



HPE REFERENCE ARCHITECTURE FOR ACCELERATED ARTIFICIAL INTELLIGENCE & MACHINE LEARNING ON HPE PROLIANT DL380 GEN10 AND HPE PROLIANT DL360 GEN10 SERVERS

Rapid deployment using Red Hat OpenShift Container Platform and
NVIDIA T4 GPUs

CONTENTS

Executive summary.....	3
Solution overview.....	3
Solution components.....	4
Hardware.....	4
Software.....	10
Solution Configuration.....	11
Deployment environment.....	11
Solution configuration.....	12
Server Configuration.....	12
Red Hat OpenShift Container Platform deployment.....	13
Step by Step OpenShift Deployment Process.....	13
RHEL CoreOS Install.....	14
Command Line Validation.....	16
OpenShift Web Consoles.....	17
NVIDIA Driver Deployment.....	18
Verify the GPU driver deployment.....	21
Testing the GPUs.....	22
NGC - Sample Application Deployment.....	23
Appendix A: Bill of Materials.....	24
Appendix B: PXE configuration.....	27
PXE server environment.....	27
Appendix C: DHCP Configuration.....	28
Appendix D: DNS Configuration.....	29
Appendix E: Example Load Balancer Configuration.....	30
Resources and additional links.....	32



EXECUTIVE SUMMARY

In today's data driven economy, to remain competitive, businesses must invest in artificial intelligence and machine learning tools and applications. Using AI, organizations are developing, and putting into production, process and industry applications that automatically learn, discover, and make recommendations or predictions that can be used to set strategic goals and provide a competitive advantage. To accomplish these strategic goals, organizations require data scientists and analysts that are skilled in developing models and performing analytics on enterprise data. These data analysts require specialized tools, applications, and compute resources to create complex models and analyze massive amounts of data. Deploying and managing these CPU intensive applications can place a burden on already taxed IT organizations. The Red Hat OpenShift Container Platform, running on HPE ProLiant DL servers and powered by NVIDIA® T4 GPUs, can provide an organization with a powerful, highly available, DevOps platform that allows these CPU intensive data analytics tools and applications to be rapidly deployed to end users through a self-service OpenShift registry and catalog. Additionally, OpenShift compute nodes equipped with NVIDIA T4 GPUs provide the compute resources to meet the performance requirements of these CPU intensive applications and queries. According to the IDC, IDC FutureScape: Worldwide Artificial Intelligence Predictions, document #US45576319 from www.IDC.com:

- By 2024, 75% of enterprises will invest in employee retraining and development, including third-party services, to address new skill needs and ways of working resulting from AI.
- By 2022, 75% of enterprises will embed intelligent automation into technology and process development, using AI-based software to discover operational and experiential insights to guide innovation.
- By 2024, AI will become the new user interface by redefining user experiences where over 50% of user touches will be augmented by computer vision, speech, natural language, and AR/VR.

This Reference Architecture describes how to:

- Setup HPE ProLiant DL380 Gen10 and HPE ProLiant DL360 Gen10 servers with NVIDIA T4 GPUs
- Install and configure Red Hat OpenShift Container Platform
- Validate the Red Hat OpenShift Container Platform installation
- Validate NVIDIA GPU module operation on Red Hat OpenShift Container Platform compute nodes

Target audience: This document is intended for Chief Information Officers (CIOs), Chief Technology Officers (CTOs), data center managers, enterprise architects, data analysts, and implementation personnel wishing to learn more about Red Hat OpenShift Container Platform on HPE ProLiant DL servers with NVIDIA T4 GPUs. Familiarity with HPE ProLiant DL servers, Red Hat OpenShift Container Platform, container-based solutions, and core networking knowledge is assumed.

Document purpose: The purpose of this document is to provide a Reference Architecture that describes the best practices and technical details for deploying a starter AI/ML platform with Red Hat OpenShift Container Platform on HPE ProLiant DL servers equipped with NVIDIA GPUs. This starter AI/ML DevOps platform can be used to automate and simplify the deployment and management of tools and server-based applications required by data scientists to perform complex data analysis for artificial intelligence and machine learning applications.

SOLUTION OVERVIEW

This Reference Architecture provides guidance for installing Red Hat OpenShift Container Platform (OCP) 4.1 on HPE ProLiant DL380 Gen10 and HPE ProLiant DL360 Gen10 servers equipped with NVIDIA T4 GPU processors. The solution consists of six (6) HPE ProLiant DL servers; three (3) HPE ProLiant DL360 Gen10 servers used for the OCP master nodes, two (2) HPE ProLiant DL380 Gen10 servers equipped with NVIDIA T4 GPU modules for the compute nodes, and one (1) HPE ProLiant DL360 Gen10 server that will be used as the OCP bootstrap node. The bootstrap node can be repurposed as an additional OCP compute node to be used for workloads that do not require GPU resources. The servers in this solution are interconnected with a 50 GB InfiniBand network.



Figure 1 shows the logical diagram.

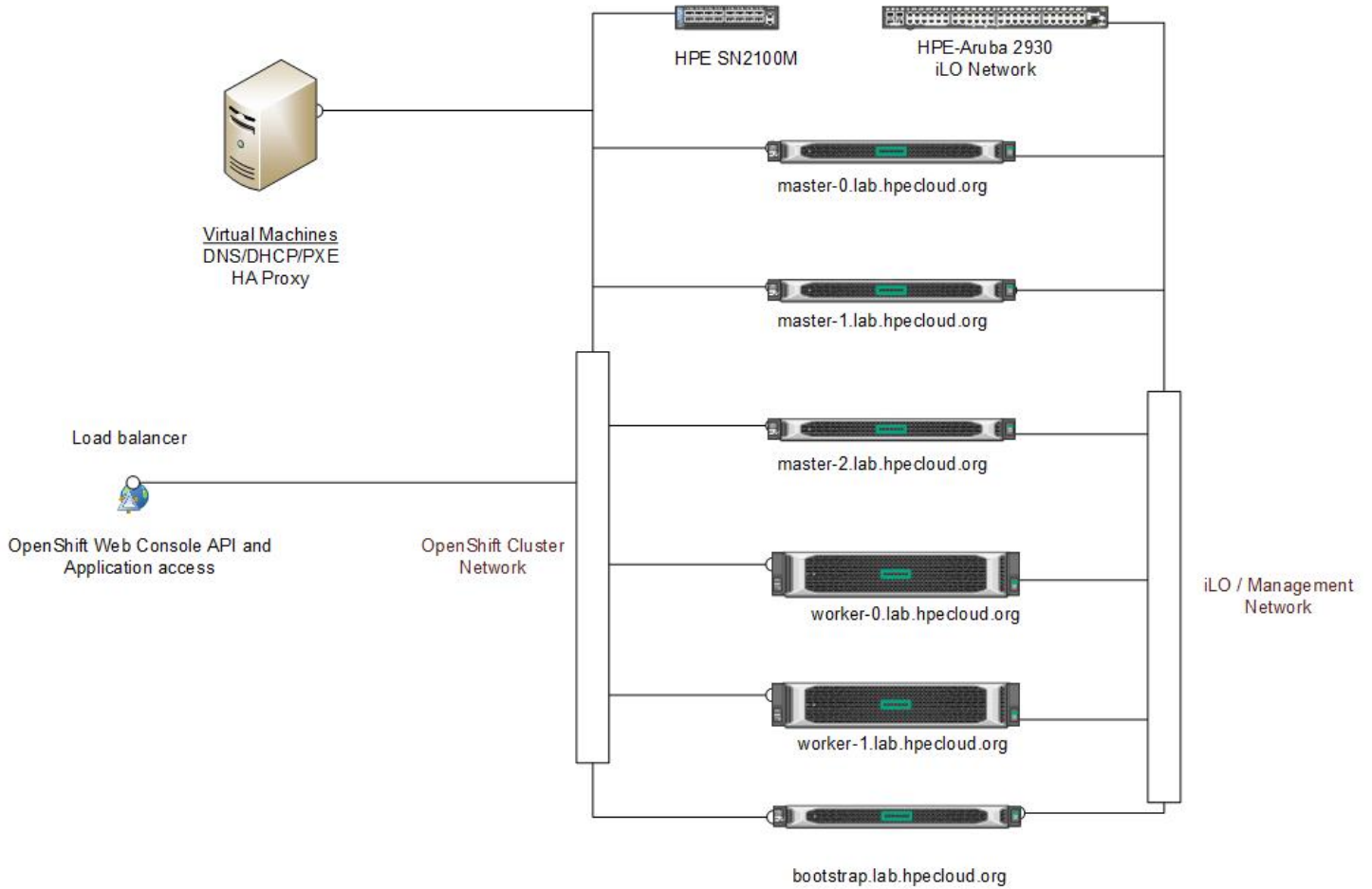


FIGURE 1. Logical diagram

SOLUTION COMPONENTS

Hardware

The configuration deployed for this solution is described in greater detail in this section. Table 1 lists the various hardware components in the solution.

TABLE 1. Components utilized in the creation of this solution

Component	Qty	Description
HPE ProLiant DL360 Gen10	4	OpenShift Master and Bootstrap/compute node
HPE ProLiant DL380 Gen10	2	OpenShift Worker Nodes
HPE SN2100M	1	Network Switch
Aruba 2930	1	Network Switch
NVIDIA T4 PCIe GPU Accelerator	8	GPU modules



Figure 2 illustrates the rack view of the HPE ProLiant DL servers deployed in this solution. Additional HPE ProLiant DL360 Gen10 or HPE ProLiant DL380 Gen10 or both servers can be added to the solution as needed.

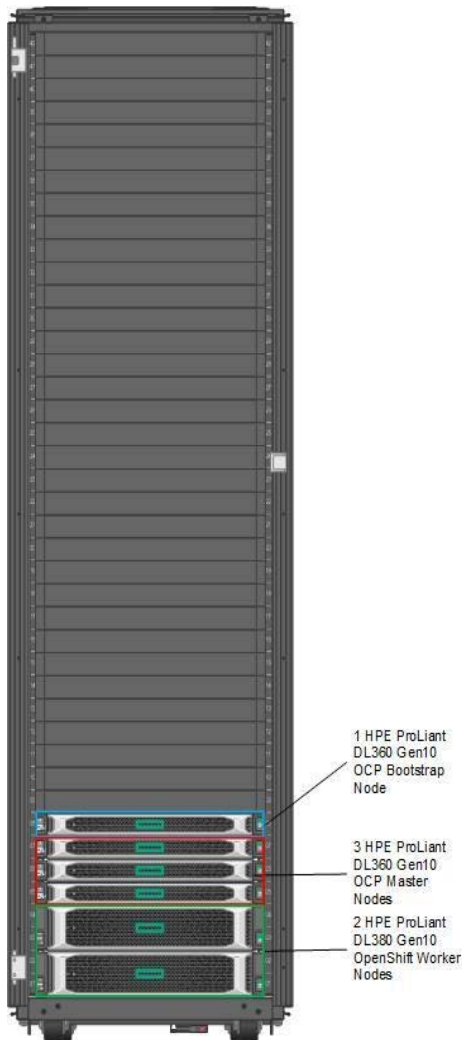


FIGURE 2. Solution rack view (front)

HPE ProLiant DL380 Gen10 server

The HPE ProLiant DL380 Gen10 server delivers the latest in security, performance, and expandability, backed by a comprehensive warranty. The server is designed securely to reduce costs and complexity, featuring the First and Second Generation Intel® Xeon® Processor Scalable Family with up to a 60% performance gain and 27% increase in cores.

The HPE ProLiant DL380 Gen10 server has an adaptable chassis, including new HPE modular drive bay configuration options with up to 30 Small Form Factor (SFF), up to 19 Large Form Factor (LFF), or up to 20 NVMe drive options along with support for up to three double-wide GPU options. Along with an embedded 4x1GbE, there is a choice of HPE FlexibleLOM or PCIe standup adapters, which offer a choice of networking bandwidth (1GbE to 40GbE) and fabric that adapt and grow to changing business needs. In addition, the HPE ProLiant DL380 Gen10 server comes with a complete set of HPE Technology Services, delivering confidence, reducing risk, and helping customers realize agility and stability.

This section describes the HPE ProLiant DL380 Gen10 servers used in the creation of this solution. Table 2 describes the individual components. Each server was equipped with 192 GB ram and dual Xeon 2.6GHz 12 Core CPUs. Individual server sizing should be based on customer needs and may not align with the configuration outlined in this document.



The HPE ProLiant DL380 Gen10 servers used in this Reference Architecture provide a robust platform to run containerized applications. The two HPE ProLiant DL380 Gen10 servers in this solution are deployed as OpenShift Compute nodes and configured with 4 NVIDIA T4 GPU Accelerators.

Figure 3 depicts the HPE ProLiant DL380 server configuration.



FIGURE 3. HPE ProLiant DL380 Gen10 server

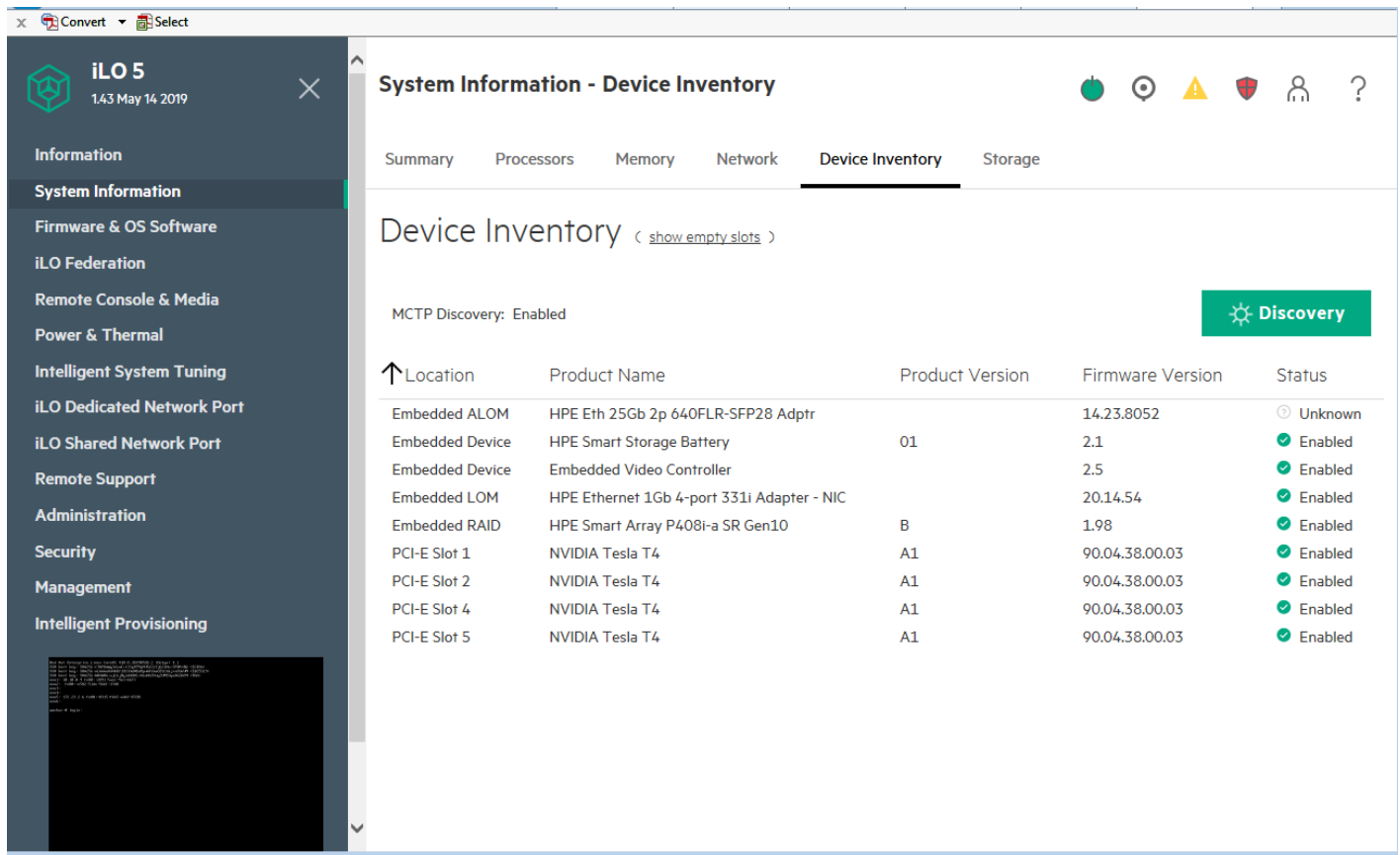
Table 2 lists the hardware components installed in the HPE ProLiant DL380 Gen10 servers.

TABLE 2. HPE ProLiant DL380 server configuration

Component	Description
Processor	2 x Intel Xeon-Gold 6126 (2.6 GHz/12-core/120 W)
Processor	2 x Intel Xeon-Gold 6126 (2.6 GHz/12-core/120 W)
Memory	12 x HPE 16GB 2Rx8 PC4-2666V-R Smart Kit
Network	HPE IB FDR/EN 40/50Gb 547FLR 2QSFP Adptr
Array Controller	HPE Smart Array E208i-a SR Gen10 Ctrlr
Disks	HPE 960GB SATA MU SFF SC DS SSD
GPUs	4 x NVIDIA T4 PCIe Accelerators



Figure 4 shows the iLO device inventory for the HPE ProLiant DL380 Gen10 servers. The NVIDIA T4 GPUs are installed in PCI slots 1, 2, 4, and 5.



The screenshot displays the iLO 5 System Information - Device Inventory page. The left sidebar shows navigation options like Information, System Information, Firmware & OS Software, etc. The main content area shows the Device Inventory table with the following data:

Location	Product Name	Product Version	Firmware Version	Status
Embedded ALOM	HPE Eth 25Gb 2p 640FLR-SFP28 Adptr		14.23.8052	Unknown
Embedded Device	HPE Smart Storage Battery	01	2.1	Enabled
Embedded Device	Embedded Video Controller		2.5	Enabled
Embedded LOM	HPE Ethernet 1Gb 4-port 331i Adapter - NIC		20.14.54	Enabled
Embedded RAID	HPE Smart Array P408i-a SR Gen10	B	1.98	Enabled
PCI-E Slot 1	NVIDIA Tesla T4	A1	90.04.38.00.03	Enabled
PCI-E Slot 2	NVIDIA Tesla T4	A1	90.04.38.00.03	Enabled
PCI-E Slot 4	NVIDIA Tesla T4	A1	90.04.38.00.03	Enabled
PCI-E Slot 5	NVIDIA Tesla T4	A1	90.04.38.00.03	Enabled

FIGURE 4. HPE ProLiant DL380 device inventory

HPE iLO

HPE Integrated Lights Out (iLO) is embedded in HPE ProLiant platforms and provides server management that enables faster deployment, simplified lifecycle operations, while maintaining end-to-end security thus increasing productivity.

HPE ProLiant DL360 Gen10 servers

[HPE ProLiant DL360 Gen10 server](#) is a secure, performance driven dense server that can be confidently deployed for virtualization, database, or high-performance computing. The HPE ProLiant DL360 Gen10 server delivers security, agility, and flexibility without compromise.

This section describes the HPE ProLiant DL360 Gen10 servers used in the creation of this solution. Each server was equipped with 32GB ram and dual Xeon 2.3 GHz 12 Core CPUs. Individual server sizing should be based on customer needs and may not align with the configuration outlined in this document.

The HPE ProLiant DL360 Gen10 servers in this Reference Architecture provide the OpenShift control plane, the OpenShift master and bootstrap nodes. The OpenShift master nodes are responsible for the OpenShift cluster health, scheduling, API access, and authentication. The etcd cluster runs on the OpenShift master nodes. The bootstrap node provides resources used by the master nodes to create the control plane for the OpenShift cluster. The bootstrap node is a temporary role that is only used during the initial OpenShift cluster installation. After the OpenShift cluster bootstrap process is complete the bootstrap node can be removed from the cluster and repurposed as an OpenShift compute node. In this Reference Architecture the bootstrap node is not equipped with NVIDIA T4 GPUs. This compute node can be used to deploy containerized applications that do not require GPU resources.



Table 3 lists the components installed in the HPE ProLiant DL360 servers.

TABLE 3. HPE ProLiant DL360 server configuration

Component	Description
Processors	2 x Intel Xeon-Gold 5118 (2.3 GHz/12-core/105 W)
Memory	4 x HPE 8GB 1Rx8 PC4-2666V-R Smart Kit
Network	HPE IB FDR/EN 40/50Gb 54-7FLR 2QSFP Adptr
Array Controller	HPE Smart Array P408i-a SR Gen10 Ctrlr
Disks	HPE 960GB SATA MU SFF SC DS SSD

HPE SN2100M Network Switch

HPE M-series SN2100M Ethernet switches are ideal for modern server and storage networks. Supporting port speeds of 1, 10, 25, 40, 50, and 100 GbE, delivering predictable performance and zero packet loss at line rate across each port and packet size. Enhanced for storage combined with efficient design, it provides enterprise-level performance with attractive economics and outstanding ROI. Networks built on HPE SN2100M are fast, reliable, and scalable while also being affordable and easy to manage. It supports primary and secondary storage, providing consistently fair, fast, low-latency connectivity even under heavy workloads or a mix of different port speeds. This makes them ideal for storage, hyperconverged, financial services, and media and entertainment deployments.



FIGURE 5. HPE SN2100M Switch

Aruba 2930

The Aruba 2930F Switch Series is designed for customers creating smart digital workplaces that are optimized for mobile users with an integrated wired and wireless approach. These convenient Layer 3 network switches include built-in uplinks and PoE power and are simple to deploy and manage with advanced security and network management tools like Aruba ClearPass Policy Manager, Aruba AirWave and cloud-based Aruba Central.

A powerful Aruba ProVision ASIC delivers performance, robust feature support and value with programmability for the latest applications. Stacking with Virtual Switching Framework (VSF) provides simplicity and scalability. The 2930F supports built-in 1GbE or 10GbE uplinks, PoE+, Access OSPF routing, Dynamic Segmentation, robust QoS, RIP routing, and IPv6 with no software licensing required.

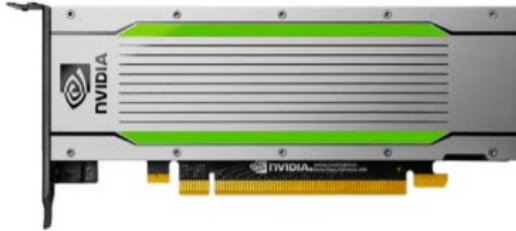


FIGURE 6. Aruba 2930F Switch



NVIDIA T4

The NVIDIA T4 GPU is based on the latest NVIDIA Turing architecture which provides support for virtualized workloads with NVIDIA virtual GPU (vGPU) software. It is a single wide card with passive cooling and offers good performance while consuming less power. The NVIDIA Turing architecture includes RT cores for real-time ray tracing acceleration, and batch rendering. It also supports GDDR6 memory which provides improved power efficiency and performance over the previous generation GDDR5 memory. With the help of Turing architecture, T4 is capable of offering the same breakthrough performance and versatility to Virtual Machines (VMs) as achieved on non-virtualized systems. This performance is achieved when T4 is used with NVIDIA vGPU software. Users can achieve a native-PC like experience in a virtualized environment when T4 is combined with NVIDIA vGPU software. The T4 is well suited for various data center workloads including virtual desktops using modern productivity applications, and virtual workstations for data scientists.¹

**FIGURE 7.** NVIDIA T4 GPU

¹ <https://www.NVIDIA.com/en-in/data-center/-t4/>



Network Overview

Each server in this solution is equipped with an HPE IB FDR/EN 40/50Gb InfiniBand adapter that is connected to an HPE StoreFabric M-series SN2100M 100Gbe Ethernet switch. The HPE ProLiant DL servers use iLO management interfaces which connect to a 1Gbe Ethernet network.

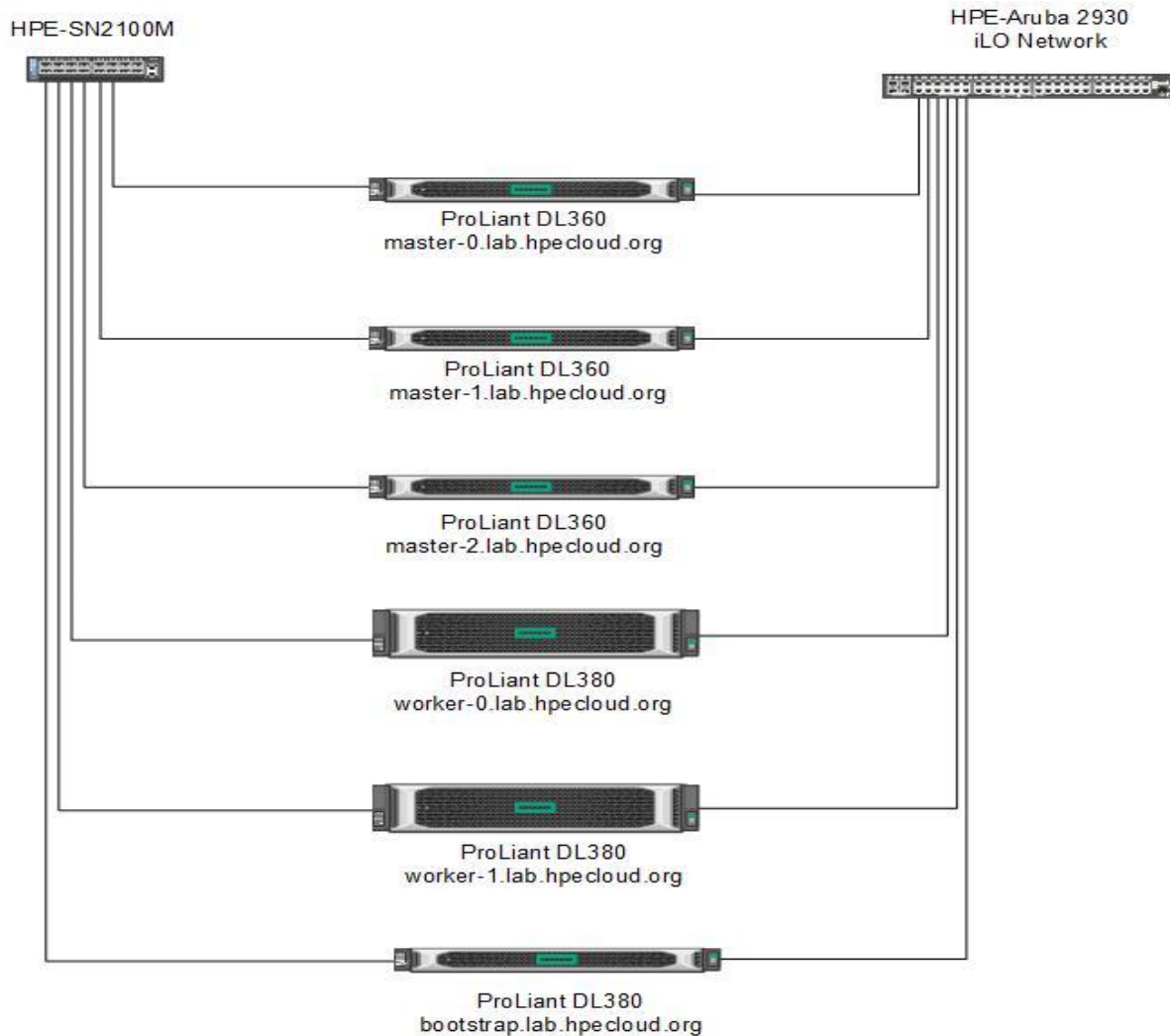


FIGURE 8. Solution Network Diagram

Software

Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform® is Red Hat's enterprise grade Kubernetes distribution that provides enterprises the ability to build, deploy, and manage container-based applications. Red Hat OpenShift Container Platform provides enterprises with a full featured Kubernetes environment that includes automated operations, cluster services, developer services, and application services to build Platform as a Service (PaaS) and Containers as a Service (CaaS) on-premises hybrid cloud solution. Red Hat OpenShift Container Platform provides integrated logging and metrics, authentication and scheduling, high availability, automated over the air updates, and an integrated application container registry.

Red Hat Enterprise Linux CoreOS

OpenShift Container Platform uses Red Hat Enterprise Linux CoreOS (RHCOS), a new container-oriented operating system that combines some of the best features and functions of the CoreOS and Red Hat Atomic Host operating systems. RHCOS is specifically designed for running



containerized applications from OpenShift Container Platform and works with new tools to provide fast installation, Operator-based management, and simplified upgrades².

RHCOS includes:

- Ignition, which is a first boot system configuration for initially bringing up and configuring OpenShift Container Platform nodes.
- CRI-O, a Kubernetes native container runtime implementation that integrates closely with the operating system to deliver an efficient and optimized Kubernetes experience.
- Kubelet, the primary node agent for Kubernetes that is responsible for launching and monitoring containers.

In OpenShift Container Platform 4.1, you must use RHCOS for all control plane machines, but you can use Red Hat Enterprise Linux (RHEL) as the operating system for compute (worker) machines.

Table 4 lists the versions of Red Hat OpenShift Container Platform and Red Hat Enterprise Linux CoreOS used in the creation of this solution. The installer should insure they have downloaded or have access to this software.

TABLE 4. Software versions

Component	Version
Red Hat CoreOS	4.1
Red Hat OpenShift Container Platform	4.1

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 will be 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	Required to provide network boot capabilities to hosts that will install via a Kickstart file.
Load Balancer	Required	Provides load balancing across the OpenShift master and worker nodes
NTP	Required	Required to insure consistent time across the solution stack.

DNS

All nodes used for the Red Hat OpenShift Container platform deployment must be registered in DNS. A sample DNS zone file is provided in Appendix D of this document.

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 collected using the HPE iLO management interface before installation has begun to

² https://access.redhat.com/documentation/en-us/openshift_container_platform/4.1/pdf/architecture/OpenShift_Container_Platform-4.1-Architecture-en-US.pdf



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. A sample DHCP configuration file is provided in Appendix C of this document.

TFTP/PXE

The hosts in this configuration were deployed via a TFTP/PXE server to provide the initial network boot services. In order to successfully complete the necessary portions of the install you will need a host that is capable of providing HTTP, TFTP, and network boot services. In the solution environment, PXE services existed on 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 Appendix B of this document. It is assumed that TFTP and network boot services are being provided from a Linux-based host.

Load Balancer

A load balancer is required for the deployment. A sample HAProxy configuration file is provided in Appendix E of this document

NTP

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

NFS

An NFS share is required to provide persistent storage for the OpenShift Registry.

Installer laptop

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

Solution configuration

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.

Server Configuration

Installing the NVIDIA GPU modules

The HPE ProLiant DL380 Gen10 servers are configured with a PCI riser card to accommodate the NVIDIA GPU cards.

BIOS Settings

Set network boot order and network boot device to boot from the HPE InfiniBand adapters. Enable the HPE 547FDR adapter for network booting. Set the network boot order to boot from the HPE 547FDR before the HDD.

BIOS Settings

Set network boot order and network boot device to boot from the HPE InfiniBand adapters

1. During the boot process, press F9 to bring up the BIOS/Platform Configuration (RBSU).
2. BIOS/Platform Configuration (RBSU) -> (at the top) Workload Profile -> selected "High Performance Compute (HPC).
3. BIOS/Platform Configuration (RBSU) -> Network Options -> Network Boot Options -> Embedded FlexibleLOM 1 Port 1 (see example below) -> Network Boot -> Also select "Disabled" for the remaining network ports -> remember to Save (F10).
4. BIOS/Platform Configuration (RBSU) -> Boot Options -> Boot Mode "Legacy BIOS Mode" -> Legacy BIOS Boot Order -> Standard Boot Order (IPL) -> Bring the "Embedded FlexibleLOM 1 Port 1 : HPE Eth (10/25/50G) Adapter to the first one in the boot order. (see example below) -> Save
5. Reboot the system.

Hard Disk Configuration

Create mirror set on each of the HPE SmartArray adapters for the RHCOS image operating system deployment

1. During the boot process, press F10 to bring up the Intelligence Provisioning allow 15 seconds for the selection of Smart Storage Administrator (SSA).



2. Select the Smart Storage Administrator. Wait for the Smart Storage Administrator to present (may take up to several minutes depending on the storage configuration).
3. Select HPE Smart Array P408i-a SR -> Configure -> Create Array.
4. Select Physical Drives for the New Array > Create Array > Create Logical Drive > Finish.
5. Exit and reboot the server.

See the [HPE Smart Storage Administrator User Guide](#) for detailed instructions.

RED HAT OPENSIFT CONTAINER PLATFORM DEPLOYMENT

The OpenShift deployment is a multi-step process that involves using the openshift-installer program to create RHCOS ignition configuration files. The ignition files are used to configure RHCOS on each of the master and worker nodes in the OpenShift cluster. This document provides installation guidelines specific to this installation. For detailed installation and configuration information refer to the official OpenShift 4.1 Documentation, <https://docs.openshift.com/container-platform/4.1/welcome/index.html>.

The components required for the installation include the openshift-installer, install-config.yml, rhcos image, a pull secret, and an ssh key. The openshift-installer, pull secret, and RHCOS image can be obtained from the OpenShift Install on Bare Metal: User-Provisioned Infrastructure web page, <https://cloud.redhat.com/openshift/install/metal/user-provisioned>. The components and their usage are described in the following section.

Step by Step OpenShift Deployment Process

On the installer machine (laptop or virtual machine) configure an installation directory. Download and unzip the openshift-install utility on the installer machine. Create a directory for the installer. The directory name is not important; however, the structure can help if performing multiple installations. The openshift-install utility will overwrite the configuration files in the directory it is executed in. For that reason, we use a small shell script to copy the install-config.yaml to the installdir directory. This directory will be deleted and recreated each time the shell script is executed.

1. Create a working directory on the installer laptop or virtual machine.

```
#mkdir /root/ocp-install
#cd /root/ocp-install
```

2. Download the openshift-installer and openshift client utilities from <https://mirror.openshift.com/pub/openshift-v4/clients/ocp/latest/>.

```
#wget https://mirror.openshift.com/pub/openshift-v4/clients/ocp/latest/openshift-install-linux-4.1.0.tar.gz
#wget https://mirror.openshift.com/pub/openshift-v4/clients/ocp/latest/openshift-client-linux-4.1.0.tar.gz
```

3. Unzip the openshift-installer in a working directory (/root/ocp-install) on the installer laptop or virtual machine.

```
#tar -xvf openshift-install-linux-4.1.0.tar.gz
#tar -xvf openshift-client-linux-4.1.0.tar.gz
```

4. Create the install-config.yaml. Review the install-config.yaml file later in this document.

```
#vi install-config.yaml

apiVersion: v1
baseDomain: hpecloud.org
compute:
- name: worker
  replicas: 2
controlPlane:
  name: master
  replicas: 3
metadata:
  name: lab
platform:
```



```
none: {}
pullSecret: ''
sshKey: ''
```

5. Create an ssh key for the install user.

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

Copy and paste the contents of the public key (~/.ssh/id_rsa.pub) to the install-config.yaml file.

6. Obtain a pull secret from the OpenShift Install on Bare Metal: User-Provisioned Infrastructure web page.

Copy and paste the contents of the pull secret into the install-config.yaml file.

7. Create a shell script to launch the openshift-install utility.

```
#vi install.sh

#!/bin/bash
rm -rf installdir
mkdir installdir
cp install-config.yaml installdir/install-config.yaml
./openshift-install --dir=/root/installer/installdir create ignition-configs
```

RHEL CoreOS Install

1. Download to the RHEL CoreOS image from the [OpenShift Install on Bare Metal: User-Provisioned Infrastructure](#) web page.

```
#wget https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/4.1/latest/rhcos-4.1.0-x86_64-metal-bios.raw.gz
#wget https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/4.1/latest/rhcos-4.1.0-x86_64-installer-initramfs.img
#https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/4.1/latest/rhcos-4.1.0-x86_64-installer-kernel
```

2. Copy the initramfs image and the kernel files to the PXE server.

```
#scp rhcos-4.1.0-x86_64-installer-initramfs.img root@<pxe server IP address>:/var/lib/tftpboot/
#scp rhcos-4.1.0-x86_64-installer-kernel root@<pxe server IP address>:/var/lib/tftpboot/
```

3. Copy the RHEL CoreOS image to the HTTP server.

```
#scp /root/ocp-install/ rhcos-4.1.0-x86_64-metal-bios.raw.gz root@<http server IP address>:/var/www/html/rhcos/
```

4. Copy the ignition files to the HTTP server.

```
#scp /root/ocp-install/installdir/*.ign root@<http server IP address>:/var/www/html/rhcos/
```

5. PXE boot the master, worker, and bootstrap nodes.



Cluster formation

The master and worker nodes will display an error message until they connect to the bootstrap node. Once connected the master and worker nodes reboot and form the OpenShift cluster. Once the cluster is formed a message will be displayed informing the installer that the bootstrap process is complete and it is safe to remove the bootstrap resources.

1. Monitor the cluster formation. Connect to the bootstrap server using the following command:

```
#ssh core@bootstrap<domain name>
#journalctl -f -u bootkube.service
```

2. From the installer workstation:

```
#!/openshift-install --dir=./installdir wait-for bootstrap-complete --log-level debug
DEBUG OpenShift Installer v4.1.18-201909201915-dirty
DEBUG Built from commit 80c0ef5e57812daf721522db78972aa557730fc4
INFO Waiting up to 30m0s for the Kubernetes API at https://api.lab.hpecloud.org:6443...
DEBUG Still waiting for the Kubernetes API: the server could not find the requested resource
DEBUG Still waiting for the Kubernetes API: the server could not find the requested resource
DEBUG Still waiting for the Kubernetes API: Get https://api.lab.hpecloud.org:6443/version?timeout=32s:
EOF
INFO API v1.13.4+c2a5caf up
INFO Waiting up to 30m0s for bootstrapping to complete...
DEBUG Bootstrap status: complete
INFO It is now safe to remove the bootstrap resources
```

Completing the Cluster installation

1. Configure the OpenShift client (oc) to use the credentials saved in /root/ocp-install/installdir/auth/

```
#export KUBECONFIG=/root/ocp-install/installdir/auth/kubeconfig
#cp /root/ocp-install/installdir/auth/kubeconfig ~/.kube/config
#cp /root/ocp-install/oc /usr/bin
```

2. Approve the CSRs.

```
#oc get csr
#oc get csr -ojson | jq -r '.items[] | select(.status == {} ) | .metadata.name' | xargs oc adm certificate
approve
#oc adm certificate approve <csr_name>
```

3. Configure the OpenShift registry persistent storage.

- a. Create the persistent volume yaml file

```
#vi registryPV.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv001
spec:
  capacity:
    storage: 50Gi
  accessModes:
    - ReadWriteMany
  nfs:
    path: <nfs share>
    server: <nfs server IP Address>
  persistentVolumeReclaimPolicy: Retain
```



b. Create PV

```
#oc create -f registryPV.yaml
```

4. Monitor the cluster operator status until the operators are ready.

```
#watch -n5 oc get clusteroperators
Every 5.0s: oc get clusteroperators
```

5. Use the openshift installer to monitor the cluster for completion.

```
#!/openshift-install --dir=./installdir wait-for install-complete
INFO Waiting up to 30m0s for the cluster at https://api.lab.hpecloud.org:6443 to initialize...
INFO Waiting up to 10m0s for the openshift-console route to be created...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export KUBECONFIG=/root/OCP-
Installer/installdir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-console.apps.lab.hpecloud.org
INFO Login to the console with user: kubeadmin, password: igbMz-4yLgw-K3m3Q-bGngz
```

6. Log into the cluster using the openshift client and default kubeadmin account. The output from step 5 above provides the installer with the default username and password along with the url for accessing the OpenShift web console.

Decommission / Repurpose Bootstrap node

1. Shutdown the Bootstrap node.
2. Remove the bootstrap server from the load balancer configuration.

```
#vi /etc/haproxy/haproxy.cfg on the load balancer and remove the bootstrap entry
```

3. Optional – change the hostname in DNS and DHCP configuration files.
4. Reboot the Bootstrap node and select the worker node option from the PXE boot menu.
5. Get the csr.

```
#oc get csr
```

6. Approve the pending csr for the Bootstrap node.

```
#oc adm certificate approve <csr_name>Validate OpenShift deployment
```

Command line validation

1. List the OpenShift Cluster Nodes

```
# oc get nodes
NAME STATUS ROLES AGE VERSION
master-0.lab.hpecloud.org Ready master 115m v1.13.4+12ee15d4a
master-1.lab.hpecloud.org Ready master 115m v1.13.4+12ee15d4a
master-2.lab.hpecloud.org Ready master 115m v1.13.4+12ee15d4a
worker-0.lab.hpecloud.org Ready worker 115m v1.13.4+12ee15d4a
worker-1.lab.hpecloud.org Ready worker 115m v1.13.4+12ee15d4a
```

2. Configure the htpasswd identity provider and the admin user. See the [Red Hat OpenShift documentation](#) for configuring alternate and additional identity providers.
3. Create users.htpasswd file.

```
htpasswd -c -B -b /root/users.htpasswd <username> <password>
```

4. Create the htpasswd secret.

```
oc create secret generic htpass-secret / --from-file=htpasswd=/root/users.htpasswd -n openshift-config
```

5. Create the htpasswd identity provider configuration file.




```

vi htpasswdOAuth.yaml
apiVersion: config.openshift.io/v1
kind: OAuth
metadata:
  name: cluster
spec:
  IdentityProviders:
  - name: my_htpasswd_provider
    mappingMethod: claim
    type: HTPasswd
    htpasswd:
      fileData:
        name: htpass-secret

```

6. Configure htpasswd identity provider.

```
oc apply -f ./htpasswdOAuth.yaml
```

7. Assign cluster admin role to user defined in htpasswd file.

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

8. Test login.

```
oc login -u <username>
```

OpenShift web consoles

The Red Hat OpenShift Container Platform web console can be used to deploy new image-based container applications and administer the OpenShift cluster. Access the OpenShift web console from the following URL <https://console-openshift-console.<domain name>>. Figure 9 shows the overall health and status of the OpenShift cluster.

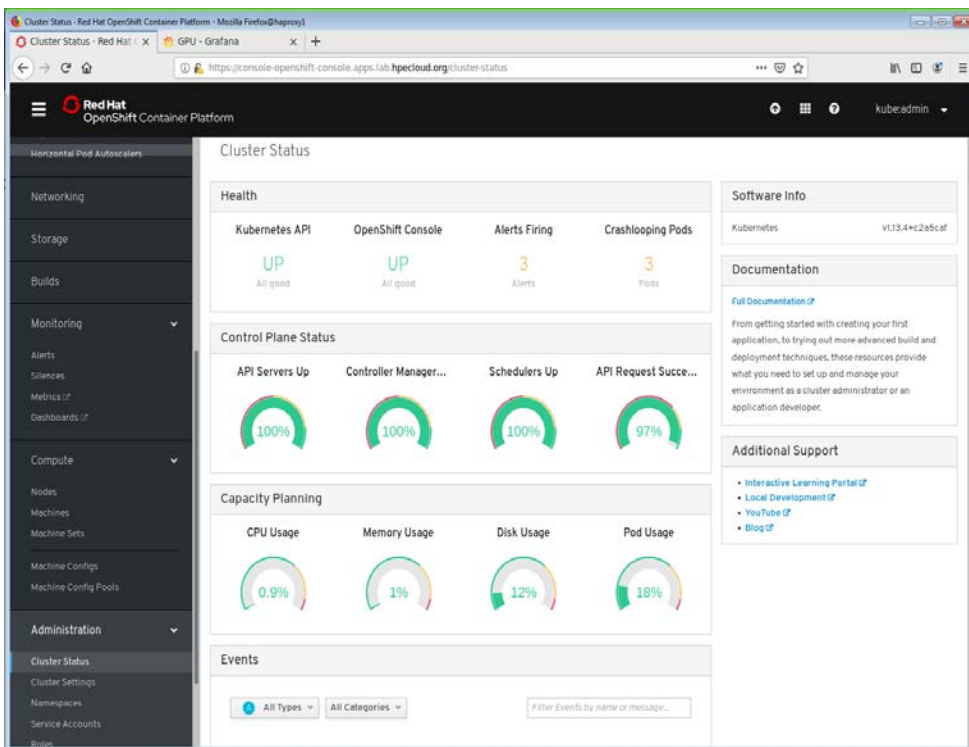


FIGURE 9. OpenShift web console Cluster status



Using the OpenShift web console the administrator can drill down into the cluster to view the status of different areas of the cluster including workloads and compute resources. Selecting Compute > Nodes provides the administrator with a list of OpenShift nodes and their status. Figure 10 shows the 5 nodes, 3 masters and 2 worker nodes, initially deployed in this solution. In this illustration the bootstrap node has not yet been repurposed into a worker node.

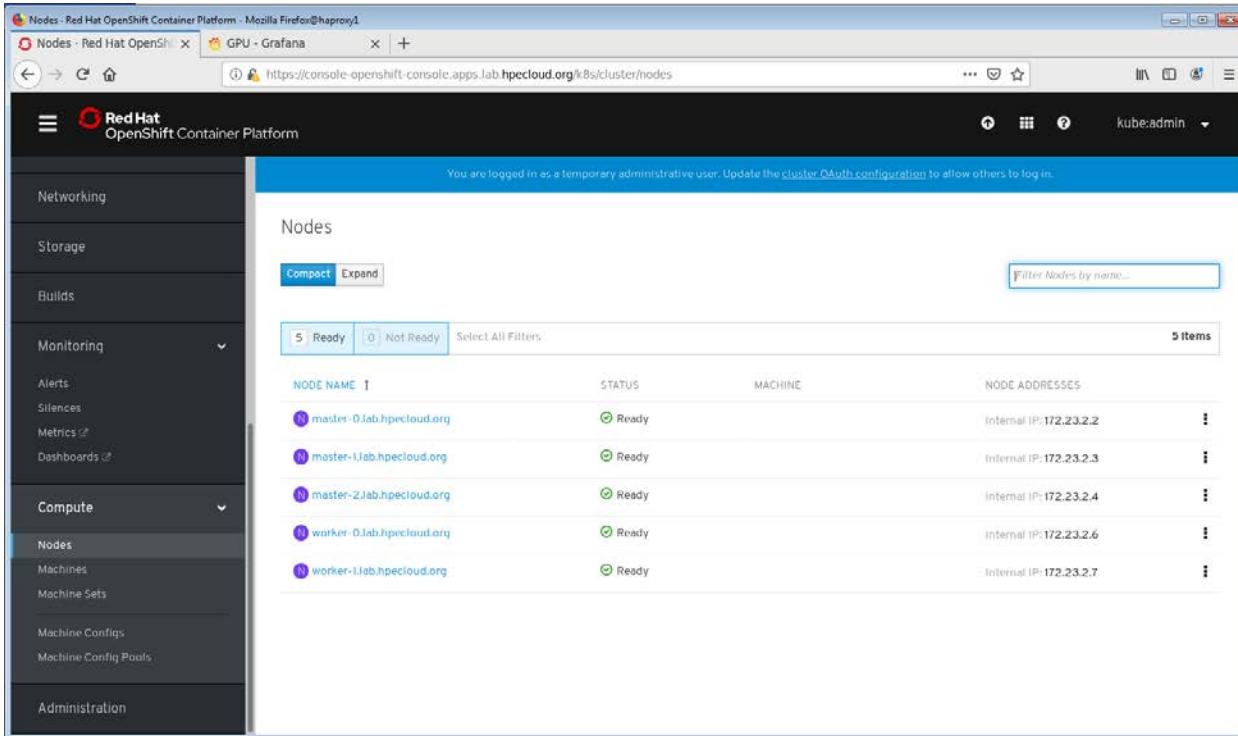


FIGURE 10. OpenShift web console Node view

NVIDIA DRIVER DEPLOYMENT

This solution describes the deployment of RHCOS worker nodes using the community supported NVIDIA drivers to provide an operator installation of containerized NVIDIA Cuda GPU drivers. The Cuda drivers used to enable the NVIDIA T4 GPUs are pulled from a private community supported GitHub repository. At the time of this writing OpenShift 4 only supports GPUs on RHEL 7 worker nodes. If Red Hat supported GPU worker nodes are required, an alternative solution would be to install the NVIDIA GPUs on RHEL 7 worker nodes and join the cluster using the procedure outlined in the [Red Hat OpenShift 4.1 documentation for adding RHEL 7 compute nodes](#).

Install the Node Feature Discovery (NFD) Operator

The Node Feature Discovery Operator detects available hardware features on nodes in the OpenShift cluster. In this case the Node Feature Discovery will identify the HPE ProLiant DL380 server nodes that have the NVIDIA T4 GPU modules installed. The Node Feature Discovery operator is available from the OperatorHub. Install the Node Feature Discovery operator using the following steps:

1. In the OpenShift console select Catalog > OperatorHub.
2. Open the Node Feature Discovery operator and select Install.
3. The Node Feature Discovery operator will be installed in the openshift-operators namespace.



Figure 11 illustrates the Node Feature Discovery operator install window that will be used to install the operator.

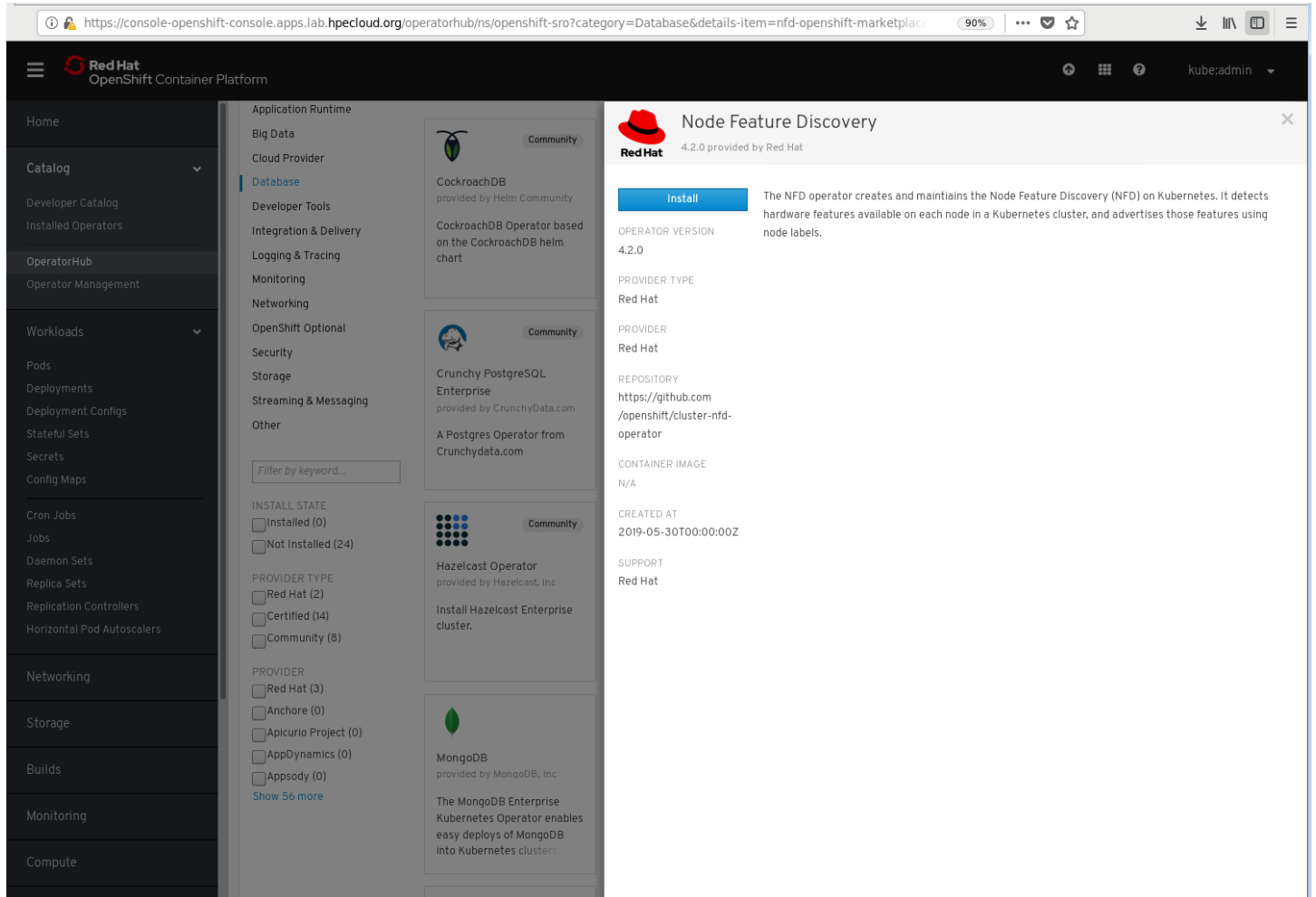


FIGURE 11. Node Feature Discovery operator



Once the Node Feature Discovery Operator is installed, select Catalog > Installed Operators to display the results of the operator installation as displayed in Figure 12.

The screenshot shows the OpenShift console interface. The left sidebar contains navigation options: Home, Catalog (with sub-items: Developer Catalog, Installed Operators), OperatorHub, Operator Management, Workloads (with sub-items: Pods, Deployments, Deployment Configs, Stateful Sets, Secrets, Config Maps), Cron Jobs, Jobs, Daemon Sets, Replica Sets, Replication Controllers, Horizontal Pod Autoscalers, Networking, Storage, Builds, Monitoring, and Compute. The main content area displays the 'Node Feature Discovery' operator details for the 'openshift-operators' project. It includes a 'Provided APIs' section with a 'Node Feature Discovery' card, a 'Description' section, and a 'ClusterServiceVersion Overview' table.

NAME	STATUS
nfd.4.2.1-201910221723	Succeeded

NAMESPACE	STATUS REASON
openshift-operators	install strategy completed with no errors

LABELS	OPERATOR DEPLOYMENTS
olm.api.7558ec538b9b19ff=provided	nfd-operator

FIGURE 12. Node Feature Discovery operator installed

The next step is to install the NVIDIA drivers for the GPU modules. At the time of this writing the NVIDIA driver operator is still in development. This paper will be updated with the procedures for installing the official NVIDIA driver operator once it has been posted to the Operator Hub. A reference implementation is available as a Special Resource Operator and can be installed using the procedure below.

1. Pull the NVIDIA Special Resource Operator.

```
#git clone https://github.com/openshift/om/-psap/special-resource-operator
#cd special-resource-operator
#git checkout release-4.2
#PULLPOLICY=Always make deploy
#make -C special-resource-operator deploy
```



Verify the GPU driver deployment

1. Run the oc project command to display the current project.

```
# oc project
Using project "openshift-sro"; on server "https://api.lab.hpecloud.org:6443";.
```

2. Run the oc get pod -o wide to display the NVIDIA containerized drivers.

```
# oc get pod -o wide
NAME READY STATUS RESTARTS AGE IP NODE
NOMINATED NODE READINESS GATES
NVIDIA-dcgm-exporter-cf1dl 2/2 Running 0 12d 172.23.2.7 worker-1.lab.hpecloud.org <none> <none>
NVIDIA-dcgm-exporter-lw7wt 2/2 Running 0 12d 172.23.2.6 worker-0.lab.hpecloud.org <none> <none>
NVIDIA-device-plugin-daemonset-7zg2v 1/1 Running 0 12d 10.131.0.27 worker-0.lab.hpecloud.org <none>
<none>
NVIDIA-device-plugin-daemonset-jp88w 1/1 Running 0 12d 10.128.2.16 worker-1.lab.hpecloud.org <none>
<none>
NVIDIA-device-plugin-validation 0/1 Completed 0 12d 10.128.2.17 worker-1.lab.hpecloud.org <none>
<none>
NVIDIA-device-plugin-validationr87 0/1 Completed 0 11d 10.128.2.26 worker-1.lab.hpecloud.org <none>
<none>
NVIDIA-driver-daemonset-58bd1 1/1 Running 0 12d 10.131.0.25 worker-0.lab.hpecloud.org <none> <none>
NVIDIA-driver-daemonset-86r8s 1/1 Running 0 12d 10.128.2.14 worker-1.lab.hpecloud.org <none> <none>
NVIDIA-driver-validation 0/1 Completed 0 12d 10.128.2.15 worker-1.lab.hpecloud.org <none> <none>
NVIDIA-feature-discovery-5b776 1/1 Running 0 12d 10.131.0.29 worker-0.lab.hpecloud.org <none> <none>
NVIDIA-feature-discovery-z5frm 1/1 Running 0 12d 10.128.2.18 worker-1.lab.hpecloud.org <none> <none>
NVIDIA-grafana-6fbddd75c-g9pm7 1/1 Running 0 12d 10.131.0.28 worker-0.lab.hpecloud.org <none> <none>
special-resource-operator-7f74c6786b-q8rmw 1/1 Running 1 12d 10.128.2.13 worker-1.lab.hpecloud.org <none>
<none>
```

The command `oc describe nodes | grep NVIDIA` will describe the nodes that are running the NVIDIA drivers. The output of the `oc describe nodes | grep NVIDIA` will display the driver information. The output below shows two (2) HPE ProLiant DL380 Gen10 servers each running 4 NVIDIA T4 GPU modules.

```
# oc describe nodes | grep NVIDIA
NVIDIA.com/cuda.driver.major=430
NVIDIA.com/cuda.driver.minor=34
NVIDIA.com/cuda.driver.rev=
NVIDIA.com/cuda.runtime.major=10
NVIDIA.com/cuda.runtime.minor=1
NVIDIA.com/gfd.timestamp=1572889382
NVIDIA.com/gpu.compute.major=7
NVIDIA.com/gpu.compute.minor=5
NVIDIA.com/gpu.family=undefined
NVIDIA.com/gpu.machine=ProLiant-DL380-Gen10
NVIDIA.com/gpu.memory=16127
NVIDIA.com/gpu.product=-T4
NVIDIA.com/gpu: 4
NVIDIA.com/gpu: 4
openshift-sro NVIDIA-dcgm-exporter-lw7wt 100m [0%] 200m [0%] 30Mi [0%] 50Mi [0%] 12d
openshift-sro NVIDIA-device-plugin-daemonset-7zg2v 0 [0%] 0 [0%] 0 [0%] 0 [0%] 12d
openshift-sro NVIDIA-driver-daemonset-58bd1 0 [0%] 0 [0%] 0 [0%] 0 [0%] 12d
openshift-sro NVIDIA-feature-discovery-5b776 0 [0%] 0 [0%] 0 [0%] 0 [0%] 12d
openshift-sro NVIDIA-grafana-6fbddd75c-g9pm7 0 [0%] 0 [0%] 0 [0%] 0 [0%] 12d
NVIDIA.com/gpu 0 0
NVIDIA.com/cuda.driver.major=430
NVIDIA.com/cuda.driver.minor=34
NVIDIA.com/cuda.driver.rev=
NVIDIA.com/cuda.runtime.major=10
```



```

NVIDIA.com/cuda.runtime.minor=1
NVIDIA.com/gfd.timestamp=1572889374
NVIDIA.com/gpu.compute.major=7
NVIDIA.com/gpu.compute.minor=5
NVIDIA.com/gpu.family=undefined
NVIDIA.com/gpu.machine=ProLiant-DL380-Gen10
NVIDIA.com/gpu.memory=16127
NVIDIA.com/gpu.product=-T4
NVIDIA.com/gpu: 4
NVIDIA.com/gpu: 4
openshift-sro NVIDIA-dcgm-exporter-cfldl 100m [0%] 200m [0%] 30Mi [0%] 50Mi [0%] 12d
openshift-sro NVIDIA-device-plugin-daemonset-jp88w 0 [0%] 0 [0%] 0 [0%] 0 [0%] 12d
openshift-sro NVIDIA-driver-daemonset-86rhs 0 [0%] 0 [0%] 0 [0%] 0 [0%] 12d
openshift-sro NVIDIA-feature-discovery-z5frm 0 [0%] 0 [0%] 0 [0%] 0 [0%] 12d
NVIDIA.com/gpu 0 0

```

Testing the GPUs

1. Test the NVIDIA GPU drivers using the `NVIDIA-smi`.
2. Create the `NVIDIA-smi.yaml` file.

```

#vi NVIDIA-smi.yaml

apiVersion: v1
kind: Pod
metadata:
  name: NVIDIA-smi
spec:
  containers:
  - image: NVIDIA/cuda
    name: NVIDIA-smi
    command: ["/bin/bash", "-c","NVIDIA-smi; exit 0" ]
    resources:
      limits:
        NVIDIA.com/gpu: 1
      requests:
        NVIDIA.com/gpu: 1

```

```
#oc create -f NVIDIA-smi.yaml
```

```
#oc logs NVIDIA-smi
```

```

Wed Oct 30 16:13:33 2019
+-----+
| NVIDIA-SMI 430.34      Driver Version: 430.34      CUDA Version: 10.1      |
+-----+-----+-----+-----+-----+-----+
| GPU  Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+-----+-----+-----+
|   0   T4             On          | 00000000:86:00:0 Off  |      0          |
| N/A   77C    P0      70W /  70W | 13603MiB / 15109MiB |    100%      Default  |
+-----+-----+-----+-----+-----+
+-----+
| Processes:                                                       GPU Memory
|  GPU       PID    Type    Process name                                             Usage
+-----+-----+-----+-----+-----+

```



NGC - Sample Application Deployment

NVIDIA maintains the NGC Accelerated Software catalog of containerized ML/AI applications that can be deployed into the OpenShift cluster. The available applications are grouped by industry, AI applications, or technology.

1. A tensorflow container was deployed from the NVIDIA catalog using the command and associated .yaml file shown below. This pod deployment launched an instance of Tensorflow and validated it with the python resnet50 job.

```
#vi tensorflow-benchmarks-gpu-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: tensorflow-benchmarks-gpu
spec:
  containers:
  - image: nvcr.io/NVIDIA/tensorflow:19.09-py3
    name: cudnn
    command: ["/bin/sh","-c"]
    args: ["git clone https://github.com/tensorflow/benchmarks.git;cd benchmarks/scripts/tf_cnn_benchmarks;python3
    tf_cnn_benchmarks.py --num_gpus=1 --data_format=NHWC --batch_size=32 --model=resnet50 --
    variable_update=parameter_server"]
  resources:
  limits:
  NVIDIA.com/gpu: 1
  requests:
  NVIDIA.com/gpu: 1
  restartPolicy: Never
```

2. Create the tensorflow and benchmark pod.

```
#oc create -f tensorflow-benchmarks-gpu-pod.yaml
```

3. Run the oc logs command to review the output from the Tensorflow pod.

```
#oc logs tensorflow-benchmarks-gpu
```

```
TensorFlow: 1.14
Model: resnet50
Dataset: imagenet [synthetic]
Mode: training
SingleSess: False
Batch size: 32 global
32 per device
Num batches: 100
Num epochs: 0.00
Devices: ['/gpu:0']
NUMA bind: False
Data format: NHWC
Optimizer: sgd
Variables: parameter_server
=====
Generating training model
Initializing graph
Running warm up
Done warm up
Step Img/sec total_loss
1 images/sec: 135.9 +/- 0.0 [jitter = 0.0] 8.108
10 images/sec: 134.9 +/- 0.3 [jitter = 1.4] 8.122
20 images/sec: 134.8 +/- 0.2 [jitter = 0.7] 7.983
```



```

30 images/sec: 134.7 +/- 0.2 [jitter = 0.8] 7.780
40 images/sec: 134.5 +/- 0.1 [jitter = 0.8] 7.848
50 images/sec: 134.3 +/- 0.1 [jitter = 0.7] 7.779
60 images/sec: 134.2 +/- 0.1 [jitter = 0.8] 7.825
70 images/sec: 134.0 +/- 0.1 [jitter = 0.8] 7.839
80 images/sec: 133.9 +/- 0.1 [jitter = 0.9] 7.818
90 images/sec: 133.7 +/- 0.1 [jitter = 1.0] 7.648
100 images/sec: 133.6 +/- 0.1 [jitter = 1.1] 7.915
-----
total images/sec: 133.52
-----
    
```

Grafana NVIDIA dashboards

The NVIDIA driver installation also implements Grafana dashboards that can be used to monitor the performance metrics of the GPUs. Login into the Grafana dashboard at the following URL: `NVIDIA-grafana-openshift-sro.<domain name>` with the username and password of `admin/admin`. You will be prompted to change the default username and password. Figure 13 illustrates the Grafana dashboard for GPU performance metrics.



FIGURE 13. NVIDIA Grafana dashboard

APPENDIX A: BILL OF MATERIALS

The following HPE UCID can be referenced to build out the configuration listed in this BOM: UCID: 5117723231-02.

TABLE A1. BOM

Qty	Part Number	Description
1	P9K10A	HPE 42U 600mmx1200mm G2 Kitted Advanced Shock Rack with Side Panels and Baying
1	P9K10A 001	HPE Factory Express Base Racking Service



Qty	Part Number	Description
4	867959-B21	HPE ProLiant DL360 Gen10 8SFF Configure-to-order Server
4	867959-B21 OD1	Factory Integrated
4	867959-B21 ABA	HPE DL360 Gen10 8SFF CTO Server
4	860663-L21	HPE DL360 Gen10 Intel Xeon-Gold 5118 (2.3GHz/12-core/105W) FIO Processor Kit
4	860663-B21	HPE DL360 Gen10 Intel Xeon-Gold 5118 (2.3GHz/12-core/105W) Processor Kit
4	860663-B21 OD1	Factory Integrated
16	815097-B21	HPE 8GB (1x8GB) Single Rank x8 DDR4-2666 CAS-19-19-19 Registered Smart Memory Kit
16	815097-B21 OD1	Factory Integrated
8	P07926-B21	HPE 960GB SATA 6G Mixed Use SFF (2.5in) SC 3yr Wty Digitally Signed Firmware SSD
8	P07926-B21 OD1	Factory Integrated
4	867982-B21	HPE DL360 Gen10 Low Profile Riser Kit
4	867982-B21 OD1	Factory Integrated
4	P01366-B21	HPE 96W Smart Storage Battery (up to 20 Devices) with 145mm Cable Kit
4	P01366-B21 OD1	Factory Integrated
4	804331-B21	HPE Smart Array P408i-a SR Gen10 (8 Internal Lanes/2GB Cache) 12G SAS Modular Controller
4	804331-B21 OD1	Factory Integrated
4	879482-B21	HPE InfiniBand FDR/Ethernet 40/50Gb 2-port 547FLR-QSFP Adapter
4	879482-B21 OD1	Factory Integrated
8	865414-B21	HPE 800W Flex Slot Platinum Hot Plug Low Halogen Power Supply Kit
8	865414-B21 OD1	Factory Integrated
4	BD505A	HPE iLO Advanced 1-server License with 3yr Support on iLO Licensed Features
4	BD505A OD1	Factory Integrated
4	874543-B21	HPE 1U Gen10 SFF Easy Install Rail Kit
4	874543-B21 OD1	Factory Integrated
2	868703-B21	HPE ProLiant DL380 Gen10 8SFF Configure-to-order Server
2	868703-B21 OD1	Factory Integrated
2	868703-B21 ABA	HPE DL380 Gen10 8SFF CTO Server
2	826862-L21	HPE DL380 Gen10 Intel Xeon-Gold 6126 (2.6GHz/12-core/120W) FIO Processor Kit
2	826862-B21	HPE DL380 Gen10 Intel Xeon-Gold 6126 (2.6GHz/12-core/120W) Processor Kit
2	826862-B21 OD1	Factory Integrated
24	835955-B21	HPE 16GB (1x16GB) Dual Rank x8 DDR4-2666 CAS-19-19-19 Registered Smart Memory Kit
24	835955-B21 OD1	Factory Integrated
4	P07926-B21	HPE 960GB SATA 6G Mixed Use SFF (2.5in) SC 3yr Wty Digitally Signed Firmware SSD
4	P07926-B21 OD1	Factory Integrated
2	826700-B21	HPE DL38X Gen10 x16 Tertiary Riser Kit
2	826700-B21 OD1	Factory Integrated
8	ROW29C	HPE NVIDIA T4 16GB Computational Accelerator
8	ROW29C OD1	Factory Integrated
2	826704-B21	HPE DL Gen10 x16/x16 GPU Riser Kit
2	826704-B21 OD1	Factory Integrated
2	871674-B21	HPE DL38X Gen10 Slot 1/2 x16/x16 FIO Riser Kit
2	804326-B21	HPE Smart Array E208i-a SR Gen10 (8 Internal Lanes/No Cache) 12G SAS Modular Controller



Qty	Part Number	Description
2	804326-B21 OD1	Factory Integrated
2	879482-B21	HPE InfiniBand FDR/Ethernet 40/50Gb 2-port 547FLR-QSFP Adapter
2	879482-B21 OD1	Factory Integrated
2	867810-B21	HPE DL38X Gen10 High Performance Temperature Fan Kit
2	867810-B21 OD1	Factory Integrated
4	830272-B21	HPE 1600W Flex Slot Platinum Hot Plug Low Halogen Power Supply Kit
4	830272-B21 OD1	Factory Integrated
2	BD505A	HPE iLO Advanced 1-server License with 3yr Support on iLO Licensed Features
2	BD505A OD1	Factory Integrated
2	733660-B21	HPE 2U Small Form Factor Easy Install Rail Kit
2	733660-B21 OD1	Factory Integrated
2	826706-B21	HPE DL380 Gen10 High Performance Heat Sink Kit
2	826706-B21 OD1	Factory Integrated
1	JL260A	Aruba 2930F 48G 4SFP Switch
1	JL260A B2E	Aruba 2930F 48G 4SFP Switch United States 220 volt
1	JL260A OD1	Factory Integrated
1	ROP75A	HPE StoreFabric SN2100M 100GbE 16QSFP28 Power to Connector Airflow Half Width TAA-compliant Switch
1	ROP75A OD1	Factory Integrated
1	H6J85A	HPE Rack Hardware Kit
1	H6J85A OD1	Factory Integrated
2	P9Q41A	HPE G2 Basic 4.9kVA/L6-30P 24A/208V Outlets (20) C13/Vertical NA/JP PDU
2	P9Q41A OD1	Factory Integrated
1	120672-B21	HPE Rack Ballast Kit
1	120672-B21 OD1	Factory Integrated
1	BW932A	HPE 600mm Rack Stabilizer Kit
1	BW932A B01	HPE 600mm Rack include with Complete System Stabilizer Kit
1	J9583A	HPE X410 1U Universal 4-post Rackmount Kit
1	J9583A OD1	Factory Integrated
1	Q9Y41AAE	HPE Network Orchestrator E-LTU
1	HA113A1	HPE Installation SVC
1	HA113A1 5BY	HPE Rack and Rack Options Install SVC
1	HA114A1	HPE Installation and Startup Service
4	HA114A1 5A0	HPE Startup Entry 300 Series OS SVC
2	HA114A1 5A6	HPE Startup 300 Series OS SVC
1	HA114A1 5SE	HPE StoreFabric M-series Eth Startup SVC
1	H1K92A3	HPE 3Y Proactive Care 24x7 SVC
6	H1K92A3 R2M	HPE iLO Advanced Non Blade Support
4	H1K92A3 WAG	HPE DL360 Gen10 Support
2	H1K92A3 WAH	HPE DL38x Gen10 Support
1	H1K92A3 RCC	HPE SN2100M Storage Switch Support
1	H1K92A3 WKL	HPE Aruba 2930F48G4SFP Switch Supp
1	H1K92A3 ZGT	HPE Smart Fabric Orchestrator SW-6 Support



Qty	Part Number	Description
2	HF385A1	HPE Training Credit Servers/Hybrid IT SVC

TABLE A2. Red Hat OpenShift Subscriptions

Qty	Part Number	Description
18		Red Hat OpenShift Container Platform, Premium, 2-Core

APPENDIX B: PXE CONFIGURATION

The RHCOS servers in this solution are installed using an external server that provides PXE boot, HTTP, and DHCP services. This server hosts the RHCOS installation image on an httpd web server. DHCP is configured to assign specific IP addresses to the RHCOS 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 using the iLO management interface on each of the respective RHCOS servers. This section provides example configuration settings for creating a network boot server. For detailed setup and configuration information, see the [Red Hat documentation](#).

PXE server environment

1. On the PXE server enable following repositories:

```
subscription-manager repos --enable="rhel-7-server-rpms" --enable="rhel-7-server-extras-rpms" --
enable="rhel-7-fast-datapath-rpms"
```

2. yum install syslinux tftp-server httpd dhcp xinetd.
3. Configure the tftp server.
 - a. cp -r /usr/share/syslinux/* /var/lib/tftpboot/
 - b. systemctl start tftp
 - c. systemctl enable tftp
 - d. mkdir /var/lib/tftpboot/pxelinux.cfg
 - e. touch /var/lib/tftpboot/pxelinux.cfg/default
 - f. vi /var/lib/tftpboot/pxelinux.cfg/default

```
default menu.c32
prompt 0
timeout 300
ONTIMEOUT local
menu title #####PXE Boot MENU #####
LABEL local
    MENU LABEL Boot local hard drive
    LOCALBOOT 0
LABEL rhcos-master
MENU LABEL Install RHCOS Master
KERNEL vmlinuz-rhcos
    APPEND ip=dhcp rd.neednet=1 initrd=initramfs.img coreos.inst=yes coreos.inst.install_dev=sda coreos.first_boot=1
coreos.inst.image_url=http://10.19.20.221/rhcos/ rhcos-4.1.0-x86_64-metal-bios.raw.gz
coreos.inst.ignition_url=http://10.19.20.221/rhcosos/master.ign
IPAPPEND 2
LABEL rhcos-worker
MENU LABEL Install RHCOS Worker
KERNEL vmlinuz-rhcos
```



```

APPEND ip=dhcp rd.neednet=1 initrd=initramfs.img coreos.inst=yes coreos.inst.install_dev=sda coreos.first_boot=1
coreos.inst.image_url=http://10.19.20.221/rhcos/ rhcos-4.1.0-x86_64-metal-bios.raw.gz
coreos.inst.ignition_url=http://10.19.20.221/rhcos/worker.ign
IPAPPEND 2
LABEL rhcos-bootstrap
MENU LABEL Install RHCOS Bootstrap
KERNEL vmlinuz-rhcos
APPEND ip=dhcp rd.neednet=1 initrd=initramfs.img coreos.inst=yes coreos.inst.install_dev=sda coreos.first_boot=1
coreos.inst.image_url=http://10.19.20.221/rhcosos/ rhcos-4.1.0-x86_64-metal-bios.raw.gz
coreos.inst.ignition_url=http://10.19.20.221/rhcos/bootstrap.ign
IPAPPEND 2

```

4. Configure the firewall to pass ftp and http

```

firewall-cmd --add-service=ftft --permanent
firewall-cmd --add-service=http --permanent

```

5. Configure httpd

```

mkdir /var/www/html/rhcos

```

6. Enable and start httpd

```

systemctl enable httpd
systemctl start httpd

```

APPENDIX C: DHCP CONFIGURATION

```

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 "lab.hpecloud.org";
option domain-name-servers 10.19.20.221;
option ntp-servers 10.16.255.1;
Allow booting;
Allow bootp;
default-lease-time 3600;
max-lease-time 7200;
log-facility local7;
subnet 10.19.20.128 netmask 255.255.255.128 {
    option broadcast-address 10.19.20.255;
    filename "pxelinux.0";
    option domain-name-servers 10.19.20.221;
    option domain-search "lab.hpecloud.org";
    option routers 10.19.20.254;
}

host master-0 {
    hardware ethernet 30:e1:71:62:8d:30;
    fixed-address 10.19.20.247;
    option host-name "master-0";
}
host master-1 {
    hardware ethernet 30:e1:71:62:4d:9c;

```



```

fixed-address 10.19.20.248;
option host-name "master-1";
}
host master-2 {
hardware ethernet 30:e1:71:63:d0:dc;
fixed-address 10.19.20.249;
option host-name "master-2";
}
host worker-0 {
hardware ethernet 30:e1:71:63:e0:ec;
fixed-address 10.19.20.250;
option host-name "worker-0";
}
host worker-1 {
hardware ethernet 30:e1:71:63:c0:98;
fixed-address 10.19.20.251;
option host-name "worker-1";
}
host bootstrap {
hardware ethernet 30:e1:71:63:c0:74;
fixed-address 10.19.20.252;
option host-name "bootstrap";
}

systemctl enable dhcpd
systemctl start dhcpd

```

APPENDIX D: DNS CONFIGURATION

```

$ORIGIN lab.hpecloud.org
$TTL 300
$ORIGIN .
$TTL 10800      ; 3 hours
lab.hpecloud.org IN SOA  lab.hpecloud.org. admin.lab.hpecloud.org. (
                                580      ; serial
                                10800    ; refresh (3 hours)
                                3600     ; retry (1 hour)
                                604800   ; expire (1 week)
                                3600     ; minimum (1 hour)
                                )
                                NS      infrastructure.hpecloud.org.
                                A      10.19.20.221
$ORIGIN lab.hpecloud.org.
api                A      10.19.20.223
api-int           A      10.19.20.223
master-0         A      10.19.20.247
master-1         A      10.19.20.248
master-2         A      10.19.20.249
worker-0         A      10.19.20.250
worker-1         A      10.19.20.251
bootstrap        A      10.19.20.252
etcd-0           CNAME   master-0
etcd-1           CNAME   master-1
etcd-2           CNAME   master-2
_etcd-server-ssl._tcp SRV   0 10 2380 etcd-0
                  SRV   0 10 2380 etcd-1
                  SRV   0 10 2380 etcd-2
$ORIGIN apps.lab.hpecloud.org.
*                 A      10.19.20.223

```



APPENDIX E: EXAMPLE LOAD BALANCER CONFIGURATION

Sample HA Proxy Configuration File (/etc/haproxy/haproxy.cfg).

```
# cd /etc/haproxy/
# cat haproxy.cfg
defaults
    mode                http
    log                 global
    option              httplog
    option              dontlognull
    option forwardfor   except 127.0.0.0/8
    option              redispatch
    retries             3
    timeout http-request 10s
    timeout queue       1m
    timeout connect     10s
    timeout client      300s
    timeout server      300s
    timeout http-keep-alive 10s
    timeout check       10s
    maxconn             20000

listen stats
    bind :9000
    mode http
    stats enable
    stats uri /

frontend openshift-api-server
    bind *:6443
    default_backend openshift-api-server
    mode tcp
    option tcplog

backend openshift-api-server
    balance source
    mode tcp
    server bootstrap.lab 10.19.20.252:6443 check
    server master-0.lab 10.19.20.247:6443 check
    server master-1.lab 10.19.20.248:6443 check
    server master-2.lab 10.19.20.249:6443 check

frontend machine-config-server
    bind *:22623
    default_backend machine-config-server
    mode tcp
    option tcplog

backend machine-config-server
    balance source
    mode tcp
    server bootstrap.lab 10.19.20.252:22623 check
    server master-0.lab 10.19.20.247:22623 check
    server master-1.lab 10.19.20.248:22623 check
    server master-2.lab 10.19.20.249:22623 check

frontend ingress-http
    bind *:80
    default_backend ingress-http
    mode tcp
    option tcplog

backend ingress-http
    balance source
    mode tcp
    server worker-0.lab 10.19.20.250:80 check
```



```
server worker-1.lab 10.19.20.251:80 check
frontend ingress-https
  bind *:443
  default_backend ingress-https
  mode tcp
  option tcplog
backend ingress-https
  balance source
  mode tcp
  server worker-0.lab 10.19.20.250:443 check
  server worker-1.lab 10.19.20.251:443 check
```



Reference Architecture

RESOURCES AND ADDITIONAL LINKS

HPE Reference Architectures, hpe.com/info/ra

HPE Servers, hpe.com/servers

HPE ProLiant DL380 Gen10 server, <https://h20195.www2.hpe.com/v2/default.aspx?cc=us&lc=en&oid=10100268180> Gen10 server

HPE ProLiant DL360 Gen10 server , <http://www.hpe.com/servers/dl360-gen10>

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

NVIDIA, <https://www.NVIDIA.com/en-us/data-center/-t4/>

NVIDIA NGC, <https://ngc.NVIDIA.com/catalog/all>

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

© Copyright 2020 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

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

a50000817enw, January 2020