HPE Deployment Guide for Red Hat OpenShift Container Platform on HPE Composable Rack Platform

Accelerate container deployment with Red Hat OpenShift Container Platform and HPE Composable Rack

Contents

Overview

Organizations are going through a digital transformation that substantially changes the way applications are developed, deployed, and maintained. At the heart of this transformation are microservices and containers that allow developers to quickly create and deploy applications in a Continuous Integration / Continuous Deployment (CI / CD) model. This CI / CD model brings together the developed communities and the operations communities within an organization, commonly referred to as "DevOps". Red Hat® OpenShift Container Platform is a proven technology for enabling container-based DevOps and CI / CD processes and providing container orchestration and scheduling. Containers provide an additional layer of abstraction between the virtual and physical resources that exist in the infrastructure. This fluid deployment model requires an infrastructure that is flexible and can seamlessly add, remove, and repair the underlying physical and virtual resources. HPE ProLiant DL Gen10 servers and HPE Composable Rack provides a secure, highly available, and flexible infrastructure that is ideal to meet the requirements of Red Hat OpenShift Container Platform. HPE OneView provides a software-defined composable network, storage, and compute resources to customers implementing container-based solutions.

Target audience: This document is intended for systems administrators, architects, and DevOps leads. The reader should have a working knowledge of Red Hat Enterprise Linux®, Red Hat OpenShift Container Platform, HPE OneView, and Red Hat Ansible.

Solution design

This document describes a solution using HPE Composable Rack that can easily scale to accommodate additional containerized workloads. This document outlines the steps required to deploy a Red Hat OpenShift Container Platform environment running on HPE Composable Rack, HPE ProLiant DL Gen10 servers, and Red Hat OpenShift Container Storage. It is meant to be used in conjunction with files and Ansible playbooks found at https://github.com/HewlettPackard/hpe-solutions-openshift. Note that the scripts described in this document and provided on the GitHub site are not supported by Hewlett Packard Enterprise or Red Hat. The procedures described in this document provide a reliable and repeatable installation process for Red Hat OpenShift Container Platform 3.11 on HPE ProLiant DL servers. It is recommended that the installer review this document in its entirety and understand all prerequisites prior to beginning an installation. Figure 1 provides a high-level overview of the workflow described in this solution.

Figure 1: Solution workflow overview

Figure 2 displays the configuration for this Red Hat OpenShift Container Platform and HPE Composable Rack infrastructure. The OpenShift master, infrastructure, and OpenShift Container Storage nodes are all deployed on individual physical servers. Red Hat OpenShift Container Storage is used to provide software defined storage and leverages the direct-attached storage internal to the HPE ProLiant DL servers.

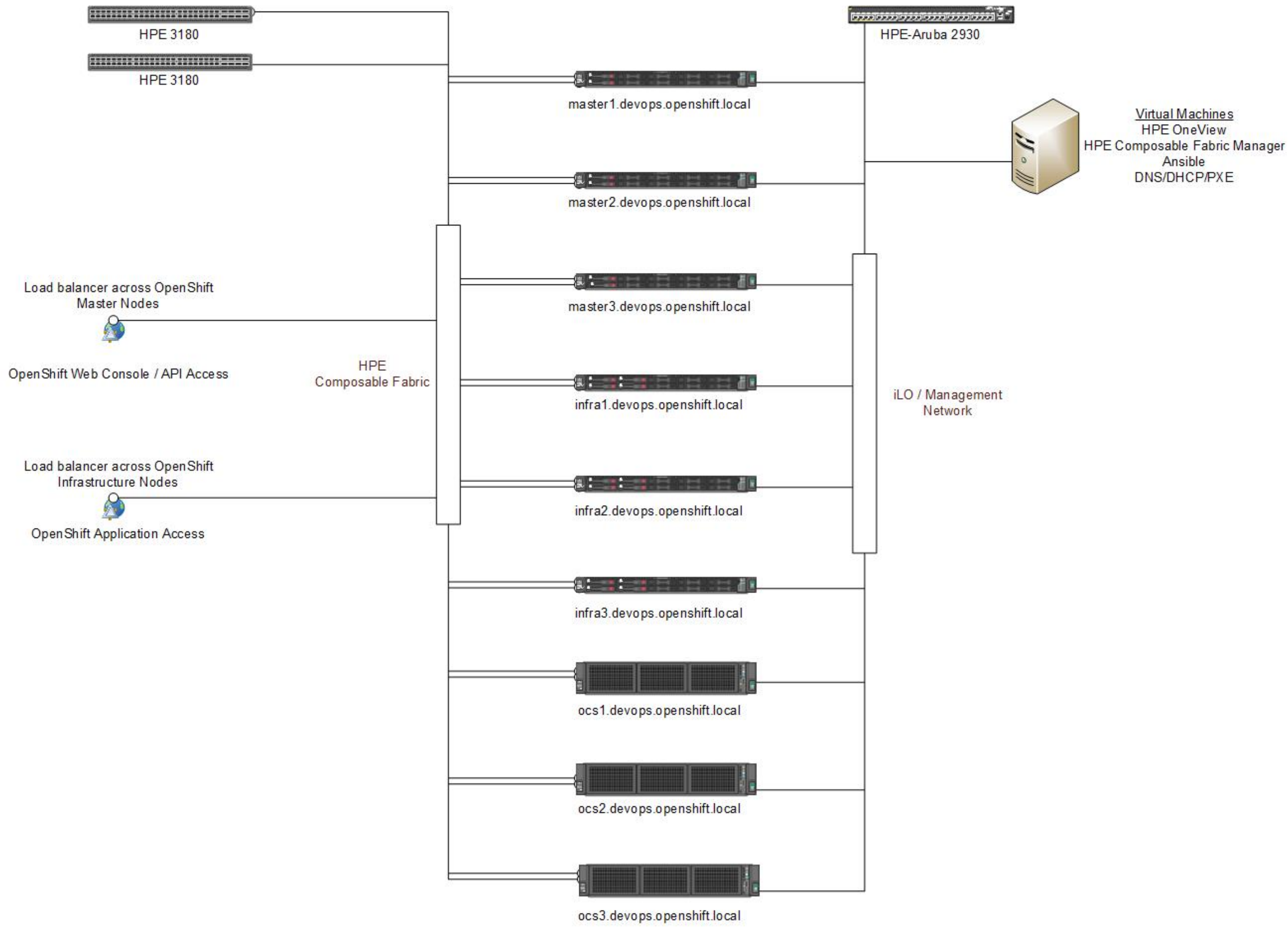


Figure 2. Solution layout

Note

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

Solution components

This solution is based on HPE Composable Rack with nine (9) HPE ProLiant DL Gen10 servers. The storage in this solution is provided by direct-attached storage located in each of the HPE ProLiant DL servers. Red Hat OpenShift Container Storage provides persistent storage for container applications running on Red Hat OpenShift Container Platform. Figure 3 illustrates the logical design of the solution.

Figure 3: Logical design

Software versions

Table 1 lists 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.

Table 1. Software versions used in solution creation

Component	Version
Red Hat Enterprise Linux	7.6
Red Hat OpenShift Container Platform	3.11
Red Hat OpenShift Container Storage	3.11
HPE OneView	5.00.00-0392358
HPE Composable Fabric Manager	5.0.2-4902
Red Hat Ansible	2.6
iLO 5	1.40
ROM	U30 v2.00

Required repositories – OpenShift nodes

- rhel-7-server-ansible-2.6-rpms
- rhel-7-fast-datapath-rpms
- rhel-7-server-extras-rpms
- rhel-7-server-rpms
- rhel-7-server-ose-3.11-rpms

Hardware components

The configuration deployed for this solution is described in greater detail in this section. Figure 4 illustrates physical rack layout and the various components in the solution. At a high level, Hewlett Packard Enterprise and Red Hat deployed:

- Nine (9) HPE ProLiant DL360 Gen10 servers
- Three (3) HPE ProLiant DL380 Gen10 servers

The user also has the flexibility of customizing the HPE components throughout this stack per their unique IT and workload requirements or building with individual components.

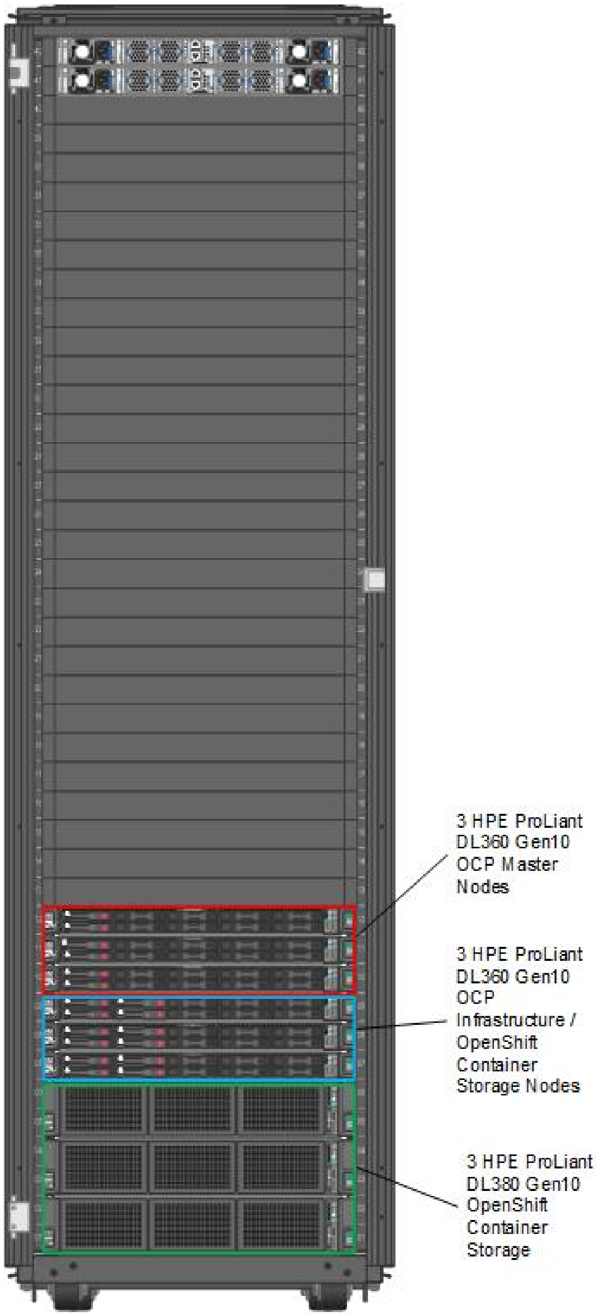


Figure 4: HPE Composable Rack components

In this solution, the OpenShift master nodes are running on the HPE ProLiant DL360 Gen10 servers. The Red Hat OpenShift infrastructure nodes — running the OpenShift registry, routing, logging and metrics services — are running on the HPE ProLiant DL360 Gen10 servers. The HPE ProLiant DL380 Gen10 servers are used for customer production applications. Additional HPE ProLiant DL360 Gen10 or HPE ProLiant DL380 Gen10 servers can be added as worker nodes to scale out the solution. The HPE Composable Rack for HPE ProLiant DL server will scale across three (3) racks and up to 120 physical servers.

Table 2. Components utilized in the creation of this solution

Component	Qty	Description
HPE ProLiant DL360 Gen10	6	OpenShift master and infrastructure nodes
HPE ProLiant DL380 Gen10	3	OpenShift Container Storage nodes
HPE Composable Fabric FM 3180	2	HPE Composable Fabric switches
HPE Aruba 2930	1	Management switch

HPE ProLiant DL380 Gen10 servers

This section describes the connectivity of the HPE ProLiant DL380 Gen10 servers used in the creation of this solution. Table 3 describes the individual components. Individual server sizing should be based on customer needs and may not align with the configuration outlined in this document.

Note

The server hardware specifications described in Table 2 and Table 3 reflect the hardware that was available in the lab environment to perform the testing of this solution. This may not reflect the hardware requirements for specific customer implementations depending on performance and capacity requirements.

Table 3. HPE ProLiant DL380 server configuration

Component	Description
Processors	Intel® Xeon® Silver 4114 CPU (2.2 GHz / 10-core
Memory	128GB 4 x 32GB DDR4
Network	HPE Eth 10/25Gb 2 P 631FLR-SFP28 Embedded ALOM HPE Ethernet 1Gb 4 -port 331i Adapter Embedded LOM
Array Controller	HPE Smart Array P408i-a SR Gen10 Embedded RAID HPE Smart Array P408i-p SR Gen10 PCI-E
Disks	2 x 400GB SAS SSD Boot Vol 2 x 2TB SATA SSD OCS Vol

HPE ProLiant DL360 Gen10 servers

This section describes the connectivity of the HPE ProLiant DL360 Gen10 servers used in the creation of this solution. Table 4 describes the individual components. Individual server sizing should be based on customer needs and may not align with the configuration outlined in this document.

Table 4. HPE ProLiant DL360 server configuration

Component	Description
Processors	Intel® Xeon® Gold 5118 CPU (2.3 GHz / 12-core
Memory	384GB 12 x 32GB DDR4
Network	HPE Eth 10/25Gb 2 P 631FLR-SFP28 Embedded ALOM HPE Ethernet 1Gb 4 -port 331i Adapter Embedded LOM
Array Controller	HPE Smart Array P408i-a SR Gen10 Embedded RAID HPE Smart Array E208i-p SR Gen10 PCI-E
Disks	2 x 400GB SAS SSD Boot Vol 2 x 400GB SAS SSD OCS Vol (infra nodes only)

Solution configuration

Deployment environment

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

Services

Table 5 lists the services utilized in the creation of this solution and provides a high-level explanation of their function.

Table 5. Services used in the creation of this solution

Service	Required/Optional	Description/Notes
DNS	Required	Provides name resolution on management and data center networks
DHCP	Required	Provides IP address leases on PXE
TFTP/PXE	Required	Provides network boot capabilities to hosts that will install via a kickstart file
NTP	Required	Ensures consistent time across the solution stack

DNS

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

DHCP

DHCP services must be in place for the PXE and management networks. DHCP services are generally in place on data center networks. The MAC address of the network interfaces on the servers can be easily collected using the HPE OneView and server profiles, thus making it is easy to create address reservations on the DHCP server for the hosts. A reservation is required for a single adapter on the PXE network of each physical server. This facilitates post-deployment configuration over SSH as well as a secure communication channel for running Ansible scripts. This solution uses the two 25Gb interfaces configured as a bonded interface. One of these interfaces (eno5) is used to PXE boot the server. After the server operating system is installed, an Ansible script is run to bond the two 25Gb interfaces (eno5 and eno6d1).

UEFI / ILo vMedia

iLO vMedia and UEFI can be configured to perform an unattended installation of Red Hat Enterprise Linux 7 by following the procedures on the following links:

- Infrastructure Automation with Operating System Deployment, https://github.com/HewlettPackard/oneview-osdeployment
- How to create a customized Golden Image or Red Hat Enterprise Linux ISO with kickstart file, https://access.redhat.com/solutions/60959

TFTP/PXE

The 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 HTTP, TFTP, and network boot services. In this solution environment, the PXE server was installed on the management network. It is beyond the scope of this document to provide instructions for building a PXE server host, however sample configuration files are included in the appendix of this document. It is assumed that TFTP and network boot services are being provided from a Linux-based host.

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

Kickstart options are covered under the Red Hat Enterprise Linux section of this document.

NTP

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

Installer laptop

A laptop system with the ability to connect to the various components within the solution stack is required. This can also be the console or SSH session connected to the Ansible host.

Ansible Engine

This document assumes that Ansible Engine 2.6 exists within the deployment environment and is accessible to the installer. Hewlett Packard Enterprise built this solution using Ansible version 2.6.

Ansible is used extensively throughout this solution to automate manual configuration tasks and to perform the Red Hat OpenShift Container Platform installation. The Ansible playbooks are used to automate the configuration as described in Table 6.

Table 6. Ansible playbooks used in this solution

Playbook	Description/Notes
subs.yml	Registers the HPE ProLiant DL server with Red Hat
prep.yml	Installs the required software on the HPE ProLiant DL servers
bonds.yml	Creates a bonded interface on the HPE ProLiant DL server
passwords.yml	Used to store sensitive customer information
prerequsites.yml	Checks the OpenShift prerequisites
deploycluster.yml	Installs OpenShift Container Platform

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

Physical environment configuration

Cabling the HPE Composable Rack

This section shows the physical cabling between the HPE ProLiant DL servers and the HPE Composable Fabric FM 3180 Switches.

Physical cabling

Table 7 below is a map of source ports to ports on the HPE switches.

Table 7. HPE 3180 port map

Host	Switch s1-fm3180-a	Switch s1-fm3180-a	Switch Port
Master-1	ALOM Port1	ALOM Port2	1
Master-2	ALOM Port1	ALOM Port2	2
Master-3	ALOM Port1	ALOM Port2	3
Infra-1	ALOM Port1	ALOM Port2	4
Infra-2	ALOM Port1	ALOM Port2	5
Infra-3	ALOM Port1	ALOM Port2	6
OCS-1	ALOM Port1	ALOM Port2	9
OCS-2	ALOM Port1	ALOM Port2	10
OCS-3	ALOM Port1	ALOM Port2	11

Figure 5 shows the cabling of the HPE ProLiant DL servers to the network switches. The specific networks contained within the Bridge Aggregation Groups are described in more detail later in this section.

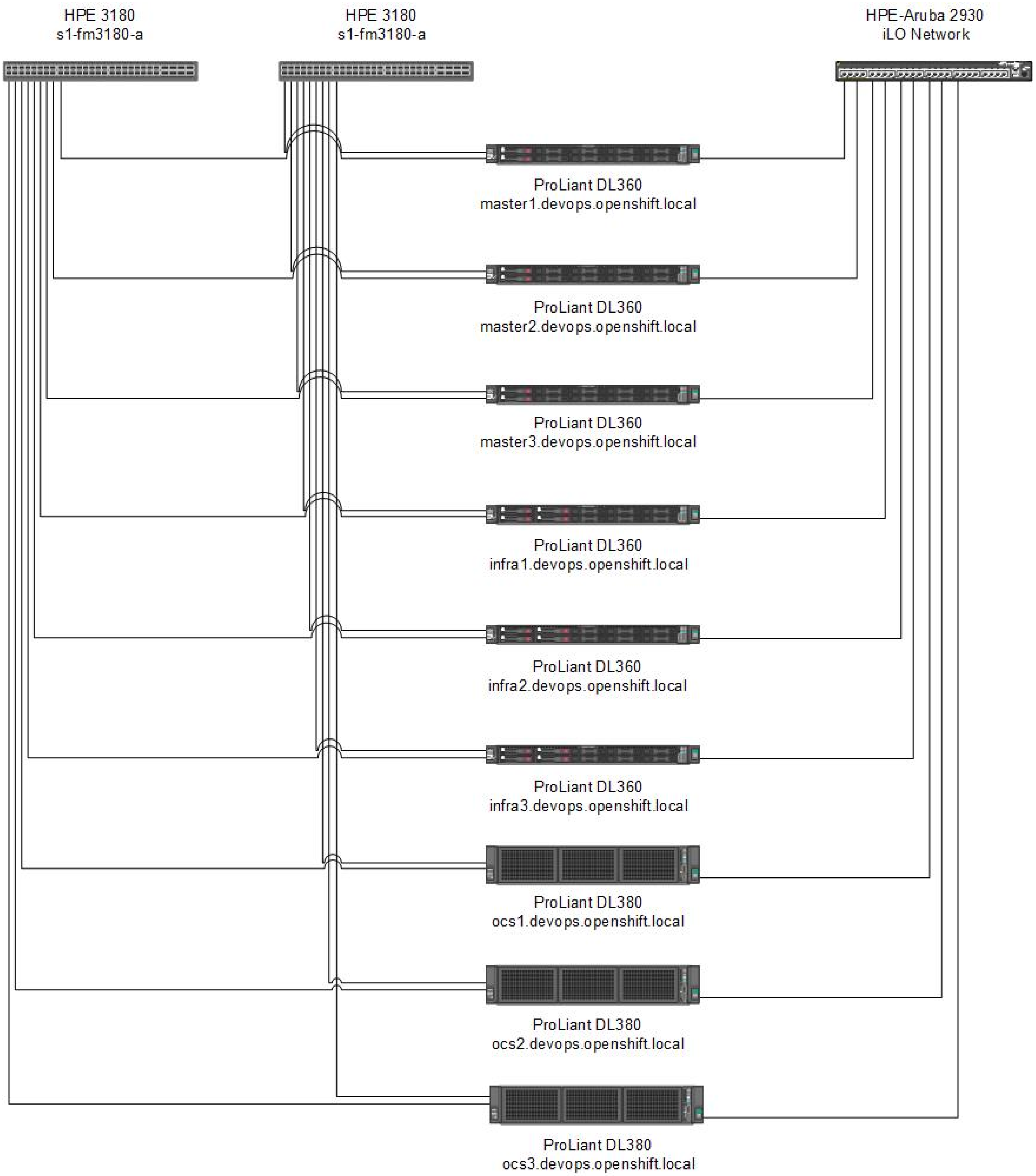


Figure 5: Cabling of the HPE ProLiant DL servers

Configuring the solution switching

The solution described in this document utilized HPE Composable Fabric. The HPE Composable Fabric Manger dashboard is shown in Figure 6. This solution consists of a single fabric and two switches connected to the HPE ProLiant DL servers as described in the previous section.

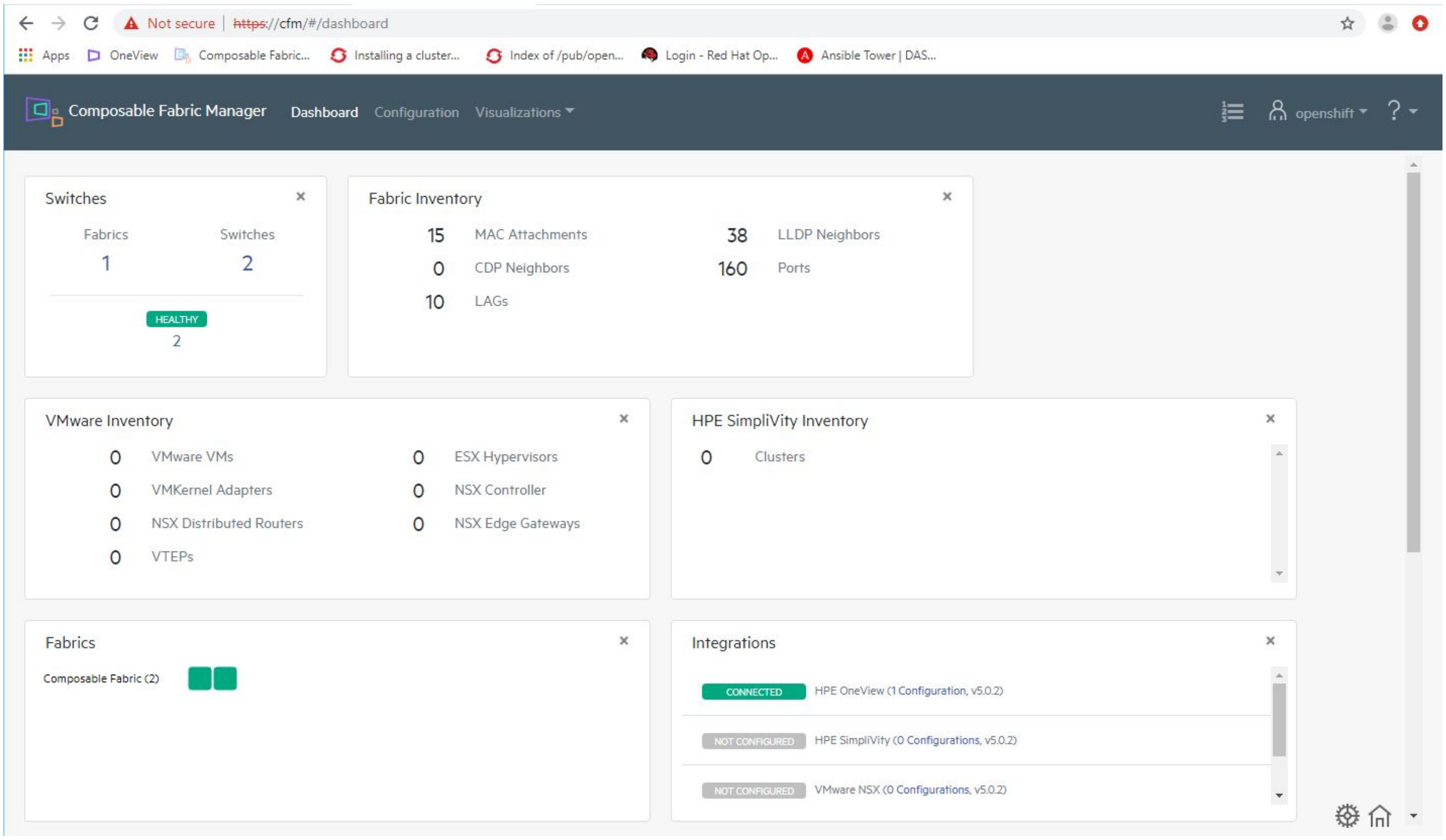
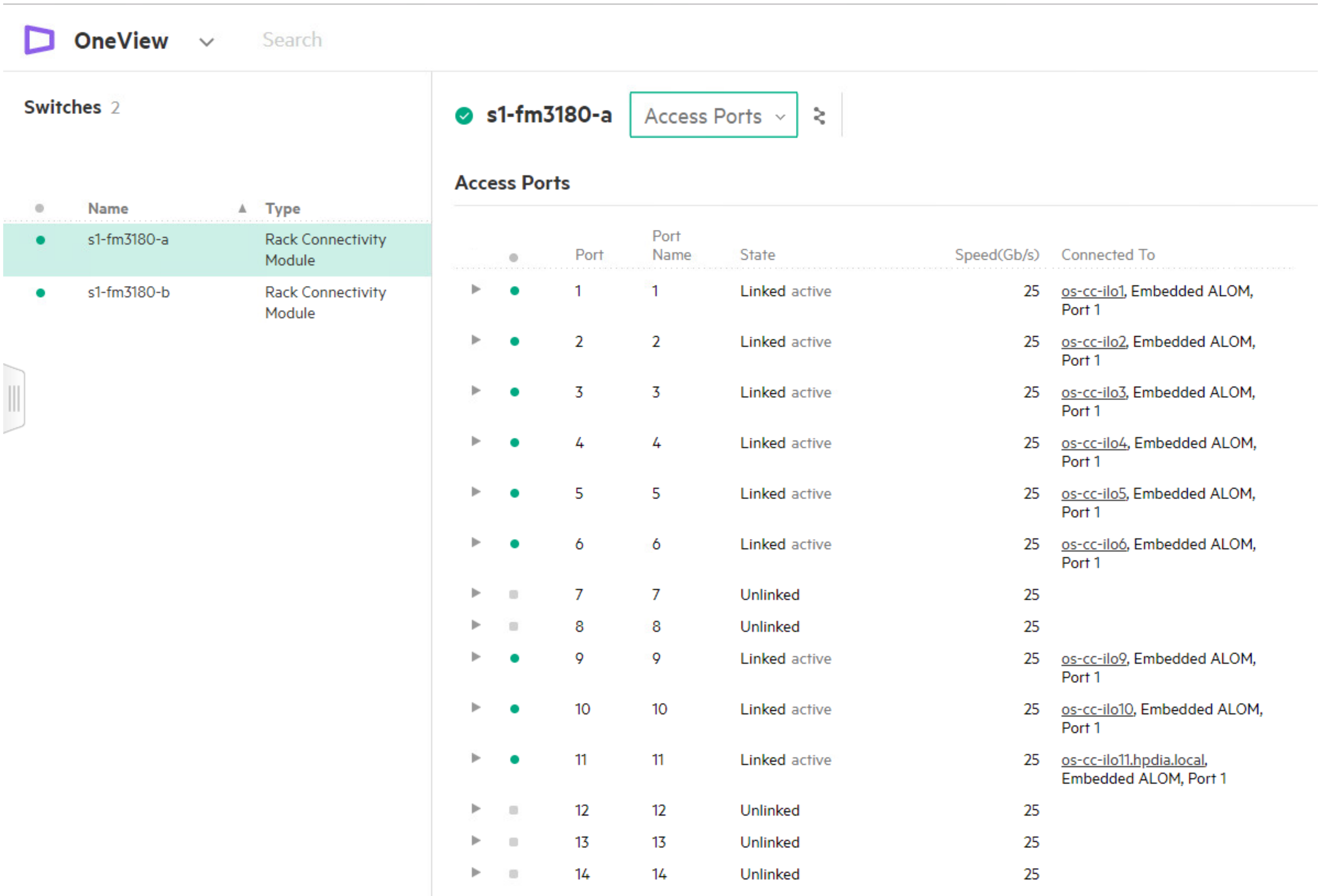


Figure 6. HPE Composable Fabric Manager dashboard

HPE Composable Fabric can also be managed through the HPE OneView interface. Figure 7 displays the HPE Composable Fabric FM 3180 Switch ports and their respective connections to the HPE ProLiant DL servers.



Port	Port Name	State	Speed(Gb/s)	Connected To
1	1	Linked active	25	os-cc-ilo1, Embedded ALOM, Port 1
2	2	Linked active	25	os-cc-ilo2, Embedded ALOM, Port 1
3	3	Linked active	25	os-cc-ilo3, Embedded ALOM, Port 1
4	4	Linked active	25	os-cc-ilo4, Embedded ALOM, Port 1
5	5	Linked active	25	os-cc-ilo5, Embedded ALOM, Port 1
6	6	Linked active	25	os-cc-ilo6, Embedded ALOM, Port 1
7	7	Unlinked	25	
8	8	Unlinked	25	
9	9	Linked active	25	os-cc-ilo9, Embedded ALOM, Port 1
10	10	Linked active	25	os-cc-ilo10, Embedded ALOM, Port 1
11	11	Linked active	25	os-cc-ilo11.hpdia.local, Embedded ALOM, Port 1
12	12	Unlinked	25	
13	13	Unlinked	25	
14	14	Unlinked	25	

Figure 7. Switch port connections

The network map can be viewed through the HPE OneView interface as shown below in Figure 8. This view displays the logical network connections originating from HPE Composable Fabric FM 3180 Switch to HPE ProLiant DL servers.

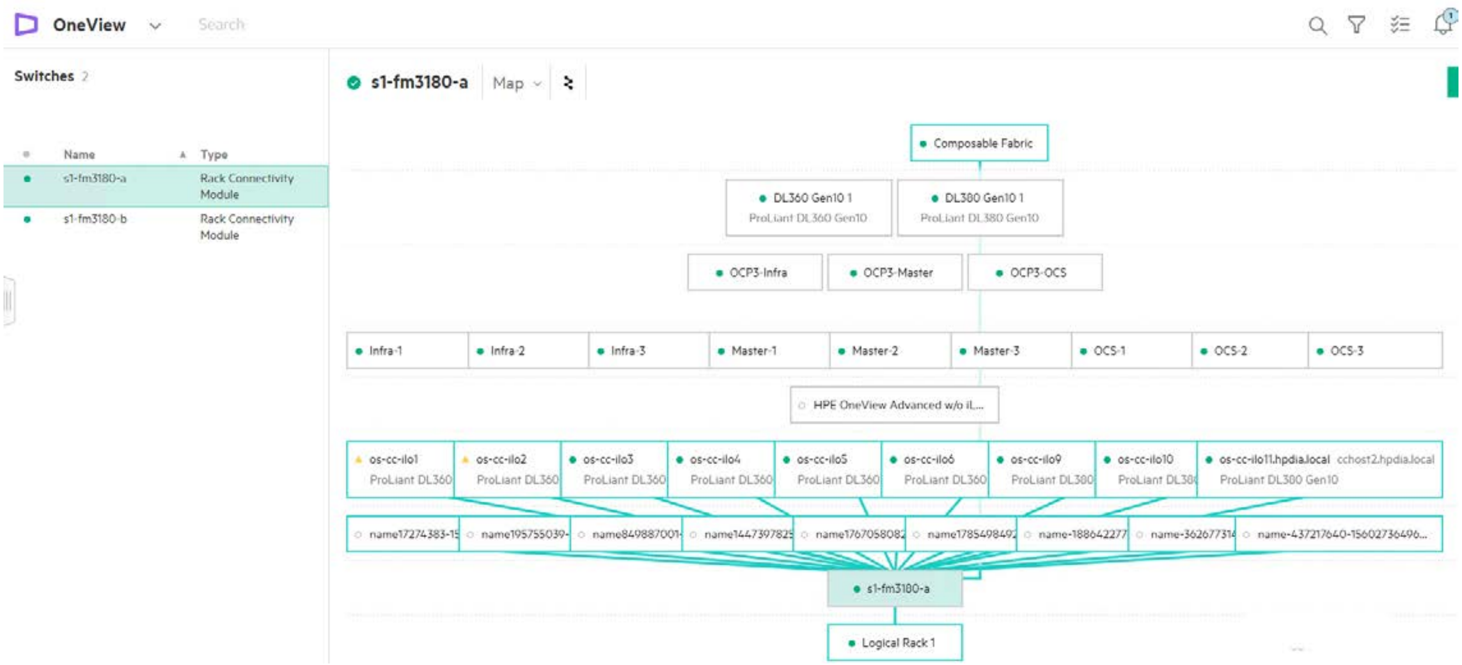


Figure 8. HPE OneView network map

Network definitions

Table 8 defines the networks configured using HPE OneView in the creation of this solution. These networks should be defined at both the first layer switch. This solution utilizes one VLAN for the data center and solution management segments. Actual VLAN IDs and number of VLANs will be determined by the requirements of your production environment.

Table 8. Networks used in this solution

Network Function	VLAN Number	Bridge Aggregation Group
Management	2000	LAG1
Data	2001	LAG1

Solution configuration

HPE OneView configuration

It is beyond the scope of this document to describe the installation and setup of HPE OneView. Refer to the HPE OneView installation and configuration procedure found at https://h20195.www2.hpe.com/V2/getpdf.aspx/4AA4-8543ENW.pdf. For the purposes of this solution, the HPE OneView configuration will focus on creating and applying server profile templates, server profiles, and applying server profiles to HPE ProLiant DL servers.

Configure networks

As explained in the network configuration section of this document, the solution utilizes at least two (2) network segments. Use the **Create network** option illustrated in Figure 9 to define the networks shown in Table 9. Your VLAN values will generally differ from those described below.

Table 9. Networks defined within HPE OneView for this solution

Network Name	Type	VLAN Number	Purpose
Management	Ethernet	2000	Solution management
Data	Ethernet	2001	Application, authentication, and other user networks

The networks created for these servers are the Data network and the Management network, as shown in Figure 9. The Management network is the subnet that provides DNS, HTTP, PXE, and DHCP services to the HPE ProLiant DL servers. This is also the subnet where the Ansible host is connected. The OpenShift web consoles will also be accessible on the subnet and VLAN residing on the Management network. The Data network can be utilized for client access to deployed applications.

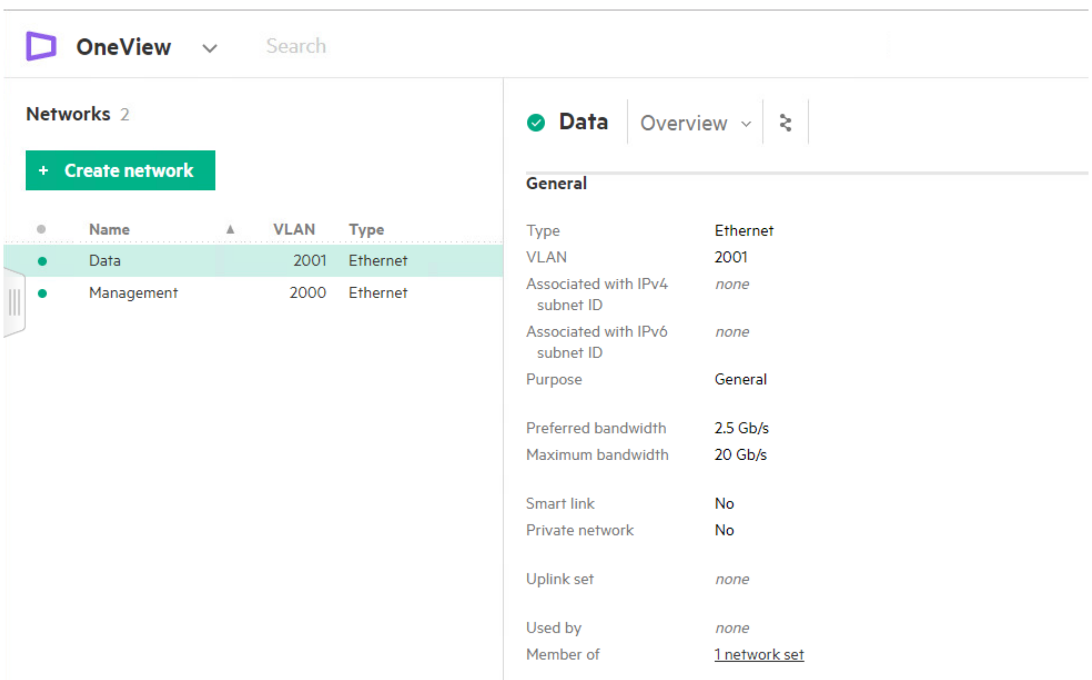


Figure 9. OneView networks

A network set was created in HPE OneView that contains the Management and Data networks. This network set will be connected to the two 25Gb interface cards on the HPE ProLiant DL servers, as shown in Figure 10.

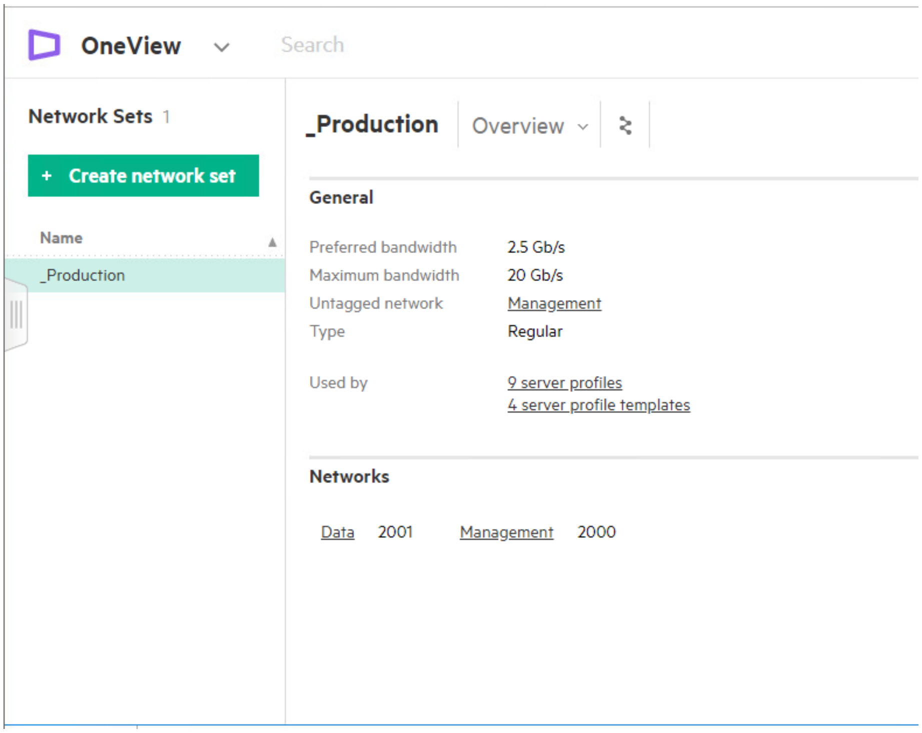


Figure 10. HPE OneView network set

The Management VLAN is untagged to facilitate network boot functions that are used to deploy RHEL to the HPE ProLiant DL servers. Figure 11 displays the option to be configured to set the Management network to untagged.

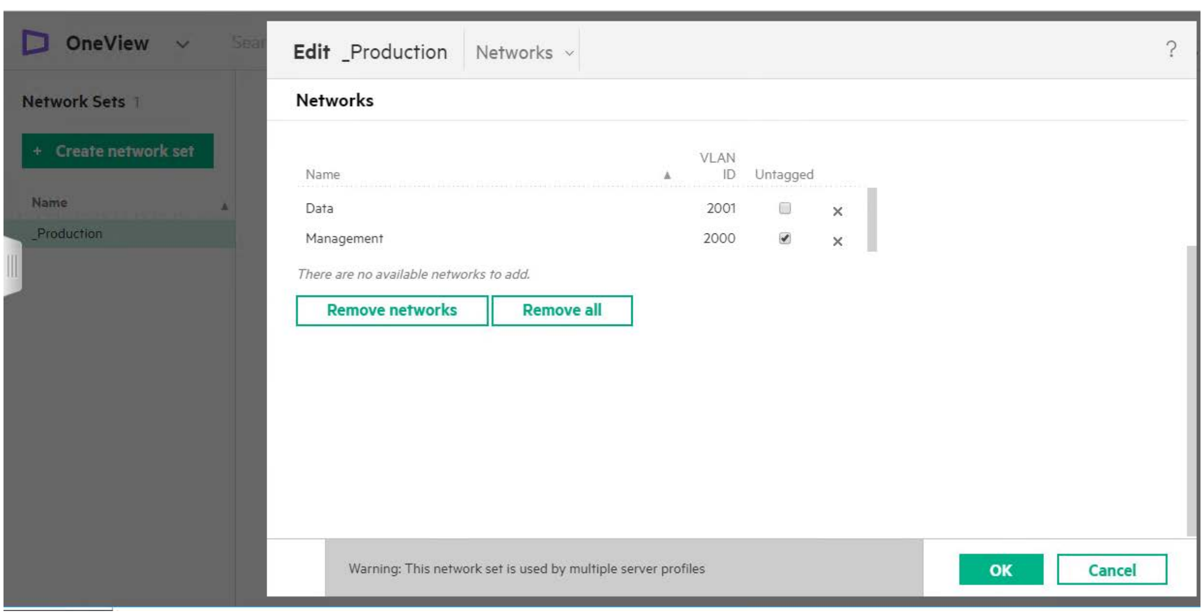


Figure 11. Untagged Management VLAN

Server profiles

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

HPE OneView provides the construct of a server profile. A server profile allows a suite of configuration parameters, including network and SAN connectivity, BIOS tuning, boot order configuration, local storage configuration and more to be templatized. These templates are the key to delivering the “infrastructure as code” capabilities of the HPE Composable Rack platform. For the purpose of this solution, templates were created for the different OpenShift server roles (master, infra, ocs, and worker). The profiles were then applied to the respective servers. Profiles can be created and applied either manually using the HPE OneView management console or automated using Ansible and the Ansible Modules for HPE OneView playbooks.

The critical items configured as part of the template were the connections and storage. Figure 12 describes the configuration of the network interfaces as part of the server profile template. There are 2 redundant networks that are defined.

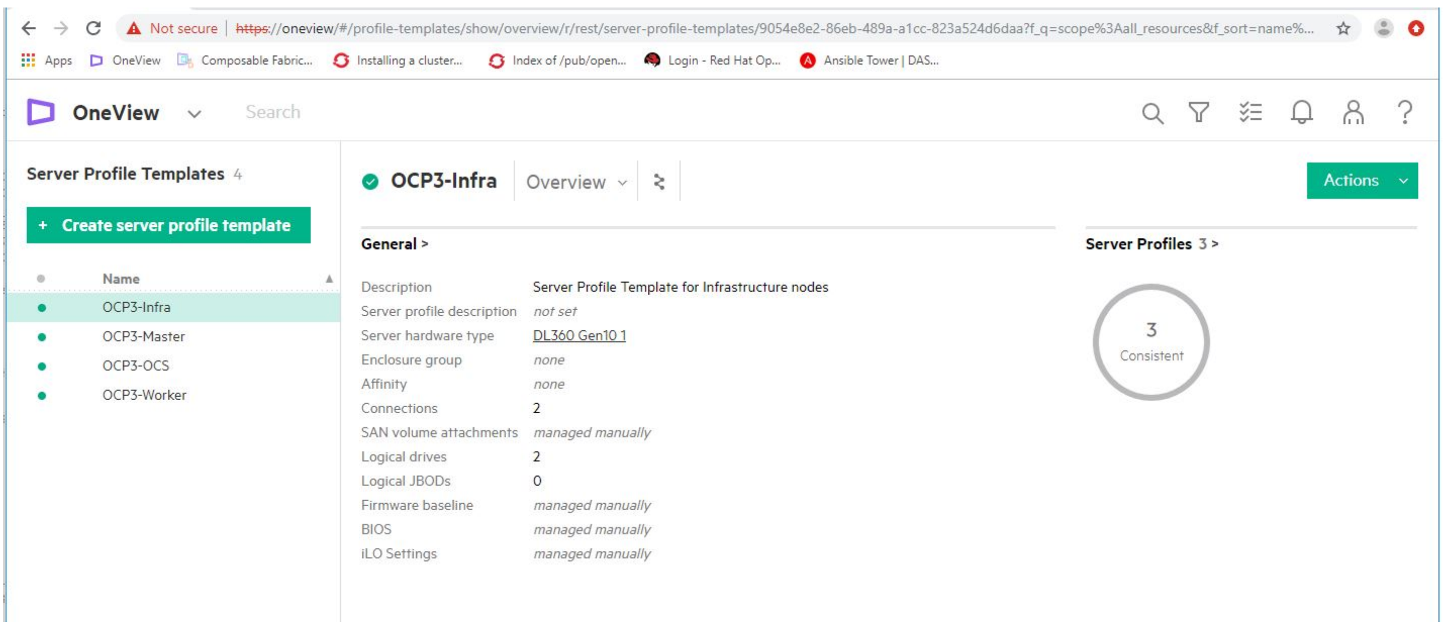


Figure 12. Server profile templates

Network connections

This solution uses two (2) 25Gb network interfaces that are connected to a network set defined in HPE OneView. The network interfaces are shown in Figure 13 as FlexibleLOM 1:1 and FlexibleLOM 1:2. These interfaces will be represented in RHEL as "eno5" and "eno6d1" respectively. Interface "eno5" will be used as a network boot device to install RHEL 7.6 and then reconfigured to be a slave interface for bond0 by an Ansible playbook (bonds.yml) described later in this document. These interfaces are connected to the network set "_Production".

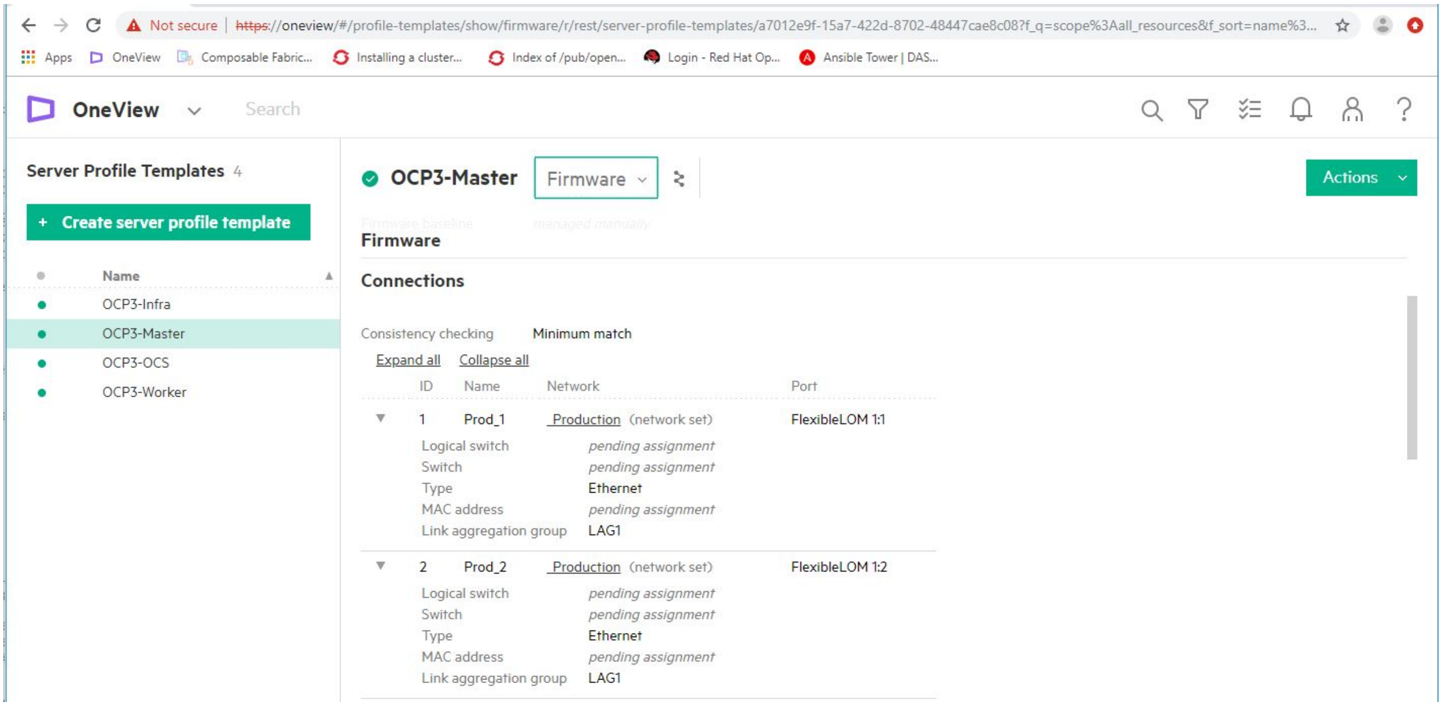


Figure 13. Server network connections defined in the profile template

---

Note

The MAC addresses are defined in the server profiles. The MAC address for the interface (FlexibleLOM1:1) connected to the PXE network is required to create a DHCP reservation as describe in Appendix A.

Storage connections

The storage for this solution is local storage provided by the internal drive bays of the HPE ProLiant DL360 and DL380 servers. The storage connections in the server profile define the volumes that will be created on the host from physical disks presented from the P408i Smart Array and the internal storage drive bays.

Each of the HPE ProLiant hosts have the following volumes:

- Bootvol: RAID1 volume provided by the internal drives in the HPE ProLiant DL servers
- OCSVol: RAID1 volume presented from the internal drive bays in the HPE ProLiant DL servers and used for Red Hat OpenShift Container Storage

All storage in this solution is local direct-attached storage. RAID1 Arrays were presented by the internal P408i Smart Array controller. Table 10 describes the storage volumes on a per-node basis. Each of the master servers had a single RAID1 array. The infrastructure and OpenShift Container Storage nodes were each configured with two (2) RAID1 arrays, providing each server with a boot volume and a volume for Red Hat OpenShift Container Storage.

Table 10. Volumes and sources used in this solution

Volume/Disk Function	Drive Qty	Size	Source	Hosts	Shared/Dedicated
Operating System	2	400 GB	Internal drive bays	All Nodes	Dedicated
Persistent Application Data – OCS	2	2 TB	Internal drive bays	OpenShift worker node	Replicated
Registry, Logging, and metrics – OCS	2	400GB	Internal drive bays	Infrastructure node	Replicated

Figure 14 shows the configuration of the local storage from the server profile used for the master nodes. In this solution, the master nodes have only a single RAID1 array used for the boot volume.

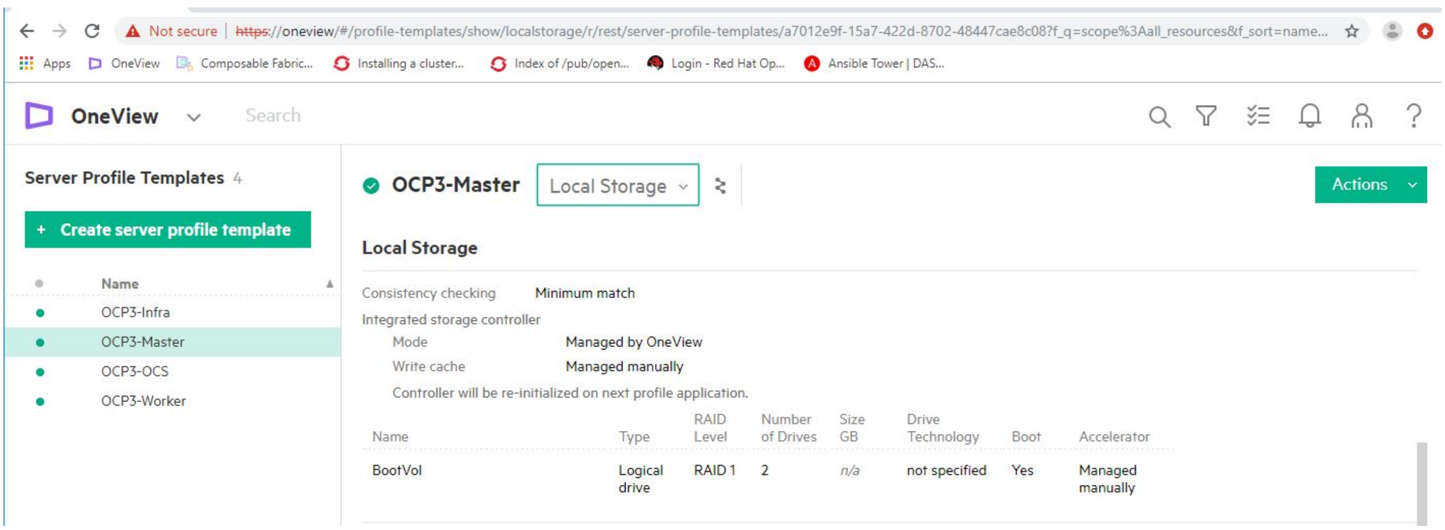


Figure 14. Master Node Storage

Additional volumes or RAID types can be added for increased capacity and performance. In this solution, the operating system will use /dev/sda for RHEL and /dev/sdb for Red Hat OpenShift Container Storage (OCS). The device names used for the OCS cluster are specified in the Ansible inventory file. If additional volumes are created for OCS, they must also be specified in the Ansible inventory file used for the Red Hat OpenShift installation.

Figure 15 shown below illustrates the storage configuration in the server profile template used for the OpenShift Container Storage nodes. In this illustration, we can see a second RAID1 volume with a name of OCSVol. This volume will be identified in the operating system as /dev/sdb and used for the OpenShift Container Storage volume.

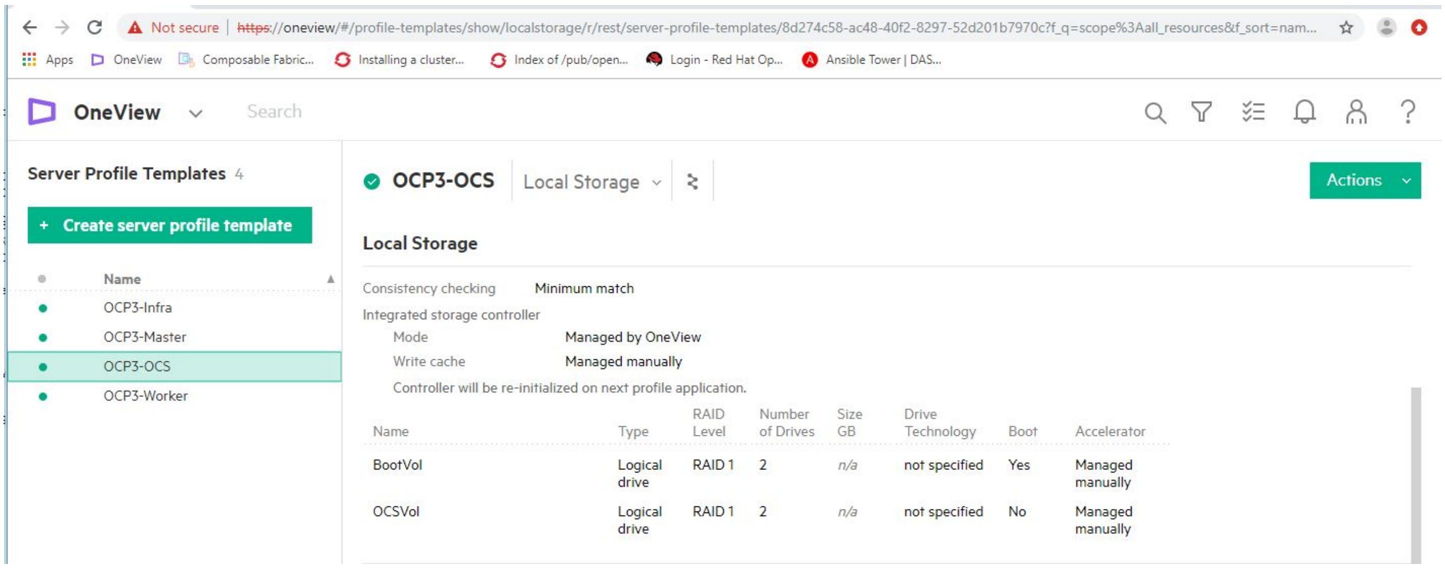


Figure 15. OCS and Infra node storage

Once the server profile templates are created for each of the OpenShift server node types, server profiles can be created and applied to the HPE ProLiant DL servers. Figure 16 illustrates the server profiles created for this solution and applied to the HPE ProLiant DL 360 Gen10 and HPE ProLiant DL380 Gen10 servers.

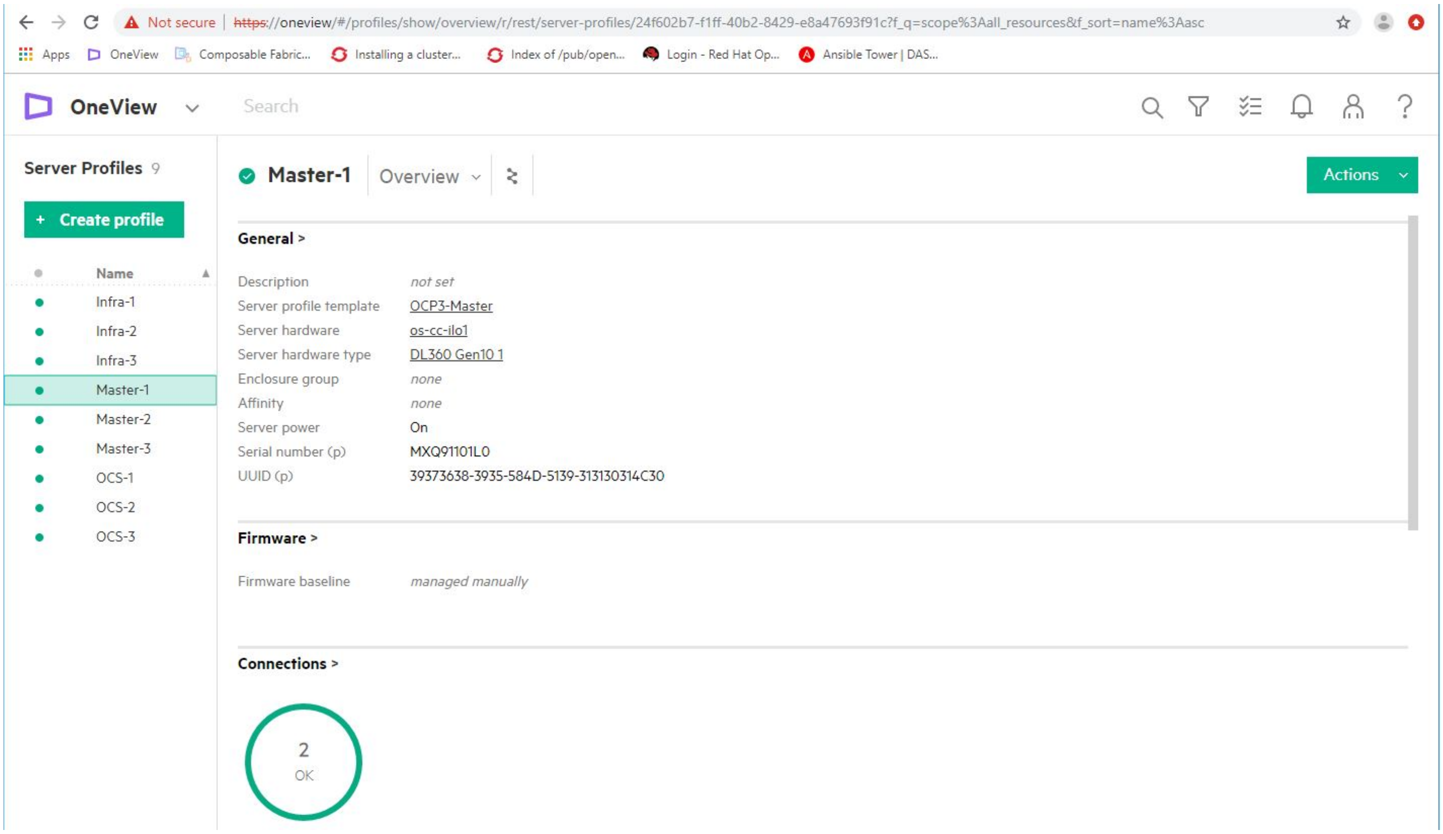


Figure 16. Server profiles

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 Enterprise Linux

This section outlines the installation and configuration of the Red Hat Enterprise Linux 7.6. In the previous section, server profiles were created and applied to the HPE ProLiant DL servers. The HPE OneView interface can be used to monitor the HPE ProLiant DL server consoles and power on the servers. The servers are configured to boot from the network adapter that is connected to the management network. The management network provides DHCP, DNS, and PXE services. The HPE ProLiant DL servers were configured for Legacy BIOS performing a network boot to one of the 25Gb interfaces (FlexibleLOM 1:1). In the server BIOS, this adapter is Port 1 of the HPE Eth 10/25Gb 2 P 631FLR-SFP28 Embedded ALOM adapter.

Once the server boots and RHEL is installed, the network interface will be reconfigured using an Ansible playbook to become a slave interface to a pair of bonded 25Gb interfaces.

Red Hat Enterprise Linux host installation

This solution utilizes a combination of kickstart and Ansible playbooks to configure Red Hat Enterprise Linux on the nine (9) HPE ProLiant DL servers.

The PXE server used in this solution resides on VLAN 2000. The NICs eno5 (FlexibleLOM 1:1) and ens6d1 (FlexibleLOM 1:2) are connected to the Management VLAN 2000. This VLAN is an untagged VLAN in the "_Production" network set as configured in HPE OneView. The servers have been configured to perform a network boot from the first 25Gb interface (FlexibleLOM 1:1) and a corresponding DHCP reservation has been created on the DHCP server for the MAC address assigned to FlexibleLOM 1:1.

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

Ensure the Management interface for each RHEL host is registered in DNS prior to proceeding. A sample DNS zone file is provided in Appendix C.

The following kickstart file is hosted over HTTP. Refer to Appendix A for an example of setting up a server that will provide PXE, HTTP, and DHCP services to enable the PXE boot environment.

```
#version=DEVEL
# Use network installation
url --url="http://172.22.0.1/rhel7-7/"
repo --name="Server-HighAvailability" --baseurl=http://172.22.0.1/rhel7-7//addons/HighAvailability
repo --name="Server-ResilientStorage" --baseurl=http://172.22.0.1/rhel7-7//addons/ResilientStorage
# Use text mode install
text
ignoredisk --only-use=sda
# Keyboard layouts
keyboard --vckeymap=us --xlayouts=''
# System language
lang en_US.UTF-8

# Network information
network  --bootproto=dhcp --device=eno1 --onboot=off --ipv6=auto --no-activate
network  --bootproto=dhcp --device=eno5 --ipv6=auto --activate
network  --bootproto=static --device=eno6d1 --onboot=off --ipv6=auto --no-activate
network  --bootproto=dhcp --device=eno2 --onboot=off --ipv6=auto --no-activate
network  --bootproto=dhcp --device=eno3 --onboot=off --ipv6=auto --no-activate
network  --bootproto=dhcp --device=eno4 --onboot=off --ipv6=auto --no-activate
#uncomment network setting for bond0 when using for ilo vMedia UEFI install
#network --device=bond0 --bondslaves=eno5,eno6d1 --bondopts=mode=802.3ad --bootproto=dhcp -activate
# Reboot after installation
reboot
# Root password
rootpw --plaintext changeme
# System services
services --enabled="chronyd"
# Do not configure the X Window System
skipx
# System timezone
timezone America/Chicago --isUtc
# System bootloader configuration
bootloader --append=" crashkernel=auto" --location=mbr --boot-drive=sda
autopart --type=plain
# Clear the Master Boot Record
zerombr
```

```
# Partition clearing information
clearpart --linux --initlabel --drives=sda
%packages
@core
chrony
kexec-tools
%end
%addon com_redhat_kdump --enable --reserve-mb='auto'
%end
```

Configure Ansible to run the playbooks

This section outlines the steps required to configure HPE ProLiant DL 360 Gen10 and HPE ProLiant DL380 Gen10 servers once RHEL 7.6 has been installed. Ansible playbooks are used to prepare the hosts for the Red Hat OpenShift Container Platform installation.

Note

The Ansible playbooks used in this solution assume that all servers have Internet connectivity and can access Red Hat Subscription Management to enable the required repositories and install Red Hat OpenShift Container Platform. Disconnected installation procedures for Red Hat OpenShift Container Platform can be found at: https://docs.openshift.com/container-platform/3.11/install/disconnected_install.html.

Ensure there is internet access and the Ansible host is able to reach the HPE GitHub site at https://github.com/HewlettPackard/hpe-solutions-openshift/tree/master/. To validate, run the following command on the Ansible Engine host:

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

It is assumed that the Ansible playbooks are run from /etc/ansible/ and the Ansible inventory is /etc/ansible/hosts. The playbooks are installed in /etc/ansible/playbooks/. Each playbook calls a corresponding Ansible role located in /etc/ansible roles.

Clone the GitHub repository to a local directory on the Ansible host and copy the contents to /etc/ansible. The playbooks and variable files used in this solution assume that the scripts are executed from /etc/ansible on the Ansible host.

From the Ansible Engine host, run the following commands:

```
# cd /etc/Ansible

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

# copy -r ./hpe-solutions-openshift/proliant/composable/ansible/*  /etc/ansible/
```

Custom Ansible playbooks were created specifically for this solution and are used to complete the preparation tasks required for the OpenShift installation. The custom playbooks are listed below:

- subs.yml - Register the HPE ProLiant DL servers with Red Hat Content Delivery Network (CDN)
- prep.yml - Configure a shared ssh key for passwordless login between the Ansible host and the HPE ProLiant DL servers, install Docker runtime on the HPE ProLiant DL servers, create non-root user
- bonds.yml - Configure the bonded interfaces
- passwords.yml - Store sensitive user account and password information in Ansible vault

The playbooks are used to call Ansible roles which contain specific tasks to prepare the OpenShift nodes. .The Ansible roles, associated tasks, and variable files can be found in the roles subdirectory. An example of the directory structure is shown in Figure 17.

```
.
├── ansible.cfg
├── hosts
├── playbooks
│   ├── bonds.yml
│   ├── prep.yml
│   ├── subs.yml
├── roles
│   ├── bonds
│   │   ├── tasks
│   │   │   ├── main.yml
│   │   │   └── netnmcli.yml
│   │   └── vars
│   │       └── main.yml
│   ├── passwords
│   │   ├── files
│   │   ├── tasks
│   │   │   └── main.yaml
│   │   └── vars
│   │       └── passwords.yml
│   ├── prep
│   │   ├── files
│   │   │   └── id_rsa.pub
│   │   ├── tasks
│   │   │   ├── hostname.yml
│   │   │   ├── keys.yml
│   │   │   ├── main.yml
│   │   │   ├── ocpdocker.yml
│   │   │   └── user.yml
│   │   └── vars
│   │       └── main.yml
│   └── subs
│       ├── tasks
│       │   ├── main.yml
│       │   └── rhn.yml
│       └── vars
│           └── main.yml
```

Figure 17. Ansible directory structure

On the Ansible host, install openshift-ansible repository using the `yum -y install openshift-ansible` command as documented in Appendix B. This will install the OpenShift 3.11 playbooks required for the Ansible installation. After the openshift-ansible repository has been installed, the openshift-ansible playbooks, prerequisites.yml, and deploycluster.yml, can be found in the /usr/share/ansible/openshift-ansible/playbooks/ subdirectory.

Additional information on running the playbooks and configuring the variable files are described later in this document in the "Red Hat OpenShift deployment" section. Refer Appendix B, for detailed information on the Ansible host configuration.

Red Hat OpenShift deployment

This section describes the process to automatically deploy Red Hat OpenShift 3.11. The Ansible playbooks mentioned later in this section are used to configure the HPE ProLiant DL server and deploy Red Hat OpenShift.

1. Create the htpasswd file:

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

   An example is shown below.

   admin :$ <hashed password>

   user :$ <hashed password>

2. Generate a key. On the Ansible Engine host, run the following command:

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

   This will create a key file at ~/.ssh/id_rsa.pub. This key will be used to provide passwordless login from the Ansible host to the openshift nodes. The key should be copied to /etc/ansible/roles/prep/files to be used by the keys.yml task in the prep role.

3. Edit the Ansible vault password file that will be used to store secrets for system user accounts and passwords.

4. Edit the following variable files to match your environment.

   ```
   # vi /etc/ansible/roles/subs/vars/main.yml
   # vi /etc/ansible/roles/bonds/vars/main.yml
   # vi /etc/ansible/roles/prep/vars/main.yml
   # vi /etc/ansible/hosts
   # ansible-vault edit /etc/ansible/roles/passwords/vars/passwords.yml (encrypted file must be edited with the ansible-vault edit command)
   ```

Install prerequisites for Red Hat OpenShift installation

To run the Ansible playbooks, execute the following command on the Ansible Engine host:

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

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

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

OpenShift-Ansible

The Ansible playbooks mentioned in this section will deploy Red Hat OpenShift Container Platform on the virtual machines that have been created by the previous Ansible playbooks. The variables for the Red Hat OpenShift deployment are maintained in the Ansible inventory file, which is located at /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/.

Install Red Hat OpenShift

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

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

   ```
   # ansible-playbook -i hosts -e@roles/passwords/vars/passwords.yml /usr/share/ansible/openshift-ansible/playbooks/prerequisites.yml --ask-vault-pass
   ```

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

```
# ansible-playbook -i hosts -e@roles/passwords/vars/passwords.yml /usr/share/ansible/openshift-ansible/playbooks/deploy_cluster.yml --ask-vault-pass
```

The Red Hat OpenShift playbooks can take approximately 1.5 - 2 hours to complete the deployment. Table 11 shows the outputs from the Ansible playbooks that illustrate the time to complete each playbook and its associated tasks. The actual time to run the Ansible playbooks may vary by environment and could be impacted by network latency and other configuration variables.

Table 11: Playbook run times

Ansible Playbook	Time (hh:mm:ss)
subs.yml	0:14:09
bonds.yml	0:02:12
prep.yml	0:04:29
prerequsites.yml	0:12:13
deploy_cluster.yml	1:08:55

A complete breakdown of the playbook timing and tasks can be found in Appendix D.

When the deployment has finished, the installer may access the Red Hat OpenShift webpage, shown in Figure 17, using the credentials provided in the htpasswd file. The URL will be https://<your_system_name.domain>:8443/login.

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

Validate Red Hat OpenShift deployment

The final step in the deployment process is to validate that the deployment succeeded. To accomplish this, this section 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 master1.openshift.devops.local and run the **oc get nodes** command to ensure all nodes have a status of **Ready**.

```
# oc get nodes
NAME                                STATUS    ROLES     AGE       VERSION

infra1.devops.openshift.local        Ready     infra    5h         v1.11.0+d4cacc0
infra2.devops.openshift.local        Ready     infra     5h         v1.11.0+d4cacc0
infra3.devops.openshift.local        Ready     infra     5h         v1.11.0+d4cacc0
master1.devops.openshift.local       Ready     master    6h         v1.11.0+d4cacc0
master2.devops.openshift.local       Ready     master    6h         v1.11.0+d4cacc0
master3.devops.openshift.local       Ready     master    6h         v1.11.0+d4cacc0
ocs1.devops.openshift.local          Ready     compute   5h         v1.11.0+d4cacc0
ocs2.devops.openshift.local          Ready     compute   5h         v1.11.0+d4cacc0
ocs3.devops.openshift.local          Ready     compute   5h         v1.11.0+d4cacc0
```

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 pods

NAME                                           READY    STATUS      RESTARTS     AGE
docker-registry-1-bm96v                         1/1      Running        0          5h
glusterblock-registry-provisioner-dc-1-kxwsw    1/1      Running        0          5h
glusterfs-registry-24kdf                        1/1      Running        0          5h
glusterfs-registry-52spt                        1/1      Running        0          5h
glusterfs-registry-bgsmx                        1/1      Running        0          5h
heketi-registry-1-8h97r                         1/1      Running        0          5h
registry-console-1-w9nz6                        1/1      Running        0          5h
router-1-4tlt9                                  1/1      Running        0          5h
router-1-4zbtj                                  1/1      Running        0          5h
router-1-7jshj                                  1/1      Running        0          5h
```

Administrator login and cluster admin role

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

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

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

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

```
* default
    glusterfs
    kube-public
    kube-service-catalog
    kube-system
    management-infra
    openshift
    openshift-ansible-service-broker
    openshift-console
    openshift-infra
    openshift-logging
    openshift-metrics-server
    openshift-monitoring
    openshift-node
    openshift-sdn
    openshift-template-service-broker
    openshift-web-console
```

Grant cluster role to user

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

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

Assigning the cluster-admin role will provide the user with access to the Red Hat OpenShift cluster web console and Grafana Dashboards. The cluster-admin role is not required to deploy applications or access the default Red Hat OpenShift web console.

Red Hat OpenShift web console

Access the Red Hat OpenShift web console via https://<your_system_name.domain>:8443/login. This will prompt you for a username and password. Use the credentials created with the htpasswd utility and specified in the Ansible inventory file.

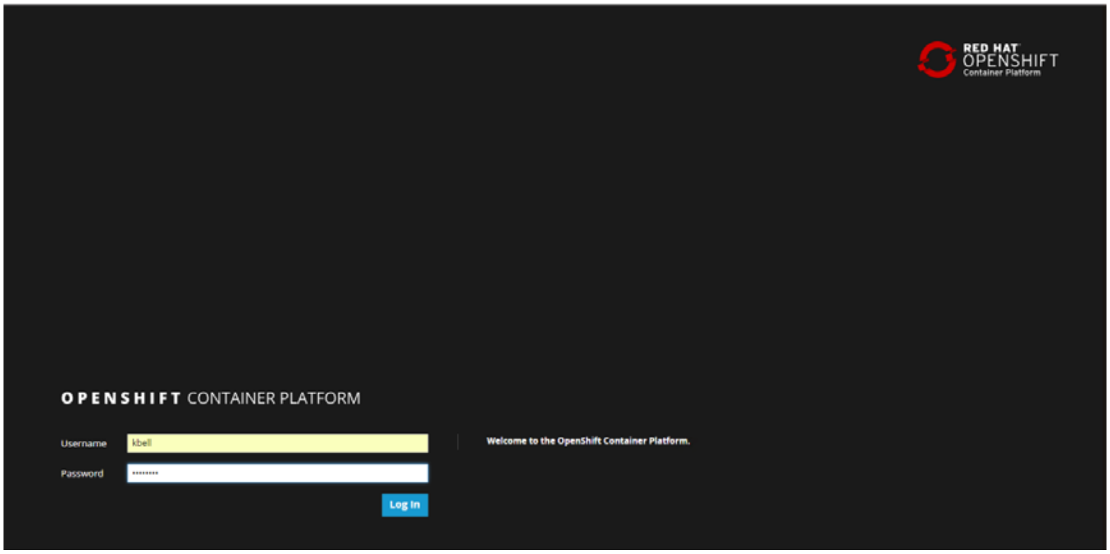


Figure 18: Red Hat OpenShift web console

Using the credentials that were granted, the cluster-admin role in the previous step provides access to the Red Hat OpenShift cluster console, administration, and monitoring web consoles. Red Hat OpenShift Cluster Console provides an overview of the cluster status and control plane as shown below in Figure 19.

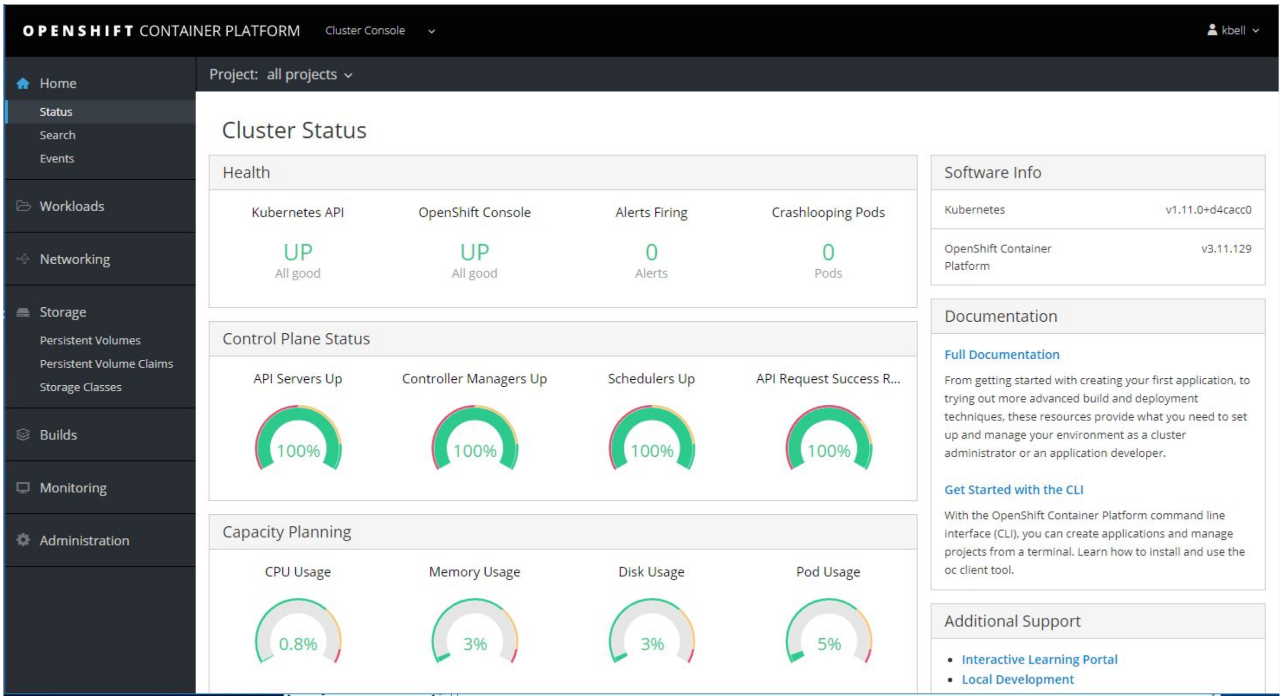


Figure 19. Red Hat OpenShift cluster status

In addition to the cluster status, the Cluster Console can be used to administer the cluster. Figure 20 shows the nodes view of the Administration section. In the nodes view, an administrator can view the status of the Red Hat OpenShift cluster nodes and drill down into each node.

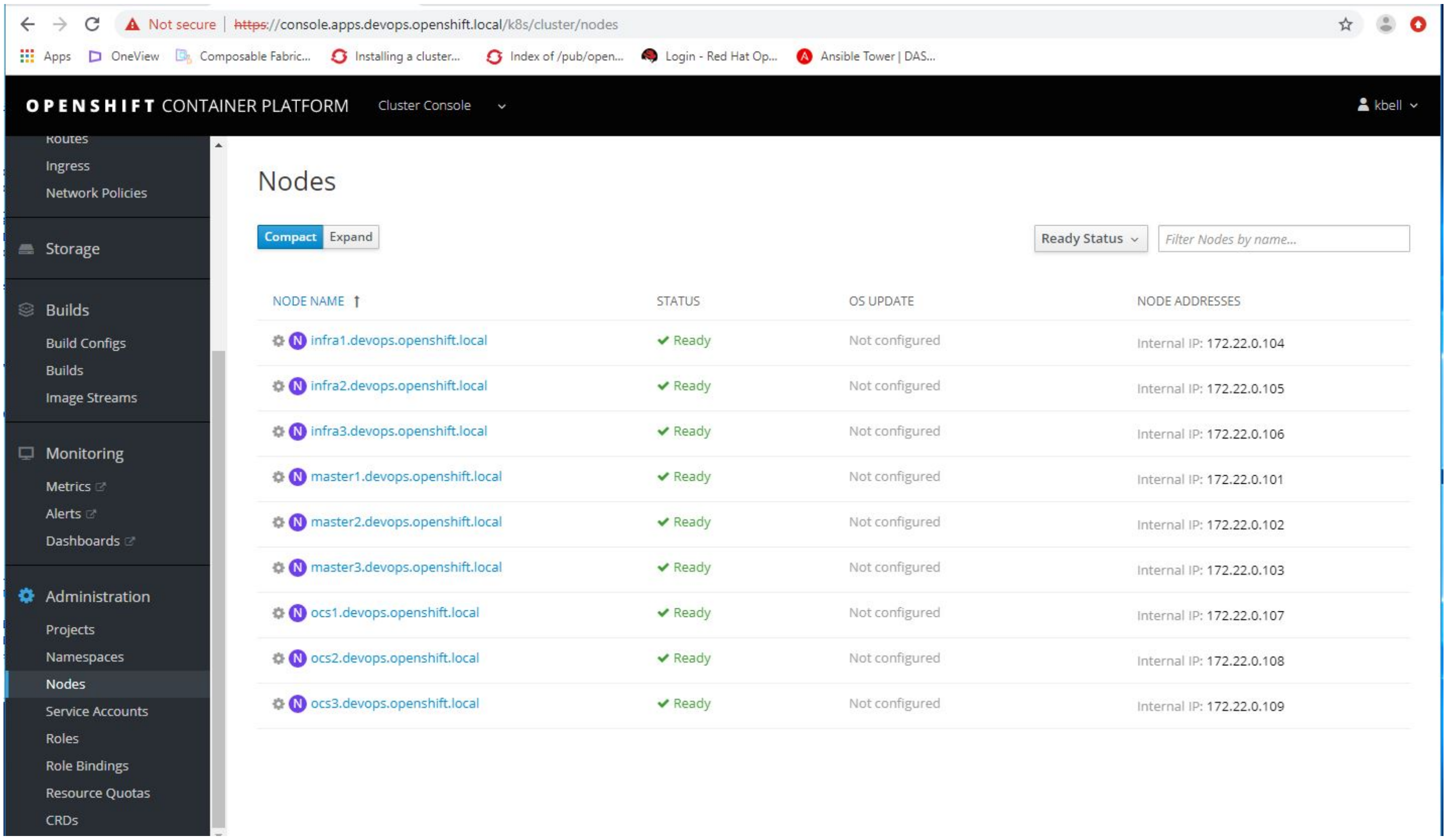


Figure 20. Red Hat OpenShift cluster nodes

The Grafana dashboard can be accessed at grafana-openshift-monitoring <openshift domain>. There are several built-in dashboards installed with Red Hat OpenShift Container Platform as shown in Figure 21.

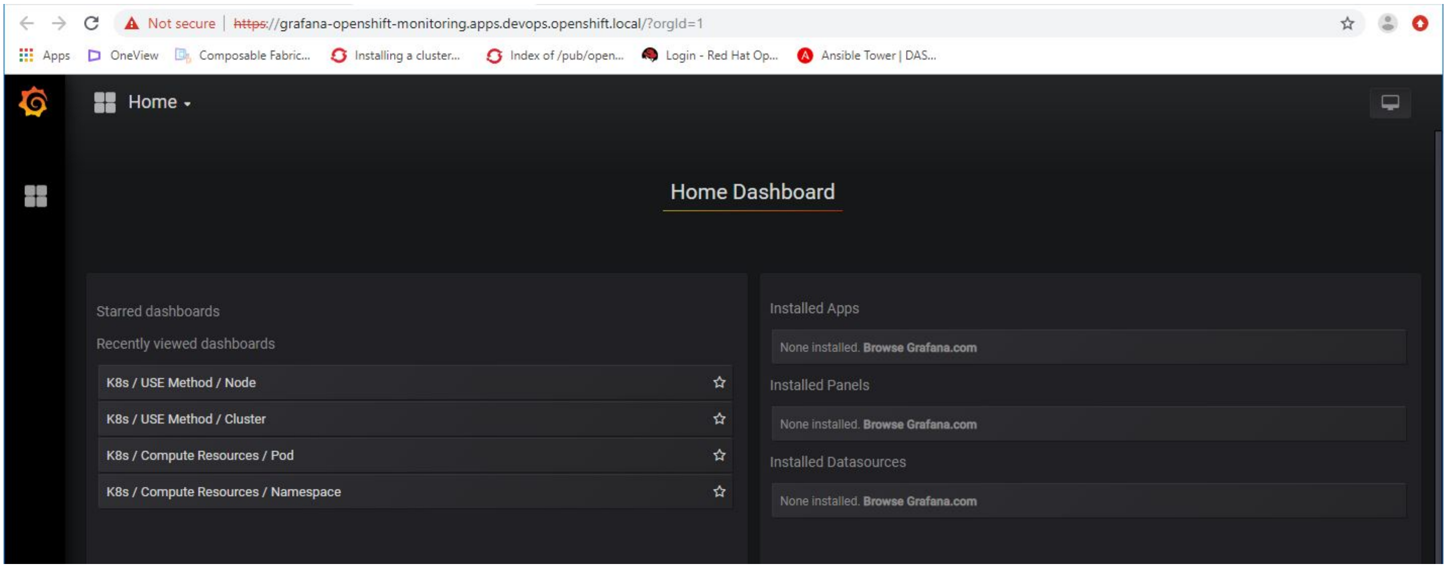


Figure 21. Grafana dashboards

Grafana can be used to display the metrics collected by Prometheus. In the Node dashboard, the performance metrics can be displayed for the Red Hat OpenShift nodes as shown in Figure 22.

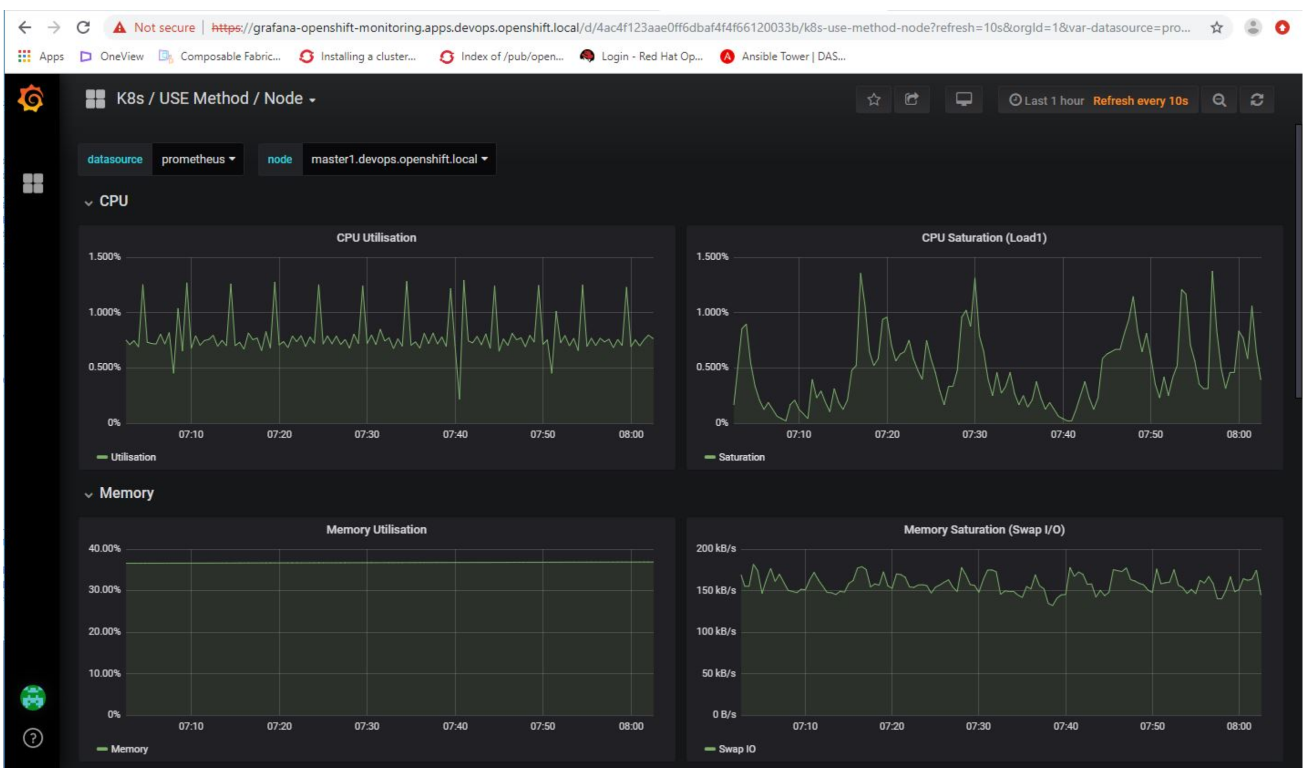


Figure 22. Node metrics

This solution uses the gluster based OpenShift Container Storage (OCS) for persistent storage. There are two instances of OCS deployed in this solution. One OCS cluster spans the infrastructure nodes and provides persistent storage for the OpenShift registry, metrics, and logging. The second OCS cluster spans the application nodes and provides persistent storage for the organization's production applications.

Figure 23 displays the storage classes that are created with deployment of OpenShift. The gluster-registry-block storage class is used to provide persistent storage for the OpenShift registry. The glusterfs-storage provides storage for the organization's applications.

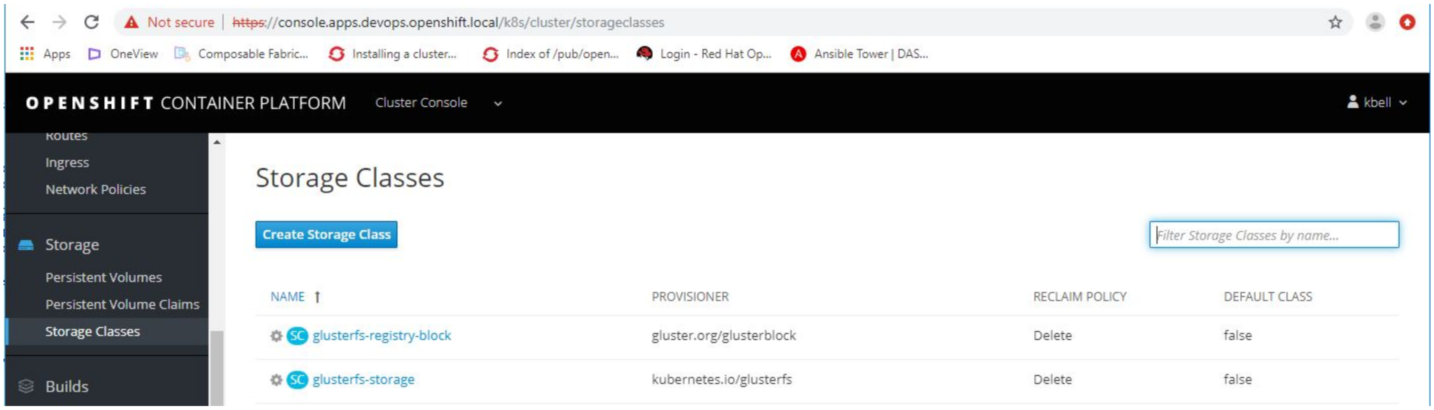


Figure 23. Persistent storage

An additional storage class is required to provide persistent storage for OpenShift logging and metrics. This storage class is called glusterfs-storage-block and must be manually created. Using the **Create Storage Class** option, create a storage class named glusterfs-storage-block as shown in Figure 24.

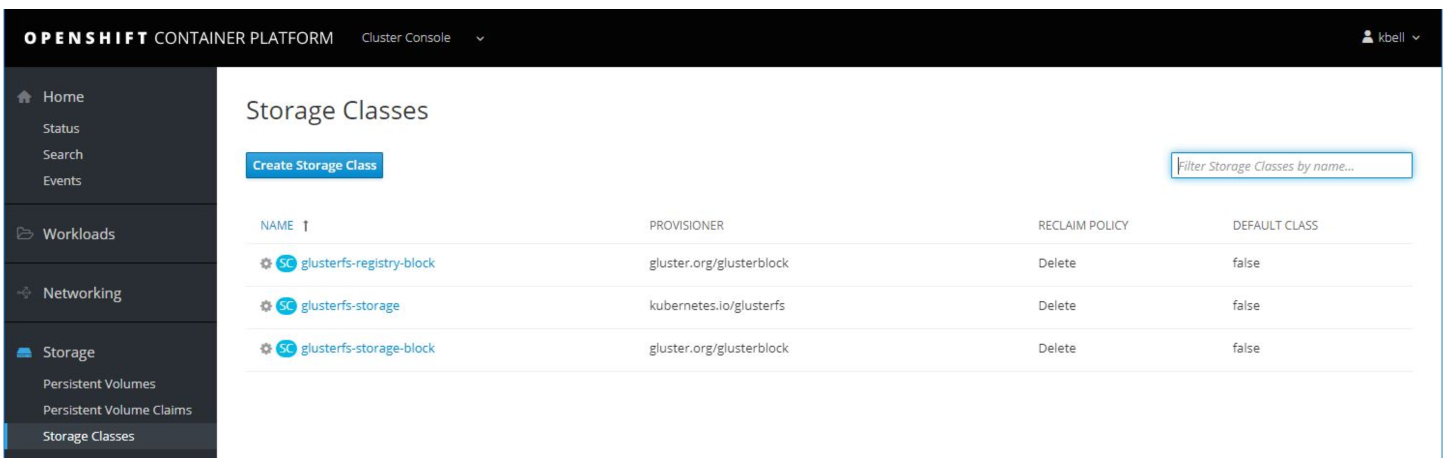


Figure 24: glusterfs-storage-block

The yaml file for the storage class should be similar to the one shown below:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: glusterfs-storage-block
  selfLink: /apis/storage.k8s.io/v1/storageclasses/glusterfs-storage-block
  uid: e108f31f-aeed-11e9-8981-20677ce3a32c
  resourceVersion: '11609'
  creationTimestamp: '2019-07-25T15:07:11Z'
provisioner: gluster.org/glusterblock
parameters:
  chapauthenabled: 'true'
  hacount: '3'
  restsecretname: heketi-registry-admin-secret-block
  restsecretnamespace: infra-storage
  resturl: 'http://heketi-registry.infra-storage.svc:8080'
  restuser: admin
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

Figure 25 shows the persistent volume claims that are used for the OpenShift registry, metrics, and logging services. A persistent storage volume must be created for the metrics. Create a persistent storage volume, as shown in Figure 25, by selecting the **Create Persistent Volume Claim** option.

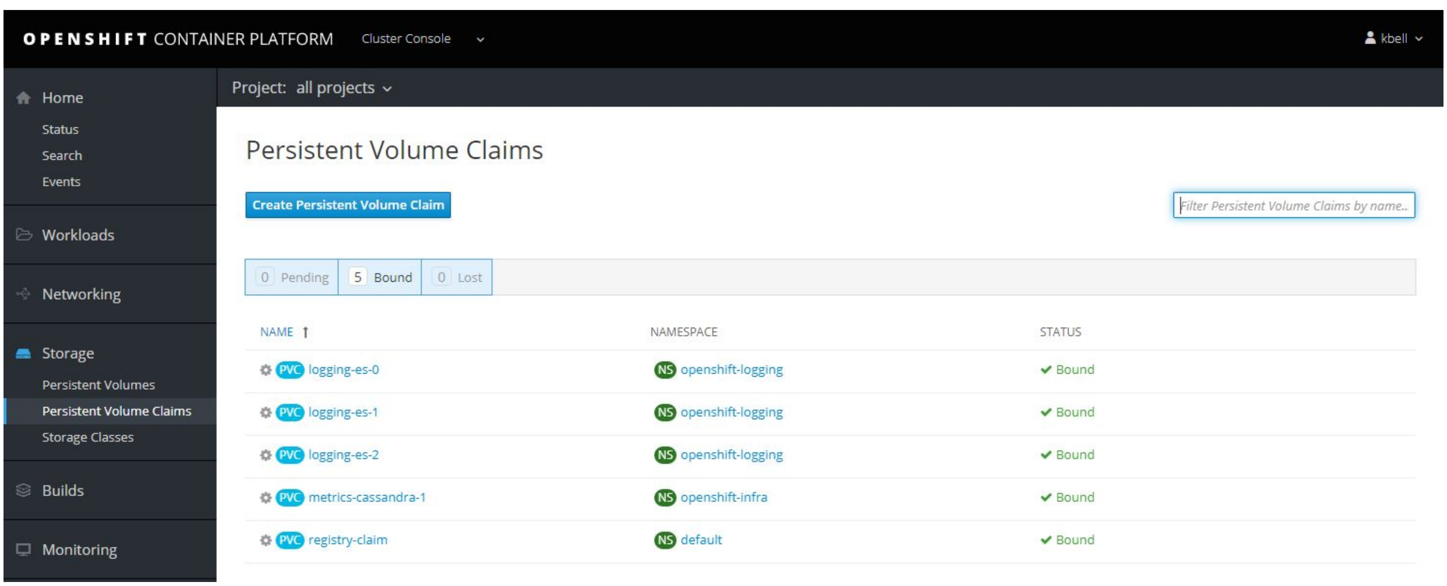


Figure 25: Persistent Volume Claims

The persistent volume claim yaml file should be similar to the example shown below:


```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  annotations:
    control-plane.alpha.kubernetes.io/leader: >-
      {"holderIdentity":"fe6f571c-aeea-11e9-8996-
0a580a820204","leaseDurationSeconds":15,"acquireTime":"2019-07-25T15:11:31Z","renewTime":"2019-07-
25T15:11:35Z","leaderTransitions":0}
    kubectl.kubernetes.io/last-applied-configuration: >

{"apiVersion":"v1","kind":"PersistentVolumeClaim","metadata":{"annotations":{},"labels":{"metrics-
infra":"hawkular-cassandra"},"name":"metrics-cassandra-1","namespace":"openshift-
infra"},"spec":{"accessModes":["ReadWriteOnce"],"resources":{"requests":{"storage":"20Gi"}}}}
    pv.kubernetes.io/bind-completed: 'yes'
    pv.kubernetes.io/bound-by-controller: 'yes'
    volume.beta.kubernetes.io/storage-provisioner: gluster.org/glusterblock
  selfLink: >-
    /api/v1/namespaces/openshift-infra/persistentvolumeclaims/metrics-cassandra-1
  resourceVersion: '12580'
  name: metrics-cassandra-1
  uid: 7ba58282-aeee-11e9-8981-20677ce3a32c
  creationTimestamp: '2019-07-25T15:11:31Z'
  namespace: openshift-infra
  finalizers:
    - kubernetes.io/pvc-protection
  labels:
    metrics-infra: hawkular-cassandra
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
  volumeName: pvc-7ba58282-aeee-11e9-8981-20677ce3a32c
  storageClassName: glusterfs-storage-block
status:
  phase: Bound
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 20Gi
```

Sample application deployment

In this section, a sample application will be deployed using persistent storage presented by Red Hat OpenShift Container Storage running on the application nodes. The sample application selected for this example is Jenkins with persistent storage. Figure 26 illustrates the default catalog that is installed with Red Hat OpenShift.

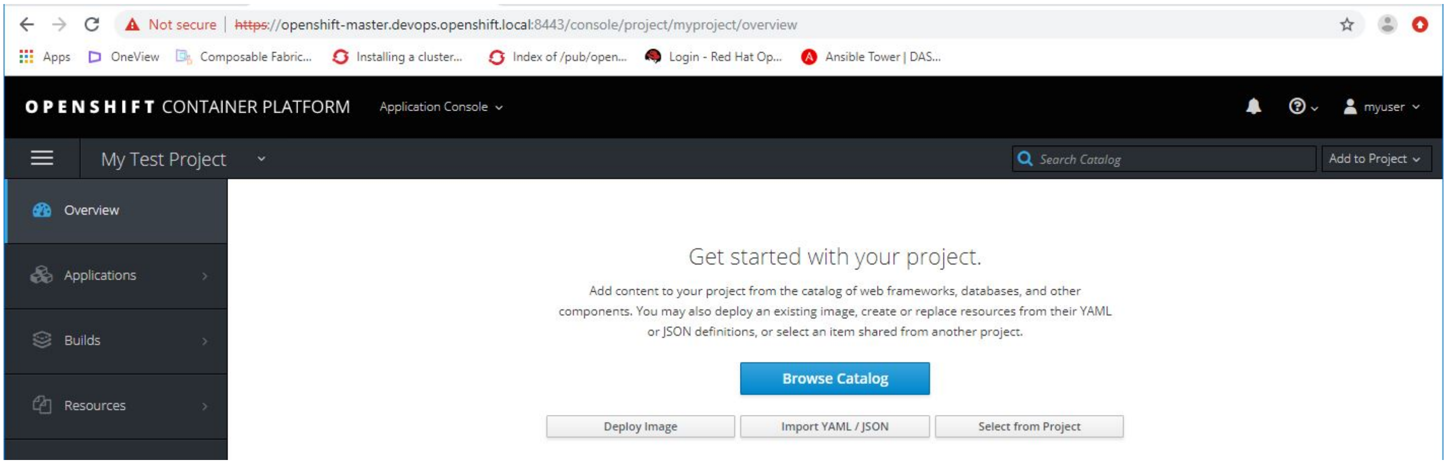


Figure 26: New Project

Under **My Projects,** notice **Development Project**, as show in Figure 27. When you deploy your first application in Red Hat OpenShift Container Platform, you will be prompted to create a project. Additional applications created by this user account will be created under that project.

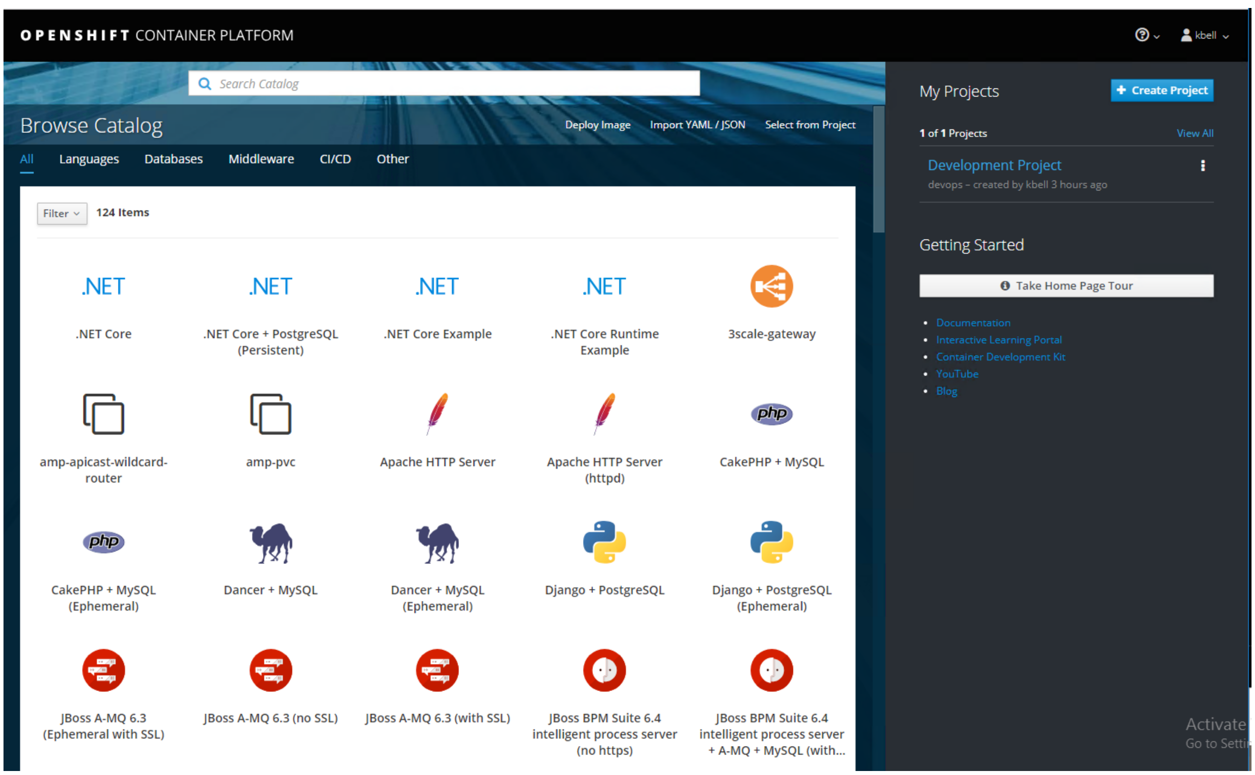


Figure 27: OpenShift Catalog

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

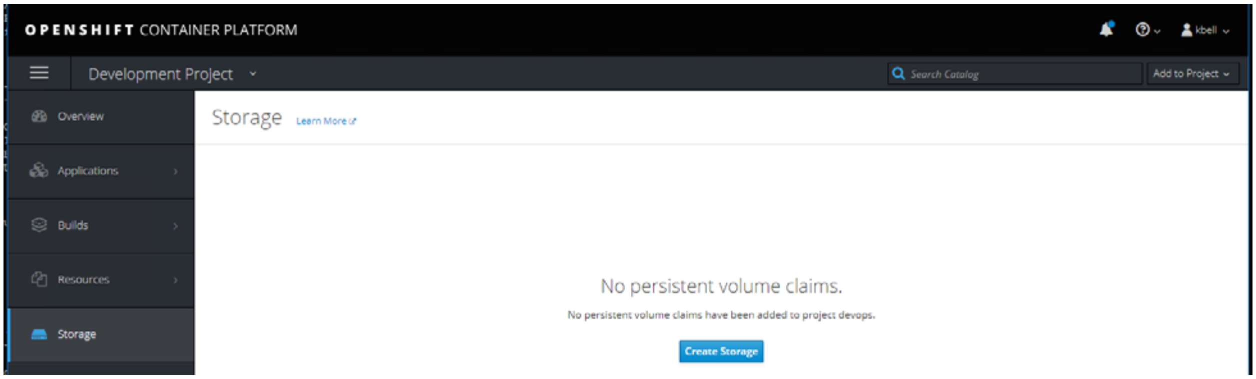


Figure 28: OpenShift Persistent Volumes

3. In the Create Storage screen, select the Storage Class glusterfs-storage. This is the storage class for applications and is being provided by the Red Hat OpenShift Container Storage that is running on the three application nodes (OCS1, OCS2, OCS3).

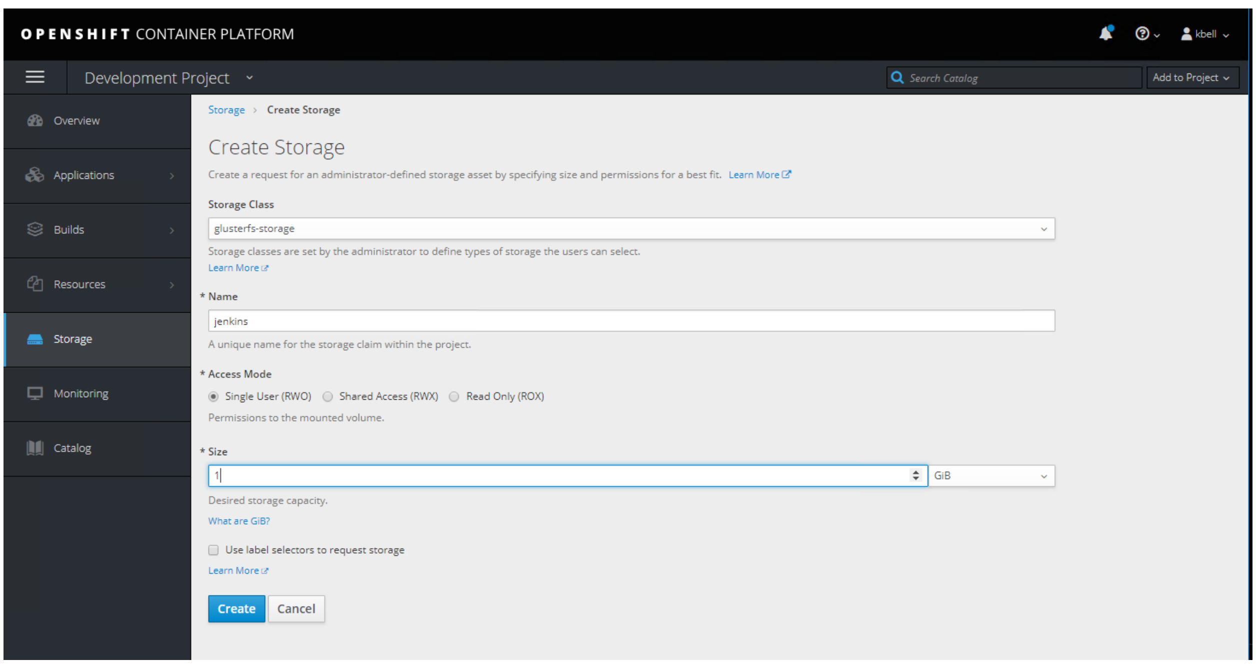


Figure 29: Red Hat OpenShift Create Persistent Volume for Jenkins

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

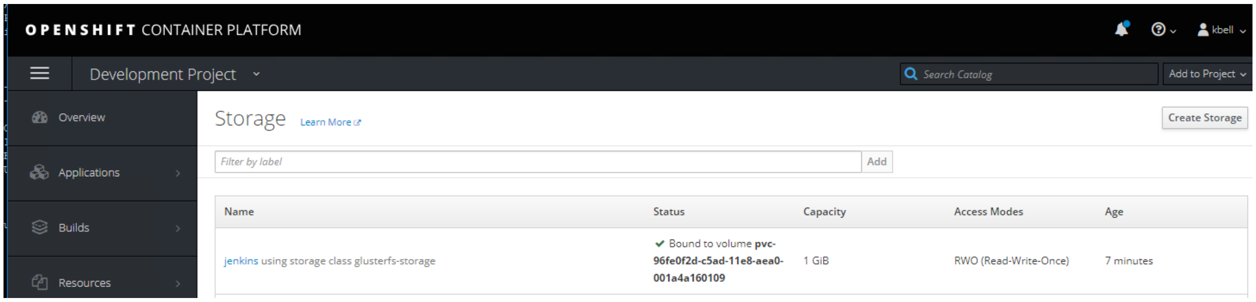


Figure 30: Red Hat OpenShift Persistent Volume Claim

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

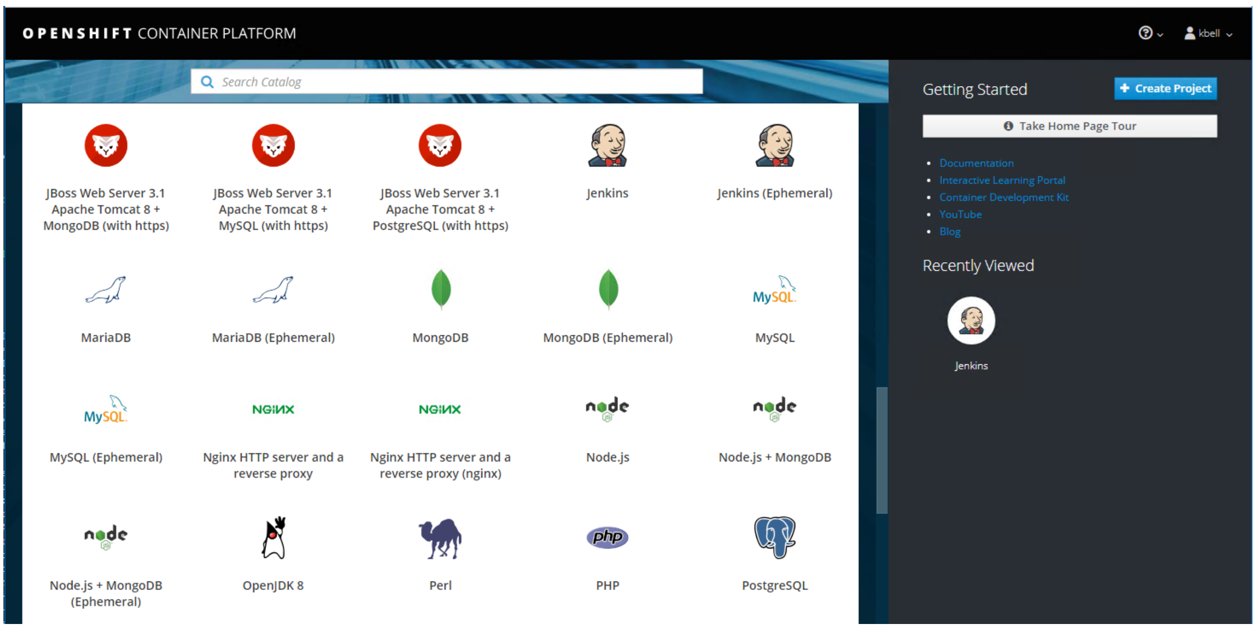


Figure 31: Red Hat OpenShift Catalog Jenkins

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

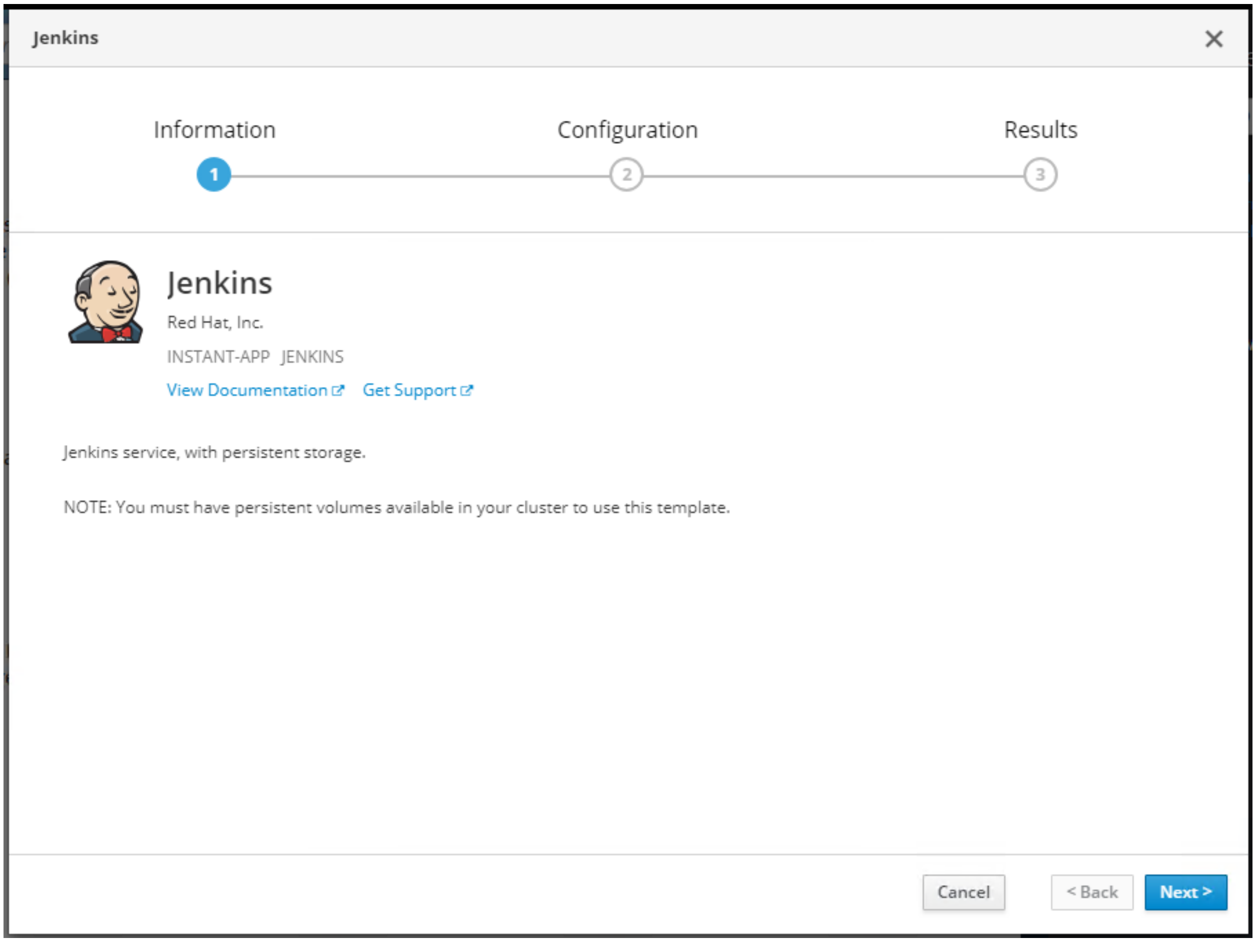


Figure 32: Red Hat OpenShift Jenkins deployment

7. Complete the Jenkins form, in this example the defaults were selected as shown in Figure 33.

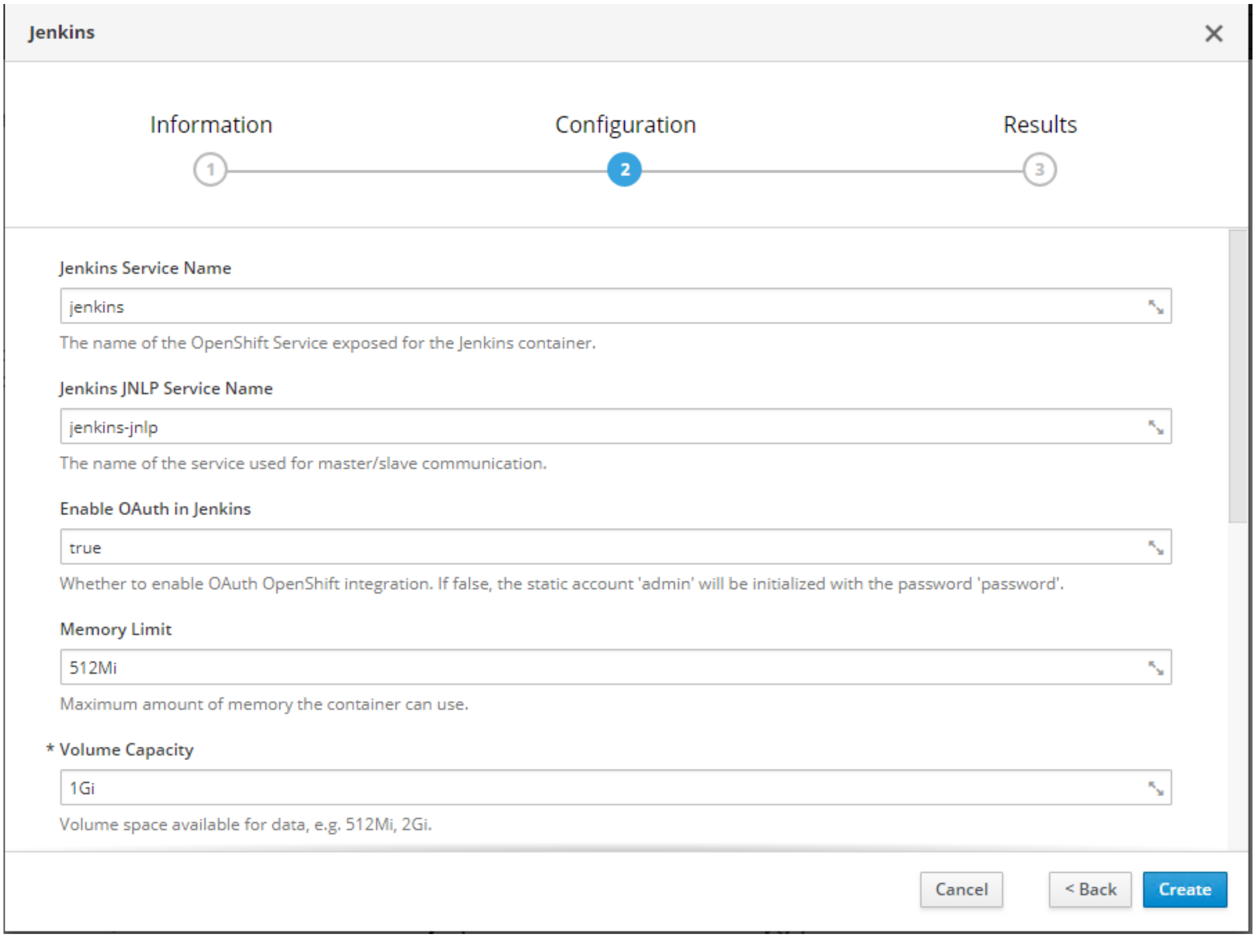


Figure 33: Red Hat OpenShift deployment Jenkins

8. Select **Create** to deploy the application.

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

9. Select **Applications > Deployments** to view the status of the application deployment.

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

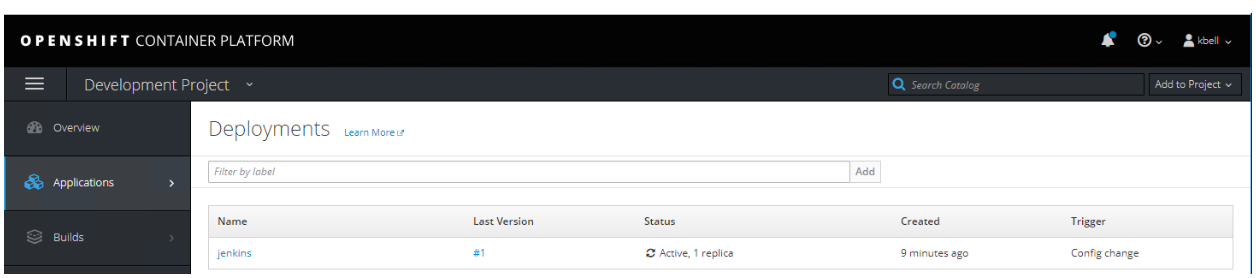


Figure 34: Red Hat OpenShift applications deployments

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

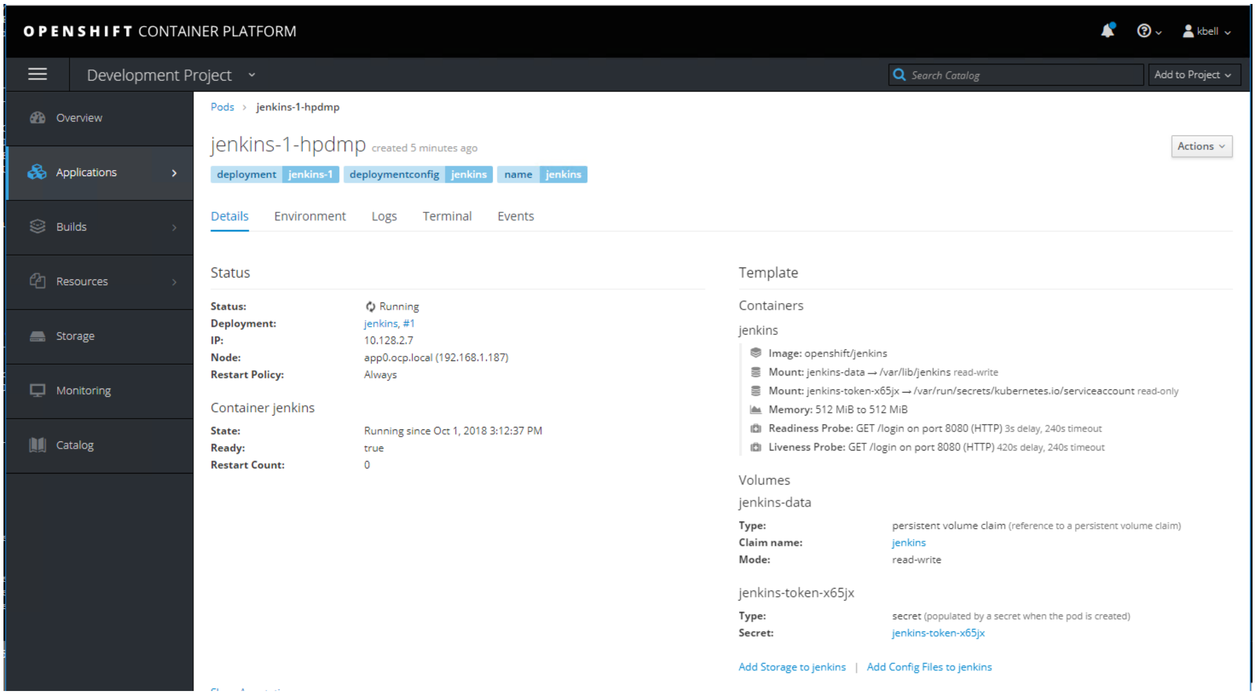


Figure 35: Red Hat OpenShift applications pods

11. Select **Services** to view the services that are associated with the pods.

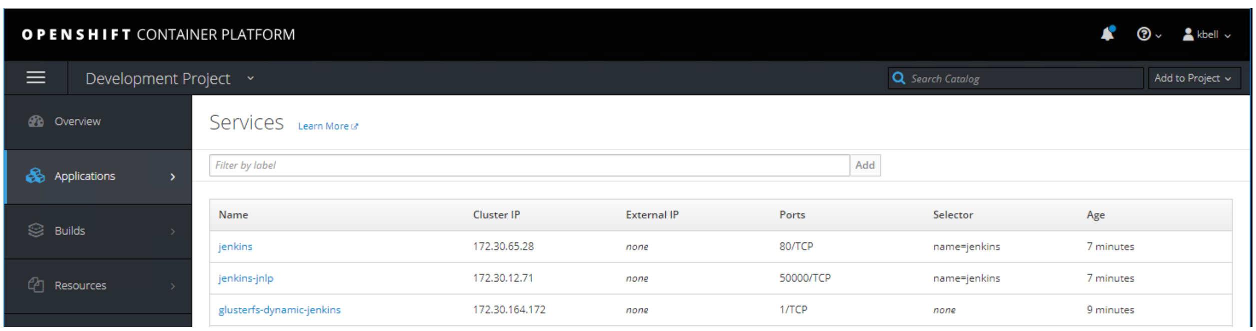


Figure 36: Red Hat OpenShift applications services

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

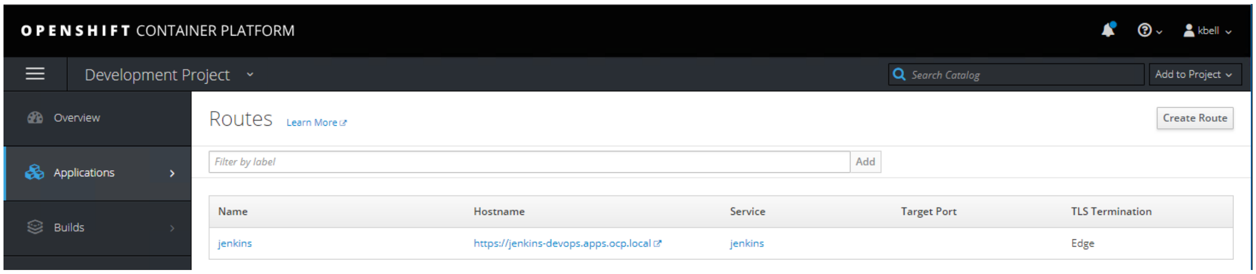


Figure 37: Red Hat OpenShift applications routes

13. Click on the URL to access Jenkins.

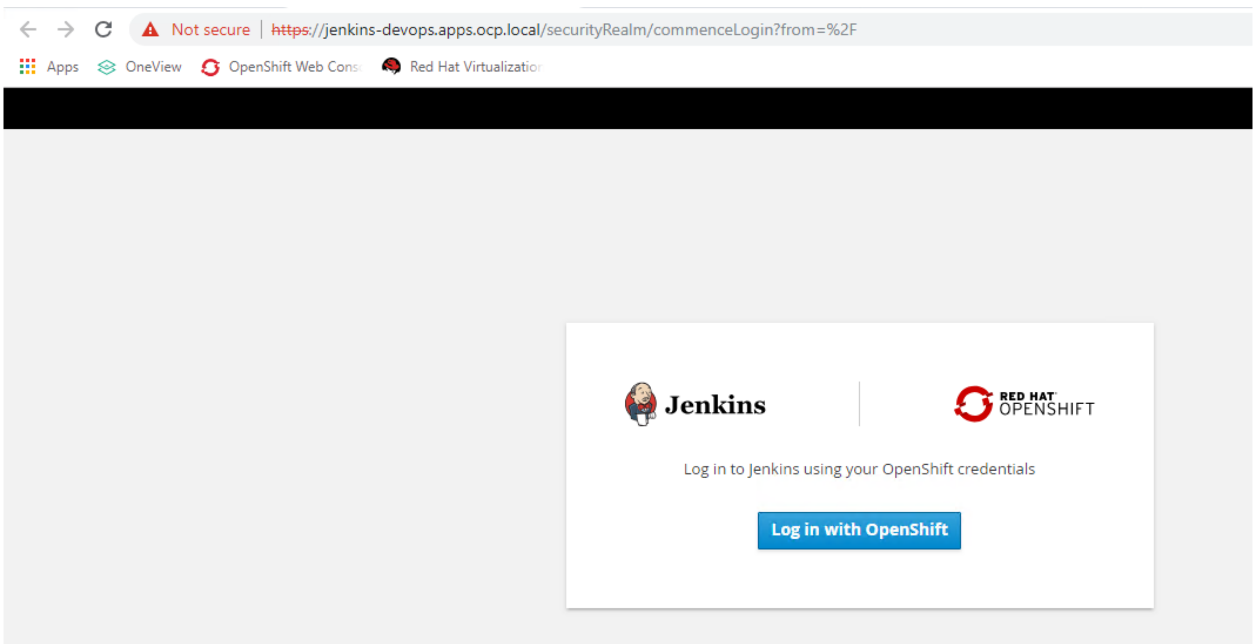


Figure 38: Jenkins Login

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

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

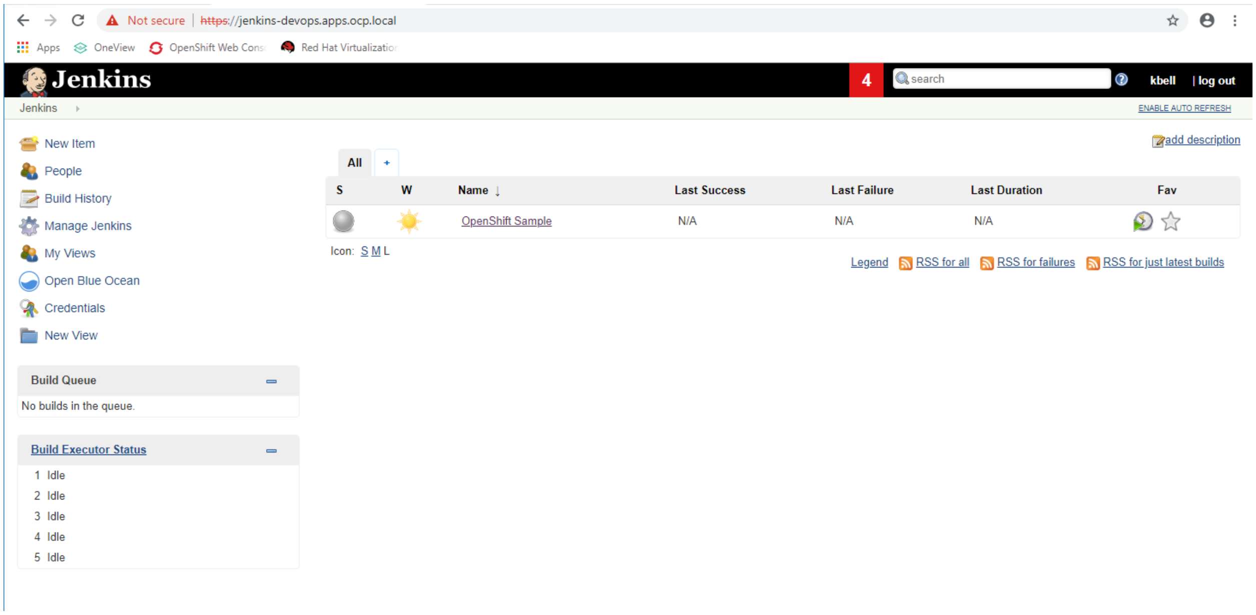


Figure 39: Jenkins user interface

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

- https://docs.openshift.com/container-platform/3.11/dev_guide/dev_tutorials/openshift_pipeline.html
- https://access.redhat.com/documentation/en-us/reference_architectures/2017/html-single/application_cicd_on_openshift_container_platform_with_jenkins/index

Appendix A: PXE configuration

Installing Red Hat Enterprise Linux using a PXE configuration

The RHEL servers in this solution are installed using an external server that provides PXE boot, HTTP, and DHCP services. This server hosts the RHEL installation distribution on a httpd web server. DHCP is configured to assign specific IP addresses to the RHEL server nodes using a DHCP reservation tied to the MAC address of the server. This is accomplished by setting a reservation of the Ethernet MAC address of the network interface that is configured on the PXE network in the DHCP configuration file. The MAC address can be found in the HPE OneView server profile for the respective RHEL server. This section provides example configuration settings for creating a network boot server. For detailed setup and configuration information, refer to the Red Hat documentation at https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/installation_guide/chap-installation-server-setup.

PXE server environment

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

```
default menu.c32
prompt 0
timeout 300
ONTIMEOUT local
Menu title ##############################PXE Boot MENU#################################
LABEL local
 MENU LABEL Boot local hard drive
 LOCALBOOT 0

LABEL RHEL76
        MENU Label Install RHEL76
        KERNEL images/rhel76/vmlinuz
        append initrd=initrd-images/rhel76/initrd.img method=http://172.22.0.1/rhel7-install/
inst.ks=http://172.22.0.1/rhel7-install/rhel7.ks devfs=nomount
```

4. Mount the RHEL 7.6 distribution image and copy the boot images. Attach the RHEL 7 iso to the server using iLO vMedia, then mount the image using the commands below.
   a. mount -o loop /dev/cdrom /mnt:
   b. cd /mnt/
   c. cp images/pxeboot/vmlinuz /var/lib/tftpboot/rhel76
   d. cp images/pxeboot/initrd.img /var/lib/tftpboot/rhel76

5. Configure the firewall to pass TFTP and HTTP:
   a. firewall-cmd --add-service=tftp --permanent
   b. firewall-cmd --add-service=http --permanent
6. Configure HTTPD:
   a. mkdir /var/www/html/rhel7-install
   b. cp -rf /mnt/* /var/www/html/rhel7-install
7. Edit the kickstart file:
   a. vi /var/www/html/ `rhel7-install` /rhel7.ks

```
#version=DEVEL
# Use network installation
url --url="http://172.22.0.1/ rhel7-install /"
repo --name="Server-HighAvailability" --baseurl=http://172.22.0.1/ rhel7-install
//addons/HighAvailability
repo --name="Server-ResilientStorage" --baseurl=http://172.22.0.1/ rhel7-install
//addons/ResilientStorage
# Use text mode install
text
ignoredisk --only-use=sda
# Keyboard layouts
keyboard --vckeymap=us --xlayouts=''
# System language
lang en_US.UTF-8

# Network information
network  --bootproto=dhcp --device=eno1 --onboot=off --ipv6=auto --no-activate
network  --bootproto=dhcp --device=eno5 --ipv6=auto --activate
network  --bootproto=static --device=eno6d1 --onboot=off --ipv6=auto --no-activate
network  --bootproto=dhcp --device=eno2 --onboot=off --ipv6=auto --no-activate
network  --bootproto=dhcp --device=eno3 --onboot=off --ipv6=auto --no-activate
network  --bootproto=dhcp --device=eno4 --onboot=off --ipv6=auto --no-activate
#uncomment network setting for bond0 when using for ilo vMedia UEFI install
#network --device=bond0 --bondslaves=eno5,eno6d1 --bondopts=mode=802.3ad --bootproto=dhcp -activate
# Reboot after installation
reboot
# Root password
rootpw --plaintext changeme
# System services
services --enabled="chronyd"
# Do not configure the X Window System
skipx
# System timezone
timezone America/Chicago --isUtc
# System bootloader configuration
bootloader --append=" crashkernel=auto" --location=mbr --boot-drive=sda
autopart --type=plain
# Clear the Master Boot Record
zerombr
# Partition clearing information
clearpart --linux --initlabel --drives=sda
```

```
%packages
@core
chrony
kexec-tools
%end
%addon com_redhat_kdump --enable --reserve-mb='auto'
%end
```

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

```
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.example
#   see dhcpd.conf(5) man page
#
option domain-name "devops.openshift.local";
option domain-name-servers 172.22.0.1;

Allow booting;
Allow bootp;
default-lease-time 600;
max-lease-time 7200;
subnet 172.22.0.0 netmask 255.255.255.0 {
range dynamic-bootp 172.22.0.111 172.22.0.150;
filename "pxelinux.0";
option broadcast-address 172.22.0.255;
option domain-name-servers 172.22.0.1;
option domain-search "devops.openshift.local";
option routers 172.22.0.254;
}
host master1 {
hardware ethernet F4:03:43:C0:6B:C0;
fixed-address 172.22.0.101;
option host-name "master1.devops.openshift.local";
}
host master2 {
hardware ethernet F4:03:43:C0:7B:B0;
fixed-address 172.22.0.102;
option host-name "master2.devops.openshift.local";
}
host master3 {
hardware ethernet F4:03:43:C0:84:E0;
fixed-address 172.22.0.103;
option host-name "master3.devops.openshift.local";
}
host infra1 {
hardware ethernet F4:03:43:C0:E3:50;
fixed-address 172.22.0.104;
option host-name "infra1.devops.openshift.local";
}

host infra2 {
hardware ethernet F4:03:43:C0:6B:60;
fixed-address 172.22.0.105;
```

```
option host-name "infra2.devops.openshift.local";
}

host infra3 {
hardware ethernet F4:03:43:C0:7C:E0;
fixed-address 172.22.0.106;
option host-name "infra3.devops.openshift.local";
}

host ocs1 {
hardware ethernet F4:03:43:C0:7C:20;
fixed-address 172.22.0.107;
option host-name "ocs1.devops.openshift.local";
}
host ocs2 {
hardware ethernet F4:03:43:C0:6B:B0;
fixed-address 172.22.0.108;
option host-name "ocs2.devops.openshift.local";
}
host ocs3 {
hardware ethernet F4:03:43:C0:7C:F0;
fixed-address 172.22.0.109;
option host-name "ocs3.devops.openshift.local";
}
```

b. systemctl enable dhcpd

c. systemctl start dhcpd

Appendix B: Configuring the Ansible host

Configuring the Ansible host

1. Register the Ansible host:

```
subscription-manager register --username=<user_name> --password=<password>
```

2. Attach the subscription pool ID:

```
subscription-manager attach --pool=<pool_id>
```

3. Enable the required repositories:

```
subscription-manager repos subscription-manager repos \
```

```
▫ rhel-7-server-ansible-2.6-rpms
▫ rhel-7-server-extras-rpms
▫ rhel-7-server-rpms
▫ rhel-7-server-ose-3.11-rpms
```

4. Install Ansible:

```
yum install ansible
```

5. Install OpenShift Ansible:

```
yum install openshift-ansible
```

Create the Ansible vault file

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

```
ansible_user: <username>
ansible_ssh_user: <username>
ansible_ssh_pass: <password>
vault_rhsub_user: <rhsm_user>
vault_rhsub_pass: <rhsm_password>
vault_rhsub_pool: <rhsm_subscription_poolID
vault_rhsub_server:
oneview_auth.username: <oneview username>
oneview_pw: <oneview password>
oneview_auth.ip: <oneview IP>
oneview_auth.api: <oneview api version>
ilo_username: <ilo username>
ilo_pw: <ilo password>
```

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

```
#ansible-vault edit /etc/ansible/roles/password/vars/passwords.yml
```

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

Ansible configuration file

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

Ansible inventory (hosts) file

A sample inventory file is provided in the GitHub repository for this solution. Edit this file to reflect settings specific to your environment.

Appendix C: DNS configuration

Sample DNS Zone File

```
$ORIGIN devops.openshift.local
$TTL 300
$ORIGIN .
$TTL 10800 ; 3 hours
devops.openshift.local       IN SOA devops.openshift.local. admin.openshift.local. (
                                        580    ; serial
                                        10800  ; refresh (3 hours)
                                        3600   ; retry   (1 hour)
                                        604800 ; expire  (1 week)
                                        3500   ; minimum (1 hour)
                                    )
                        NS          infra.openshift.local.
                                    A     172.22.0.1
$ORIGIN devops.openshift.local.
lb                                  A      172.22.0.4
openshift-master        A      172.22.0.4
master1                             A      172.22.0.101
master2                             A     172.22.0.102
master3                              A    172.22.0.103
infra1                              A     172.22.0.104
infra2                              A     172.22.0.105
infra3                              A     172.22.0.106
```

```
ocs1                          A    172.22.0.107
ocs2                          A    172.22.0.108
ocs3                          A    172.22.0.109
$ORIGIN apps.devops.openshift.local.
*                              A    172.22.0.4
```

Appendix D: Ansible playbook timing

```
ansible-playbook -i hostsGFS playbooks/subs.yml --ask-vault-pass
Thursday 29 August 2019  07:46:09 -0500 (0:06:57.240)       0:14:09.242 *******
===============================================================================
../roles/subs : Perform an Update ------------------------------------------------------------------------------ 417.24s
../roles/subs : Install Basic Utils ---------------------------------------------------------------------------- 336.98s
../roles/subs : Register machines to RHSM ---------------------------------------------------------------------- 36.32s
../roles/subs : Disable all repos ------------------------------------------------------------------------------ 36.26s
../roles/subs : Enable Required repos -------------------------------------------------------------------------- 19.04s
Gathering Facts ------------------------------------------------------------------------------------------------ 2.95s
../roles/passwords : include_vars ------------------------------------------------------------------------------ 0.38s

ansible-playbook -i hostsGFS playbooks/bonds.yml --ask-vault-pass
Thursday 29 August 2019  08:02:23 -0500 (0:00:00.502)       0:02:12.721 *******
===============================================================================
../roles/bonds : restart network ------------------------------------------------------------------------------- 67.88s
../roles/bonds : wait for bonds -------------------------------------------------------------------------------- 60.73s
../roles/bonds : add bond-slave -------------------------------------------------------------------------------- 0.99s
../roles/bonds : add bond -------------------------------------------------------------------------------------- 0.80s
../roles/bonds : replace bootp eno5 ---------------------------------------------------------------------------- 0.72s
../roles/bonds : add vlan -------------------------------------------------------------------------------------- 0.50s
../roles/bonds : replace bootp bond0 --------------------------------------------------------------------------- 0.36s
../roles/bonds : replace PREFIX -------------------------------------------------------------------------------- 0.35s
../roles/passwords : include_vars ------------------------------------------------------------------------------ 0.33s


ansible-playbook -i hostsGFS playbooks/prep.yml --ask-vault-pass
Thursday 29 August 2019  08:07:58 -0500 (0:00:00.773)       0:04:29.390 *******
===============================================================================
../roles/prep : Install Basic Utils ---------------------------------------------------------------------------- 258.58s
../roles/prep : enable and start docker ------------------------------------------------------------------------ 5.93s
Gathering Facts ------------------------------------------------------------------------------------------------ 1.71s
../roles/prep : hostname --------------------------------------------------------------------------------------- 1.24s
../roles/prep : Set authorized key took from file -------------------------------------------------------------- 0.77s
../roles/prep : debug ------------------------------------------------------------------------------------------ 0.73s
../roles/passwords : include_vars ------------------------------------------------------------------------------ 0.36s


ansible-playbook -i hostsGFS -e@roles/passwords/vars/passwords.yml /usr/share/ansible/openshift-
ansible/playbooks/prerequisites.yml --ask-vault-pass
NSTALLER STATUS ******************************************************************************************
Initialization  : Complete (0:04:17)
Thursday 29 August 2019  08:22:09 -0500 (0:00:00.316)       0:12:13.730 *******
===============================================================================
container_runtime : Install Docker ----------------------------------------------------------------------------- 284.31s
Ensure openshift-ansible installer package deps are installed -------------------------------------------------- 204.67s
openshift_excluder : Install docker excluder - yum ------------------------------------------------------------- 82.36s
os_firewall : Install iptables packages ------------------------------------------------------------------------ 21.49s
openshift_repos : Ensure libselinux-python is installed -------------------------------------------------------- 16.48s
container_runtime : Create credentials for oreg_url ------------------------------------------------------------ 14.89s
os_firewall : need to pause here, otherwise the iptables service starting can sometimes cause ssh to fail -- 10.10s
```

```
os_firewall : Wait 10 seconds after disabling firewalld ------------------------------------------------ 10.08s
openshift_repos : refresh cache --------------------------------------------------------------------------- 6.87s
container_runtime : Get current installed Docker version ------------------------------------------------- 6.11s
os_firewall : Start and enable iptables service ------------------------------------------------------------ 3.88s
container_runtime : restart container runtime -------------------------------------------------------------- 3.16s
os_firewall : Ensure firewalld service is not enabled ----------------------------------------------------- 1.78s
container_runtime : Configure Docker service unit file ---------------------------------------------------- 1.29s
container_runtime : Place additional/blocked/insecure registries in /etc/containers/registries.conf -------- 1.07s
container_runtime : Fixup SELinux permissions for docker ------------------------------------------------- 0.93s
Gather Cluster facts ---------------------------------------------------------------------------------------- 0.89s
Detecting Operating System from ostree_booted ---------------------------------------------------------------- 0.85s
container_runtime : Configure Docker Network OPTIONS -------------------------------------------------------- 0.81s
Initialize openshift.node.sdn_mtu ---------------------------------------------------------------------------- 0.80s

ansible-playbook -i hostsGFS -e@roles/passwords/vars/passwords.yml /usr/share/ansible/openshift-
ansible/playbooks/deploy_cluster.yml --ask-vault-pass
Thursday 29 August 2019  09:33:19 -0500 (0:00:00.037)       1:08:55.397 *******
===============================================================================
cockpit : Install cockpit-ws -------------------------------------------------------------------------- 558.66s
openshift_node : install needed rpm(s) ---------------------------------------------------------------- 475.68s
openshift_node : Install node, clients, and conntrack packages ---------------------------------------- 322.84s
openshift_node : Install iSCSI storage plugin dependencies ------------------------------------------- 213.54s
openshift_storage_glusterfs : Wait for GlusterFS pods ------------------------------------------------ 104.21s
openshift_storage_glusterfs : Wait for GlusterFS pods ------------------------------------------------ 104.16s
openshift_node : Install NFS storage plugin dependencies --------------------------------------------- 98.25s
openshift_ca : Install the base package for admin tooling -------------------------------------------- 85.40s
openshift_excluder : Install openshift excluder - yum ------------------------------------------------ 82.51s
openshift_node : Install dnsmasq --------------------------------------------------------------------- 81.30s
openshift_node : Install Ceph storage plugin dependencies -------------------------------------------- 77.03s
openshift_control_plane : Wait for all control plane pods to become ready ---------------------------- 75.72s
openshift_cluster_monitoring_operator : Wait for the ServiceMonitor CRD to be created ---------------- 61.01s
openshift_service_catalog : Wait for Controller Manager rollout success ------------------------------ 56.36s
openshift_node : Install GlusterFS storage plugin dependencies --------------------------------------- 46.88s
template_service_broker : Verify that TSB is running ------------------------------------------------- 46.64s
openshift_control_plane : Wait for control plane pods to appear -------------------------------------- 45.78s
openshift_storage_glusterfs : Wait for deploy-heketi pod --------------------------------------------- 41.82s
openshift_storage_glusterfs : Wait for heketi pod ---------------------------------------------------- 41.79s
openshift_storage_glusterfs : Wait for deploy-heketi pod --------------------------------------------- 41.77s
```

Appendix E: Troubleshooting

Playbooks

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

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

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

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

Docker Logs

If the OpenShift control plane fails to start or a container is in a crash loop, review the docker logs for the container. From the console of a Master node, run the following commands:

- To list the running containers on a host, run: `docker ps`.
- To review the logs associated with a specific container, run: `docker logs <<container id>>`.

Routing

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

Reinstall

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

Resources and additional links

HPE Reference Architectures, hpe.com/info/ra

HPE Servers, hpe.com/servers

HPE Storage, hpe.com/storage

HPE Networking, hpe.com/networking

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

Red Hat OpenShift Container Platform, openshift.com

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

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

November 2018