Dell EMC Ready Stack for Red Hat OpenShift Container Platform 4.3
Enabled by Dell EMC PowerEdge R-Series Servers and PowerSwitch Networking

January 2021
H18212.1

Deployment Guide

Abstract
This deployment guide provides a validated procedure for deploying Red Hat OpenShift Container Platform 4.3 on Dell EMC PowerEdge servers.

Dell Technologies Solutions
The information in this publication is provided as is. Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

Copyright © 2021 Dell Inc. or its subsidiaries. All Rights Reserved. Dell Technologies, Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Intel, the Intel logo, the Intel Inside logo and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries. Other trademarks may be trademarks of their respective owners. Published in the USA 01/21 Deployment Guide H18212.1.

Dell Inc. believes the information in this document is accurate as of its publication date. The information is subject to change without notice.
Contents

Chapter 1  Introduction  ................................................................. 5
Solution overview ............................................................................ 6
Document purpose ............................................................................. 6
Audience .......................................................................................... 7
We value your feedback ................................................................. 7

Chapter 2  Configuring Switches .................................................... 8
Introduction ....................................................................................... 9
Customizing the Dell EMC switches ..................................................... 10
Configuring the Dell EMC switches ..................................................... 11

Chapter 3  Setting Up the CSAH Node .......................................... 12
Introduction ....................................................................................... 13
Preparing the CSAH node ................................................................. 13
Preparing and running Ansible playbooks ........................................ 15

Chapter 4  Deploying OpenShift 4.3 ............................................. 20
Introduction ....................................................................................... 21
Creating a bootstrap node ............................................................... 21
Installing control plane nodes ........................................................... 22
Installing worker nodes ................................................................. 25
Completing the cluster setup ............................................................ 26
Removing the bootstrap node ........................................................... 27
Validating and approving Certificate Signing Requests .................... 27
Validating the cluster operators ....................................................... 29
Accessing the OpenShift web console ............................................. 30
Configuring authentication .............................................................. 30
Configuring image registry storage ............................................... 32

Chapter 5  Adding Worker Nodes ................................................ 36
Adding a Red Hat Enterprise Linux worker node .............................. 37
Adding Red Hat Enterprise Linux CoreOS worker node .................... 41

Chapter 6  Deploying and Scaling Applications ................................ 42
Deploying applications ..................................................................... 43
Scaling applications ......................................................................... 45
Chapter 7  Provisioning Storage  47
Introduction ......................................................................................................................... 48
Prerequisites for using Unity storage ................................................................................... 48
Adding worker nodes in Unity hosts .................................................................................... 48
Creating a LUN and mapping it to the worker nodes .......................................................... 49
Creating a pod using Unity storage ..................................................................................... 50

Chapter 8  Monitoring the Cluster  53
OpenShift monitoring overview ............................................................................................ 54
Enabling Grafana dashboards ............................................................................................... 54
Viewing alerts ....................................................................................................................... 55
Viewing cluster metrics ........................................................................................................ 55
This chapter presents the following topics:

- **Solution overview** ................................................................. 6
- **Document purpose** ............................................................... 6
- **Audience** ............................................................................... 7
- **We value your feedback** ........................................................... 7
Solution overview

Red Hat OpenShift Container Platform is an open-source application deployment platform that is based on Kubernetes container orchestration technology. Containers are stand-alone processes that run within their own environment and runtime context, independent of the underlying infrastructure. Red Hat OpenShift Container Platform helps you develop, deploy, and manage container-based applications.

Note: While you can rely on Red Hat Enterprise Linux security and container technologies to prevent intrusions and protect your data, some security vulnerabilities might persist. For information about security vulnerabilities in OpenShift Container Platform, see OCP Errata. For a general listing of Red Hat vulnerabilities, see the RH Security Home Page.

As part of Red Hat OpenShift Container Platform, Kubernetes manages containerized applications across a set of containers or hosts and provides mechanisms for the deployment, maintenance, and scaling of applications. The container runtime engine packages, instantiates, and runs containerized applications.

A Kubernetes cluster consists of one or more control plane nodes and a set of worker nodes. Kubernetes allocates an IP address from an internal network to each pod so that all containers within the pod behave as if they were on the same host. Giving each pod its own IP address means that pods can be treated like physical hosts or virtual machines for port allocation, networking, naming, service discovery, load balancing, application configuration, and migration. Dell Technologies recommends creating a Kubernetes service that will enable your application pods to interact, rather than requiring that the pods communicate directly using their IP addresses.

A fully functioning Domain Name System (DNS) residing outside the OpenShift Container Platform is crucial in the deployment and operation of your container ecosystem. Red Hat OpenShift Container Platform has an integrated DNS so that the services can be found through DNS service entries or through the service IP/port registrations.

Dell EMC Ready Stack for Red Hat OpenShift Container Platform is a proven design to help organizations accelerate their container deployments and cloud-native adoption. Dell Technologies delivers tested, validated, and documented design guidance to help customers rapidly deploy Red Hat OpenShift on Dell EMC infrastructure by minimizing time and effort. For more information, see the Red Hat OpenShift Container Platform 4.3 solution design guide, which is available at Red Hat OpenShift Container Platform on the Dell Technologies Info Hub.

Document purpose

This deployment guide describes the infrastructure that is required for deploying and operating Red Hat OpenShift Container Platform and provides a validated process for deploying a production-ready OpenShift Container Platform cluster. This guide provides information to facilitate readiness for Day 2 operations.

This guide provides procedures for deploying Red Hat OpenShift Container Platform 4.3 on Dell EMC PowerEdge servers and with Dell EMC PowerSwitch switches. Dell Technologies strongly recommends that you complete the validation steps that are
described in this guide and ensure that you are satisfied that your application will operate smoothly before proceeding with development or production use.

For more information about OpenShift Container Platform, see the OpenShift Container Platform 4.3 Documentation.

This guide may contain language that is not consistent with Dell's current guidelines. Dell plans to update the guide over subsequent future releases to revise the language accordingly.

Audience

This deployment guide is for system administrators and system architects. Some experience with Docker and Red Hat OpenShift Container Platform technologies is recommended. Review the solution design guide to familiarize yourself with the solution architecture and design before planning your deployment.

We value your feedback

Dell Technologies and the authors of this document welcome your feedback on the solution and the solution documentation. Contact the Dell Technologies Solutions team by email or provide your comments by completing our documentation survey.

Author: Umesh Sunnapu

Contributors: Scott Powers, John Terpstra, Karen Johnson

Note: For links to additional documentation for this solution, see Red Hat OpenShift Container Platform on the Dell Technologies Info Hub.
This chapter presents the following topics:

**Introduction** .................................................................................................................. 9

**Customizing the Dell EMC switches** .............................................................................. 10

**Configuring the Dell EMC switches** ............................................................................. 11
Introduction

Overview

Dell Technologies has provided sample switch configuration files in GitHub at [dell-esg/openshift-bare-metal](https://github.com/dell-esg/openshift-bare-metal). These files enable you to easily configure the switches that are used for the OpenShift Container Platform cluster. This chapter describes how to customize these configuration files.

**Note:** Clone the repository using `git clone https://github.com/dell-esg/openshift-bare-metal.git` and change to the `examples` directory.

**CAUTION:** If you use different hardware or need different configurations, modify the configuration files accordingly.

Typographical conventions

Configuration instructions use certain typographical conventions to designate commands and screen output.

Command syntax is identified by *Courier New* font. Information that is specific to your environment is placed inside `<>` symbols and in *italics*. For example:

- Deployment guide command reference: `OS10 (config)# hostname <hostname>`
- On the top S5232F-ON switch, enter: `OS10 (config)# hostname SW1`

Screen output is in *Italic* *Courier New* type. Any user action or commands are in *Courier New*.

Sample port configuration between switches

Dell EMC PowerSwitch S5232F switches are in Virtual Link Trunking (VLT) mode and uplinked to core switches in the environment. PowerSwitch S3048 has an uplink to S5232F switches. The following table shows the port connections between switches. These connections are reflected in the sample switch configuration files.

<table>
<thead>
<tr>
<th>Switch name</th>
<th>Port</th>
<th>Description</th>
<th>S5232F-1</th>
<th>S5232F-2</th>
<th>S3048</th>
</tr>
</thead>
<tbody>
<tr>
<td>S5232F-1</td>
<td>1/1/30</td>
<td>Uplink to core switch</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S5232F-1</td>
<td>1/1/31</td>
<td>VLTi</td>
<td>X</td>
<td>1/1/31</td>
<td>X</td>
</tr>
<tr>
<td>S5232F-1</td>
<td>1/1/32</td>
<td>VLTi</td>
<td>X</td>
<td>1/1/32</td>
<td>X</td>
</tr>
<tr>
<td>S3252F-1</td>
<td>1/1/34</td>
<td>Uplink to S3048</td>
<td>X</td>
<td>X</td>
<td>1/1/51</td>
</tr>
<tr>
<td>S5232F-2</td>
<td>1/1/30</td>
<td>Uplink to core switch</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S5232F-2</td>
<td>1/1/34</td>
<td>Uplink to S3048</td>
<td>X</td>
<td>x</td>
<td>1/1/52</td>
</tr>
</tbody>
</table>
Customizing the Dell EMC switches

The following table shows connections from each server to the switch ports with a 100 GbE NIC in PCI slot 2.

- SW1 (S5232F-1) is connected to port 1 in PCI slot 2.
- SW2 (S5232F-2) is connected to port 2 in PCI slot 2.
- S3048 is connected to iDRAC.

<table>
<thead>
<tr>
<th>Role</th>
<th>S5232F-1</th>
<th>S5232F-2</th>
<th>S3048</th>
<th>iDRAC IP VLAN 34 100.82.34.0/24</th>
<th>Public IP VLAN 461 100.82.46.0/26</th>
</tr>
</thead>
<tbody>
<tr>
<td>csah</td>
<td>1/1/1</td>
<td>1/1/1</td>
<td>1/1/1</td>
<td>100.82.34.20</td>
<td>100.82.46.20</td>
</tr>
<tr>
<td>etcd-0</td>
<td>1/1/2</td>
<td>1/1/2</td>
<td>1/1/2</td>
<td>100.82.34.21</td>
<td>100.82.46.21</td>
</tr>
<tr>
<td>etcd-1</td>
<td>1/1/3</td>
<td>1/1/3</td>
<td>1/1/3</td>
<td>100.82.34.22</td>
<td>100.82.46.22</td>
</tr>
<tr>
<td>etcd-2</td>
<td>1/1/4</td>
<td>1/1/4</td>
<td>1/1/4</td>
<td>100.82.34.23</td>
<td>100.82.46.23</td>
</tr>
<tr>
<td>worker1</td>
<td>1/1/5</td>
<td>1/1/5</td>
<td>1/1/5</td>
<td>100.82.34.24</td>
<td>100.82.46.24</td>
</tr>
<tr>
<td>worker2</td>
<td>1/1/6</td>
<td>1/1/6</td>
<td>1/1/6</td>
<td>100.82.34.25</td>
<td>100.82.46.25</td>
</tr>
<tr>
<td>bootstrap/worker3</td>
<td>1/1/7</td>
<td>1/1/7</td>
<td>1/1/7</td>
<td>100.82.34.26</td>
<td>100.82.46.26</td>
</tr>
</tbody>
</table>

The following table shows the firmware versions running in each switch model:

<table>
<thead>
<tr>
<th>PowerSwitch model</th>
<th>Out-of-band management IP</th>
<th>Firmware version</th>
<th>Default username</th>
<th>Default password</th>
</tr>
</thead>
<tbody>
<tr>
<td>S5232F-1</td>
<td>100.82.33.44</td>
<td>10.5.1.0</td>
<td>admin</td>
<td>admin</td>
</tr>
<tr>
<td>S5232F-2</td>
<td>100.82.33.45</td>
<td>10.5.1.0</td>
<td>admin</td>
<td>admin</td>
</tr>
<tr>
<td>S3048</td>
<td>100.82.33.46</td>
<td>10.5.1.0</td>
<td>admin</td>
<td>admin</td>
</tr>
</tbody>
</table>

To modify the switches for your environment:

1. Download the switch configuration files from GitHub:
   ```
   git clone https://github.com/dell-esg/openshift-bare-metal.git
   ```

2. Modify the sample switch configuration files in `<git clone dir>/containers/examples` to match your VLAN and IP schemes.

   The deployment uses untagged VLANs that use switchport access for nodes and tagged port channels for switch uplinks.
The deployment sample uses:

- VLAN_461 configured public network
- VLAN_34 configured for management network
- Single 100 GbE Mellanox X5 DP NIC in PCI slot 2

**Note:** The serial-port baud rate is 115200.

**Note:** Ethernet ports ens2f0 and ens2f1 in R640 servers are used in the rest of the document for CoreOS, p2p1, and p2p2 for Red Hat Enterprise Linux 7.x.

### Configuring the Dell EMC switches

This section describes Dell EMC Networking OS10 initial OOB management IP setup and provides sample switch configuration directions copied to `running-configuration`.

Follow these steps:

1. Power on the switches, connect to the serial debug port, set the hostname, and configure a static IP address for management 1/1/1.

   The following code sample shows an S5232F-1 switch. The same process applies for S5232F-2 and S3048 switches.

   ```
   OS# configure terminal
   OS(config)# hostname S5232F-1
   S5232F-1(config)# interface mgmt 1/1/1
   S5232F-1(conf-if-ma-1/1/1)# no shutdown
   S5232F-1(conf-if-ma-1/1/1)# no ip address dhcp
   S5232F-1(conf-if-ma-1/1/1)# ip address 100.82.33.44/24
   S5232F-1(conf-if-ma-1/1/1)# exit
   S5232F-1(config)# management route 0.0.0.0/0 100.82.33.1
   ```

2. Copy the modified sample switch configuration to `running-configuration` and configure the switch:

   ```
   S5232F-1# copy scp://<user>@<hostip>/<path to downloaded S5232F config file> running-configuration
   S5232F-1# write memory
   ```

3. Repeat the preceding steps for switch S5232F-2 and S3048, ensuring that the appropriate switch configuration file is copied and IP addresses are modified accordingly.
This chapter presents the following topics:

- **Introduction**........................................................................................................... 13
- **Preparing the CSAH node**......................................................................................... 13
- **Preparing and running Ansible playbooks** ............................................................... 15
Chapter 3: Setting Up the CSAH Node

Introduction

This chapter describes the prerequisites for creating an OpenShift Container Platform cluster. Services required to create the cluster are set up in the CSAH. This chapter provides information about installing Red Hat Enterprise Linux 7.6 in the CSAH node and running the OpenShift Container Platform cluster prerequisites.

Preparing the CSAH node

To install Red Hat 7.6 in the CSAH node:

1. Follow the guidelines in the Red Hat Enterprise Linux 7 Installation Guide.
   In the Red Hat UI, ensure that, under Software Selection, Server with GUI is selected, as shown in the following figure:

   ![Figure 1. Operating system installation options in UI](image)

2. After the installation is complete, perform the following tasks as user root unless specified otherwise:
   a. Set the hostname to reflect the naming standards:
      
      ```
      [root@csah ~]# hostnamectl set-hostname <hostname>.<base domain>
      ```
Chapter 3: Setting Up the CSAH Node

b. Create a bonded interface, add secondary interfaces to it, and then assign an IP address to the bonded interface.

As part of our validation, we created bond0 using 100 Gb NIC ports in slot 2 (p2p1 and p2p2):

```
# Create bond0 interface
[root@csah ~]# nmcli connection add type bond con-name bond0 ifname bond0 bond.options "lacp_rate=1,miimon=100,mode=802.3ad,xmit_hash_policy=layer3+4"
# Add slaves to bond0 interface
[root@csah ~]# nmcli con add type ethernet ifname p2p1 master bond0
[root@csah ~]# nmcli con add type ethernet ifname p2p2 master bond0
# Set IP Address to bond0 interface
[root@csah ~]# nmcli connection modify bond0 ipv4.method manual ipv4.address <ipaddress/cidr> connection.autoconnect yes ipv4.gateway <gateway> ipv4.dns <dns-server> ipv4.dns-search <base domain>
ex: nmcli connection modify bond0 ipv4.method manual ipv4.addresses 100.82.46.20/26 connection.autoconnect yes ipv4.gateway 100.82.46.1 ipv4.dns 100.82.32.10 ipv4.dns-search example.com
# Bring up bond0 interface
[root@csah ~]# nmcli connection up bond0
```

**Note:** Check the `/proc/net/bonding/bond0` file and ensure that the secondary interfaces (p2p1 and p2p2) are listed along with the bonding configuration.

**Note:** The assigned IP address must be able to reach the Internet, and the DNS must be able to resolve `subscription.rhsm.redhat.com`.

c. Add the newly created hostname in the `/etc/hosts` file together with its IP address, as shown in the following output:

```
100.82.46.20  csah  csah.example.com
```

d. Enable the `rhel-7-server-ansible-2.9-rpms` repository using `subscription-manager`:

```
[root@csah ~]# subscription-manager register --username <subscription.user> --password <subscription.password> --force
[root@csah ~]# subscription-manager attach --pool=<pool id>
[root@csah ~]# subscription-manager repos --enable=rhel-7-server-ansible-2.9-rpms
```
Chapter 3: Setting Up the CSAH Node

e. Install the following RPMs:

```
[root@csah ~]# yum install -y git ansible
```

f. Create a user to run playbooks.

**Note:** Do not use the username `core`. User `core` is used as part of the OpenShift Container Platform cluster setup and is a predefined user in CoreOS.

```
[root@csah ~]# useradd <user>
```

**Note:** The remainder of this guide assumes that user `ansible` is created to run playbooks.

3. As user `ansible`, set up passwordless access to the CSAH FQDN:

```
[ansible@csah ~]$ ssh-keygen (press enter and go by defaults for the next set of questions)
[ansible@csah ~]$ cat .ssh/id_rsa.pub > .ssh/authorized_keys
[ansible@csah ~]$ chmod 600 .ssh/authorized_keys
```

4. As user `root`, provide permissions to the user that you just created (`ansible` in our example) to run all commands without being prompted for a password:

```
visudo
# add the following line after # %wheel ALL=(ALL) NOPASSWD: ALL
ansible ALL=(ALL) NOPASSWD: ALL
```

5. Download the Ansible playbooks from GitHub:

```
[ansible@csah ~]$ git clone https://github.com/dell-esg/openshift-bare-metal.git
```

### Preparing and running Ansible playbooks

As user `ansible`, unless otherwise specified, prepare and run Ansible playbooks as follows:

1. Run the following commands, which are prerequisites for running the Python scripts:

```
[ansible@csah python]$ sudo yum install python3
[ansible@csah python]$ sudo pip3 install pyyaml requests
```

2. To automatically create an inventory file for Ansible playbooks, run the Python scripts in the `<git clone directory>/containers/python directory>`:

```
[ansible@csah python]$ python3 generate_inventory_file.py
```
Select the number for each task in the following list and provide inputs:

![Figure 2. Inventory file generation input tasks menu](image)

**Note:** If in doubt, accept the default values, if any, that are listed for each option.

a. For option 1, specify the directory to which the files are to be downloaded:

   `- provide complete path of directory to download OCP 4.3 software bits
   `- default [/home/ansible/files]:

   Option 1 downloads OpenShift Container Platform 4.3 software from RedHat into a directory for which user `ansible` has permissions. The instructions in this document assume that the directory is specified as `/home/ansible/files`.

b. For option 2:

   i. Enter bootstrap node details including node names, the IP address assigned for bond0, and the iDRAC IP address and credentials.

   ii. Using iDRAC credentials, from the list of available network devices, select one interface whose MAC addresses are used by DHCP and PXE boot.

   **Note:** Ensure that the iDRAC IP address and credentials are accurate. If they are not accurate, an empty value `''` is set as the MAC address, which results in the failure of the Ansible playbooks. A manual change is then necessary to add MAC address to ensure that Ansible playbooks can run.

   `- enter the bootstrap node name
   `- default [bootstrap]:

   `- ip address for os in bootstrap node: 100.82.46.26
   `- ip address for idrac in bootstrap node: 100.82.34.26
   `- enter the idrac user for bootstrap: root
Chapter 3: Setting Up the CSAH Node

Dell EMC Ready Stack for Red Hat OpenShift Container Platform 4.3
Enabled by Dell EMC PowerEdge R-Series Servers and PowerSwitch Networking
Deployment Guide

enter idrac password for bootstrap:
1 -> NIC.Integrated.1-1-1
2 -> NIC.Integrated.1-2-1
3 -> NIC.Slot.2-1-1
4 -> NIC.Slot.2-2-1
Select the interface used by DHCP: 3

c. For options 3 and 4:
   i Add control plane and worker node details, including node names, the IP address that is assigned for bond0, and the iDRAC IP address and credentials.

   *Note:* You must provide the IP address for bond0, node names, and other details for each node one at a time.

   ii Using iDRAC credentials, from the list of available network devices, select one interface whose MAC addresses are used by DHCP and PXE boot.

d. For option 5, provide details about the bond name, interfaces used by the bond, and bond options.

   The bond options default value is based on best practices. Do not change it. You can change the primary interface name if it is different from the default value.

   Interface names are based on the slot in which the NIC is placed in the server. This document assumes that RHCOS is used for both control plane and worker nodes and that the slot 2 NIC ports are used. The remaining instructions assume that the interface names are ens2f0 and ens2f1.

   *Note:* Port enumeration in RHCOS is based on Red Hat Enterprise Linux 8 standards.

   enter bond name
default [bond0]:
   enter bond interfaces separated by '','
default [ens2f0,ens2f1]:
   enter bond options
default [mode=active-backup,miimon=100,primary=ens2f0]:

e. For option 6, provide details about the disks that are used in bootstrap, control plane, and worker nodes.

   *Note:* This document assumes that the nvme drive in the first slot is used for the OpenShift installation.

   ensure disknames are absolutely available. Otherwise OpenShift install fails
specify the master device that will be installed
default [nvme0n1]:
specify the bootstrap device that will be installed
default [nvme0n1]:
specify the worker device that will be installed
default [nvme0n1]:

f. For option 7, provide the cluster name and DNS zones file name:
specify zone file
default [/var/named/ocp.zones]:
specify cluster name
default [ocp]:

g. For option 8, provide details about the HTTP web server setup and directory names that will be created under /var/www/html:
enter http port
default [8080]:
specify dir where ignition files will be placed
directory will be created under /var/www/html
default [ignition]:
specify the version of ocp
default [4.3]:

h. For option 9, provide lease times for dhcpd configuration:
enter a default lease time for dhcp
default [800]:
enter max lease time for dhcp
default [7200]:

i. For option 10, provide details about the default user that will be used to install the OpenShift Container Platform cluster, service network CIDR, pod network CIDR, and other details that will be added in the install-config.yaml file:
enter the user used to install openshift
DONOT CHANGE THIS VALUE
default [core]:
enter the directory where openshift installs
directory will be created under /home/core
default [openshift]:
enter the pod network cidr
default [10.128.0.0/14]:
pod network cidr: 10.128.0.0/14
specify cidr notation for number of ips in each node:
cidr number should be an integer and less than 32
default [23]:
specify the service network cidr
default [172.30.0.0/16]:

Note: Do not change the user value from core. Only the core user is allowed to SSH into cluster nodes.
j. To print all the inputs that you have provided up to this point, select option 11.
   To modify any incorrect values, rerun the related option and correct the values.

k. Select option 12 to perform a YAML dump of all the displayed contents into a file called inventory_file in the current directory.
   To review a sample file, see <git clone directory>/containers/ansible/hosts.

4. Log in to Red Hat, download the pull secret file, and copy the contents into a file called pullsecret under the directory containing the OpenShift Container Platform 4.3 software bits.

   **Note:** You must have the appropriate Red Hat Customer Portal credentials to download the pull secret file.

   **Note:** This document uses the /home/ansible/files directory containing the software bits.

5. Edit inventory_file, adding the following contents under the software_src key, and save the file:

   ```yaml
   vars:
   software_src: /home/ansible/files
   pull_secret_file: pullsecret
   ```

   **Note:** Copy inventory_file from <git clone dir>/containers/python/ to <git clone dir>/containers/ansible/hosts.

6. As user ansible, run the playbooks:

   ```bash
   [ansible@csah ansible] $ pwd
   /home/ansible/containers/ansible
   [ansible@csah ansible] $ ansible-playbook -i hosts <git clone dir>/containers/ansible/ocp.yml
   ```

   The CSAH node is installed and configured with HTTP, DHCP, DNS, and PXE services. Also, the install-config.yaml file is generated, and the ignition config files are created and made available over HTTP.
This chapter presents the following topics:

- Introduction ........................................................................................................................................ 21
- Creating a bootstrap node .................................................................................................................. 21
- Installing control plane nodes ......................................................................................................... 22
- Installing worker nodes ................................................................................................................... 25
- Completing the cluster setup ........................................................................................................... 26
- Removing the bootstrap node ........................................................................................................... 27
- Validating and approving Certificate Signing Requests .................................................................... 27
- Validating the cluster operators ......................................................................................................... 29
- Accessing the OpenShift web console ............................................................................................... 30
- Configuring authentication .................................................................................................................. 30
- Configuring image registry storage .................................................................................................... 32
**Introduction**

To create an OpenShift Container Platform cluster, you must first create a bootstrap node, control plane nodes, and worker nodes in a specific order, as set out in this chapter.

---

**Note:** The remainder of the document assumes that NIC in Slot 2 Port 1 is used for PXE installation. Replace the interface, if necessary, according to your setup.

**Note:** All the nodes must be running in UEFI mode so that the playbooks that are run in CSAH node work effectively.

---

**Creating a bootstrap node**

Installation of the cluster begins with the creation of a bootstrap node. The bootstrap node is necessary to create the persistent control plane that is managed by the control plane nodes. After the initial minimum cluster with at least three control plane and two worker nodes is operational, you can remove the bootstrap node and convert it to a worker node if required.

To create a bootstrap node:

1. Connect to the iDRAC of the bootstrap node and open the virtual console.
2. From the iDRAC console, power on the bootstrap node.
3. Ensure that the ens2f0 interface is set for PXE boot:
   a. Press F2 to enter System Setup.
   b. Select Device Settings > NIC in Slot 2 Port 1 > NIC Configuration.
   c. From the Legacy Boot Protocol menu, select PXE.
   d. Select Finish to return to System Setup.
   e. Select System BIOS > Network Settings.
   f. Under UEFI PXE Settings, select Enable PXE Device and 1. Select PXE Device1 Settings.
   g. From the interface drop-down menu, select NIC in Slot 2 Port 1 Partition 1.
   h. Save your changes and reboot the node.

The system boots automatically into the PXE network and displays the PXE menu, as shown in the following figure:

![PXE menu]

**Figure 3. iDRAC console PXE menu**
4. Select bootstrap, and then perform the following steps after node installation is complete but before the node automatically reboots into the PXE:
   a. Press F2 to enter System Setup.
   b. Select System BIOS > Boot Settings > UEFI Boot Settings > UEFI Boot Sequence.
   c. Select PXE Device 1 and click -.
   d. Repeat the preceding step until PXE Device 1 is at the bottom of the boot menu.
   e. Click OK and then click Back.
   f. Click Finish and save the changes.

5. Let the node boot into the hard drive where the operating system is installed.

6. After the node comes up, check that the hostname in the iDRAC console is displayed as bootstrap.

![Figure 4. iDRAC console bootstrap node](image)

7. As user core in CSAH, run ssh bootstrap to ensure that the proper IP address is assigned to bond0.

8. From the CSAH node, as user core, ssh to the bootstrap node and verify that ports 6443 and 22623 are listening.

   Allow approximately 30 minutes for the ports to show up as listening. If the ports are not up and listening after that time, reinstall the bootstrap by repeating the preceding steps.

   ```
   [core@csah ~]$ ssh bootstrap sudo ss -tulpn | grep -E '6443|22623'
   tcp    LISTEN   0        128                     *:6443
   *:*     users:(("kube-etc\-signe",pid=9150,fd=3))
   tcp    LISTEN   0        128                     *:22623
   *:*     users:(("machine-config\",pid=8924,fd=5))
   ```

**Installing control plane nodes**

To install the control plane nodes:

1. Connect to the iDRAC of a control plane node and open the virtual console.
2. Power on the control plane node.
3. To ensure that the `ens2f0` interface is set for PXE boot:
   a. Press F2 to enter System Setup.
   b. Select Device Settings > NIC in Slot 2 Port 1 > NIC Configuration.
   c. From the Legacy Boot Protocol menu, select PXE.
   d. Select Finish to return to System Setup.
   e. Select System BIOS > Network Settings.
   f. Under UEFI PXE Settings, select Enable PXE Device and 1. Select PXE Device1 Settings.
   g. From the interface drop-down menu, select NIC in Slot2 Port1 Partition 1.
   h. Save your changes and reboot the node.

   The system automatically boots into the PXE network and displays the PXE menu, as shown in the following figure:

   ![Figure 5: iDRAC console PXE menu](image)

4. Select etcd-0 (the first node), and after installation is complete but before the node reboots into the PXE, ensure that the hard disk is placed above the PXE interface in the boot order, as follows:
   a. Press F2 to enter System Setup.
   b. Select System BIOS > Boot Settings > UEFI Boot Settings > UEFI Boot Sequence.
   c. Select PXE Device 1 and click -.
   d. Repeat the preceding step until PXE Device 1 is at the bottom of the boot menu.
   e. Click OK and then click Back.
   f. Click Finish and save the changes.

5. Let the node boot into the hard drive where the operating system is installed.
6. After the node comes up, ensure that the hostname is displayed as `etcd-0` in the iDRAC console.

![Red Hat Enterprise Linux CoreOS 43.81.202001142154.0 (Dotpa) 4.3](image)

**Figure 6.** Control plane (etcd-0) iDRAC console

7. As user `core` in CSAH, run `ssh etcd-0` to ensure that the proper IP address is assigned to bond0.

After the installation is complete, the node reboots to fetch the control plane node configuration file.

8. Repeat the preceding steps for the remaining two control plane nodes, selecting `etcd-1` for the second control plane node and `etcd-2` for the third control plane node.

When all three control plane (etcd-*) nodes are installed and running, port 22623 listens in control plane nodes. When worker nodes are installed, control plane nodes provision the worker nodes.

Similarly, port 6443 listens in control plane (etcd-*) nodes.

```
[core@csah ~]$ for masternode in etcd-0 etcd-1 etcd-2;do
echo $masternode;ssh $masternode sudo ss -tulpn | grep -E '6443|22623';done
etcd-0
tcp    LISTEN   0        128                     *:6443
*:6443 users:(("hyperkube",pid=29503,fd=5))
tcp    LISTEN   0        128                     *:22623
*:22623 users:(("machine-config",pid=10931,fd=6))
etcd-1
tcp    LISTEN   0        128                     *:6443
*:6443 users:(("hyperkube",pid=35122,fd=5))
tcp    LISTEN   0        128                     *:22623
*:22623 users:(("machine-config",pid=12743,fd=6))
etcd-2
tcp    LISTEN   0        128                     *:6443
*:6443 users:(("hyperkube",pid=18445,fd=5))
tcp    LISTEN   0        128                     *:22623
*:22623 users:(("machine-config",pid=9853,fd=6))
```
Installing worker nodes

To install worker nodes:

1. Connect to the iDRAC of a worker node and open the virtual console.
2. Power on the worker node.
3. To ensure that the `ens2f0` interface is set for PXE boot:
   a. Press F2 to enter System Setup.
   b. Select Device Settings > NIC in Slot 2 Port 1 > NIC Configuration.
   c. From the Legacy Boot Protocol menu, select PXE.
   d. Select Finish to return to System Setup.
   e. Select System BIOS > Network Settings.
   f. Under UEFI PXE Settings, select Enable PXE Device and 1. Select PXE Device1 Settings.
   g. From the interface drop-down menu, select NIC in Slot2 Port1 Partition 1.
   h. Save your changes and reboot the node.
      The system automatically boots into the PXE network, as shown in the following figure:

![image]

Figure 7. iDRAC console PXE menu

4. Select worker-0 (the first worker node), and let the system reboot after the installation; before the node reboots into the PXE, ensure that the hard disk is placed above the PXE interface in the boot order, as follows:
   a. Press F2 to enter System Setup.
   b. Select System BIOS > Boot Settings > UEFI Boot Settings > UEFI Boot Sequence.
   c. Select PXE Device 1 and click −.
   d. Repeat the preceding step until PXE Device 1 is at the bottom of the boot menu.
   e. Click OK and then click Back.
   f. Click Finish and save the changes.
5. Let the node boot into the hard drive where the operating system is installed.
6. After the node comes up, ensure that the hostname is displayed as `worker-0` in the iDRAC console.

![Worker (worker-0) iDRAC console](image)

7. As user `core` in CSAH, run `ssh worker-0` to ensure that the proper IP address is assigned to bond0.

After the installation is complete, the node reboots to fetch the worker configuration file from the master nodes.

8. Repeat the preceding steps for the second worker node, but, in step 4, select `worker-1` from the PXE menu.

### Completing the cluster setup

**Note:** This section uses `openshift` for the `install_dir` variable. See the inventory file for the value specified for the `install_dir` variable.

After the bootstrap, control plane, and worker nodes are installed, as user `core`, switch to the home directory of `core` and run the following command:

```bash
[core@csah ~]$ cd /home/core
[core@csah ~]$ ./openshift-install --dir=openshift wait-for bootstrap-complete --log-level debug
DEBUG OpenShift Installer v4.3.1
DEBUG Built from commit 2055609f95b19322ee6cfdd0bea73399297c4a3e
INFO Waiting up to 30m0s for the Kubernetes API at https://api.ocp.example.com:6443...
INFO API v1.16.2 up
INFO Waiting up to 30m0s for bootstrapping to complete...
DEBUG Bootstrap status: complete
INFO It is now safe to remove the bootstrap resources
```

You may now reuse the bootstrap node as a worker node.
Removing the bootstrap node

We created a bootstrap node as part of the deployment procedure. You can remove this node now because the OpenShift Container Platform cluster is running.

To remove the bootstrap node:

1. Remove the bootstrap node entries along with the names, IP addresses, and MAC addresses.
   
   For example, in the following sample entry for bootstrap_node in the inventory file, remove all entries along with the bootstrap_node line:

   ```bash
   [ansible@csah ansible]$ cat hosts | grep -A 3 -i bootstrap_node
   bootstrap_node:
     - name: bootstrap
       ip: 100.82.46.26
       mac: B8:59:9F:C0:35:86
   ```

2. Run playbooks as user ansible in the CSAH node.

   ```bash
   [ansible@csah ansible]$ ansible-playbook -i hosts ocp.yml
   ```

Validating and approving Certificate Signing Requests

Validate and approve Certificate Signing Requests (CSRs) as follows. Unless otherwise specified, run the commands as user core.

1. Ensure that all the control plane nodes are visible and in Ready status.

   ```bash
   [core@csah ~]$ oc get nodes
   NAME                        STATUS ROLES    AGE    VERSION
   etcd-0.example.com          Ready master 52m    v1.16.2
   etcd-1.example.com          Ready master 52m    v1.16.2
   etcd-2.example.com          Ready master 52m    v1.16.2
   ```

2. Check for any pending CSRs:

   ```bash
   [core@csah ~]$ oc get csr
   NAME        AGE REQUESTOR       CONDITION
   csr-2hkn9    42m system:node:etcd-1.example.com m
   csr-9gdsn    15m system:serviceaccount:openshift-machine-config-operator:node-bootstrapper Pending
   csr-cfl81    12m system:serviceaccount:openshift-machine-config-operator:node-bootstrapper Pending
   csr-f7lf4    30m system:serviceaccount:openshift-machine-config-operator:node-bootstrapper Pending
   csr-mk6kk    42m system:serviceaccount:openshift-machine-config-operator:node-bootstrapper Approved,Issued
   ```
3. **Approve all pending CSRs as follows:**

   ```
   [core@csah ~]$ oc get csr | grep -i pending | cut -f 1 -d ' ' | xargs -n 1 oc adm certificate approve certificatesigningrequest.certificates.k8s.io/csr
   approved
   certificatesigningrequest.certificates.k8s.io/csr
   approved
   certificatesigningrequest.certificates.k8s.io/csr
   approved
   certificatesigningrequest.certificates.k8s.io/csr
   approved
   ```

4. **Ensure that the worker node certificates show as pending and approve them by running the command specified in step 3.**

   ```
   [core@csah ~]$ oc get csr | grep -i pending
   csr-8zxkk 52s system:node:worker-0.example.com Pending
   csr-qjztw 53s system:node:worker-1.example.com Pending
   ```

   ```
   [core@csah ~]$ oc get csr | grep -i pending | cut -f 1 -d ' ' | xargs -n 1 oc adm certificate approve certificatesigningrequest.certificates.k8s.io/csr
   approved
   certificatesigningrequest.certificates.k8s.io/csr
   approved
   ```

5. **Ensure that the worker nodes are part of the cluster nodes:**

   ```
   [core@csah ~]$ oc get nodes
   NAME                   STATUS   ROLES    AGE     VERSION
   etcd-0.example.com     Ready    master   56m     v1.16.2
   etcd-1.example.com     Ready    master   56m     v1.16.2
   etcd-2.example.com     Ready    master   56m     v1.16.2
   worker-0.example.com   Ready    worker   4m50s   v1.16.2
   worker-1.example.com   Ready    worker   4m51s   v1.16.2
   ```
Validating the cluster operators

The OpenShift Container Platform cluster consists of multiple cluster operators. Ensure that all cluster operators are available:

1. Verify that the AVAILABLE column displays True for all the cluster operators:

```bash
[core@csah ~]$ oc get clusteroperators
NAME            VERSION    AVAILABLE   PROGRESSING   DEGRADED   SINCE
authentication   4.3.1       True        False        False      10h
cloud-credential 4.3.1       True        False        False      16h
cluster-autoscaler4.3.1 True        False        False      19m
console          4.3.1       True        False        False      10m
dns              4.3.1       True        False        False      16m
ingress          4.3.1       True        False        False      11m
insights         4.3.1       True        False        False      16m
kube-api-server  4.3.1       True        False        False      16m
kube-controller-manager4.3.1 True        False        False      16m
kube-scheduler    4.3.1       True        False        False      16m
machine-api      4.3.1       True        False        False      16m
machine-config   4.3.1       True        False        False      16m
marketplace       4.3.1       True        False        False      16m
monitoring       4.3.1       True        False        False      16m
network          4.3.1       True        False        False      16m
node-tuning      4.3.1       True        False        False      16m
openshift-api-server4.3.1 True        False        False      16m
openshift-controller-manager4.3.1 True        False        False      16m
openshift-samples4.3.1 True        False        False      16m
operator-lifecycle-manager 4.3.1 True        False        False      16m
operator-lifecycle-manager-catalog4.3.1 True        False        False      16m
operator-lifecycle-manager-package-server4.3.1 True        False        False      16m
openshift-cs    4.3.1       True        False        False      16m
service-catalog-api-server4.3.1 True        False        False      16m
service-catalog-controller-manager4.3.1 True        False        False      16m
storage          4.3.1       True        False        False      15m
```

Figure 9. Cluster operators in OpenShift Container Platform 4.3 cluster

2. Complete the cluster installation:

```bash
[core@csah ~]$ pwd
/home/core

[core@csah ~]$ ./openshift-install --dir=openshift wait-for install-complete --log-level debug

DEBUG OpenShift Installer v4.3.1
DEBUG Built from commit 2055609f95b19322ee6cf60bea73399297c4a3e
INFO Waiting up to 30m0s for the cluster at https://api.ocp.example.com:6443 to initialize...
DEBUG Cluster is initialized
INFO Waiting up to 10m0s for the openshift-console route to be created...
DEBUG Route found in openshift-console namespace: console
DEBUG Route found in openshift-console namespace: downloads
DEBUG OpenShift console route is created
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export KUBECONFIG=/home/core/openshift/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-console.apps.ocp.example.com
```
INFO Login to the console with user: kubeadmin, password: xxxxx-xxxxx-xxxxx-xxxxx-xxxx

Accessing the OpenShift web console

The OpenShift web console provides access to all cluster functionality, including POD creation and application deployment.

To access OpenShift through a web browser:

1. Obtain the routes console URL.
2. Obtain the existing routes in all namespaces:

```bash
[core@csah ~]$ oc get routes --all-namespaces | grep -i console-openshift
console-openshift-console console openshift-console-openshift-console.apps.ocp.example.com
console https reencrypt/Redirect None
```

Note: The URL in the openshift-console namespace is console-openshift-console.apps.ocp.example.com.

3. Open a web browser and paste the URL.
4. Log in as kubeadmin, using the password that was saved in /home/core/<install dir>/auth/kubeadmin-password.

Configuring authentication

Introduction

OpenShift supports different authentication methods based on the Identity provider. For more information about supported authentication providers, see Understanding authentication in the OpenShift Container Platform documentation.

This section describes how to configure identity providers by using htpasswd.

Creating an admin user

Create an admin user as follows. Unless otherwise specified, run the commands in this section in CSAH node as user core.

1. Create an htpasswd file on the CSAH node:

```bash
[core@csah ~]$ cd /home/core/<install directory>/

[core@csah openshift]$ htpasswd -c -B -b htpasswd ocpadmin Password1

[core@csah openshift]$ htpasswd -b htpasswd ocpuser Password2
```
2. **Create a secret for** htpasswd:

   [core@csah openshift]$ oc create secret generic htpass-secret --from-file=htpasswd=/home/core/openshift/htpasswd -n openshift-config

3. **Create a custom resource (CR), and save the following contents in a file:**

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

4. **Apply the CR:**

   [core@csah ~]$ oc apply -f <file name>

5. **Log in as a user created with** htpasswd:

   [core@csah ~]$ oc login -u <username>
   Authentication required for https://api.ocp.example.com:6443 (openshift)
   Username: <username>
   Password: <password>
   Login successful.
   You don't have any projects. You can try to create a new project, by running `oc new-project <projectname>`

---

**Assigning a cluster-admin role to a user**

1. **Log in as kubeadmin to assign cluster-admin access:**

   [core@csah openshift]$ oc login -u kubeadmin -p xxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx Login successful.
   You have access to 53 projects, the list has been suppressed. You can list all projects with 'oc projects'
   Using project "default".

2. **Run the following command and ensure that the user is listed:**

   [core@csah ~]$ oc get users
   NAME         UID                                    FULL NAME
   IDENTITIES
   ocpadmin 273ccf25-9b32-4b4d-aad4-503c5aa27eee
   htpasswd:ocpadmin
3. Get a list of all available cluster roles:
   
   ```
   oc get clusterrole --all-namespaces
   ```

4. Assign the cluster-admin role to the user ocpadmin as follows:
   
   ```
   [core@csah ~]$ oc adm policy add-cluster-role-to-user
   cluster-admin ocpadmin
   clusterrole.rbac.authorization.k8s.io/cluster-admin added:
   "ocpadmin"
   ```

## Configuring image registry storage

### Introduction

By default, OpenShift Container Platform 4.3 does not have any storage configured for the image registry, and the image registry operator is in a nonmanaged (Removed) state. Follow the guidelines in this section to configure image registry storage.

The details in this document are based on the use of a Dell EMC Unity XT380F-based NAS server to host the image registry.

Consider the following information:

- Dell Technologies recommends using the Dell EMC Unity 380 All Flash (380F) array for image registry storage.
- Red Hat does not recommend using the NFS server that it ships with Red Hat Enterprise Linux for image registry storage, although using Linux-backed NFS shares is possible for proof-of-concept (POC) implementations.
- Although POC implementations can use emptyDir for image registry storage, images pushed to the image registry are not saved following a reboot.

---

**Note:** At the time that this document was written, the Dell EMC Unity CSI driver was not yet available. We used generic drivers of iSCSI and NFS to provision storage.

### NFS-based image registry

The prerequisites for creating the image registry storage are as follows:

- The Unity array must be configured and accessible through the management IP address.
- The NAS server must be created and reachable from the OpenShift Container Platform cluster.
- The disk pool must be created.

### NAS server and NFS share details

The NAS server and NFS share details are as follows:

- NAS server name—`unitynas`
- File system name—`unitynfs`
- NFS share path—`100.82.47.8:/unitynfs`
- Default access—`RW, Allow root`
To provision storage for the image registry, complete the following steps in the Unity array. Run the commands as user `core` in CSAH unless otherwise specified.

1. Ensure that the environment meets the prerequisites that are listed in the preceding section.

2. Create a persistent volume (PV) file `nfspv.yml`, using following content:

   ```yaml
   [core@csah ~]$ cat nfspv.yaml
   apiVersion: v1
   kind: PersistentVolume
   metadata:
     name: nfs-image-registry
     namespace: openshift-image-registry
   spec:
     capacity:
       storage: 100Gi
     accessModes:
       - ReadWriteMany
     nfs:
       path: /unitynfs
       server: 100.82.47.8
     persistentVolumeReclaimPolicy: Retain
   
   3. Create the following PV:

   ```bash
   [core@csah ~]$ oc create -f nfsimageregpv.yml
   ```

4. Create a persistent volume claim (PVC) YAML file, using the following content:

   ```yaml
   [core@csah ~]$ cat nfspvc.yaml
   apiVersion: v1
   kind: PersistentVolumeClaim
   metadata:
     name: nfspvc
     namespace: openshift-image-registry
   spec:
     accessModes:
       - ReadWriteMany
     resources:
       requests:
         storage: 100G
   
   5. Create a PVC:

   ```bash
   [core@csah ~]$ oc create -f nfspvc.yaml
   persistentvolumeclaim/nfspvc created
   
   6. Edit the registry configuration to use the PV:

   ```bash
   [core@csah ~]$ oc edit configs.imageregistry.operator.openshift.io
   ```
7. In the YAML output, under `spec`, change `managementState` from the default value, `Removed`, to `Managed`, and add the PVC name as the `storage` key, as follows.

```
spec:
  managementState: Managed
  storage:
    pvc:
      claim: nfspvc
```

**Note:** To use the storage that was created in the Unity array, ensure that the `managementState` value is set to `Managed`.

8. Type `:wq` to save the changes.

```
[core@csah ~]$ oc edit
configs.imageregistry.operator.openshift.io
config.imageregistry.operator.openshift.io/cluster
```

Operators such as `image-registry`, `apiserver`, and so on are in a PROGRESSING True state for few minutes before they become AVAILABLE True.

**Note:** Receiving a permissions error during an attempt to push an image to an NFS-based image registry is a known issue. For more information, see the Knowledgebase in the Red Hat Customer Portal.

**Note:** Perform these steps regardless of whether you use NFS-based or iSCSI-based Unity storage.

Validate the image registry as follows:

1. Verify that the AVAILABLE column displays True for all the cluster operators:

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>AVAILABLE</th>
<th>PROGRESSING</th>
<th>DEGRADED</th>
<th>SINCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>authn</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>cloud-credential</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>cluster-autoscaler</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>console</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>image-registry</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>231m</td>
</tr>
<tr>
<td>ingress</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>ingress-tcp</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>kube-apiserver</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>kube-controller-manager</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>kube-scheduler</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>machine-api</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>machine-config</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>marketplace</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>monitoring</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>network</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>node-tuning</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>openshift-apiserver</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>openshift-controller-manager</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>openshift-samples</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>operator-lifecycle-manager</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>operator-lifecycle-manager-catalog</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>operator-lifecycle-manager-package-server</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>249h</td>
</tr>
<tr>
<td>service-catalog</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>service-catalog-controller</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
<tr>
<td>storage</td>
<td>4.3.1</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3d3h</td>
</tr>
</tbody>
</table>

**Figure 10.** Cluster operators status check
2. Ensure that the image registry pods are all in the Running state, as shown here:

![Image Registry Pod Status](image)

Figure 11. Image registry pod status

3. Verify that the registry storage ClaimName that is used for the image registry pod in the preceding output matches the PVC name:

   [core@csah ~]$ oc describe pod image-registry-54d58569dc-tlvr4 -n openshift-image-registry | grep -i volumes -A 4
   
   Volumes:
   registry-storage:
     Type: PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
     ClaimName: nfspvc
     ReadOnly: false

4. Connect to any control plane or worker node in the OpenShift Container Platform cluster by running the following command as user core in CSAH node:

   [core@csah ~]$ oc debug nodes/etcd-0.example.com
   Starting pod/etcd-0examplecom-debug ...
   To use host binaries, run `chroot /host`
   Pod IP: 100.82.46.21
   If you don't see a command prompt, try pressing enter.
   sh-4.2# chroot /host

5. Log in as kubeadmin:

   sh-4.4# oc login -u kubeadmin -p xxxxxxxx
   Login successful.
   You have access to 53 projects, the list has been suppressed. You can list all projects with 'oc projects'
   Using project "default".

6. Test the connection to the image registry service listening in port 5000:

   sh-4.4# podman login -u kubeadmin -p $(oc whoami -t) image-registry.openshift-image-registry.svc:5000
   Login Succeeded!
Chapter 5 Adding Worker Nodes

This chapter presents the following topics:

Adding a Red Hat Enterprise Linux worker node .............................................. 37
Adding Red Hat Enterprise Linux CoreOS worker node .................................. 41
Adding a Red Hat Enterprise Linux worker node

**Introduction**

Worker nodes in OpenShift Container Platform 4.3 can run either CoreOS or Red Hat Enterprise Linux 7.6 or later. This section describes how to add a worker node that is running Red Hat Enterprise Linux 7.6 or later.

**Prerequisites**

The prerequisites for adding a worker node are as follows:

- Red Hat Enterprise Linux 7.6 must be installed and assigned an IP address.
- The node must be subscribed to the following repositories:
  - rhel-7-server-rpms
  - rhel-7-extras-rpms
  - rhel-7-server-ose-4.3-rpms

The Red Hat Enterprise Linux worker node details are as follows:

- The hostname is set to worker-2.example.com.
- The IP address is assigned 100.82.46.26/26 (bond0), and DNS is 100.82.46.20 (CSAH node IP address).

**Adding the Red Hat Enterprise Linux worker node**

To add the worker node, complete the following steps. Run the commands as user root in CSAH unless otherwise specified.

1. Update the subscription repositories:
   
   ```
   [root@csah ~]# subscription-manager repos --enable="rhel-7-server-extras-rpms" --enable="rhel-7-server-ose-4.3-rpms"
   Repository 'rhel-7-server-os-rpms' is enabled for this system.
   Repository 'rhel-7-server-extras-rpms' is enabled for this system.
   ```

2. Install the openshift-ansible and openshift-clients RPMs:
   
   ```
   [root@csah ~]# yum install openshift-ansible openshift-clients
   ```

3. Copy kubeconfig to the user ansible home directory:
   
   ```
   [root@csah ~]# cp /home/core/openshift/auth/kubeconfig /home/ansible/
   [root@csah ~]# chown ansible:ansible /home/ansible/kubeconfig
   ```

4. Update the DNS entries for forward lookup (A and CNAME records) and reverse lookup of the Red Hat Enterprise Linux worker node as follows.

   **Note:** DNS files are located under the /var/named directory. The forward lookup file is example.com, and the reverse lookup file is 46.82.100.in-addr.arpa.

   ```
   [root@csah named]# pwd
   ```
5. Restart the named service and validate the DNS entries:

[root@csah named]# systemctl restart named
[...]
[root@csah named]# nslookup worker-2
Server: 100.82.46.20
Address: 100.82.46.20#53
Name: worker-2.example.com
Address: 100.82.46.26

6. As user `ansible`, create an inventory file with the Red Hat Enterprise Linux worker node information:

[ansible@csah ~]$ cat rhelworker
[all:vars]
ansible_user=ansible
ansible_become=True
openshift_kubeconfig_path="/home/ansible/kubeconfig"
[new_workers]
worker-2.example.com

Note: Run the commands in the remaining steps as user root in Red Hat Enterprise Linux worker node unless specified otherwise.

7. Use subscription manager to add repositories:

[root@worker-2 ~]# subscription-manager repos --enable="rhel-7-server-extras-rpms" --enable="rhel-7-server-rpms" --enable="rhel-7-server-ose-4.3-rpms"
Chapter 5: Adding Worker Nodes

Repository 'rhel-7-server-extras-rpms' is enabled for this system.
Repository 'rhel-7-server-rpms' is enabled for this system.
Repository 'rhel-7-server-ose-4.3-rpms' is enabled for this system.

8. Disable the firewall:

   [root@worker-2 ~]# systemctl disable firewalld
   Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
   Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
   [root@worker-2 ~]# systemctl stop firewalld

   **Note:** Do not enable firewalld later. If you do, you cannot access OpenShift Container Platform logs on the worker node.

9. Create user ansible:

   [root@worker-2 ~]# useradd ansible

10. Provide sudoers access to user ansible, and add the following content after running the visudo command:

    [root@worker-2 ~]# visudo
    root    ALL=(ALL) ALL (This line already exists)
    ansible ALL=(ALL) NOPASSWD: ALL

11. Copy the contents of /home/ansible/.ssh/id_rsa.pub in CSAH node to /home/ansible/.ssh/authorized_keys in the Red Hat Enterprise Linux worker node:

    [ansible@csah ~]$ mkdir .ssh
    [ansible@csah ~]$ vi authorized_keys

12. Set permissions of 700 to /home/ansible/.ssh and 600 to /home/ansible/.ssh/authorization_keys:

    [ansible@worker-2 ~]$ chmod 700 .ssh/
    [ansible@worker-2 ~]$ chmod 600 .ssh/authorized_keys

13. As user ansible in CSAH node, copy kubeconfig to the Red Hat Enterprise Linux worker node:

    [ansible@csah ~]$ scp kubeconfig worker-2:~/

   100% 17KB 44.4MB/s 00:00

   **Note:** You are not prompted for a password because the authorized keys from CSAH have already been added.

As user ansible in CSAH node, you must list all existing nodes in the cluster and, in some cases, might receive the following error:
Chapter 5: Adding Worker Nodes

[ansible@csah tasks]$ oc get nodes --config=/home/ansible/.auth/kubeconfig
error: You must be logged in to the server (Unauthorized)

If you receive the login error when logging in as ansible, log in as kubeadmin:

[ansible@csah ~]$ oc login -u kubeadmin -p xxxxx-xxxxxx-xxxxxx-xxxxxx
Login successful.
You have access to 53 projects, the list has been suppressed. You can list all projects with 'oc projects'
Using project "default".

14. As user ansible, run ansible-playbook in CSAH node, which adds the new worker node into an existing cluster:

[ansible@csah ~]$ cd /usr/share/ansible/openshift-ansible/
[ansible@csah openshift-ansible]$ ansible-playbook -i /home/ansible/rhelworker playbooks/scaleup.yml

The following figure shows the task summary and execution times:

Figure 12. Red Hat Enterprise Linux worker node Ansible task summary

15. Edit haproxy.cfg under /etc/haproxy to include the new worker node IP address under sections backend http and backend https:

backend http
   balance roundrobin
   mode tcp
   server worker-0 100.82.46.24:80 check
   server worker-1 100.82.46.25:80 check
   server worker-2 100.82.46.26:80 check

backend https
   balance roundrobin
   mode tcp
   server worker-0 100.82.46.24:443 check
   server worker-1 100.82.46.25:443 check
   server worker-2 100.82.46.26:443 check

The line server worker-2 is now in haproxy.cfg. The output that is shown here is truncated for documentation purposes.
16. As user root, restart the haproxy service:

   [root@csah named]# systemctl restart haproxy

17. Verify that the worker node (worker-2.example.com) is now part of the cluster:

   [core@csah ~]$ oc get nodes

   NAME                   STATUS   ROLES    AGE    VERSION
   etcd-0.example.com     Ready    master   4d2h   v1.16.2
   etcd-1.example.com     Ready    master   4d2h   v1.16.2
   etcd-2.example.com     Ready    master   4d2h   v1.16.2
   worker-0.example.com   Ready    worker   4d1h   v1.16.2
   worker-1.example.com   Ready    worker   4d1h   v1.16.2
   worker-2.example.com   Ready    worker   89m    v1.16.2

Adding Red Hat Enterprise Linux CoreOS worker node

Introduction

Worker nodes in OpenShift Container Platform 4.3 can run either Red Hat Enterprise CoreOS (RHCOS) or Red Hat Enterprise Linux 7.6. This section describes how to add a worker node that is running RHCOS.

Adding the CoreOS worker node

To add worker nodes that are running RHCOS, use Ansible playbooks that were initially used to create worker nodes. Follow these steps:

1. Edit the Ansible inventory file that you generated in Preparing and running Ansible playbooks, and add information about the new worker node.

   Note: You cannot use a Python script to update an existing inventory file. Manually add the worker node as an item in the list with the keys name, ip, and mac. Update the value of the key number_of_workers in the inventory file.

2. As user ansible in CSAH node, run the playbooks:

   [ansible@csah ~]$ ansible-playbook -i <updated inventory file path> <git clone dir>/containers/ansible/ocp.yml

   Note: Re-add any manual entries in DNS, HAProxy for Red Hat Enterprise Linux worker nodes for both forward lookup and reverse lookup. The playbook updates the entries based on the inventory files and deletes any other entries. See steps 4, 5, 16, and 17 in Adding a Red Hat Enterprise Linux worker node.

3. Boot the worker node and ensure that it has been added to the cluster.

   For instructions, see Installing worker nodes.

4. Accept the new worker node certificate and, as user core in CSAH, run oc get nodes and verify that the node is visible.

   Follow the guidelines in Validating and approving Certificate Signing Requests.
This chapter presents the following topics:

**Deploying applications** ................................................................. 43

**Scaling applications** .................................................................... 45
Deploying applications

Introduction

You can use multiple methods to deploy applications in an OpenShift cluster. This guide provides some examples. For more information, see Creating applications using the Developer perspective in the OpenShift Container Platform documentation.

Deploying application images

OpenShift supports application deployment using an image that is stored in an external image registry. Images have the necessary packages and program tools to run the applications by default.

To deploy an application that is already part of an image, complete the following steps. Unless specified otherwise, run all the commands as user core in CSAH.

1. Log in to the OpenShift cluster:
   
   [core@csah ~]$ oc login -u <username>

2. Create a project:
   
   [core@csah ~]$ oc new-project <project name>

3. Create an application:
   
   [core@csah ~]$ oc new-app <image-name>

   This guide uses openshift/hello-openshift for the image name that is being tested.

4. After the image is deployed, identify all the objects that are created as part of the deployment by running the oc get all command, as shown in the following figure:

   ![Figure 13. Sample application deployment status](image)

Deploying S2I

OpenShift supports application deployment by using a source from GitHub and specifying an image. One good application example is Source-to-Image (S2I), a toolkit and workflow for building reproducible container images from source code. A build configuration is generated for the S2I deployment in a new pod called Build Pod. In the build configuration, you can configure the triggers that are required to automate the new build process every time a condition meets the specifications that you defined. After the deployment is complete, a new image with injected source code is created automatically.

Follow these steps to deploy an application using a source from GitHub. The source in the sample deployment is httpd-ex.
1. Log in to the OpenShift cluster:
   [core@csah ~]$ oc login -u <user name>

2. Create a project:
   [core@csah ~]$ oc new-project <project name>

3. Create an application by using the GitHub source and specifying the image of which the application will be a part:
   [core@csah ~]$ oc new-app centos/httpd-24-centos7~https://github.com/sclorg/httpd-ex.git

   **Note:** The image is centos/httpd-24-centos7. The GitHub source is https://github.com/sclorg/httpd-ex.git. You can obtain build logs by running oc logs -f bc/httpd-ex for this example.

4. After the image is deployed, identify all the objects that were created as part of the deployment by running the command `oc get all`:

   ![Sample S2I deployment status](image)

   **Figure 14. Sample S2I deployment status**

5. Obtain triggers for this deployment by checking the YAML template of the build configuration:
   [core@csah ~]$ oc get buildconfig httpd-ex -o yaml

---

### Accessing applications from an external network

You can access applications that are deployed within the OpenShift cluster using images or source code from GitHub by using the service IP address that is associated with the deployments. External access to the applications is not available by default.

To enable access to the applications from an external network:

1. Log in to the OpenShift cluster:
   [core@csah ~]$ oc login -u <user name>

2. Switch to the project under which the application is running:
   [core@csah ~]$ oc project sample
   Now using project "sample" on server "https://api.ocp.example.com:6443".

3. Identify the service that is associated with the application.

   **Note:** Typically, the name of the service is the same as the name of the deployment.
Chapt[368x746]er 6
Deploying and Scaling Applications

Dell EMC Ready Stack for Red Hat OpenShift Container Platform 4.3
Enabled by Dell EMC PowerEdge R-Series Servers and PowerSwitch Networking

Deployment Guide

[core@csah ~]$ oc get svc
NAME          TYPE       CLUSTER-IP     EXTERNAL-IP
PORT(S)       AGE
hello-openshift  ClusterIP  172.30.93.229   <none>
8080/TCP,8888/TCP  23m

4. Expose the route for service of your application:

[core@csah yaml]$ oc expose svc/hello-openshift
route.route.openshift.io/hello-openshift exposed

5. Obtain the routes that were created:

[core@csah ~]$ oc get routes
NAME          HOST/PORT
PATH          SERVICES          PORT       TERMINATION   WILDCARD
hello-openshift sample.apps.ocp.example.com          hello-openshift  8080-tcp
None

6. Open a web browser, enter hello-openshift-
sample.apps.ocp.example.com, and press Enter.

7. Repeat the preceding steps to expose the service for S2I deployment.

Scaling applications

Applications are designed and created to meet the demands of customers and can be
scaled up or down based on business needs.

Scale an application as follows. This example uses hello-openshift.

1. Log in to the OpenShift cluster:

[core@csah ~]$ oc login -u <user name>

2. Switch to the project under which the application is running:

[core@csah ~]$ oc project sample
Now using project "sample" on server
"https://api.ocp.example.com:6443".

3. Identify the deployment configuration that is associated with the application:

[core@csah ~]$ oc get dc
NAME          REVISION DESIRED CURRENT TRIGGERED
BY
hello-openshift   1      3     3
config,image(hello-openshift:latest)

4. Increase the DESIRED count to 3:

[core@csah ~]$ oc scale --replicas=3 dc/hello-openshift
deploymentconfig.apps.openshift.io/hello-openshift scaled

[core@csah ~]$ oc get dc
### Chapter 6: Deploying and Scaling Applications

<table>
<thead>
<tr>
<th>NAME</th>
<th>REVISION</th>
<th>DESIRED</th>
<th>CURRENT</th>
<th>TRIGGERED BY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello-openshift</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>config,image(hello-openshift:latest)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** OpenShift supports the autoscaling of pods if cluster metrics are installed. Run `oc autoscale dc/hello-openshift --min=1 --min=10 --cpu-percent=80`. This feature is under Technology Preview. For more information about enabling cluster metrics for pods, see [Exposing custom application metrics for autoscaling](https://docs.openshift.com/container-platform/4.3/monitoring_and_lightning反腐/autoscaling.html) in the OpenShift Container Platform documentation.
Chapter 7  Provisioning Storage

This chapter presents the following topics:

Introduction ..................................................................................................................................................................... 48
Prerequisites for using Unity storage .......................................................................................................................... 48
Adding worker nodes in Unity hosts ............................................................................................................................ 48
Creating a LUN and mapping it to the worker nodes ................................................................................................ 49
Creating a pod using Unity storage ............................................................................................................................ 50
Chapter 7: Provisioning Storage

Introduction

OpenShift Container Platform cluster administrators can map storage to containers. For a list of supported PV plug-ins, see Types of PVs in the OpenShift Container Platform documentation. This chapter describes how to use Dell EMC Unity iSCSI storage to create a PV, claim the PV, and map the storage claim to the pod.

Prerequisites for using Unity storage

The prerequisites for using Unity iSCSI storage for PVs are as follows:

- The Unity array must be set up with iSCSI interfaces and pools.
- You must know the Unity iSCSI interface IP address and iSCSI Qualified Name (IQN).
- Network connectivity must exist between the Unity array and the CSAH node.

Adding worker nodes in Unity hosts

To provide LUN access to the worker nodes, add the worker nodes in the Unity hosts:

1. Log in to the worker nodes as user core from CSAH and obtain the IQN that is specified under /etc/iscsi/initiatorname.iscsi.

   [core@worker-0 ~]$ cat /etc/iscsi/initiatorname.iscsi
   InitiatorName=iqn.1994-05.com.redhat:877f6bdd1b

   [core@worker-1 ~]$ cat /etc/iscsi/initiatorname.iscsi
   InitiatorName=iqn.1994-05.com.redhat:fc795ced1fc3

2. Log in to Dell EMC Unisphere storage management using the Unity management IP address.

3. Under Access, select Hosts, click the plus sign (+), and select Host.

4. Enter the details that are shown in the following figure, and then click Next.

   ![Add a Host](image)

   Figure 15. Adding the worker node in Unisphere

   5. Under Manually Added Initiators, click the plus sign (+) and select Create iSCSI Initiator.
6. Provide the IQN of worker-0, and then click Add.

**Add iSCSI Initiator**

![IQN](iqn.1994-05.com.redhat:877f6bdd1)

**Figure 16. Adding iSCSI initiator for worker node**

7. Click Next, verify the details, and then click Finish.

**Figure 17. Details of iSCSI initiator for worker node**

8. Repeat steps 3 through 7 for all the remaining worker nodes.

### Creating a LUN and mapping it to the worker nodes

After you add worker nodes in the Unity hosts, create a LUN and map it to the worker nodes:

1. In the Unisphere UI, under STORAGE, select Block, and click the plus sign (+). The **Configure LUN(s)** screen is displayed.

**Figure 18. Configuring LUNs**
2. Click Next, click the plus sign (+), add the worker nodes, and then click OK.

```plaintext
Figure 19. Mapping LUNs to worker nodes
```

3. Click Next and accept the defaults for the remaining options.

## Creating a pod using Unity storage

### Introduction

A Unity LUN that is provisioned to the worker nodes cannot be used directly as a storage option for the image registry. You must create a PV and PVC mappings to the Unity LUN.

Ensure that the prerequisites have been met (see Prerequisites for using Unity storage) and then perform the tasks in the following procedures:

- Adding worker nodes in Unity hosts
- Creating a LUN and mapping it to the worker nodes
- Creating a PV using a Unity LUN
- Creating a PVC and pod

### Creating a PV using a Unity LUN

In the following sample, the IP address for `targetPortal` and the IQN for `iqn` is from the Unity Unisphere UI, as shown in the following figure.

<table>
<thead>
<tr>
<th>Name</th>
<th>Operating System</th>
<th>Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>worker</td>
<td>Linux</td>
<td>iSCSI, File</td>
</tr>
<tr>
<td>worker-0</td>
<td>Linux</td>
<td>iSCSI, File</td>
</tr>
<tr>
<td>worker-1</td>
<td>Linux</td>
<td>iSCSI, File</td>
</tr>
</tbody>
</table>

---

**Note:** In the Unisphere UI, under STORAGE, select Block, and then click the iSCSI Interfaces tab.
Chapter 7: Provisioning Storage

Dell EMC Ready Stack for Red Hat OpenShift Container Platform 4.3
Enabled by Dell EMC PowerEdge R-Series Servers and PowerSwitch Networking
Deployment Guide

Figure 20. iSCSI IP address and IQN info in Unisphere

To create a PV:

1. Create a PV YAML file using the following sample:

   ```
   [core@csah ~]$ cat podpv.yaml
   apiVersion: v1
   kind: PersistentVolume
   metadata:
     name: podpv
   spec:
     capacity:
       storage: 20G
     accessModes:
     - ReadWriteOnce
     iscsi:
       targetPortal: 100.82.47.29:3260
       lun: 1
       fsType: ext4
       readOnly: false
   ```

2. Create a PV:

   ```
   [core@csah ~]$ oc create -f podpv.yaml
   persistentvolume/podpv created
   ```

Creating a PVC and pod

A PVC is used in the image registry operator as a storage option for the image registry.
Create a PVC and then the pod as follows:

1. Create a PVC YAML file using the following sample file:

   ```
   [core@csah ~]$ cat podpvc.yaml
   apiVersion: v1
   kind: PersistentVolumeClaim
   metadata:
     name: podpvc
   spec:
     accessModes:
     - ReadWriteOnce
     resources:
       requests:
       storage: 20G
   ```
2. Create a PVC:

   [core@csah ~]$ oc create -f podpv.yaml
   persistentvolume/podpv created

3. Create a YAML file with the following content to create a pod that will use the PVC as a mount point:

   [core@csah ~]$ cat podunity.yaml
   apiVersion: v1
   kind: Pod
   metadata:
     name: unitypod
   spec:
     containers:
     - name: nginx
       image: nginx
       volumeMounts:
       - mountPath: "/var/www/html"
         name: unitypvc
     volumes:
     - name: unitypvc
       persistentVolumeClaim:
         claimName: podpvc

4. Create a pod:

   [core@csah ~]$ oc create -f podunity.yaml
   pod/unitypod created

   If the pod creation fails, run oc describe pod unitypod to check for errors. Under Events, a FailedMount error is displayed if the block storage is not visible in the worker node where the pod will be started. To address this issue, run sudo rescan-scsi-bus.sh in all worker nodes as user core.

   Ensure that the LUN number that is displayed matches the LUN number on the worker node (ls /dev/disk/by-path).

5. Validate the pod status and mount point:

   [core@csah ~]$ oc get pods
   NAME       READY   STATUS    RESTARTS   AGE
   unitypod   1/1     Running   0          31m

   [core@csah ~]$ oc exec -it unitypod -- df -h /var/www/html
   Filesystem      Size  Used Avail Use% Mounted on
   /dev/sdc         20G   45M   20G   1% /var/www/html
This chapter presents the following topics:

- **OpenShift monitoring overview** .............................................................. 54
- **Enabling Grafana dashboards** ............................................................... 54
- **Viewing alerts** ...................................................................................... 55
- **Viewing cluster metrics** ....................................................................... 55
OpenShift monitoring overview

By default, OpenShift Container Platform includes a monitoring cluster operator based on the Prometheus open source project. Multiple pods run in the cluster to monitor the state of the cluster and immediately raise any alerts in the OpenShift web console. Grafana dashboards provide cluster metrics.

For more information, see About cluster monitoring in the OpenShift Container Platform documentation.

Enabling Grafana dashboards

To enable viewing of cluster metrics in the OpenShift web console, enable the Grafana dashboards as follows. Unless specified otherwise, run all the commands as user core.

1. Log in to the CSAH node.
2. Obtain the Grafana route:
   ```bash
   [core@csah ~]$ oc get routes --all-namespaces |grep -i grafana
   openshift-monitoring grafana grafana-opensource-monitoring.apps.ocp.example.com
   grafana https reencrypt/Redirect None
   ```
3. Open a web browser and paste in the URL (grafana-opensource-monitoring.apps.ocp.example.com from the preceding output example).
4. Log in as kubeadmin or as a user with cluster admin privileges.
   A list of available components in the cluster is displayed.
5. Click etcd.

The dashboard shows the active streams, number of etcd nodes up, and other details, as shown in the following figure:
Chapter 8: Monitoring the Cluster

Viewing alerts

To view the alerts in the OpenShift web console:

1. Log in to the CSAH node.
2. Get the Alert Manager route:
   
   ```bash
   [core@csah ~]$ oc get routes --all-namespaces | grep -i alertmanager
   openshift-monitoring alertmanager-main alertmanager-main-openshift-monitoring.apps.ocp.example.com
   alertmanager-main web reencrypt/Redirect None
   ```
3. Open a web browser and paste in the URL (alertmanager-main-openshift-monitoring.apps.ocp.example.com from the preceding output example).
4. Log in as kubeadmin or as a cluster admin user.

   **Note:** To temporarily mute the notification of alerts, see Silencing Alerts in the OpenShift Container Platform documentation.

Viewing cluster metrics

To view cluster metrics in the OpenShift web console:

1. Log in to the CSAH node.
2. Obtain the cluster metrics route:
   
   ```bash
   [core@csah auth]$ oc get routes --all-namespaces | grep -i prometheus
   openshift-monitoring prometheus-k8s prometheus-k8s-openshift-monitoring.apps.ocp.example.com
   prometheus-k8s web reencrypt/Redirect None
   ```
3. Open a web browser and paste in the URL (prometheus-k8s-openshift-monitoring.apps.ocp.example.com from the preceding output example).
4. Log in as kubeadmin or as a Microsoft Active Directory user.
5. From the Execute menu, select one of the available queries and click Execute.
   
   A graph for the selected query is displayed.