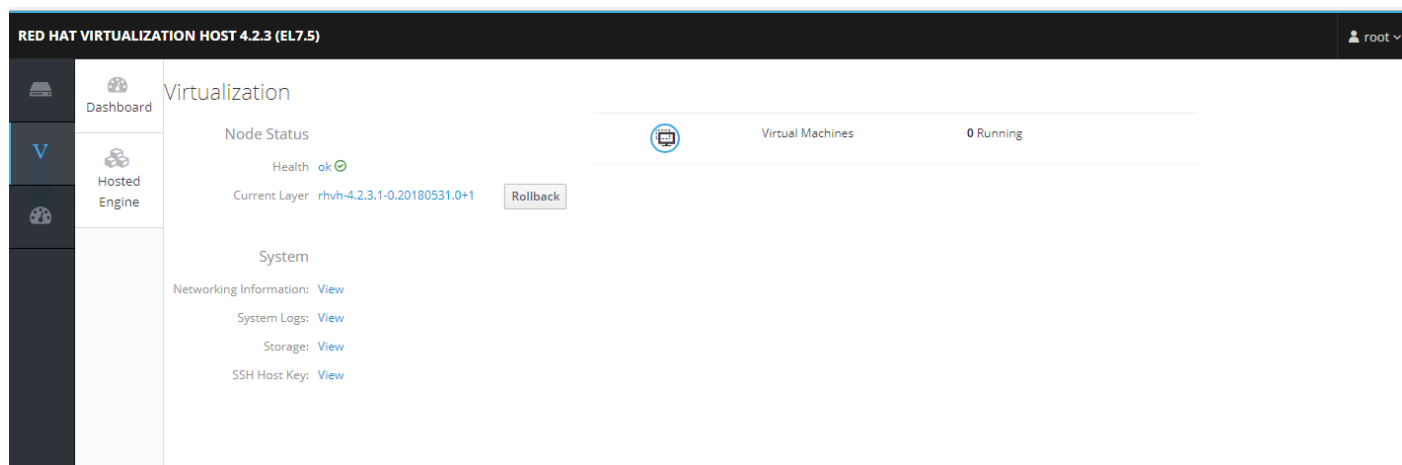


Figure 16. Individual host web interface

2. From the left navigation menu, choose **Localhost > Networking**. **Error! Reference source not found.** shows the list of bond IPs. This should align to the settings you configured in the Ansible playbooks.



Interfaces				Add Bond	Add Team	Add Bridge	Add VLAN
Name	IP Address	Sending		Receiving			
bond0	10.0.9.7/16	16.0 Kbps		7.51 Kbps			
bond1	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 plays (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 Nimble 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 *

rhvh-hosts

SUBNETS

Use all configured subnets

INITIATORS

NAME

buranrhvh01

IQN

iqn.1994-05.com.redhat:buranrhvh01

IP ADDRESS

*

x

NAME

buranrhvh02

IQN

iqn.1994-05.com.redhat:buranrhvh02

IP ADDRESS

*

x

NAME

buranrhvh03

IQN

iqn.1994-05.com.redhat:buranrhvh03

IP ADDRESS

*

x

ADD

CREATE

CANCEL

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. Choose **Manage > Data Storage**.
 - b. Select the **+** icon.
 - c. Create the volume by providing the required details such as name, storage size, and so on.
 - d. Select **Create**.



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

```
# iscsiadm -m discovery -t sendtargets -p 40.0.0.10
# iscsiadm -m discovery -t sendtargets -p 30.0.0.10
# iscsiadm -m node -T iqn.1994-05.com.redhat:<your initiator> -p 30.0.0.10 --login
# iscsiadm -m node -T iqn.1994-05.com.redhat:<your initiator> -p 40.0.0.10 --login
```

5. Restart the host by selecting the **System** link and then choosing **Restart**.

Configuring the virtualization environment

Once the three hosts have been configured as above, the Red Hat 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.

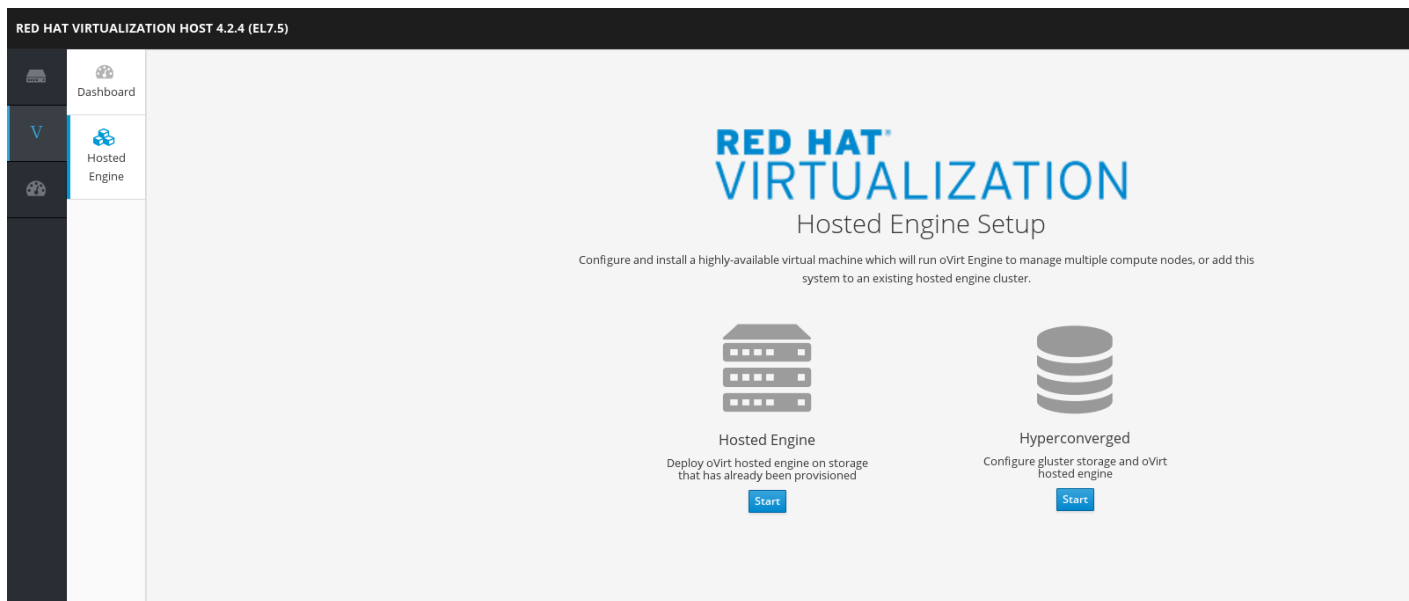


Figure 19. Hosted Engine Setup screen of the Cockpit interface.

1. If not logged on, reconnect to the cockpit user interface using the instructions in the prior section. This step should only be carried out on one host. Log into the cockpit user interface with the host's local account credentials.
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, select **Next**.

Hosted Engine Deployment

VM

Engine

Prepare VM

Storage

Finish

1

2

3

4

5

VM Settings

Engine VM FQDN

ovirtnim.tennet.local

MAC Address

00:16:3E:6A:7A:F9

Network Configuration

Static

VM IP Address

10.0.4.1

/

16

Gateway Address

10.0.1.1

DNS Servers

10.0.1.254

-

+

Bridge Interface

ens3f2

Root Password

.....

Root SSH Access

Yes

Number of Virtual CPUs

4

Memory Size (MiB)

16348

254,716MB available

Cancel

< Back

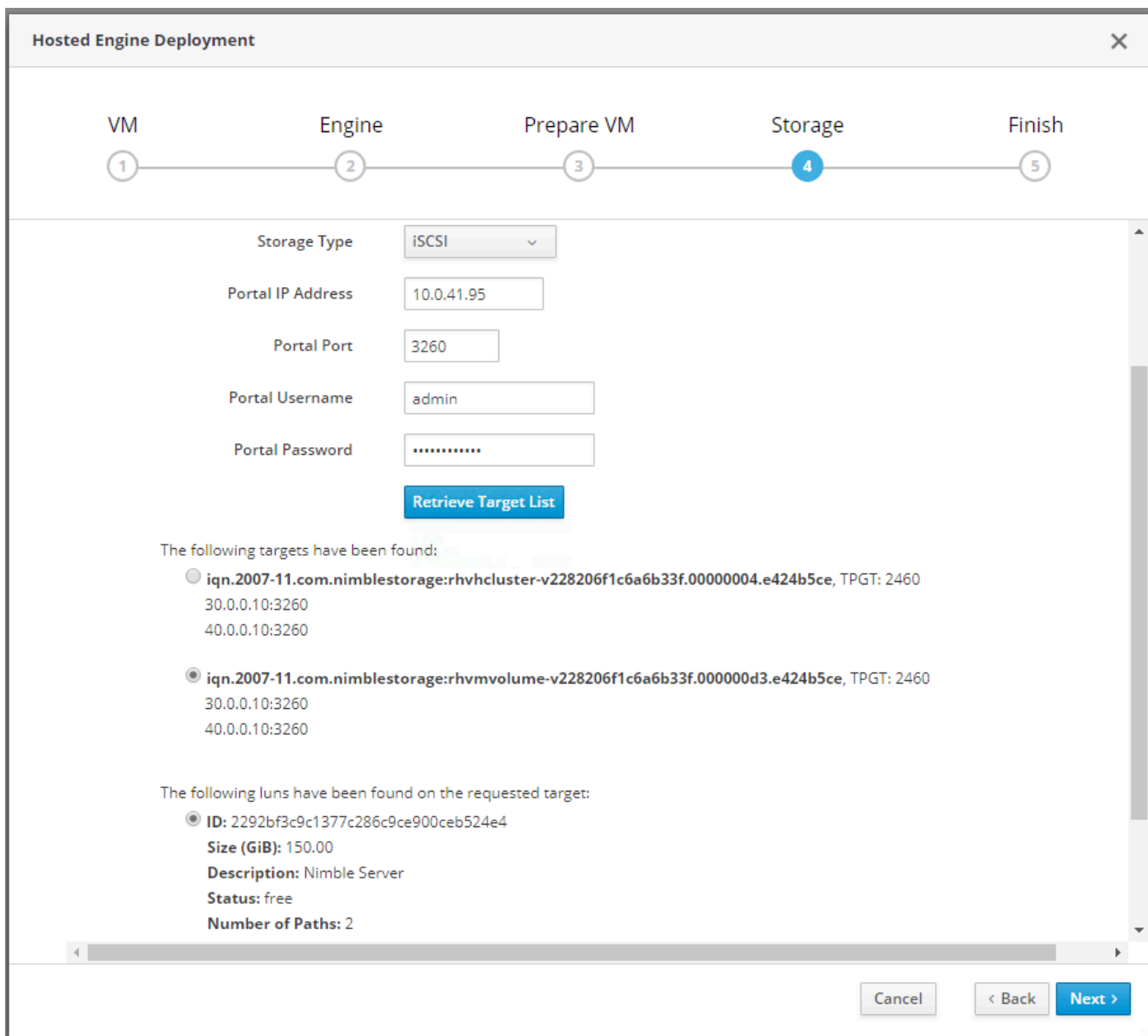
Next >

Figure 20. Hosted Engine Deployment configuration example.

4. On the Engine screen, enter the engine credentials for your environment and select **Next**.
5. On the Prepare VM screen, review the summary and select **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, select **Next**.
7. On the Storage screen, choose **iSCSI** as the storage type, and enter the management IP address of the HPE Nimble SAN along with the portal username and portal password. When completed, select **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 select **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, currently active), and Finish (5). The Storage step contains the following fields and options:

- Storage Type: iSCSI (dropdown menu)
- Portal IP Address: 10.0.41.95
- Portal Port: 3260
- Portal Username: admin
- Portal Password: (masked with dots)
- Retrieve Target List (button)

Below these fields, 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

Below the targets, 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

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

Figure 21. Storage selection example.

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



Configuring Hosted Engine

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.

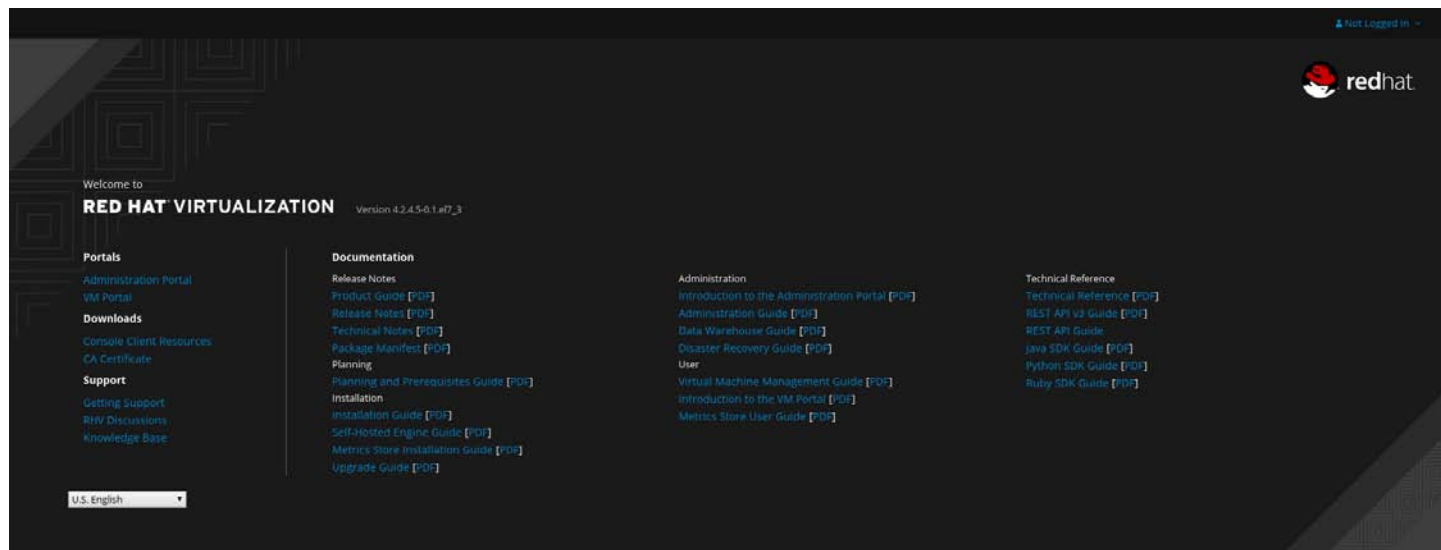


Figure 22. Red Hat Virtualization Welcome screen.

You can view the details of the hosted engine that you configured by selecting the **Administration Portal** link under the **Portals** section.

The following steps are required to configure the Hosted Engine and create the RHV cluster:

1. Create an account on each iLO to facilitate power management.
2. Add additional RHVH hosts to the cluster.
3. Configure host networking.
4. 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 admin 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**, select **New**.

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**. **Error! Reference source not found.** shows the account creation process.

The screenshot displays the HPE iLO 5 Administration interface for User Administration. The left sidebar lists various system components, with 'Administration' highlighted. The main panel shows the 'Add/Edit Local User' form. The 'User Information' section has 'Login Name' and 'User Name' both set to 'rhvadmin'. The 'Change password' checkbox is unchecked, and the 'Service Account' checkbox is checked. Under 'User Privileges', 'Login' and 'Virtual Power and Reset' are selected. The 'IPMI/DCMI Privilege based on above settings:' field is set to 'user'.

Figure 23. HPE iLO Administration – User Administration Create 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 - 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.

The screenshot shows the 'Edit Host' window with the 'Power Management' tab active. A sub-dialog titled 'Edit fence agent' is open, displaying the following configuration:

- Address:** 192.168.1.18
- User Name:** rhvadmin
- Password:** (masked with dots)
- Type:** ilo_ssh (selected from a dropdown)
- Port:** (empty field)
- Options:** (empty field)

Below the fields, there is a 'Test' button and a status message: 'Test successful: power on'. The background window shows a sidebar with tabs: General, Power Management (selected), SPM, Console, Kernel, and Affinity Labels. The 'Power Management' section includes checkboxes for 'Enable Power Management' and 'Kdump integration'.

Figure 24. Red Hat Virtualization Host Power Management Configuration



- 4. Select **Test**.
- 5. Once the test is successful, select **OK**.
- 6. On the **Hosted Engine** tab, from the **Choose hosted engine deployment action** drop down, select **Deploy**. This enables the hosted engine to run on the new host. **Error! Reference source not found.** 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.

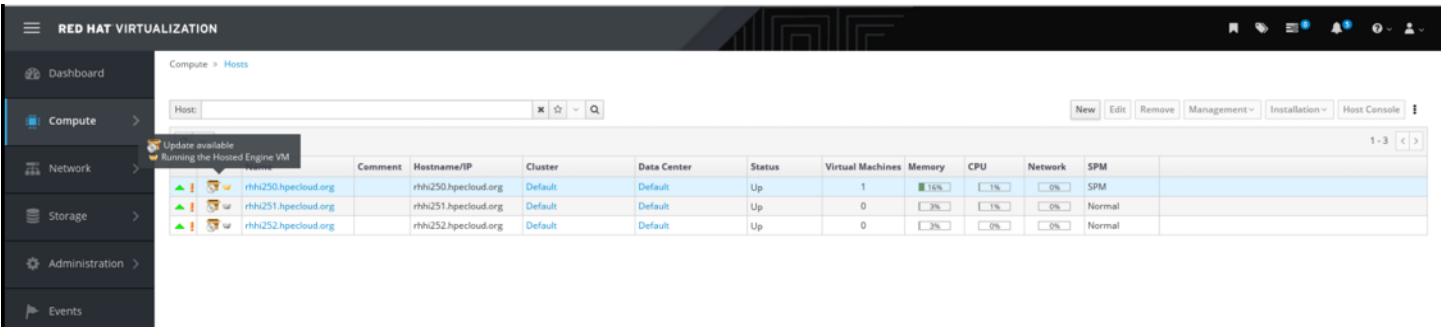


Figure 25. Virtualized hosts as added to the default cluster.

- 7. Once hosts are added, select **Network > Networks > New** as in Figure 26.

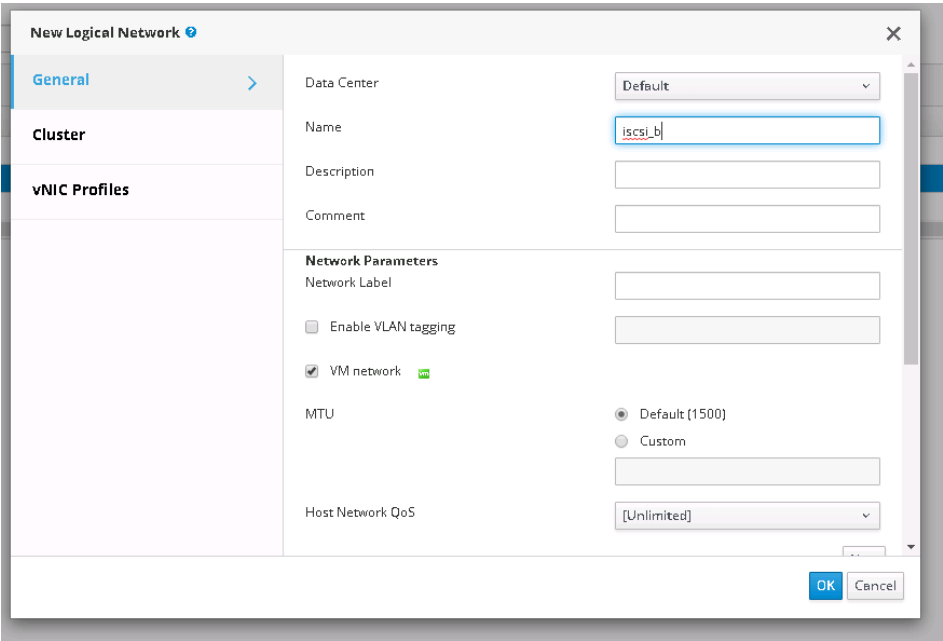


Figure 26. Adding a new logical network

- 8. On the **General** tab, one at a time, add your datacenter and iscsi networks ensuring you provide labels and descriptions.
- 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.

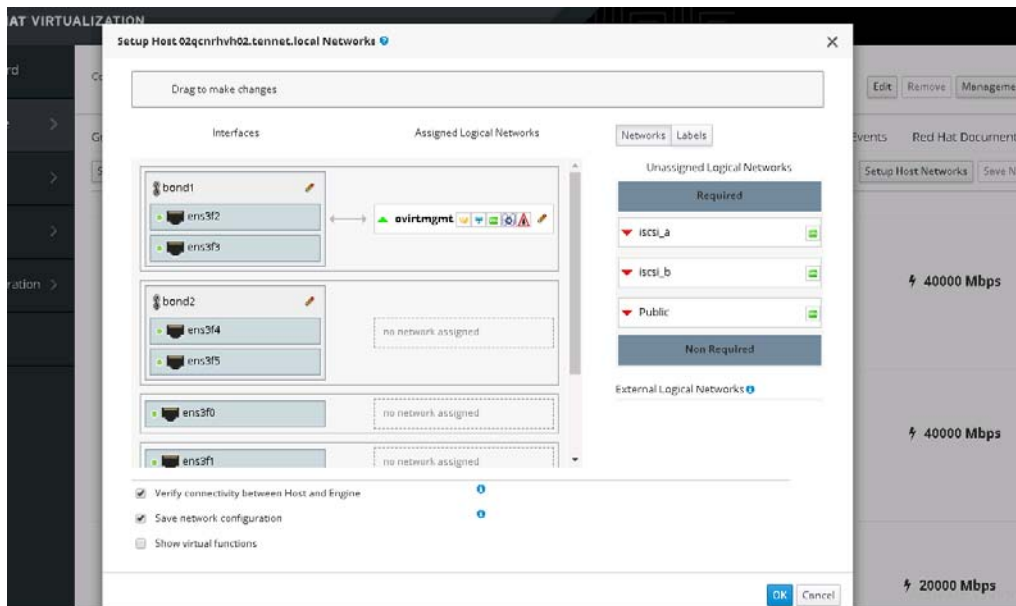


Figure 27. Selecting network interfaces

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

This section describes the deployment and configuration of the Red Hat OpenShift Worker Nodes. These nodes are responsible for running the container workloads within the solution environment. These nodes are deployed using a combination of HPE Synergy Image Streamer and Ansible playbooks. In order to successfully deploy worker nodes, the installer will complete the following steps.

1. Build a worker node image.
2. Capture a Golden Image from the worker node image that HPE Synergy Image Streamer will use to deploy the hosts.
3. Copy and edit the NIC teaming plan script.
4. Copy and edit the OS Build Plan.
5. Create a deployment plan.
6. Create a Server Profile Template.
7. Deploy the hosts.
8. Post-deployment Ansible configuration.

Prior to beginning the steps in this section ensure that you have downloaded and installed the latest artifact bundles 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/tree/V4.1/artifact-bundles/>. Sample foundation artifact bundles should be downloaded from <https://github.com/HewlettPackard/image-streamer-tools/tree/v4.0/foundation/artifact-bundles>.



Build a worker node image

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

9. Create a temporary server profile and attach it to a host. Adjust the following settings.
 - a. Create a 40GB (40960MB) Image Streamer volume that will be used to install RHEL by selecting HPE-Create Empty volume from the OS deployment plan dropdown.
 - b. Configure the connections
 - 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. 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 select **Create**.
10. From the Actions drop down found on the server profile's OneView page, select Launch console.
11. Attach the RHEL 7.6 image and select Power Switch and select Momentary Press.
12. Check your media and then select "e" at the Install Red Hat Enterprise Linux screen
13. Append the following to the Install kernel boot parameter:
rd.iscsi.ibft=1
14. Press **Ctrl-X** to boot the system
15. Fill in the appropriate language and localization settings insuring you set your Timezone to align with your environment practices.
16. Select **Installation Destination** to choose the appropriate location to install the Operating System.



9. On the Installation Destination screen, select **Add a disk** as in Figure 28.

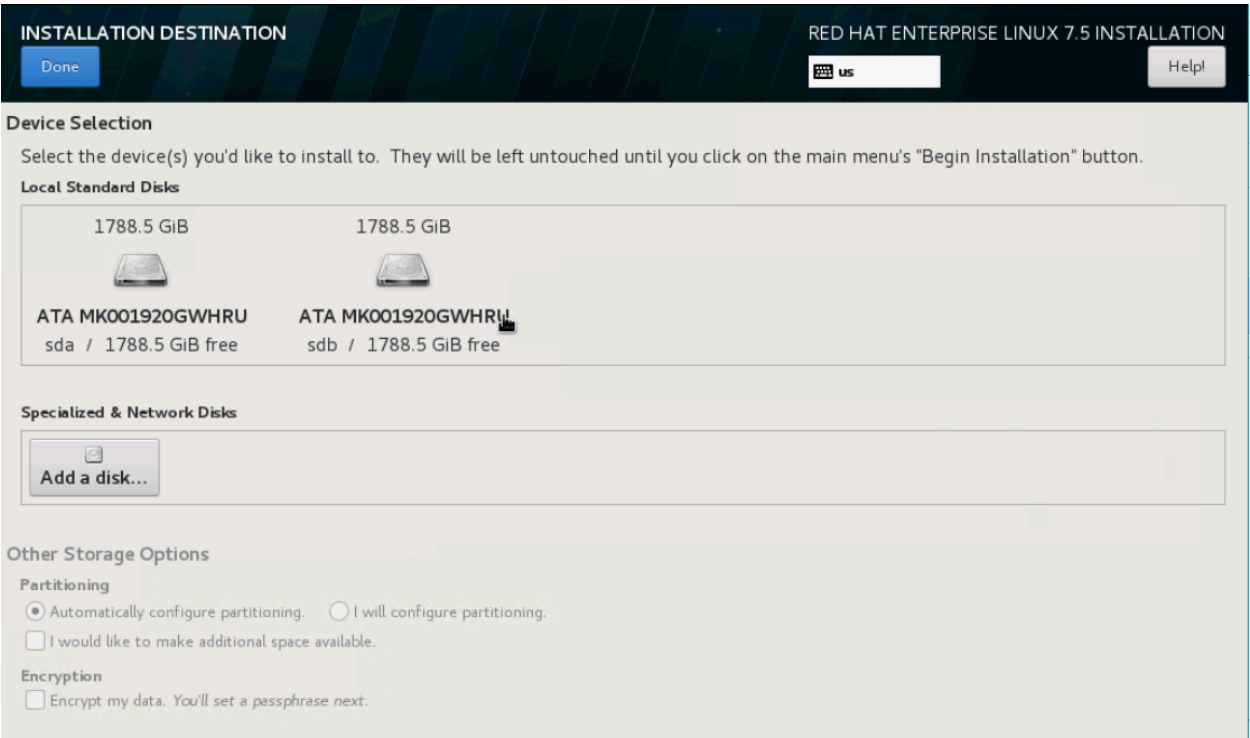


Figure 28. Installation Destination screen.

10. Select the Image Streamer disk as in Figure 29 and then select **Done**.

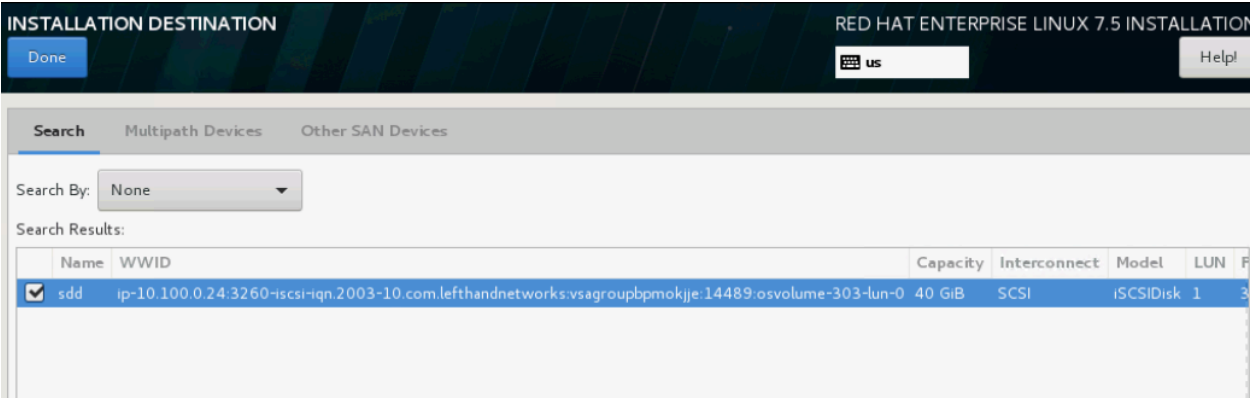


Figure 29. The Image Streamer volume as seen from within the Installation Destination screen.

17. Once back on the Installation Destination screen, select **I will configure Partitioning** and then select **Done**.



18. On the Manual Partitioning screen, select **Click here to create them automatically** as seen below.

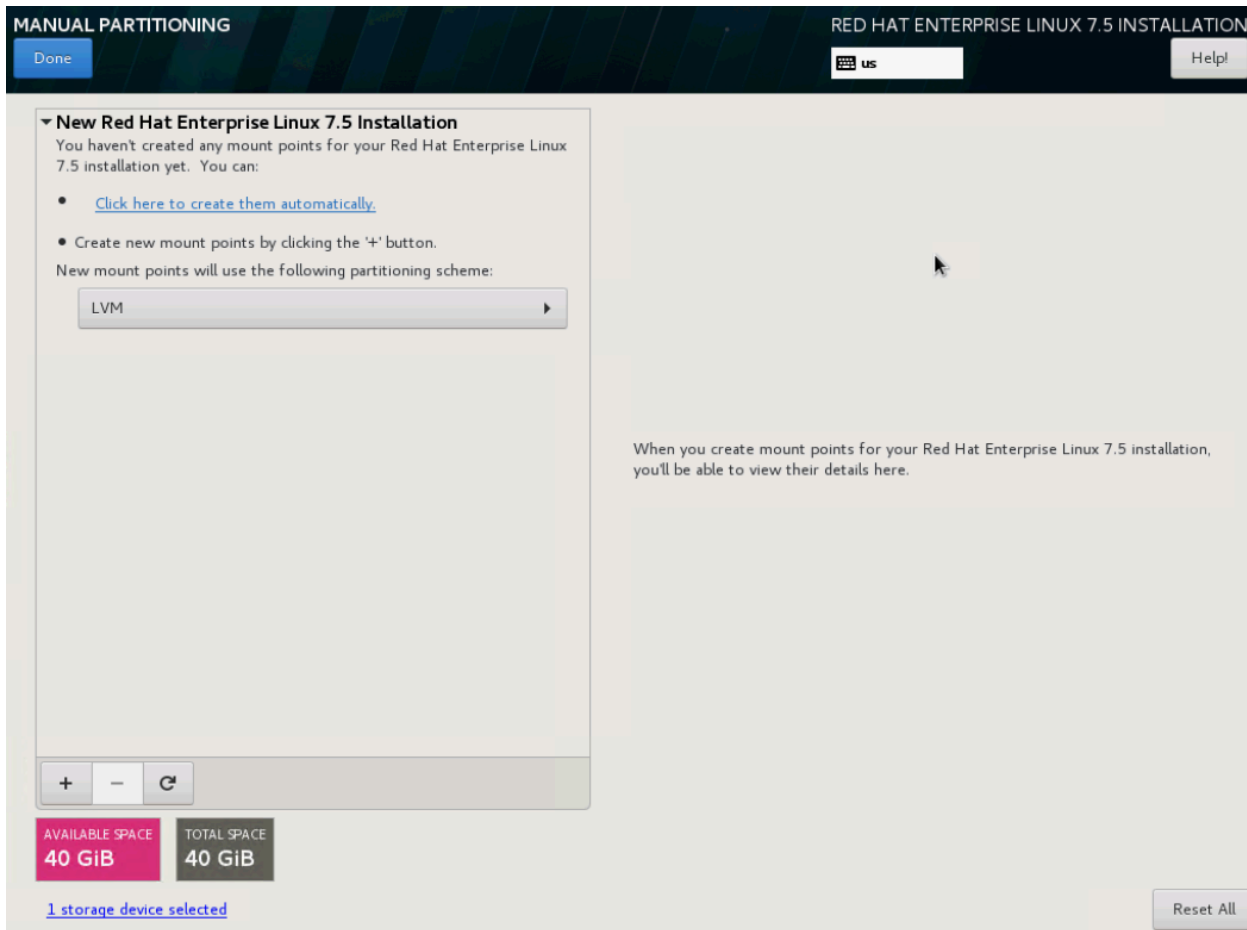


Figure 30. Manual Partitioning screen.



19. Create or alter the partitions as follows
- a. /boot 1024MiB ext4
 - b. / 18.8 GiB ext4
 - c. Select the + sign and add
 - /var - Leave capacity empty
 - Change to ext4 once complete

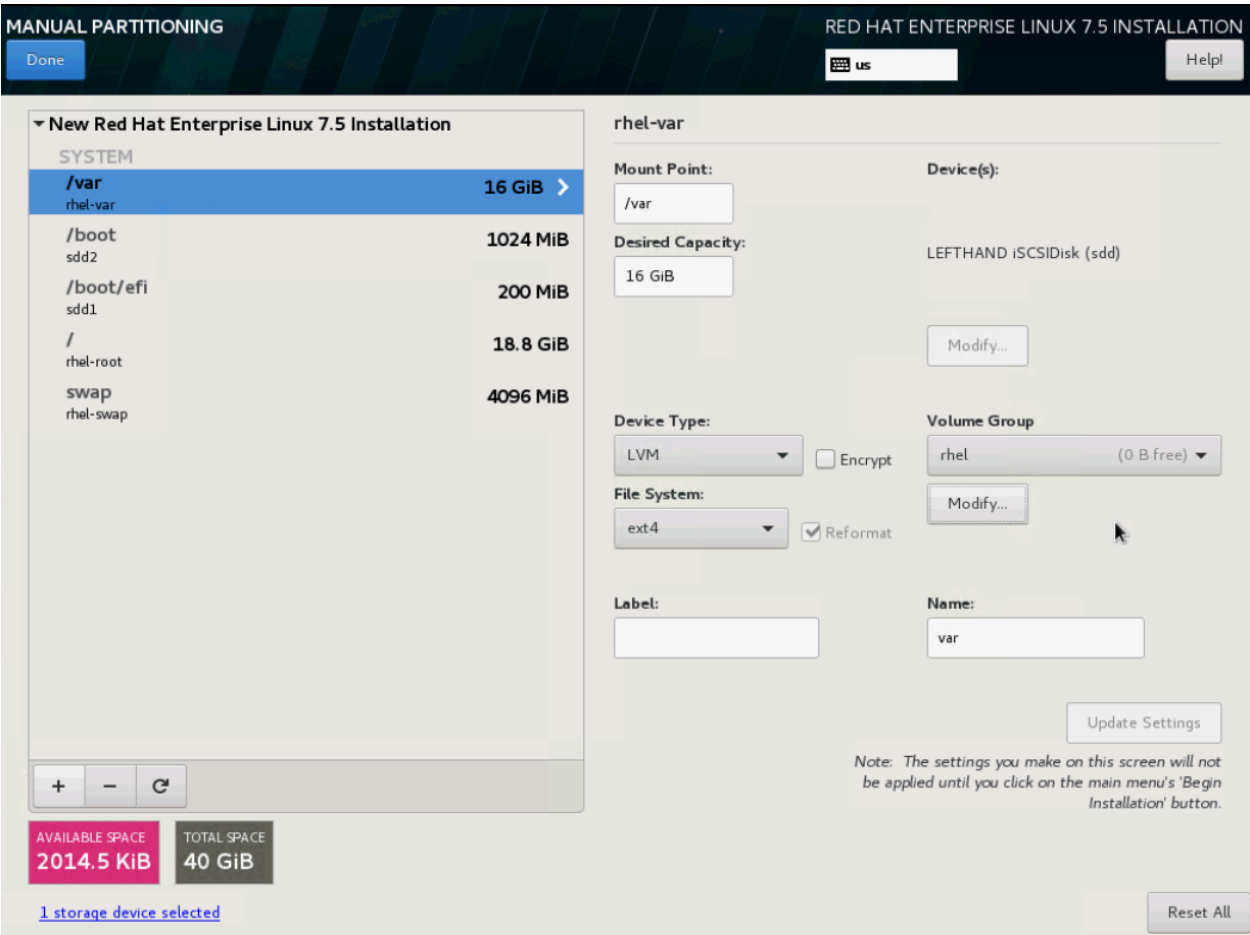


Figure 31. Partitions used during installation.

20. Select **Done** and then select **Accept Changes**.
21. Select **Network & Hostname**.



22. Select **Ethernet (ens3f2)** and switch it to **ON**. Provide a temporary hostname to the host and select **Apply**. When complete, select **Done**.

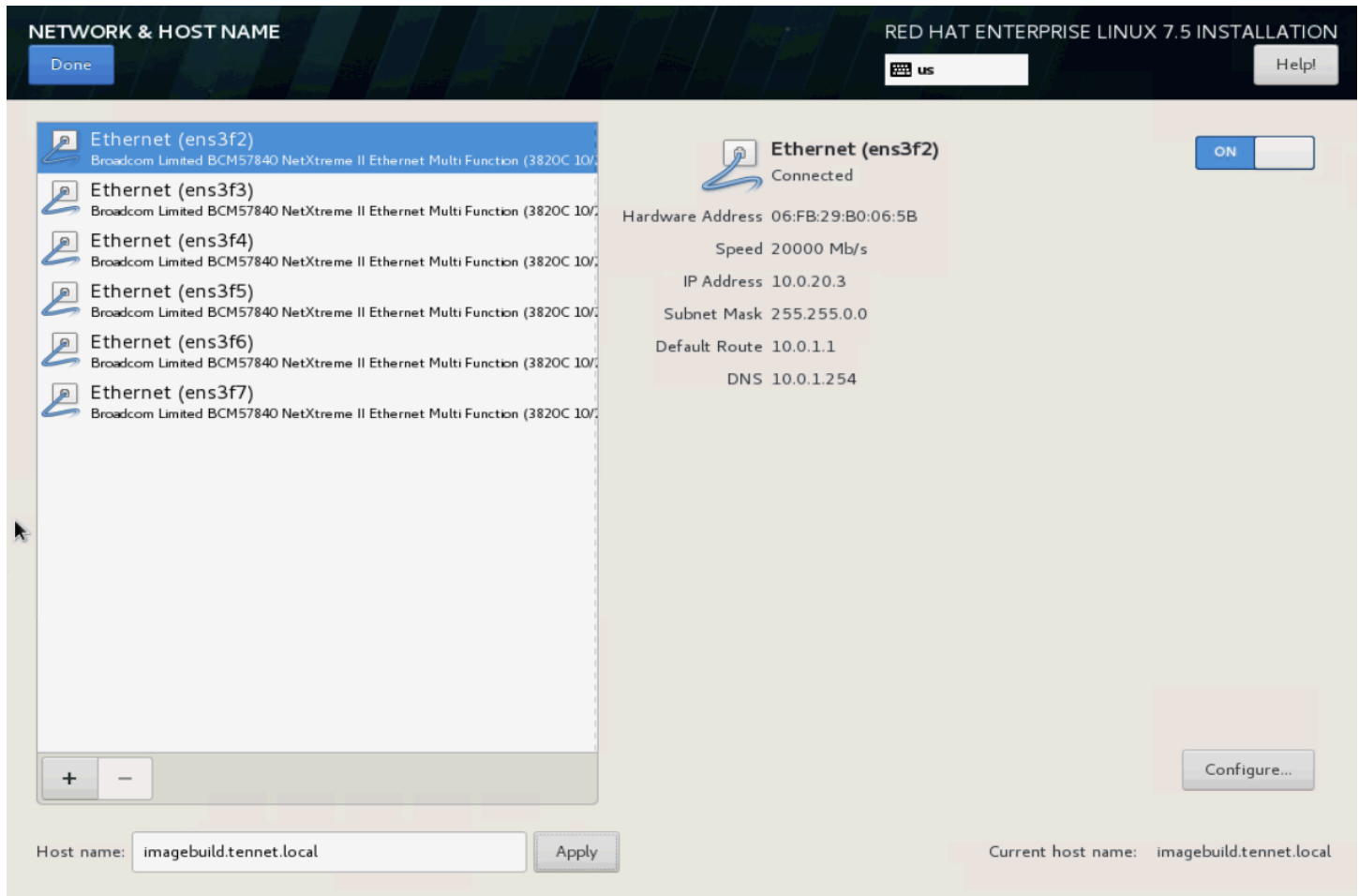


Figure 32. Network & Host Name screen.

23. Select **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 count later).
24. Once the installation is complete, you can reboot the host by selecting **Reboot** when prompted.
25. Once the system is rebooted, log on as root and type in the following:
- ```
subscription-manager register
```
26. Enter your RH username and password when prompted.
27. Attach the subscriptions for RHEL and Red Hat Cloud Infrastructure to enable the host to be updated. For each subscription, run the following command:
- ```
# subscription-manager attach-pool=<your-pool-id>
```
28. Disable all repos and then enable the required repos 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-2.6-rpms
```



29. Install teamd by running the following command:

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

30. Update the host by running the following command:

```
# yum update -y
```

31. Reboot the host by typing the following:

```
# reboot
```

32. Once the host has restarted, log on 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
```

33. Verify your settings by running the following command:

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

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

```
# subscription-manager unregister
```

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

```
# shutdown -h now
```

Capture a Golden Image

- 1. From within HPE OneView, highlight the server profile of the system you just installed and scroll in the right window until you see the OS volume. Record the ID of the volume.
- 2. From the OneView drop down, select **OS Deployment Servers** and launch the Image Streamer UI by selecting the link as shown in Figure 333.

✔ Image Streamer

General ▾

⌵

General

State	Connected
Type	Image Streamer
Image Streamer UI	<u>10.0.0.31</u>
Description	Deployment

Figure 33. 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 follows:
 - a. Name – Provide a name for the golden image
- 5. Description – Provide a short description about the golden image
- 6. OS volume – Select the OS volume.
- 7. Capture OS build plan – Use the HPE-Capture-RHEL75-Image-2018-07-13 build plan.
 - a. Select **Create**.

Create Golden Image?

Name

Nimble Worker Image

Description

Image for Nimble worker node.

OS volume

OSVolume-303

x

Capture OS build plan

HPE-Capture-RHEL75-Image-2018-07-13

x

4

Changed: Capture OS build plan to "HPE-Capture-RHEL75-Image...

Create

Create +

Cancel

Figure 34. Golden Image creation screen.

- 8. Once the image is successfully captured, return to the OneView interface and delete the server profile from the host you used to capture the image. You can 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 NIC teaming plan script

- 1. From within the Image Streamer UI, select Plan Scripts and then highlight the latest version of the RHEL-configure-management-NIC-teaming-<date>.
- 2. Select **Actions** and then **Copy** to make a copy of the plan script. Assign it the name nworker-configure-management-NIC-teaming.
- 3. In the content text box, scroll until you find the line:

```
nmcli connection add type team con-name $team_name ifname $team_name config '{"runner": {"name": "activebackup"}, "link_watch": {"name": "ethtool"}}'
```
- 4. Select **OK**.

Copy and edit the OS Build Plan

- 1. From within the Image Streamer UI, select **OS Build Plans** from the Image Streamer drop down.
- 2. Select the **HPE-Deploy-RHEL75 Build Plan <Date>** 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. Select the **X** next to HPE-RHEL_NETWORK_DEPLOY
- 6. Select **Add steps** and locate nworker-configure-management-NIC-teaming.
- 7. Set the order to be prior to HPE-RHEL-HOSTNAME-DEPLOY as in Figure 35.

Steps

	Name	Type	
⇅ 1	HPE_RHEL_MOUNT_DEPLOY	Deploy	×
⇅ 2	HPE_RHEL_USER_DEPLOY	Deploy	×
⇅ 3	RHEL-configure-management-NIC-teaming-2018-02-09	Deploy	×
⇅ 4	HPE_RHEL_HOSTNAME-DEPLOY	Deploy	×
⇅ 5	HPE_RHEL_SECURITY_DEPLOY	Deploy	×
⇅ 6	HPE_RHEL_UNMOUNT_DEPLOY	Deploy	×

Add steps

Remove all

Figure 35. Order for build plan steps

- 8. Under Custom attributes, select Team1NIC1 and select the edit icon. Check the box for **Allow no network connection** and select **OK**. Repeat this process for Team1NIC2.
- 9. Select **OK** when done



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 36 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

DeploymentNworker02072019

Description

Plan Attributes

OS build plan

NworkerDeploy02072019

Custom attributes

Name	Type	Constraint	Visible on deployment	Value
NewRootPassword	password	maxlen:20	<input checked="" type="checkbox"/>	<div>Confirm password</div>
NewUser	string		<input checked="" type="checkbox"/>	<div></div>
NewUserPassword	password	maxlen:20	<input checked="" type="checkbox"/>	<div>Confirm password</div>
NICTeam0Name	string		<input checked="" type="checkbox"/>	<div>team0</div>
NICTeam1Name	string		<input checked="" type="checkbox"/>	<div>team1</div>
ServerFQDN	fqdn		<input checked="" type="checkbox"/>	<div></div>
SSH	option	options:Enabled Disabled	<input checked="" type="checkbox"/>	<div>Enabled</div>

Changed: OS build plan to "NworkerDeploy02072019"

Create

Create +

Cancel

Figure 36. Create Deployment Plan screen from within Image Streamer.

3. In the Golden Image dropdown, select the Golden image created in the “Capture a Golden image from the compute module” section in the **Golden image** drop down.
4. Select **Create** to finish creating the Deployment Plan.



Create the server profile template

1. From the OneView drop down, 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 dropdown.
4. Scroll to Connections and add the following in order (you will see two deployment network connections in place prior to adding these networks). Select **Add+** until the final network and then select **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 1Gb.
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 TeamONIC1 select Management_a and fill in the network information requested.
 - d. For TeamONIC2 select Management_b.
7. Leave ServerFQDN blank.
8. Fill in any RBSU customizations and then select **Create**.



Deploying the worker nodes

1. Deploy a new server profile from your server profile template to the first worker node. Ensure that the deployment plan that you previously created is selected from the dropdown in the **OS Deployment** portion of the server profile.
2. Fill in the remaining settings as in Figure 377, inserting your own values. For IPv4 address you should enter a temporary IP on your management network. This will be deleted as part of the Ansible play, but initial connectivity from Ansible Engine will be directed to this IP. When complete, select **Create**.

Create Server Profile

General

?

General

Server hardware typeSY 480 Gen10 3

Enclosure groupEnclosure_Group

AffinityDevice bay

OS Deployment

OS deployment planOCP Worker Node Deploy

Deployment Settings

Setting	Value
ManagementNIC1	Management01
IPv4 configuration	IPv4 settings must be entered because there is no subnet associated with the selected network
IPv4 address	10.0.0.131
Netmask	255.255.0.0
Gateway	10.0.1.1
DNS 1	10.0.1.254
DNS 2	10.0.1.253
Domain	tennet.local
MAC address	pending assignment
NewRootPassword	••••••••

Changed: OS deployment plan to "OCP Worker Node Deploy"

Create

Create +

Cancel

Figure 37. Create server profile

3. Repeat these 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 were 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.
- `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 repos 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.
 - `services.yaml` - this file starts and enables the Docker service.
 - `iscsi.yaml` - this file configures and activates the NICs used for iSCSI connectivity within the environment and pulls information from the `hosts` file.
 - `dcbond.yaml` - configures the datacenter bond for the host. This file uses variables defined in both the `hosts` and `vars/main.yaml` files.
 - `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 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/Openshift-Synergy-RA/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 assumes you have already cloned the required repository. 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 RHHL cluster.
- Make sure that both 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. You can create the htpasswd file using the tool available at <http://www.htaccessstools.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. Generate a key:

On the Ansible Engine host run the following command:

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

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

2. 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"
```

3. 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 and load balancer) for OpenShift deployment.

To create a virtual machine template:

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 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 complete, you should see a template appear in the Red Hat Virtualization Administration Portal as in Figure 388.

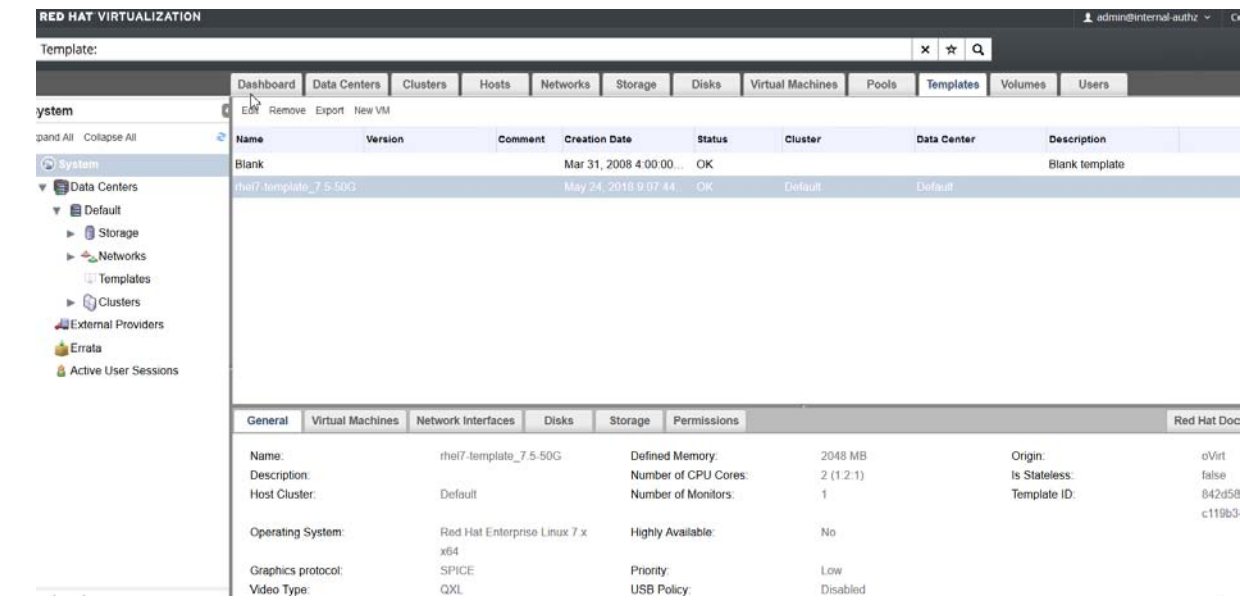


Figure 38. VM Template view from the Red Hat Virtualization Administration Portal

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:

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

The file should look like the example provided in `/etc/ansible/Openshift-Synergy-RA/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.

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 file `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 passwordless sudo rights.
- Creates a Sudo user and adds the user to the passwordless sudo group.
- Uploads the public SSH key to allow secure access without credentials.
- Registers the host using Subscription Manager.
- Enables the required repos.
- Installs the basic utilities.
- Performs a yum update to 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' repo, 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 /usr/share/ansible/openshift-ansible/playbooks/prerequisites.yml playbook:

```
# ansible-playbook -i /etc/ansible/Openshift-Synergy-RA/synergy/scalable/nimble/hosts/usr/share/ansible/openshift-ansible/playbooks/prerequisites.yml
```

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

```
# ansible-playbook -i /etc/ansible/Openshift-Synergy-RA/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 399, using the credentials provided in the htpasswd file. The URL for the webpage is <https://openshift-master:8443>.

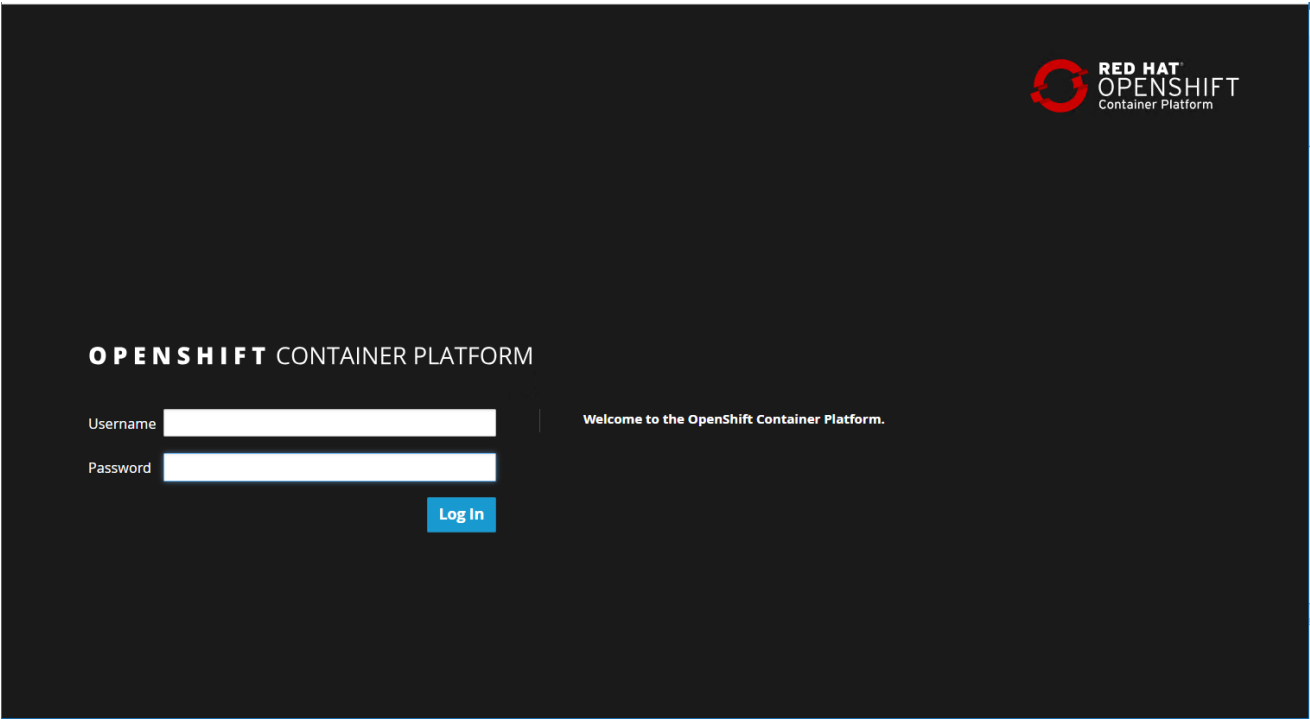


Figure 39. OpenShift User Interface

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

Administrator Login and Cluster Admin Role

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

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

2. Once logged in, the system 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
Using project "default".
```

Grant Cluster Role to User

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

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

Assigning the cluster-admin role is not required to deploy applications.



Figure 4040 shows the final VM layout once OpenShift has been deployed. Validate that your environment aligns to this design.

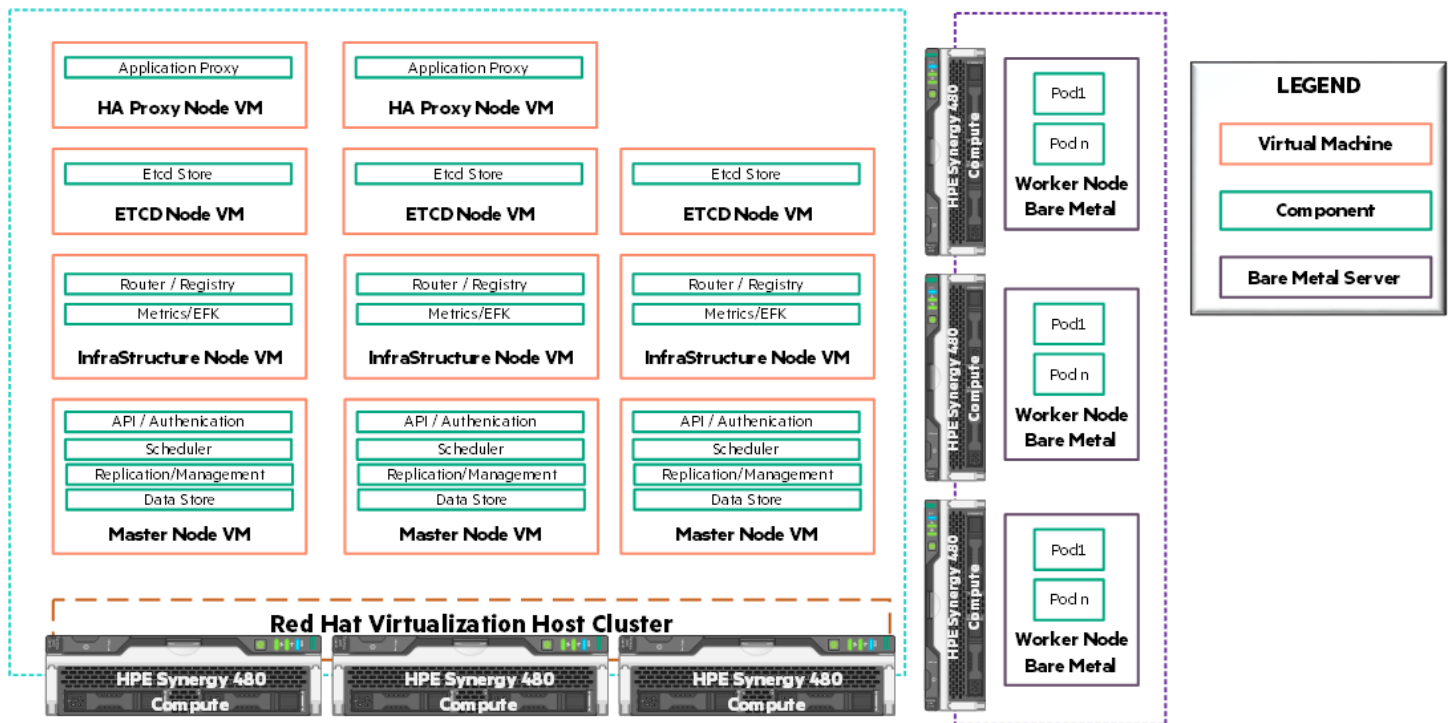


Figure 40. Final OpenShift layout, post deployment

Install the Nimble Linux Toolkit on the worker nodes

If you have used Ansible playbooks to automate the installation of the array(s) in the environment, skip this section.

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.##
# ./nlt_installer_2.4.## --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 kubelet after installing NLT using the following command:

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

Note

The NLT and multipath should be re-installed after any upgrades of OpenShift to ensure the correct multipath.conf file is being used.

Install the HPE Nimble Kube Storage Controller

1. On a master node, verify that you are logged on as a user with the system:admin role by running the following command:



```
# oc whoami -c
```

```
default/masternodename.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-doryd	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 following 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 and an example deployment. Please 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 Persistent Volume Claim from the StorageClass by running the following command:

```
# oc create -f pvc-mariadb.yaml
```

```
persistentvolumeclaim "mariadb-claim" created
```

3. Create a Deployment with a Persistent Volume Claim 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" Persistent Volume Claims as below:

```
# oc create -f sc-default.yaml
```

```
storageclass "general" created
```



5. Create a persistent volume claim 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

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

Unregister OpenShift components and delete deployed VMs

```
# 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	Datacenter 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	Datacenter 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#usage>.

Prerequisites

- Make sure your Ansible server has Python 3 installed and the Ansible engine running is using Python 3.

To check the python version used by Ansible engine, run the below 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. Consult [https:// http://jmespath.org/](https://http://jmespath.org/) for 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 their environment.

Array Setup

1. Download the Nimble Windows Toolkit from HPE Infocight and install it on the installer laptop. With the laptop connected to the same switch as the Nimble array, use the tool to discover the array. Record the serial number for the array as it will be required.

The Ansible playbook for Nimble Array setup uses variables defined in `default/main.yml` under the section `'nimble_array_config'`. An example of this configuration appears below. Copy this file outside of the role structure and create a new variables file for your project with what you need.

```
nimble_array_config:
  AF-200145:
    group_name: group-af-145
    name: <array name>
    domainname: tennet.local
    ntpserver: time.nimblestorage.com
    timezone: America/Chicago
    mgmt_ipaddr: 10.0.2.95

    # This is the serial number of the new array
    # Name of the group.
    # Name of the array.
    # DNS domain name of this system.
    # Hostname or IP address of NTP server.
    # Time zone that the array is in.
    # Management IP address.
```



```

default_gateway: 100.1.1          # Default gateway IP address.

dnsservers:                       # IP addresses of the DNS servers.
- 10.0.1.254
- 20.1.1.254

iscsi_automatic_connection_method: "yes" # Redirect connections from the specified
                                         # iSCSI target IP appropriately.

iscsi_connection_rebalancing: "yes"    # Rebalance iSCSI connections by periodically
                                         # breaking existing connections that are
                                         # out-of-balance.

subnets:                          # Subnets to configure (In NIC order)
- label: mgmt-data                 # Subnet Label on NIC.
  subnet_addr: 10.0.2.95/16        # Subnet on NIC.
  subnet_type: management         # Type of subnet -- data, management, or data
                                  # and management
  subnet_mtu: 1500                # Size of MTU - standard, jumbo, other.

- label: mgmt-data                 # Subnet Label on NIC.
  subnet_addr: 10.0.2.95/16        # Subnet on NIC.
  subnet_type: management         # Type of subnet -- data, management, or data
                                  # and management
  subnet_mtu: 1500                # Size of MTU - standard, jumbo, other.

- label: data1                     # Subnet Label on NIC.
  data_ipaddr: 30.0.2.95           # Data IP address setting.
  subnet_addr: 30.0.2.95/16        # Subnet on NIC.
  subnet_type: data               # Type of subnet -- data, management, or data
                                  # and management
  subnet_mtu: 1500                # Size of MTU - standard, jumbo, other.

- label: data2                     # Subnet Label on NIC.
  data_ipaddr: 40.0.2.95           # Data IP address setting.
  subnet_addr: 40.0.2.95/16        # Subnet on NIC.
  subnet_type: data               # Type of subnet -- data, management, or data
                                  # and management
  subnet_mtu: 1500                # Size of MTU - standard, jumbo, other.

discovery_ipaddrs:                # Discovery IP address setting.
- 30.0.2.96
- 40.0.2.96

support_ipaddrs:                  # Support IP address setting node A.
- 10.0.2.96                       # Support IP address setting node B.
- 10.0.2.97

```

2. 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
```

```
# Please store the password with Ansible Vault.
```

```
nimble_group_password: admin
```

3. Run the following command to install the 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 edit the following files.

`/etc/ansible/hpe-solutions-openshift/synergy/scalable/nimble/roles/deploy-vm/vars/main.yml`
Add lines to the section that outlines the FQDN of the VMs as follows.

```
fqdnnode1: <fqdn of hostname for node 1>
```

```
fqdnnode2: <fqdn of hostname for node 2>
```

```
fqdnnode3: <fqdn of hostname for node 3>
```

For each worker node VM, add a section as follows above the section that starts with **vms:**

```
<vm 1 name>_ip: <ip address>
```

```
<vm 1 name>_ip2: <ip address>
```

```
<vm 1 name>_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 to the deploy-vm play needed.

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

Need to run the play `'playbooks/virtual-hostprepare.yml'`, but no changes to the play required.

Ansible hosts file

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

Ensure that the worker nodes are listed under the `[virtual-nodes]` section and comment out any hosts that are present in the `[physical-nodes]` section.

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 the `/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 done:
# 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 have been stood up you can run the final plays prior to installing OpenShift:



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

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



Change Tracker

Version	Release Date	Changes
3.0.0	07/03/2019	Initial release



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>

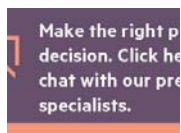
HPE Synergy, <https://www.hpe.com/info/synergy>

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>



Sign up for updates

© Copyright 2018-2019 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.

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

Version 3.0, July 2019

