



OpenShift Container Platform 4.4

Installing on IBM Power

Installing OpenShift Container Platform 4.4 IBM Power clusters

OpenShift Container Platform 4.4 Installing on IBM Power

Installing OpenShift Container Platform 4.4 IBM Power clusters

Legal Notice

Copyright © 2021 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document provides instructions for installing OpenShift Container Platform 4.4 clusters on IBM Power.

Table of Contents

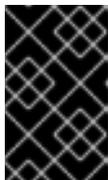
| | |
|--|----------|
| CHAPTER 1. INSTALLING ON IBM POWER | 4 |
| 1.1. INSTALLING A CLUSTER ON IBM POWER | 4 |
| 1.1.1. Prerequisites | 4 |
| 1.1.2. Internet and Telemetry access for OpenShift Container Platform | 4 |
| 1.1.3. Machine requirements for a cluster with user-provisioned infrastructure | 5 |
| 1.1.3.1. Required machines | 5 |
| 1.1.3.2. Network connectivity requirements | 5 |
| 1.1.3.3. Minimum resource requirements | 5 |
| 1.1.3.4. Certificate signing requests management | 6 |
| 1.1.4. Creating the user-provisioned infrastructure | 6 |
| 1.1.4.1. Networking requirements for user-provisioned infrastructure | 6 |
| Network topology requirements | 7 |
| Load balancers | 8 |
| 1.1.4.2. User-provisioned DNS requirements | 9 |
| 1.1.5. Generating an SSH private key and adding it to the agent | 12 |
| 1.1.6. Obtaining the installation program | 13 |
| 1.1.7. Installing the CLI by downloading the binary | 14 |
| 1.1.7.1. Installing the CLI on Linux | 14 |
| 1.1.7.2. Installing the CLI on Windows | 14 |
| 1.1.7.3. Installing the CLI on macOS | 15 |
| 1.1.8. Manually creating the installation configuration file | 15 |
| 1.1.8.1. Sample install-config.yaml file for IBM Power | 16 |
| 1.1.9. Creating the Kubernetes manifest and Ignition config files | 18 |
| 1.1.10. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines | 19 |
| 1.1.10.1. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines using an ISO image | 19 |
| 1.1.10.2. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines by PXE booting | 21 |
| 1.1.11. Creating the cluster | 23 |
| 1.1.12. Logging in to the cluster | 24 |
| 1.1.13. Approving the certificate signing requests for your machines | 25 |
| 1.1.14. Initial Operator configuration | 27 |
| 1.1.14.1. Image registry storage configuration | 28 |
| 1.1.14.1.1. Configuring registry storage for bare metal | 28 |
| 1.1.14.1.2. Configuring storage for the image registry in non-production clusters | 29 |
| 1.1.15. Completing installation on user-provisioned infrastructure | 29 |
| 1.1.16. Next steps | 31 |
| 1.2. INSTALLING A CLUSTER ON IBM POWER IN A RESTRICTED NETWORK | 32 |
| 1.2.1. Prerequisites | 32 |
| 1.2.2. About installations in restricted networks | 32 |
| 1.2.2.1. Additional limits | 33 |
| 1.2.3. Machine requirements for a cluster with user-provisioned infrastructure | 33 |
| 1.2.3.1. Required machines | 33 |
| 1.2.3.2. Network connectivity requirements | 33 |
| 1.2.3.3. Minimum resource requirements | 34 |
| 1.2.3.4. Certificate signing requests management | 34 |
| 1.2.4. Creating the user-provisioned infrastructure | 34 |
| 1.2.4.1. Networking requirements for user-provisioned infrastructure | 34 |
| Network topology requirements | 35 |
| Load balancers | 36 |
| 1.2.4.2. User-provisioned DNS requirements | 37 |
| 1.2.5. Generating an SSH private key and adding it to the agent | 40 |
| 1.2.6. Manually creating the installation configuration file | 41 |

| | |
|---|----|
| 1.2.6.1. Sample install-config.yaml file for IBM Power | 42 |
| 1.2.6.2. Configuring the cluster-wide proxy during installation | 43 |
| 1.2.7. Creating the Kubernetes manifest and Ignition config files | 45 |
| 1.2.8. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines | 46 |
| 1.2.8.1. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines using an ISO image | 46 |
| 1.2.8.2. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines by PXE booting | 48 |
| 1.2.9. Creating the cluster | 50 |
| 1.2.10. Logging in to the cluster | 51 |
| 1.2.11. Approving the certificate signing requests for your machines | 52 |
| 1.2.12. Initial Operator configuration | 54 |
| 1.2.12.1. Image registry storage configuration | 55 |
| 1.2.12.1.1. Configuring registry storage for bare metal | 55 |
| 1.2.12.1.2. Configuring storage for the image registry in non-production clusters | 56 |
| 1.2.13. Completing installation on user-provisioned infrastructure | 56 |
| 1.2.14. Next steps | 58 |

CHAPTER 1. INSTALLING ON IBM POWER

1.1. INSTALLING A CLUSTER ON IBM POWER

In OpenShift Container Platform version 4.4, you can install a cluster on IBM Power infrastructure that you provision.



IMPORTANT

Additional considerations exist for non-bare metal platforms. Review the information in the [guidelines for deploying OpenShift Container Platform on non-tested platforms](#) before you install an OpenShift Container Platform cluster.

1.1.1. Prerequisites

- Provision [persistent storage using NFS](#) for your cluster. To deploy a private image registry, your storage must provide **ReadWriteMany** access modes.
- Review details about the [OpenShift Container Platform installation and update](#) processes.
- If you use a firewall, you must [configure it to allow the sites](#) that your cluster requires access to.



NOTE

Be sure to also review this site list if you are configuring a proxy.

1.1.2. Internet and Telemetry access for OpenShift Container Platform

In OpenShift Container Platform 4.4, you require access to the Internet to install your cluster. The Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, also requires Internet access. If your cluster is connected to the Internet, Telemetry runs automatically, and your cluster is registered to the [Red Hat OpenShift Cluster Manager \(OCM\)](#).

Once you confirm that your Red Hat OpenShift Cluster Manager inventory is correct, either maintained automatically by Telemetry or manually using OCM, [use subscription watch](#) to track your OpenShift Container Platform subscriptions at the account or multi-cluster level.

You must have Internet access to:

- Access the [Red Hat OpenShift Cluster Manager](#) page to download the installation program and perform subscription management. If the cluster has Internet access and you do not disable Telemetry, that service automatically entitles your cluster.
- Access [Quay.io](#) to obtain the packages that are required to install your cluster.
- Obtain the packages that are required to perform cluster updates.



IMPORTANT

If your cluster cannot have direct Internet access, you can perform a restricted network installation on some types of infrastructure that you provision. During that process, you download the content that is required and use it to populate a mirror registry with the packages that you need to install a cluster and generate the installation program. With some installation types, the environment that you install your cluster in will not require Internet access. Before you update the cluster, you update the content of the mirror registry.

1.1.3. Machine requirements for a cluster with user-provisioned infrastructure

For a cluster that contains user-provisioned infrastructure, you must deploy all of the required machines.

1.1.3.1. Required machines

The smallest OpenShift Container Platform clusters require the following hosts:

- One temporary bootstrap machine
- Three control plane, or master, machines
- At least two compute machines, which are also known as worker machines



NOTE

The cluster requires the bootstrap machine to deploy the OpenShift Container Platform cluster on the three control plane machines. You can remove the bootstrap machine after you install the cluster.



IMPORTANT

To maintain high availability of your cluster, use separate physical hosts for these cluster machines.

The bootstrap, control plane, and compute machines must use the Red Hat Enterprise Linux CoreOS (RHCOS) as the operating system.

Note that RHCOS is based on Red Hat Enterprise Linux 8 and inherits all of its hardware certifications and requirements. See [Red Hat Enterprise Linux technology capabilities and limits](#) .

1.1.3.2. Network connectivity requirements

All the Red Hat Enterprise Linux CoreOS (RHCOS) machines require network in **initramfs** during boot to fetch Ignition config files from the Machine Config Server. During the initial boot, the machines require either a DHCP server or that static IP addresses be set in order to establish a network connection to download their Ignition config files.

1.1.3.3. Minimum resource requirements

Each cluster machine must meet the following minimum requirements:

| Machine | Operating System | vCPU ¹ | Virtual RAM | Storage |
|---------------|------------------|-------------------|-------------|---------|
| Bootstrap | RHCOS | 2 | 16 GB | 120 GB |
| Control plane | RHCOS | 2 | 16 GB | 120 GB |
| Compute | RHCOS | 2 | 8 GB | 120 GB |

¹1 physical core provides 2 vCPUs when hyper-threading is enabled. 1 physical core provides 1 vCPU when hyper-threading is not enabled.

1.1.3.4. Certificate signing requests management

Because your cluster has limited access to automatic machine management when you use infrastructure that you provision, you must provide a mechanism for approving cluster certificate signing requests (CSRs) after installation. The **kube-controller-manager** only approves the kubelet client CSRs. The **machine-approver** cannot guarantee the validity of a serving certificate that is requested by using kubelet credentials because it cannot confirm that the correct machine issued the request. You must determine and implement a method of verifying the validity of the kubelet serving certificate requests and approving them.

1.1.4. Creating the user-provisioned infrastructure

Before you deploy an OpenShift Container Platform cluster that uses user-provisioned infrastructure, you must create the underlying infrastructure.

Prerequisites

- Review the [OpenShift Container Platform 4.x Tested Integrations](#) page before you create the supporting infrastructure for your cluster.

Procedure

1. Configure DHCP or set static IP addresses on each node.
2. Provision the required load balancers.
3. Configure the ports for your machines.
4. Configure DNS.
5. Ensure network connectivity.

1.1.4.1. Networking requirements for user-provisioned infrastructure

All the Red Hat Enterprise Linux CoreOS (RHCOS) machines require network in **initramfs** during boot to fetch Ignition config from the machine config server.

During the initial boot, the machines require either a DHCP server or that static IP addresses be set on each host in the cluster in order to establish a network connection, which allows them to download their Ignition config files.

It is recommended to use the DHCP server to manage the machines for the cluster long-term. Ensure that the DHCP server is configured to provide persistent IP addresses and host names to the cluster machines.

The Kubernetes API server, which runs on each master node after a successful cluster installation, must be able to resolve the node names of the cluster machines. If the API servers and worker nodes are in different zones, you can configure a default DNS search zone to allow the API server to resolve the node names. Another supported approach is to always refer to hosts by their fully-qualified domain names in both the node objects and all DNS requests.

You must configure the network connectivity between machines to allow cluster components to communicate. Each machine must be able to resolve the host names of all other machines in the cluster.

Table 1.1. All machines to all machines

| Protocol | Port | Description |
|----------|--------------------|---|
| ICMP | N/A | Network reachability tests |
| TCP | 9000-9999 | Host level services, including the node exporter on ports 9100-9101 and the Cluster Version Operator on port 9099 . |
| | 10250-10259 | The default ports that Kubernetes reserves |
| | 10256 | openshift-sdn |
| UDP | 4789 | VXLAN and Geneve |
| | 6081 | VXLAN and Geneve |
| | 9000-9999 | Host level services, including the node exporter on ports 9100-9101 . |
| TCP/UDP | 30000-32767 | Kubernetes node port |

Table 1.2. All machines to control plane

| Protocol | Port | Description |
|----------|------------------|--------------------------------------|
| TCP | 2379-2380 | etcd server, peer, and metrics ports |
| | 6443 | Kubernetes API |

Network topology requirements

The infrastructure that you provision for your cluster must meet the following network topology requirements.

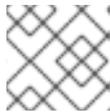
**IMPORTANT**

OpenShift Container Platform requires all nodes to have internet access to pull images for platform containers and provide telemetry data to Red Hat.

Load balancers

Before you install OpenShift Container Platform, you must provision two load balancers that meet the following requirements:

1. **API load balancer.** Provides a common endpoint for users, both human and machine, to interact with and configure the platform. Configure the following conditions:
 - Layer 4 load balancing only. This can be referred to as Raw TCP, SSL Passthrough, or SSL Bridge mode. If you use SSL Bridge mode, you must enable Server Name Indication (SNI) for the API routes.
 - A stateless load balancing algorithm. The options vary based on the load balancer implementation.

**NOTE**

Session persistence is not required for the API load balancer to function properly.

Configure the following ports on both the front and back of the load balancers:

Table 1.3. API load balancer

| Port | Back-end machines (pool members) | Internal | External | Description |
|--------------|---|----------|----------|-----------------------|
| 6443 | Bootstrap and control plane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane. You must configure the /readyz endpoint for the API server health check probe. | X | X | Kubernetes API server |
| 22623 | Bootstrap and control plane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane. | X | | Machine config server |

**NOTE**

The load balancer must be configured to take a maximum of 30 seconds from the time the API server turns off the **/readyz** endpoint to the removal of the API server instance from the pool. Within the time frame after **/readyz** returns an error or becomes healthy, the endpoint must have been removed or added. Probing every 5 or 10 seconds, with two successful requests to become healthy and three to become unhealthy, are well-tested values.

2. **Application Ingress load balancer.** Provides an Ingress point for application traffic flowing in from outside the cluster. Configure the following conditions:
 - Layer 4 load balancing only. This can be referred to as Raw TCP, SSL Passthrough, or SSL Bridge mode. If you use SSL Bridge mode, you must enable Server Name Indication (SNI) for the Ingress routes.
 - A connection-based or session-based persistence is recommended, based on the options available and types of applications that will be hosted on the platform.

Configure the following ports on both the front and back of the load balancers:

Table 1.4. Application Ingress load balancer

| Port | Back-end machines (pool members) | Internal | External | Description |
|------------|--|----------|----------|---------------|
| 443 | The machines that run the Ingress router pods, compute, or worker, by default. | X | X | HTTPS traffic |
| 80 | The machines that run the Ingress router pods, compute, or worker, by default. | X | X | HTTP traffic |

TIP

If the true IP address of the client can be seen by the load balancer, enabling source IP-based session persistence can improve performance for applications that use end-to-end TLS encryption.



NOTE

A working configuration for the Ingress router is required for an OpenShift Container Platform cluster. You must configure the Ingress router after the control plane initializes.

1.1.4.2. User-provisioned DNS requirements

DNS is used for name resolution and reverse name resolution. DNS A/AAAA or CNAME records are used for name resolution and PTR records are used for reverse name resolution. The reverse records are important because Red Hat Enterprise Linux CoreOS (RHCOS) uses the reverse records to set the host name for all the nodes. Additionally, the reverse records are used to generate the certificate signing requests (CSR) that OpenShift Container Platform needs to operate.

The following DNS records are required for an OpenShift Container Platform cluster that uses user-provisioned infrastructure. In each record, **<cluster_name>** is the cluster name and **<base_domain>** is the cluster base domain that you specify in the **install-config.yaml** file. A complete DNS record takes the form: **<component>.<cluster_name>.<base_domain>.**

Table 1.5. Required DNS records

| Component | Record | Description |
|----------------|---|--|
| Kubernetes API | api.<cluster_name>.<base_domain> | Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the load balancer for the control plane machines. These records must be resolvable by both clients external to the cluster and from all the nodes within the cluster. |
| | api-int.<cluster_name>.<base_domain> | Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the load balancer for the control plane machines. These records must be resolvable from all the nodes within the cluster.  IMPORTANT The API server must be able to resolve the worker nodes by the host names that are recorded in Kubernetes. If the API server cannot resolve the node names, then proxied API calls can fail, and you cannot retrieve logs from pods. |
| Routes | *.apps.<cluster_name>.<base_domain> | Add a wildcard DNS A/AAAA or CNAME record that refers to the load balancer that targets the machines that run the Ingress router pods, which are the worker nodes by default. These records must be resolvable by both clients external to the cluster and from all the nodes within the cluster. |
| Bootstrap | bootstrap.<cluster_name>.<base_domain> | Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the bootstrap machine. These records must be resolvable by the nodes within the cluster. |
| Master hosts | <master><n>.<cluster_name>.<base_domain> | Add DNS A/AAAA or CNAME records and DNS PTR records to identify each machine for the master nodes. These records must be resolvable by the nodes within the cluster. |
| Worker hosts | <worker><n>.<cluster_name>.<base_domain> | Add DNS A/AAAA or CNAME records and DNS PTR records to identify each machine for the worker nodes. These records must be resolvable by the nodes within the cluster. |

TIP

You can use the **nslookup <hostname>** command to verify name resolution. You can use the **dig -x <ip_address>** command to verify reverse name resolution for the PTR records.

The following example of a BIND zone file shows sample A records for name resolution. The purpose of the example is to show the records that are needed. The example is not meant to provide advice for choosing one name resolution service over another.

Example 1.1. Sample DNS zone database

```

$TTL 1W
@ IN SOA ns1.example.com. root (
    2019070700 ; serial
    3H ; refresh (3 hours)
    30M ; retry (30 minutes)
    2W ; expiry (2 weeks)
    1W ) ; minimum (1 week)
IN NS ns1.example.com.
IN MX 10 smtp.example.com.
;
;
ns1 IN A 192.168.1.5
smtp IN A 192.168.1.5
;
helper IN A 192.168.1.5
helper.ocp4 IN A 192.168.1.5
;
; The api identifies the IP of your load balancer.
api.ocp4 IN A 192.168.1.5
api-int.ocp4 IN A 192.168.1.5
;
; The wildcard also identifies the load balancer.
*.apps.ocp4 IN A 192.168.1.5
;
; Create an entry for the bootstrap host.
bootstrap.ocp4 IN A 192.168.1.96
;
; Create entries for the master hosts.
master0.ocp4 IN A 192.168.1.97
master1.ocp4 IN A 192.168.1.98
master2.ocp4 IN A 192.168.1.99
;
; Create entries for the worker hosts.
worker0.ocp4 IN A 192.168.1.11
worker1.ocp4 IN A 192.168.1.7
;
;EOF

```

The following example BIND zone file shows sample PTR records for reverse name resolution.

Example 1.2. Sample DNS zone database for reverse records

```

$TTL 1W
@ IN SOA ns1.example.com. root (
    2019070700 ; serial
    3H ; refresh (3 hours)
    30M ; retry (30 minutes)
    2W ; expiry (2 weeks)
    1W ) ; minimum (1 week)
IN NS ns1.example.com.
;
; The syntax is "last octet" and the host must have an FQDN

```

```

; with a trailing dot.
97 IN PTR master0.ocp4.example.com.
98 IN PTR master1.ocp4.example.com.
99 IN PTR master2.ocp4.example.com.
;
96 IN PTR bootstrap.ocp4.example.com.
;
5 IN PTR api.ocp4.ocp4.example.com.
5 IN PTR api-int.ocp4.ocp4.example.com.
;
11 IN PTR worker0.ocp4.example.com.
7 IN PTR worker1.ocp4.example.com.
;
;EOF

```

1.1.5. Generating an SSH private key and adding it to the agent

If you want to perform installation debugging or disaster recovery on your cluster, you must provide an SSH key to both your **ssh-agent** and the installation program. You can use this key to access the bootstrap machine in a public cluster to troubleshoot installation issues.



NOTE

In a production environment, you require disaster recovery and debugging.

You can use this key to SSH into the master nodes as the user **core**. When you deploy the cluster, the key is added to the **core** user's `~/.ssh/authorized_keys` list.



NOTE

You must use a local key, not one that you configured with platform-specific approaches such as [AWS key pairs](#).

Procedure

1. If you do not have an SSH key that is configured for password-less authentication on your computer, create one. For example, on a computer that uses a Linux operating system, run the following command:

```

$ ssh-keygen -t ed25519 -N "" \
-f <path>/<file_name> 1

```

- 1 Specify the path and file name, such as `~/.ssh/id_rsa`, of the SSH key. Do not specify an existing SSH key, as it will be overwritten.

Running this command generates an SSH key that does not require a password in the location that you specified.

2. Start the **ssh-agent** process as a background task:

```
$ eval "$(ssh-agent -s)"
```

```
Agent pid 31874
```

3. Add your SSH private key to the **ssh-agent**:

```
$ ssh-add <path>/<file_name> 1
```

```
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

- 1** Specify the path and file name for your SSH private key, such as `~/.ssh/id_rsa`

Next steps

- When you install OpenShift Container Platform, provide the SSH public key to the installation program.

1.1.6. Obtaining the installation program

Before you install OpenShift Container Platform, download the installation file on a local computer.

Prerequisites

- You must install the cluster from a computer that uses Linux or macOS.
- You need 500 MB of local disk space to download the installation program.

Procedure

1. Access the [Infrastructure Provider](#) page on the Red Hat OpenShift Cluster Manager site. If you have a Red Hat account, log in with your credentials. If you do not, create an account.
2. Navigate to the page for your installation type, download the installation program for your operating system, and place the file in the directory where you will store the installation configuration files.



IMPORTANT

The installation program creates several files on the computer that you use to install your cluster. You must keep both the installation program and the files that the installation program creates after you finish installing the cluster.



IMPORTANT

Deleting the files created by the installation program does not remove your cluster, even if the cluster failed during installation. You must complete the OpenShift Container Platform uninstallation procedures outlined for your specific cloud provider to remove your cluster entirely.

3. Extract the installation program. For example, on a computer that uses a Linux operating system, run the following command:

```
$ tar xvf <installation_program>.tar.gz
```

4. From the [Pull Secret](#) page on the Red Hat OpenShift Cluster Manager site, download your installation pull secret as a **.txt** file. This pull secret allows you to authenticate with the services that are provided by the included authorities, including Quay.io, which serves the container images for OpenShift Container Platform components.

1.1.7. Installing the CLI by downloading the binary

You can install the OpenShift CLI (**oc**) in order to interact with OpenShift Container Platform from a command-line interface. You can install **oc** on Linux, Windows, or macOS.



IMPORTANT

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.4. Download and install the new version of **oc**.

1.1.7.1. Installing the CLI on Linux

You can install the OpenShift CLI (**oc**) binary on Linux by using the following procedure.

Procedure

1. Navigate to the [Infrastructure Provider](#) page on the Red Hat OpenShift Cluster Manager site.
2. Select your infrastructure provider, and, if applicable, your installation type.
3. In the **Command-line interface** section, select **Linux** from the drop-down menu and click **Download command-line tools**.
4. Unpack the archive:

```
$ tar xvzf <file>
```

5. Place the **oc** binary in a directory that is on your **PATH**. To check your **PATH**, execute the following command:

```
$ echo $PATH
```

After you install the CLI, it is available using the **oc** command:

```
$ oc <command>
```

1.1.7.2. Installing the CLI on Windows

You can install the OpenShift CLI (**oc**) binary on Windows by using the following procedure.

Procedure

1. Navigate to the [Infrastructure Provider](#) page on the Red Hat OpenShift Cluster Manager site.
2. Select your infrastructure provider, and, if applicable, your installation type.

3. In the **Command-line interface** section, select **Windows** from the drop-down menu and click **Download command-line tools**.
4. Unzip the archive with a ZIP program.
5. Move the **oc** binary to a directory that is on your **PATH**.
To check your **PATH**, open the command prompt and execute the following command:

```
C:\> path
```

After you install the CLI, it is available using the **oc** command:

```
C:\> oc <command>
```

1.1.7.3. Installing the CLI on macOS

You can install the OpenShift CLI (**oc**) binary on macOS by using the following procedure.

Procedure

1. Navigate to the [Infrastructure Provider](#) page on the Red Hat OpenShift Cluster Manager site.
2. Select your infrastructure provider, and, if applicable, your installation type.
3. In the **Command-line interface** section, select **MacOS** from the drop-down menu and click **Download command-line tools**.
4. Unpack and unzip the archive.
5. Move the **oc** binary to a directory on your PATH.
To check your **PATH**, open a terminal and execute the following command:

```
$ echo $PATH
```

After you install the CLI, it is available using the **oc** command:

```
$ oc <command>
```

1.1.8. Manually creating the installation configuration file

For installations of OpenShift Container Platform that use user-provisioned infrastructure, you must manually generate your installation configuration file.

Prerequisites

- Obtain the OpenShift Container Platform installation program and the access token for your cluster.

Procedure

1. Create an installation directory to store your required installation assets in:

```
$ mkdir <installation_directory>
```

**IMPORTANT**

You must create a directory. Some installation assets, like bootstrap X.509 certificates have short expiration intervals, so you must not reuse an installation directory. If you want to reuse individual files from another cluster installation, you can copy them into your directory. However, the file names for the installation assets might change between releases. Use caution when copying installation files from an earlier OpenShift Container Platform version.

2. Customize the following **install-config.yaml** file template and save it in the **<installation_directory>**.

**NOTE**

You must name this configuration file **install-config.yaml**.

3. Back up the **install-config.yaml** file so that you can use it to install multiple clusters.

**IMPORTANT**

The **install-config.yaml** file is consumed during the next step of the installation process. You must back it up now.

1.1.8.1. Sample install-config.yaml file for IBM Power

You can customize the **install-config.yaml** file to specify more details about your OpenShift Container Platform cluster's platform or modify the values of the required parameters.

```

apiVersion: v1
baseDomain: example.com 1
compute:
- hyperthreading: Enabled 2 3
  name: worker
  replicas: 0 4
controlPlane:
  hyperthreading: Enabled 5 6
  name: master 7
  replicas: 3 8
metadata:
  name: test 9
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14 10
    hostPrefix: 23 11
  networkType: OpenShiftSDN
  serviceNetwork: 12
  - 172.30.0.0/16
platform:
  none: {} 13

```

```
fips: false 14
pullSecret: '{"auths": ...}' 15
sshKey: 'ssh-ed25519 AAAA...' 16
```

- 1** The base domain of the cluster. All DNS records must be sub-domains of this base and include the cluster name.
- 2 5** The **controlPlane** section is a single mapping, but the compute section is a sequence of mappings. To meet the requirements of the different data structures, the first line of the **compute** section must begin with a hyphen, -, and the first line of the **controlPlane** section must not. Although both sections currently define a single machine pool, it is possible that future versions of OpenShift Container Platform will support defining multiple compute pools during installation. Only one control plane pool is used.
- 3 6 7** Whether to enable or disable simultaneous multithreading, or **hyperthreading**. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores. You can disable it by setting the parameter value to **Disabled**. If you disable simultaneous multithreading in some cluster machines, you must disable it in all cluster machines.



IMPORTANT

If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.

- 4** You must set the value of the **replicas** parameter to **0**. This parameter controls the number of workers that the cluster creates and manages for you, which are functions that the cluster does not perform when you use user-provisioned infrastructure. You must manually deploy worker machines for the cluster to use before you finish installing OpenShift Container Platform.
- 8** The number of control plane machines that you add to the cluster. Because the cluster uses this values as the number of etcd endpoints in the cluster, the value must match the number of control plane machines that you deploy.
- 9** The cluster name that you specified in your DNS records.
- 10** A block of IP addresses from which pod IP addresses are allocated. This block must not overlap with existing physical networks. These IP addresses are used for the pod network. If you need to access the pods from an external network, you must configure load balancers and routers to manage the traffic.
- 11** The subnet prefix length to assign to each individual node. For example, if **hostPrefix** is set to **23**, then each node is assigned a **/23** subnet out of the given **cidr**, which allows for 510 ($2^{(32 - 23)} - 2$) pod IPs addresses. If you are required to provide access to nodes from an external network, configure load balancers and routers to manage the traffic.
- 12** The IP address pool to use for service IP addresses. You can enter only one IP address pool. If you need to access the services from an external network, configure load balancers and routers to manage the traffic.
- 13** You must set the platform to **none**. You cannot provide additional platform configuration variables for IBM Power infrastructure.
- 14** Whether to enable or disable FIPS mode. By default, FIPS mode is not enabled. If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography

modules that are provided with RHCOS instead.

- 15 The pull secret that you obtained from the [Pull Secret](#) page on the Red Hat OpenShift Cluster Manager site. This pull secret allows you to authenticate with the services that are provided by the included authorities, including Quay.io, which serves the container images for OpenShift Container Platform components.
- 16 The public portion of the default SSH key for the **core** user in Red Hat Enterprise Linux CoreOS (RHCOS).



NOTE

For production OpenShift Container Platform clusters on which you want to perform installation debugging or disaster recovery, specify an SSH key that your **ssh-agent** process uses.

1.1.9. Creating the Kubernetes manifest and Ignition config files

Because you must modify some cluster definition files and manually start the cluster machines, you must generate the Kubernetes manifest and Ignition config files that the cluster needs to make its machines.



IMPORTANT

The Ignition config files that the installation program generates contain certificates that expire after 24 hours, which are then renewed at that time. If the cluster is shut down before renewing the certificates and the cluster is later restarted after the 24 hours have elapsed, the cluster automatically recovers the expired certificates. The exception is that you must manually approve the pending **node-bootstrapper** certificate signing requests (CSRs) to recover kubelet certificates. See the documentation for *Recovering from expired control plane certificates* for more information.

Prerequisites

- Obtain the OpenShift Container Platform installation program.
- Create the **install-config.yaml** installation configuration file.

Procedure

1. Generate the Kubernetes manifests for the cluster:

```
$ ./openshift-install create manifests --dir=<installation_directory> 1
```

```
INFO Consuming Install Config from target directory
WARNING Making control-plane schedulable by setting MastersSchedulable to true for Scheduler cluster settings
```

- 1 For **<installation_directory>**, specify the installation directory that contains the **install-config.yaml** file you created.

Because you create your own compute machines later in the installation process, you can safely ignore this warning.

2. Modify the `<installation_directory>/manifests/cluster-scheduler-02-config.yml` Kubernetes manifest file to prevent pods from being scheduled on the control plane machines:
 - a. Open the `<installation_directory>/manifests/cluster-scheduler-02-config.yml` file.
 - b. Locate the `mastersSchedulable` parameter and set its value to **False**.
 - c. Save and exit the file.



NOTE

Currently, due to a [Kubernetes limitation](#), router Pods running on control plane machines will not be reachable by the ingress load balancer. This step might not be required in a future minor version of OpenShift Container Platform.

3. Obtain the Ignition config files:

```
$ ./openshift-install create ignition-configs --dir=<installation_directory> 1
```

- 1 For `<installation_directory>`, specify the same installation directory.

The following files are generated in the directory:

```
.
├── auth
│   ├── kubeadmin-password
│   └── kubeconfig
├── bootstrap.ign
├── master.ign
├── metadata.json
└── worker.ign
```

1.1.10. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines

Before you install a cluster on IBM Power infrastructure that you provision, you must create RHCOS machines for it to use. Follow either the steps to use an ISO image or network PXE booting to create the machines.

1.1.10.1. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines using an ISO image

Before you install a cluster on IBM Power infrastructure that you provision, you must create RHCOS machines for it to use. You can use an ISO image to create the machines.

Prerequisites

- Obtain the Ignition config files for your cluster.
- Have access to an HTTP server that you can access from your computer and that the machines that you create can access.

Procedure

1. Upload the control plane, compute, and bootstrap Ignition config files that the installation program created to your HTTP server. Note the URLs of these files.



IMPORTANT

If you plan to add more compute machines to your cluster after you finish installation, do not delete these files.

2. Obtain the RHCOS images that are required for your preferred method of installing operating system instances from the [Product Downloads](#) page on the Red Hat customer portal or the [RHCOS image mirror](#) page.



IMPORTANT

The RHCOS images might not change with every release of OpenShift Container Platform. You must download images with the highest version that is less than or equal to the OpenShift Container Platform version that you install. Use the image versions that match your OpenShift Container Platform version if they are available. Only use ISO images for this procedure. RHCOS qcow2 images are not supported for bare metal installs.

You must download the ISO file and the RAW disk file. Those file names resemble the following examples:

- ISO: **rhcos-`<version>`-installer.`<architecture>`.iso**
- Compressed metal RAW: **rhcos-`<version>`-metal.`<architecture>`.raw.gz**

3. Upload either the RAW RHCOS image file to your HTTP server and note its URL.



IMPORTANT

If you plan to add more compute machines to your cluster after you finish installation, do not delete these files.

4. Use the ISO to start the RHCOS installation. Use one of the following installation options:
 - Burn the ISO image to a disk and boot it directly.
 - Use ISO redirection via a LOM interface.
5. After the instance boots, press the **TAB** or **E** key to edit the kernel command line.
6. Add the parameters to the kernel command line:

```
coreos.inst=yes
coreos.inst.install_dev=sda 1
coreos.inst.image_url=<image_URL> 2
coreos.inst.ignition_url=http://example.com/config.ign 3
ip=<dhcp or static IP address> 4 5
bond=<bonded_interface> 6
```

- 1 Specify the block device of the system to install to.

2. Specify the URL of the RAW image that you uploaded to your server.
 3. Specify the URL of the Ignition config file for this machine type.
 4. Set **ip=dhcp** or set an individual static IP address (**ip=**) and DNS server (**nameserver=**) on each node. See *Static IP address examples for RHCOS kernel parameters* for details.
 5. If you use multiple network interfaces or DNS servers, see *Static IP address examples for RHCOS kernel parameters* for details on how to configure them.
 6. Optionally, you can bond multiple network interfaces to a single interface using the **bond=** option, as described in *Static IP address examples for RHCOS kernel parameters*.
7. Press Enter to complete the installation. After RHCOS installs, the system reboots. After the system reboots, it applies the Ignition config file that you specified.
 8. Continue to create the machines for your cluster.



IMPORTANT

You must create the bootstrap and control plane machines at this time. Because some pods are deployed on compute machines by default, also create at least two compute machines before you install the cluster.

1.1.10.2. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines by PXE booting

Before you install a cluster on IBM Power infrastructure that you provision, you must create RHCOS machines for it to use. You can use PXE booting to create the machines.

Prerequisites

- Obtain the Ignition config files for your cluster.
- Have access to an HTTP server and TFTP server that you can access from your computer.

Procedure

1. Upload the master, worker, and bootstrap Ignition config files that the installation program created to your HTTP server. Note the URLs of these files.



IMPORTANT

If you plan to add more compute machines to your cluster after you finish installation, do not delete these files.

2. Obtain the compressed metal RAW image, **kernel** and **initramfs** files from the [Product Downloads](#) page on the Red Hat customer portal or the [RHCOS image mirror](#) page.



IMPORTANT

The RHCOS images might not change with every release of OpenShift Container Platform. You must download images with the highest version that is less than or equal to the OpenShift Container Platform version that you install. Use the image versions that match your OpenShift Container Platform version if they are available. Only use RAW images for this procedure. RHCOS qcow2 images are not supported for bare metal installs.

The file names contain the OpenShift Container Platform version number. They resemble the following examples:

- Compressed metal RAW image: **rhcos-<version>-<architecture>-metal.<architecture>.raw.gz**
 - **kernel:** **rhcos-<version>-<architecture>-installer-kernel-<architecture>**
 - **initramfs:** **rhcos-<version>-<architecture>-installer-initramfs.<architecture>.img**
3. Upload the RAW image to your HTTP server.
 4. Upload the additional files that are required for your booting method:
 - For traditional PXE, upload the **kernel** and **initramfs** files to your TFTP server.
 - For iPXE, upload the **kernel** and **initramfs** files to your HTTP server.



IMPORTANT

If you plan to add more compute machines to your cluster after you finish installation, do not delete these files.

5. Configure the network boot infrastructure so that the machines boot from their local disks after RHCOS is installed on them.
6. Configure PXE installation for the RHCOS images.
Modify one of the following example menu entries for your environment and verify that the image and Ignition files are properly accessible:

- For PXE:

```

DEFAULT pxeboot
TIMEOUT 20
PROMPT 0
LABEL pxeboot
  KERNEL rhcos-<version>-<architecture>-installer-kernel-<architecture> 1
  APPEND ip=dhcp rd.neednet=1 initrd=rhcos-<version>-<architecture>-installer-
initramfs.<architecture>.img console=tty0 console=ttyS0 coreos.inst=yes
coreos.inst.install_dev=sda coreos.inst.image_url=http://<HTTP_server>/rhcos-
<version>-<architecture>-metal.<architecture>.raw.gz
coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign 2 3

```

- 1** Specify the location of the **kernel** file available on your TFTP server.
- 2** If you use multiple NICs, specify a single interface in the **ip** option. For example, to use

- 3 Specify locations of the RHCOS files that you uploaded to your HTTP or TFTP server. The **initrd** parameter value is the location of the **initramfs** file on your TFTP server.

7. If you use UEFI, perform the following actions:

- a. Provide the EFI binaries and **grub.cfg** file that are required for booting the system. You need the **shim.efi** binary and the **grubx64.efi** binary.
 - Extract the necessary EFI binaries by mounting the RHCOS ISO on your host and then mounting the **images/efiboot.img** file to your host. From the **efiboot.img** mount point, you then copy the **EFI/redhat/shimx64.efi** and **EFI/redhat/grubx64.efi** files to your TFTP server.

```
# mkdir -p /mnt/{iso,efiboot}
# mount -o loop rhcos-installer.x86_64.iso /mnt/iso
# mount -o loop,ro /mnt/iso/images/efiboot.img /mnt/efiboot
# cp /mnt/efiboot/EFI/redhat/{shimx64.efi,grubx64.efi} .
# umount /mnt/{efiboot,iso}
```

- b. Copy the **EFI/redhat/grub.cfg** file that is included in the RHCOS ISO to your TFTP server.
- c. Edit the **grub.cfg** file to include the following arguments:

```
menuentry 'Install Red Hat Enterprise Linux CoreOS' --class fedora --class gnu-linux --
class gnu --class os {
  linux rhcos-<version>-<architecture>-installer-kernel-<architecture> nomodeset
  rd.neednet=1 coreos.inst=yes coreos.inst.install_dev=sda
  coreos.inst.image_url=http://<HTTP_server>/rhcos-<version>-<architecture>-metal.
  <architecture>.raw.gz coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign 1
  initrd rhcos-<version>-<architecture>-installer-initramfs.<architecture>.img 2
}
```

- 1 The first argument to the **linux** line item is the location of the **kernel** file that you uploaded to your TFTP server. For the **coreos.inst.image_url** parameter value, specify the location of the compressed metal RAW image that you uploaded to your HTTP server. For the **coreos.inst.ignition_url** parameter, specify the location of the bootstrap Ignition config file that you uploaded to your HTTP server.

- 2 Specify the location of the **initramfs** file that you uploaded to your TFTP server.

8. Continue to create the machines for your cluster.



IMPORTANT

You must create the bootstrap and control plane machines at this time. Because some pods are deployed on compute machines by default, also create at least two compute machine before you install the cluster.

1.1.11. Creating the cluster

To create the OpenShift Container Platform cluster, you wait for the bootstrap process to complete on the machines that you provisioned by using the Ignition config files that you generated with the installation program.

Prerequisites

- Create the required infrastructure for the cluster.
- You obtained the installation program and generated the Ignition config files for your cluster.
- You used the Ignition config files to create RHCOS machines for your cluster.
- Your machines have direct Internet access or have an HTTP or HTTPS proxy available.

Procedure

1. Monitor the bootstrap process:

```
$ ./openshift-install --dir=<installation_directory> wait-for bootstrap-complete \ 1
--log-level=info 2
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com...
INFO API v1.17.1 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
```

- 1 For **<installation_directory>**, specify the path to the directory that you stored the installation files in.
- 2 To view different installation details, specify **warn**, **debug**, or **error** instead of **info**.

The command succeeds when the Kubernetes API server signals that it has been bootstrapped on the control plane machines.

2. After bootstrap process is complete, remove the bootstrap machine from the load balancer.



IMPORTANT

You must remove the bootstrap machine from the load balancer at this point. You can also remove or reformat the machine itself.

1.1.12. Logging in to the cluster

You can log in to your cluster as a default system user by exporting the cluster **kubeconfig** file. The **kubeconfig** file contains information about the cluster that is used by the CLI to connect a client to the correct cluster and API server. The file is specific to a cluster and is created during OpenShift Container Platform installation.

Prerequisites

- Deploy an OpenShift Container Platform cluster.
- Install the **oc** CLI.

Procedure

1. Export the **kubeadmin** credentials:

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

- 1 For `<installation_directory>`, specify the path to the directory that you stored the installation files in.

2. Verify you can run `oc` commands successfully using the exported configuration:

```
$ oc whoami
system:admin
```

1.1.13. Approving the certificate signing requests for your machines

When you add machines to a cluster, two pending certificate signing requests (CSRs) are generated for each machine that you added. You must confirm that these CSRs are approved or, if necessary, approve them yourself. The client requests must be approved first, followed by the server requests.

Prerequisites

- You added machines to your cluster.

Procedure

1. Confirm that the cluster recognizes the machines:

```
# oc get nodes

NAME                STATUS  ROLES  AGE  VERSION
master-01.example.com Ready  master  40d  v1.17.1
master-02.example.com Ready  master  40d  v1.17.1
master-03.example.com Ready  master  40d  v1.17.1
worker-01.example.com Ready  worker  40d  v1.17.1
worker-02.example.com Ready  worker  40d  v1.17.1
```

The output lists all of the machines that you created.

2. Review the pending CSRs and ensure that you see the client requests with the **Pending** or **Approved** status for each machine that you added to the cluster:

```
$ oc get csr

NAME      AGE  REQUESTOR                                     CONDITION
csr-8b2br 15m  system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending
csr-8vnps 15m  system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending
...
```

In this example, two machines are joining the cluster. You might see more approved CSRs in the list.

3. If the CSRs were not approved, after all of the pending CSRs for the machines you added are in **Pending** status, approve the CSRs for your cluster machines:



NOTE

Because the CSRs rotate automatically, approve your CSRs within an hour of adding the machines to the cluster. If you do not approve them within an hour, the certificates will rotate, and more than two certificates will be present for each node. You must approve all of these certificates. After you approve the initial CSRs, the subsequent node client CSRs are automatically approved by the cluster **kube-controller-manager**. You must implement a method of automatically approving the kubelet serving certificate requests.

- To approve them individually, run the following command for each valid CSR:

```
$ oc adm certificate approve <csr_name> 1
```

- 1** **<csr_name>** is the name of a CSR from the list of current CSRs.

- To approve all pending CSRs, run the following command:

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{"\n"}\n{{end}}' | xargs oc adm certificate approve
```

- Now that your client requests are approved, you must review the server requests for each machine that you added to the cluster:

```
$ oc get csr
```

Example output

```
NAME      AGE   REQUESTOR                                     CONDITION
csr-bfd72 5m26s system:node:ip-10-0-50-126.us-east-2.compute.internal
Pending
csr-c57lv 5m26s system:node:ip-10-0-95-157.us-east-2.compute.internal
Pending
...
```

- If the remaining CSRs are not approved, and are in the **Pending** status, approve the CSRs for your cluster machines:

- To approve them individually, run the following command for each valid CSR:

```
$ oc adm certificate approve <csr_name> 1
```

- 1** **<csr_name>** is the name of a CSR from the list of current CSRs.

- To approve all pending CSRs, run the following command:

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{"\n"}\n{{end}}' | xargs oc adm certificate approve
```

- After all client and server CSRs have been approved, the machines have the **Ready** status. Verify this by running the following command:

■

```
$ oc get nodes
```

Example output

```
NAME      STATUS    ROLES    AGE   VERSION
master-0  Ready    master   73m   v1.20.0
master-1  Ready    master   73m   v1.20.0
master-2  Ready    master   74m   v1.20.0
worker-0  Ready    worker   11m   v1.20.0
worker-1  Ready    worker   11m   v1.20.0
```



NOTE

It can take a few minutes after approval of the server CSRs for the machines to transition to the **Ready** status.

Additional information

- For more information on CSRs, see [Certificate Signing Requests](#).

1.1.14. Initial Operator configuration

After the control plane initializes, you must immediately configure some Operators so that they all become available.

Prerequisites

- Your control plane has initialized.

Procedure

1. Watch the cluster components come online:

```
$ watch -n5 oc get clusteroperators
```

| NAME | VERSION | AVAILABLE | PROGRESSING | DEGRADED | SINCE |
|-------------------------|---------|-----------|-------------|----------|-------|
| authentication | 4.4.0 | True | False | False | 69s |
| cloud-credential | 4.4.0 | True | False | False | 12m |
| cluster-autoscaler | 4.4.0 | True | False | False | 11m |
| console | 4.4.0 | True | False | False | 46s |
| dns | 4.4.0 | True | False | False | 11m |
| image-registry | 4.4.0 | True | False | False | 5m26s |
| ingress | 4.4.0 | True | False | False | 5m36s |
| kube-apiserver | 4.4.0 | True | False | False | 8m53s |
| kube-controller-manager | 4.4.0 | True | False | False | 7m24s |
| kube-scheduler | 4.4.0 | True | False | False | 12m |
| machine-api | 4.4.0 | True | False | False | 12m |
| machine-config | 4.4.0 | True | False | False | 7m36s |
| marketplace | 4.4.0 | True | False | False | 7m54m |
| monitoring | 4.4.0 | True | False | False | 7h54s |
| network | 4.4.0 | True | False | False | 5m9s |
| node-tuning | 4.4.0 | True | False | False | 11m |
| openshift-apiserver | 4.4.0 | True | False | False | 11m |

| | | | | | |
|------------------------------------|-------|------|-------|-------|--------|
| openshift-controller-manager | 4.4.0 | True | False | False | 5m943s |
| openshift-samples | 4.4.0 | True | False | False | 3m55s |
| operator-lifecycle-manager | 4.4.0 | True | False | False | 11m |
| operator-lifecycle-manager-catalog | 4.4.0 | True | False | False | 11m |
| service-ca | 4.4.0 | True | False | False | 11m |
| service-catalog-apiserver | 4.4.0 | True | False | False | 5m26s |
| service-catalog-controller-manager | 4.4.0 | True | False | False | 5m25s |
| storage | 4.4.0 | True | False | False | 5m30s |

2. Configure the Operators that are not available.

1.1.14.1. Image registry storage configuration

The Image Registry Operator is not initially available for platforms that do not provide default storage. After installation, you must configure your registry to use storage so the Registry Operator is made available.

Instructions for both configuring a persistent volume, which is required for production clusters, and for configuring an empty directory as the storage location, which is available for only non-production clusters, are shown.

1.1.14.1.1. Configuring registry storage for bare metal

As a cluster administrator, following installation you must configure your registry to use storage.

Prerequisites

- Cluster administrator permissions.
- A cluster on bare metal.
- Persistent storage provisioned for your cluster, such as Red Hat OpenShift Container Storage.



IMPORTANT

OpenShift Container Platform supports **ReadWriteOnce** access for image registry storage when you have only one replica. To deploy an image registry that supports high availability with two or more replicas, **ReadWriteMany** access is required.

- Must have 100Gi capacity.

Procedure

1. To configure your registry to use storage, change the **spec.storage.pvc** in the **configs.imageregistry/cluster** resource.



NOTE

When using shared storage, review your security settings to prevent outside access.

2. Verify that you do not have a registry pod:
 -

```
$ oc get pod -n openshift-image-registry
```

**NOTE**

If the storage type is **emptyDIR**, the replica number cannot be greater than **1**.

3. Check the registry configuration:

```
$ oc edit configs.imageregistry.operator.openshift.io
```

Example output

```
storage:
  pvc:
    claim:
```

Leave the **claim** field blank to allow the automatic creation of an **image-registry-storage** PVC.

4. Check the **clusteroperator** status:

```
$ oc get clusteroperator image-registry
```

1.1.14.1.2. Configuring storage for the image registry in non-production clusters

You must configure storage for the Image Registry Operator. For non-production clusters, you can set the image registry to an empty directory. If you do so, all images are lost if you restart the registry.

Procedure

- To set the image registry storage to an empty directory:

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec": {"storage":{"emptyDir":{}}}'
```

**WARNING**

Configure this option for only non-production clusters.

If you run this command before the Image Registry Operator initializes its components, the **oc patch** command fails with the following error:

```
Error from server (NotFound): configs.imageregistry.operator.openshift.io "cluster" not found
```

Wait a few minutes and run the command again.

1.1.15. Completing installation on user-provisioned infrastructure

After you complete the Operator configuration, you can finish installing the cluster on infrastructure that you provide.

Prerequisites

- Your control plane has initialized.
- You have completed the initial Operator configuration.

Procedure

1. Confirm that all the cluster components are online:

```
$ watch -n5 oc get clusteroperators
```

| NAME | VERSION | AVAILABLE | PROGRESSING | DEGRADED | SINCE |
|--|---------|-----------|-------------|----------|-------|
| authentication | 4.4.3 | True | False | False | 7m56s |
| cloud-credential | 4.4.3 | True | False | False | 31m |
| cluster-autoscaler | 4.4.3 | True | False | False | 16m |
| console | 4.4.3 | True | False | False | 10m |
| csi-snapshot-controller | 4.4.3 | True | False | False | 16m |
| dns | 4.4.3 | True | False | False | 22m |
| etcd | 4.4.3 | False | False | False | 25s |
| image-registry | 4.4.3 | True | False | False | 16m |
| ingress | 4.4.3 | True | False | False | 16m |
| insights | 4.4.3 | True | False | False | 17m |
| kube-apiserver | 4.4.3 | True | False | False | 19m |
| kube-controller-manager | 4.4.3 | True | False | False | 20m |
| kube-scheduler | 4.4.3 | True | False | False | 20m |
| kube-storage-version-migrator | 4.4.3 | True | False | False | 16m |
| machine-api | 4.4.3 | True | False | False | 22m |
| machine-config | 4.4.3 | True | False | False | 22m |
| marketplace | 4.4.3 | True | False | False | 16m |
| monitoring | 4.4.3 | True | False | False | 10m |
| network | 4.4.3 | True | False | False | 23m |
| node-tuning | 4.4.3 | True | False | False | 23m |
| openshift-apiserver | 4.4.3 | True | False | False | 17m |
| openshift-controller-manager | 4.4.3 | True | False | False | 15m |
| openshift-samples | 4.4.3 | True | False | False | 16m |
| operator-lifecycle-manager | 4.4.3 | True | False | False | 22m |
| operator-lifecycle-manager-catalog | 4.4.3 | True | False | False | 22m |
| operator-lifecycle-manager-packageserver | 4.4.3 | True | False | False | 18m |
| service-ca | 4.4.3 | True | False | False | 23m |
| service-catalog-apiserver | 4.4.3 | True | False | False | 23m |
| service-catalog-controller-manager | 4.4.3 | True | False | False | 23m |
| storage | 4.4.3 | True | False | False | 17m |

When all of the cluster Operators are **AVAILABLE**, you can complete the installation.

2. Monitor for cluster completion:

```
$ ./openshift-install --dir=<installation_directory> wait-for install-complete 1
INFO Waiting up to 30m0s for the cluster to initialize...
```

- 1 For `<installation_directory>`, specify the path to the directory that you stored the installation files in.

The command succeeds when the Cluster Version Operator finishes deploying the OpenShift Container Platform cluster from Kubernetes API server.



IMPORTANT

The Ignition config files that the installation program generates contain certificates that expire after 24 hours, which are then renewed at that time. If the cluster is shut down before renewing the certificates and the cluster is later restarted after the 24 hours have elapsed, the cluster automatically recovers the expired certificates. The exception is that you must manually approve the pending **node-bootstrapper** certificate signing requests (CSRs) to recover kubelet certificates. See the documentation for *Recovering from expired control plane certificates* for more information.

3. Confirm that the Kubernetes API server is communicating with the pods.
 - a. To view a list of all pods, use the following command:

```
$ oc get pods --all-namespaces
```

| NAMESPACE | NAME | READY | STATUS |
|-----------------------------------|---|-------|---------|
| openshift-apiserver-operator | openshift-apiserver-operator-85cb746d55-zqhs8 | 1/1 | Running |
| openshift-apiserver | apiserver-67b9g | 1/1 | Running |
| openshift-apiserver | apiserver-ljcmx | 1/1 | Running |
| openshift-apiserver | apiserver-z25h4 | 1/1 | Running |
| openshift-authentication-operator | authentication-operator-69d5d8bf84-vh2n8 | 1/1 | Running |
| ... | ... | ... | ... |

- b. View the logs for a pod that is listed in the output of the previous command by using the following command:

```
$ oc logs <pod_name> -n <namespace> 1
```

- 1 Specify the pod name and namespace, as shown in the output of the previous command.

If the pod logs display, the Kubernetes API server can communicate with the cluster machines.

1.1.16. Next steps

- [Customize your cluster.](#)
- If necessary, you can [opt out of remote health reporting](#).

1.2. INSTALLING A CLUSTER ON IBM POWER IN A RESTRICTED NETWORK

In OpenShift Container Platform version 4.4, you can install a cluster on IBM Power infrastructure that you provision in a restricted network.



IMPORTANT

Additional considerations exist for non-bare metal platforms. Review the information in the [guidelines for deploying OpenShift Container Platform on non-tested platforms](#) before you install an OpenShift Container Platform cluster.

1.2.1. Prerequisites

- [Create a mirror registry on your bastion host](#) and obtain the **imageContentSources** data for your version of OpenShift Container Platform.



IMPORTANT

Because the installation media is on the bastion host, use that computer to complete all installation steps.

- Provision [persistent storage](#) for your cluster. To deploy a private image registry, your storage must provide **ReadWriteMany** access modes.
- Review details about the [OpenShift Container Platform installation and update](#) processes.
- If you use a firewall and plan to use telemetry, you must [configure the firewall to allow the sites](#) that your cluster requires access to.



NOTE

Be sure to also review this site list if you are configuring a proxy.

1.2.2. About installations in restricted networks

In OpenShift Container Platform 4.4, you can perform an installation that does not require an active connection to the Internet to obtain software components. You complete an installation in a restricted network on only infrastructure that you provision, not infrastructure that the installation program provisions, so your platform selection is limited.

To complete a restricted network installation, you must create a registry that mirrors the contents of the OpenShift Container Platform registry and contains the installation media. You can create this registry on a mirror host, which can access both the Internet and your closed network, or by using other methods that meet your restrictions.



IMPORTANT

Restricted network installations always use user-provisioned infrastructure. Because of the complexity of the configuration for user-provisioned installations, consider completing a standard user-provisioned infrastructure installation before you attempt a restricted network installation. Completing this test installation might make it easier to isolate and troubleshoot any issues that might arise during your installation in a restricted network.

1.2.2.1. Additional limits

Clusters in restricted networks have the following additional limitations and restrictions:

- The **ClusterVersion** status includes an **Unable to retrieve available updates** error.
- By default, you cannot use the contents of the Developer Catalog because you cannot access the required image stream tags.

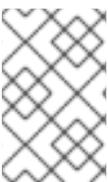
1.2.3. Machine requirements for a cluster with user-provisioned infrastructure

For a cluster that contains user-provisioned infrastructure, you must deploy all of the required machines.

1.2.3.1. Required machines

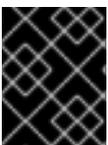
The smallest OpenShift Container Platform clusters require the following hosts:

- One temporary bootstrap machine
- Three control plane, or master, machines
- At least two compute machines, which are also known as worker machines



NOTE

The cluster requires the bootstrap machine to deploy the OpenShift Container Platform cluster on the three control plane machines. You can remove the bootstrap machine after you install the cluster.



IMPORTANT

To maintain high availability of your cluster, use separate physical hosts for these cluster machines.

The bootstrap, control plane, and compute machines must use the Red Hat Enterprise Linux CoreOS (RHCOS) as the operating system.

Note that RHCOS is based on Red Hat Enterprise Linux 8 and inherits all of its hardware certifications and requirements. See [Red Hat Enterprise Linux technology capabilities and limits](#) .

1.2.3.2. Network connectivity requirements

All the Red Hat Enterprise Linux CoreOS (RHCOS) machines require network in **initramfs** during boot to fetch Ignition config files from the Machine Config Server. During the initial boot, the machines require either a DHCP server or that static IP addresses be set in order to establish a network

connection to download their Ignition config files.

1.2.3.3. Minimum resource requirements

Each cluster machine must meet the following minimum requirements:

| Machine | Operating System | vCPU ¹ | Virtual RAM | Storage |
|---------------|------------------|-------------------|-------------|---------|
| Bootstrap | RHCOS | 2 | 16 GB | 120 GB |
| Control plane | RHCOS | 2 | 16 GB | 120 GB |
| Compute | RHCOS | 2 | 8 GB | 120 GB |

¹1 physical core provides 2 vCPUs when hyper-threading is enabled. 1 physical core provides 1 vCPU when hyper-threading is not enabled.

1.2.3.4. Certificate signing requests management

Because your cluster has limited access to automatic machine management when you use infrastructure that you provision, you must provide a mechanism for approving cluster certificate signing requests (CSRs) after installation. The **kube-controller-manager** only approves the kubelet client CSRs. The **machine-approver** cannot guarantee the validity of a serving certificate that is requested by using kubelet credentials because it cannot confirm that the correct machine issued the request. You must determine and implement a method of verifying the validity of the kubelet serving certificate requests and approving them.

1.2.4. Creating the user-provisioned infrastructure

Before you deploy an OpenShift Container Platform cluster that uses user-provisioned infrastructure, you must create the underlying infrastructure.

Prerequisites

- Review the [OpenShift Container Platform 4.x Tested Integrations](#) page before you create the supporting infrastructure for your cluster.

Procedure

1. Configure DHCP or set static IP addresses on each node.
2. Provision the required load balancers.
3. Configure the ports for your machines.
4. Configure DNS.
5. Ensure network connectivity.

1.2.4.1. Networking requirements for user-provisioned infrastructure

All the Red Hat Enterprise Linux CoreOS (RHCOS) machines require network in **iniramfs** during boot to fetch Ignition config from the machine config server.

During the initial boot, the machines require either a DHCP server or that static IP addresses be set on each host in the cluster in order to establish a network connection, which allows them to download their Ignition config files.

It is recommended to use the DHCP server to manage the machines for the cluster long-term. Ensure that the DHCP server is configured to provide persistent IP addresses and host names to the cluster machines.

The Kubernetes API server, which runs on each master node after a successful cluster installation, must be able to resolve the node names of the cluster machines. If the API servers and worker nodes are in different zones, you can configure a default DNS search zone to allow the API server to resolve the node names. Another supported approach is to always refer to hosts by their fully-qualified domain names in both the node objects and all DNS requests.

You must configure the network connectivity between machines to allow cluster components to communicate. Each machine must be able to resolve the host names of all other machines in the cluster.

Table 1.6. All machines to all machines

| Protocol | Port | Description |
|----------|--------------------|---|
| ICMP | N/A | Network reachability tests |
| TCP | 9000-9999 | Host level services, including the node exporter on ports 9100-9101 and the Cluster Version Operator on port 9099 . |
| | 10250-10259 | The default ports that Kubernetes reserves |
| | 10256 | openshift-sdn |
| UDP | 4789 | VXLAN and Geneve |
| | 6081 | VXLAN and Geneve |
| | 9000-9999 | Host level services, including the node exporter on ports 9100-9101 . |
| TCP/UDP | 30000-32767 | Kubernetes node port |

Table 1.7. All machines to control plane

| Protocol | Port | Description |
|----------|------------------|--------------------------------------|
| TCP | 2379-2380 | etcd server, peer, and metrics ports |
| | 6443 | Kubernetes API |

Network topology requirements

The infrastructure that you provision for your cluster must meet the following network topology requirements.

Load balancers

Before you install OpenShift Container Platform, you must provision two load balancers that meet the following requirements:

1. **API load balancer.** Provides a common endpoint for users, both human and machine, to interact with and configure the platform. Configure the following conditions:
 - Layer 4 load balancing only. This can be referred to as Raw TCP, SSL Passthrough, or SSL Bridge mode. If you use SSL Bridge mode, you must enable Server Name Indication (SNI) for the API routes.
 - A stateless load balancing algorithm. The options vary based on the load balancer implementation.



NOTE

Session persistence is not required for the API load balancer to function properly.

Configure the following ports on both the front and back of the load balancers:

Table 1.8. API load balancer

| Port | Back-end machines (pool members) | Internal | External | Description |
|--------------|---|----------|----------|-----------------------|
| 6443 | Bootstrap and control plane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane. You must configure the /readyz endpoint for the API server health check probe. | X | X | Kubernetes API server |
| 22623 | Bootstrap and control plane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane. | X | | Machine config server |



NOTE

The load balancer must be configured to take a maximum of 30 seconds from the time the API server turns off the **/readyz** endpoint to the removal of the API server instance from the pool. Within the time frame after **/readyz** returns an error or becomes healthy, the endpoint must have been removed or added. Probing every 5 or 10 seconds, with two successful requests to become healthy and three to become unhealthy, are well-tested values.

2. **Application Ingress load balancer.** Provides an Ingress point for application traffic flowing in from outside the cluster. Configure the following conditions:

- Layer 4 load balancing only. This can be referred to as Raw TCP, SSL Passthrough, or SSL Bridge mode. If you use SSL Bridge mode, you must enable Server Name Indication (SNI) for the Ingress routes.
- A connection-based or session-based persistence is recommended, based on the options available and types of applications that will be hosted on the platform.

Configure the following ports on both the front and back of the load balancers:

Table 1.9. Application Ingress load balancer

| Port | Back-end machines (pool members) | Internal | External | Description |
|------------|--|----------|----------|---------------|
| 443 | The machines that run the Ingress router pods, compute, or worker, by default. | X | X | HTTPS traffic |
| 80 | The machines that run the Ingress router pods, compute, or worker, by default. | X | X | HTTP traffic |

TIP

If the true IP address of the client can be seen by the load balancer, enabling source IP-based session persistence can improve performance for applications that use end-to-end TLS encryption.



NOTE

A working configuration for the Ingress router is required for an OpenShift Container Platform cluster. You must configure the Ingress router after the control plane initializes.

1.2.4.2. User-provisioned DNS requirements

DNS is used for name resolution and reverse name resolution. DNS A/AAAA or CNAME records are used for name resolution and PTR records are used for reverse name resolution. The reverse records are important because Red Hat Enterprise Linux CoreOS (RHCOS) uses the reverse records to set the host name for all the nodes. Additionally, the reverse records are used to generate the certificate signing requests (CSR) that OpenShift Container Platform needs to operate.

The following DNS records are required for an OpenShift Container Platform cluster that uses user-provisioned infrastructure. In each record, **<cluster_name>** is the cluster name and **<base_domain>** is the cluster base domain that you specify in the **install-config.yaml** file. A complete DNS record takes the form: **<component>.<cluster_name>.<base_domain>..**

Table 1.10. Required DNS records

| Component | Record | Description |
|-----------|--------|-------------|
|-----------|--------|-------------|

| Component | Record | Description |
|----------------|---|---|
| Kubernetes API | api.<cluster_name>.<base_domain> | Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the load balancer for the control plane machines. These records must be resolvable by both clients external to the cluster and from all the nodes within the cluster. |
| | api-int.<cluster_name>.<base_domain> | Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the load balancer for the control plane machines. These records must be resolvable from all the nodes within the cluster.  IMPORTANT The API server must be able to resolve the worker nodes by the host names that are recorded in Kubernetes. If the API server cannot resolve the node names, then proxied API calls can fail, and you cannot retrieve logs from pods. |
| Routes | *.apps.<cluster_name>.<base_domain> | Add a wildcard DNS A/AAAA or CNAME record that refers to the load balancer that targets the machines that run the Ingress router pods, which are the worker nodes by default. These records must be resolvable by both clients external to the cluster and from all the nodes within the cluster. |
| Bootstrap | bootstrap.<cluster_name>.<base_domain> | Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the bootstrap machine. These records must be resolvable by the nodes within the cluster. |
| Master hosts | <master><n>.<cluster_name>.<base_domain> | Add DNS A/AAAA or CNAME records and DNS PTR records to identify each machine for the master nodes. These records must be resolvable by the nodes within the cluster. |
| Worker hosts | <worker><n>.<cluster_name>.<base_domain> | Add DNS A/AAAA or CNAME records and DNS PTR records to identify each machine for the worker nodes. These records must be resolvable by the nodes within the cluster. |

TIP

You can use the **nslookup <hostname>** command to verify name resolution. You can use the **dig -x <ip_address>** command to verify reverse name resolution for the PTR records.

The following example of a BIND zone file shows sample A records for name resolution. The purpose of the example is to show the records that are needed. The example is not meant to provide advice for choosing one name resolution service over another.

■

Example 1.3. Sample DNS zone database

```

$TTL 1W
@ IN SOA ns1.example.com. root (
    2019070700 ; serial
    3H ; refresh (3 hours)
    30M ; retry (30 minutes)
    2W ; expiry (2 weeks)
    1W ) ; minimum (1 week)
IN NS ns1.example.com.
IN MX 10 smtp.example.com.
;
;
ns1 IN A 192.168.1.5
smtp IN A 192.168.1.5
;
helper IN A 192.168.1.5
helper.ocp4 IN A 192.168.1.5
;
; The api identifies the IP of your load balancer.
api.ocp4 IN A 192.168.1.5
api-int.ocp4 IN A 192.168.1.5
;
; The wildcard also identifies the load balancer.
*.apps.ocp4 IN A 192.168.1.5
;
; Create an entry for the bootstrap host.
bootstrap.ocp4 IN A 192.168.1.96
;
; Create entries for the master hosts.
master0.ocp4 IN A 192.168.1.97
master1.ocp4 IN A 192.168.1.98
master2.ocp4 IN A 192.168.1.99
;
; Create entries for the worker hosts.
worker0.ocp4 IN A 192.168.1.11
worker1.ocp4 IN A 192.168.1.7
;
;EOF

```

The following example BIND zone file shows sample PTR records for reverse name resolution.

Example 1.4. Sample DNS zone database for reverse records

```

$TTL 1W
@ IN SOA ns1.example.com. root (
    2019070700 ; serial
    3H ; refresh (3 hours)
    30M ; retry (30 minutes)
    2W ; expiry (2 weeks)
    1W ) ; minimum (1 week)
IN NS ns1.example.com.
;
; The syntax is "last octet" and the host must have an FQDN

```

```

; with a trailing dot.
97 IN PTR master0.ocp4.example.com.
98 IN PTR master1.ocp4.example.com.
99 IN PTR master2.ocp4.example.com.
;
96 IN PTR bootstrap.ocp4.example.com.
;
5 IN PTR api.ocp4.ocp4.example.com.
5 IN PTR api-int.ocp4.ocp4.example.com.
;
11 IN PTR worker0.ocp4.example.com.
7 IN PTR worker1.ocp4.example.com.
;
;EOF

```

1.2.5. Generating an SSH private key and adding it to the agent

If you want to perform installation debugging or disaster recovery on your cluster, you must provide an SSH key to both your **ssh-agent** and the installation program. You can use this key to access the bootstrap machine in a public cluster to troubleshoot installation issues.



NOTE

In a production environment, you require disaster recovery and debugging.

You can use this key to SSH into the master nodes as the user **core**. When you deploy the cluster, the key is added to the **core** user's `~/.ssh/authorized_keys` list.



NOTE

You must use a local key, not one that you configured with platform-specific approaches such as [AWS key pairs](#).

Procedure

1. If you do not have an SSH key that is configured for password-less authentication on your computer, create one. For example, on a computer that uses a Linux operating system, run the following command:

```
$ ssh-keygen -t ed25519 -N "" \
-f <path>/<file_name> 1
```

- 1 Specify the path and file name, such as `~/.ssh/id_rsa`, of the SSH key. Do not specify an existing SSH key, as it will be overwritten.

Running this command generates an SSH key that does not require a password in the location that you specified.

2. Start the **ssh-agent** process as a background task:

```
$ eval "$(ssh-agent -s)"
```

```
Agent pid 31874
```

3. Add your SSH private key to the **ssh-agent**:

```
$ ssh-add <path>/<file_name> 1
```

```
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

- 1** Specify the path and file name for your SSH private key, such as `~/.ssh/id_rsa`

Next steps

- When you install OpenShift Container Platform, provide the SSH public key to the installation program.

1.2.6. Manually creating the installation configuration file

For installations of OpenShift Container Platform that use user-provisioned infrastructure, you must manually generate your installation configuration file.

Prerequisites

- Obtain the OpenShift Container Platform installation program and the access token for your cluster.

Procedure

1. Create an installation directory to store your required installation assets in:

```
$ mkdir <installation_directory>
```



IMPORTANT

You must create a directory. Some installation assets, like bootstrap X.509 certificates have short expiration intervals, so you must not reuse an installation directory. If you want to reuse individual files from another cluster installation, you can copy them into your directory. However, the file names for the installation assets might change between releases. Use caution when copying installation files from an earlier OpenShift Container Platform version.

2. Customize the following **install-config.yaml** file template and save it in the **<installation_directory>**.



NOTE

You must name this configuration file **install-config.yaml**.

3. Back up the **install-config.yaml** file so that you can use it to install multiple clusters.



IMPORTANT

The **install-config.yaml** file is consumed during the next step of the installation process. You must back it up now.

1.2.6.1. Sample install-config.yaml file for IBM Power

You can customize the **install-config.yaml** file to specify more details about your OpenShift Container Platform cluster's platform or modify the values of the required parameters.

```

apiVersion: v1
baseDomain: example.com 1
compute:
- hyperthreading: Enabled 2 3
  name: worker
  replicas: 0 4
controlPlane:
  hyperthreading: Enabled 5 6
  name: master 7
  replicas: 3 8
metadata:
  name: test 9
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14 10
    hostPrefix: 23 11
  networkType: OpenShiftSDN
  serviceNetwork: 12
  - 172.30.0.0/16
platform:
  none: {} 13
fips: false 14
pullSecret: '{"auths": ...}' 15
sshKey: 'ssh-ed25519 AAAA...' 16

```

- 1 The base domain of the cluster. All DNS records must be sub-domains of this base and include the cluster name.
- 2 5 The **controlPlane** section is a single mapping, but the compute section is a sequence of mappings. To meet the requirements of the different data structures, the first line of the **compute** section must begin with a hyphen, -, and the first line of the **controlPlane** section must not. Although both sections currently define a single machine pool, it is possible that future versions of OpenShift Container Platform will support defining multiple compute pools during installation. Only one control plane pool is used.
- 3 6 7 Whether to enable or disable simultaneous multithreading, or **hyperthreading**. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores. You can disable it by setting the parameter value to **Disabled**. If you disable simultaneous multithreading in some cluster machines, you must disable it in all cluster machines.



IMPORTANT

If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.

- 4 You must set the value of the **replicas** parameter to **0**. This parameter controls the number of workers that the cluster creates and manages for you, which are functions that the cluster does not perform when you use user-provisioned infrastructure. You must manually deploy worker machines for the cluster to use before you finish installing OpenShift Container Platform.
- 8 The number of control plane machines that you add to the cluster. Because the cluster uses this values as the number of etcd endpoints in the cluster, the value must match the number of control plane machines that you deploy.
- 9 The cluster name that you specified in your DNS records.
- 10 A block of IP addresses from which pod IP addresses are allocated. This block must not overlap with existing physical networks. These IP addresses are used for the pod network. If you need to access the pods from an external network, you must configure load balancers and routers to manage the traffic.
- 11 The subnet prefix length to assign to each individual node. For example, if **hostPrefix** is set to **23**, then each node is assigned a **/23** subnet out of the given **cidr**, which allows for 510 ($2^{(32 - 23)} - 2$) pod IPs addresses. If you are required to provide access to nodes from an external network, configure load balancers and routers to manage the traffic.
- 12 The IP address pool to use for service IP addresses. You can enter only one IP address pool. If you need to access the services from an external network, configure load balancers and routers to manage the traffic.
- 13 You must set the platform to **none**. You cannot provide additional platform configuration variables for IBM Power infrastructure.
- 14 Whether to enable or disable FIPS mode. By default, FIPS mode is not enabled. If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.
- 15 The pull secret that you obtained from the [Pull Secret](#) page on the Red Hat OpenShift Cluster Manager site. This pull secret allows you to authenticate with the services that are provided by the included authorities, including Quay.io, which serves the container images for OpenShift Container Platform components.
- 16 The public portion of the default SSH key for the **core** user in Red Hat Enterprise Linux CoreOS (RHCOS).



NOTE

For production OpenShift Container Platform clusters on which you want to perform installation debugging or disaster recovery, specify an SSH key that your **ssh-agent** process uses.

1.2.6.2. Configuring the cluster-wide proxy during installation

Production environments can deny direct access to the Internet and instead have an HTTP or HTTPS proxy available. You can configure a new OpenShift Container Platform cluster to use a proxy by configuring the proxy settings in the **install-config.yaml** file.

Prerequisites

- An existing **install-config.yaml** file.
- Review the sites that your cluster requires access to and determine whether any need to bypass the proxy. By default, all cluster egress traffic is proxied, including calls to hosting cloud provider APIs. Add sites to the **Proxy** object's **spec.noProxy** field to bypass the proxy if necessary.



NOTE

The **Proxy** object **status.noProxy** field is populated with the values of the **networking.machineNetwork[].cidr**, **networking.clusterNetwork[].cidr**, and **networking.serviceNetwork[]** fields from your installation configuration.

For installations on Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure, and Red Hat OpenStack Platform (RHOSP), the **Proxy** object **status.noProxy** field is also populated with the instance metadata endpoint (**169.254.169.254**).

Procedure

1. Edit your **install-config.yaml** file and add the proxy settings. For example:

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: http://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
  additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  ...
```

- 1 A proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**. If you use an MITM transparent proxy network that does not require additional proxy configuration but requires additional CAs, you must not specify an **httpProxy** value.
- 2 A proxy URL to use for creating HTTPS connections outside the cluster. If this field is not specified, then **httpProxy** is used for both HTTP and HTTPS connections. If you use an MITM transparent proxy network that does not require additional proxy configuration but requires additional CAs, you must not specify an **httpsProxy** value.
- 3 A comma-separated list of destination domain names, domains, IP addresses, or other network CIDRs to exclude proxying. Preface a domain with **.** to include all subdomains of that domain. Use ***** to bypass proxy for all destinations.
- 4 If provided, the installation program generates a config map that is named **user-ca-bundle**

**NOTE**

The installation program does not support the proxy **readinessEndpoints** field.

2. Save the file and reference it when installing OpenShift Container Platform.

The installation program creates a cluster-wide proxy that is named **cluster** that uses the proxy settings in the provided **install-config.yaml** file. If no proxy settings are provided, a **cluster Proxy** object is still created, but it will have a nil **spec**.

**NOTE**

Only the **Proxy** object named **cluster** is supported, and no additional proxies can be created.

1.2.7. Creating the Kubernetes manifest and Ignition config files

Because you must modify some cluster definition files and manually start the cluster machines, you must generate the Kubernetes manifest and Ignition config files that the cluster needs to make its machines.

**IMPORTANT**

The Ignition config files that the installation program generates contain certificates that expire after 24 hours, which are then renewed at that time. If the cluster is shut down before renewing the certificates and the cluster is later restarted after the 24 hours have elapsed, the cluster automatically recovers the expired certificates. The exception is that you must manually approve the pending **node-bootstrapper** certificate signing requests (CSRs) to recover kubelet certificates. See the documentation for *Recovering from expired control plane certificates* for more information.

Prerequisites

- Obtain the OpenShift Container Platform installation program.
- Create the **install-config.yaml** installation configuration file.

Procedure

1. Generate the Kubernetes manifests for the cluster:

```
$ ./openshift-install create manifests --dir=<installation_directory> 1
```

```
INFO Consuming Install Config from target directory
WARNING Making control-plane schedulable by setting MastersSchedulable to true for
Scheduler cluster settings
```

- 1** For **<installation_directory>**, specify the installation directory that contains the **install-config.yaml** file you created.

Because you create your own compute machines later in the installation process, you can safely ignore this warning.

2. Modify the `<installation_directory>/manifests/cluster-scheduler-02-config.yml` Kubernetes manifest file to prevent pods from being scheduled on the control plane machines:
 - a. Open the `<installation_directory>/manifests/cluster-scheduler-02-config.yml` file.
 - b. Locate the `mastersSchedulable` parameter and set its value to **False**.
 - c. Save and exit the file.



NOTE

Currently, due to a [Kubernetes limitation](#), router Pods running on control plane machines will not be reachable by the ingress load balancer. This step might not be required in a future minor version of OpenShift Container Platform.

3. Obtain the Ignition config files:

```
$ ./openshift-install create ignition-configs --dir=<installation_directory> 1
```

- 1** For `<installation_directory>`, specify the same installation directory.

The following files are generated in the directory:

```
.
├── auth
│   ├── kubeadmin-password
│   └── kubeconfig
├── bootstrap.ign
├── master.ign
├── metadata.json
└── worker.ign
```

1.2.8. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines

Before you install a cluster on IBM Power infrastructure that you provision, you must create RHCOS machines for it to use. Follow either the steps to use an ISO image or network PXE booting to create the machines.

1.2.8.1. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines using an ISO image

Before you install a cluster on IBM Power infrastructure that you provision, you must create RHCOS machines for it to use. You can use an ISO image to create the machines.

Prerequisites

- Obtain the Ignition config files for your cluster.
- Have access to an HTTP server that you can access from your computer and that the machines that you create can access.

Procedure

1. Upload the control plane, compute, and bootstrap Ignition config files that the installation program created to your HTTP server. Note the URLs of these files.



IMPORTANT

If you plan to add more compute machines to your cluster after you finish installation, do not delete these files.

2. Obtain the RHCOS images that are required for your preferred method of installing operating system instances from the [Product Downloads](#) page on the Red Hat customer portal or the [RHCOS image mirror](#) page.



IMPORTANT

The RHCOS images might not change with every release of OpenShift Container Platform. You must download images with the highest version that is less than or equal to the OpenShift Container Platform version that you install. Use the image versions that match your OpenShift Container Platform version if they are available. Only use ISO images for this procedure. RHCOS qcow2 images are not supported for bare metal installs.

You must download the ISO file and the RAW disk file. Those file names resemble the following examples:

- ISO: **rhcos-<version>-installer.<architecture>.iso**
 - Compressed metal RAW: **rhcos-<version>-metal.<architecture>.raw.gz**
3. Upload either the RAW RHCOS image file to your HTTP server and note its URL.



IMPORTANT

If you plan to add more compute machines to your cluster after you finish installation, do not delete these files.

4. Use the ISO to start the RHCOS installation. Use one of the following installation options:
 - Burn the ISO image to a disk and boot it directly.
 - Use ISO redirection via a LOM interface.
5. After the instance boots, press the **TAB** or **E** key to edit the kernel command line.
6. Add the parameters to the kernel command line:

```
coreos.inst=yes
coreos.inst.install_dev=sda 1
coreos.inst.image_url=<image_URL> 2
coreos.inst.ignition_url=http://example.com/config.ign 3
ip=<dhcp or static IP address> 4 5
bond=<bonded_interface> 6
```

- 1** Specify the block device of the system to install to.

2. Specify the URL of the RAW image that you uploaded to your server.
 3. Specify the URL of the Ignition config file for this machine type.
 4. Set **ip=dhcp** or set an individual static IP address (**ip=**) and DNS server (**nameserver=**) on each node. See *Static IP address examples for RHCOS kernel parameters* for details.
 5. If you use multiple network interfaces or DNS servers, see *Static IP address examples for RHCOS kernel parameters* for details on how to configure them.
 6. Optionally, you can bond multiple network interfaces to a single interface using the **bond=** option, as described in *Static IP address examples for RHCOS kernel parameters*.
7. Press Enter to complete the installation. After RHCOS installs, the system reboots. After the system reboots, it applies the Ignition config file that you specified.
 8. Continue to create the machines for your cluster.



IMPORTANT

You must create the bootstrap and control plane machines at this time. Because some pods are deployed on compute machines by default, also create at least two compute machines before you install the cluster.

1.2.8.2. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines by PXE booting

Before you install a cluster on IBM Power infrastructure that you provision, you must create RHCOS machines for it to use. You can use PXE booting to create the machines.

Prerequisites

- Obtain the Ignition config files for your cluster.
- Have access to an HTTP server and TFTP server that you can access from your computer.

Procedure

1. Upload the master, worker, and bootstrap Ignition config files that the installation program created to your HTTP server. Note the URLs of these files.



IMPORTANT

If you plan to add more compute machines to your cluster after you finish installation, do not delete these files.

2. Obtain the compressed metal RAW image, **kernel** and **initramfs** files from the [Product Downloads](#) page on the Red Hat customer portal or the [RHCOS image mirror](#) page.



IMPORTANT

The RHCOS images might not change with every release of OpenShift Container Platform. You must download images with the highest version that is less than or equal to the OpenShift Container Platform version that you install. Use the image versions that match your OpenShift Container Platform version if they are available. Only use RAW images for this procedure. RHCOS qcow2 images are not supported for bare metal installs.

The file names contain the OpenShift Container Platform version number. They resemble the following examples:

- Compressed metal RAW image: **rhcos-`<version>`-`<architecture>`-metal.`<architecture>`.raw.gz**
 - **kernel:** **rhcos-`<version>`-`<architecture>`-installer-kernel-`<architecture>`**
 - **initramfs:** **rhcos-`<version>`-`<architecture>`-installer-initramfs.`<architecture>`.img**
3. Upload the RAW image to your HTTP server.
 4. Upload the additional files that are required for your booting method:
 - For traditional PXE, upload the **kernel** and **initramfs** files to your TFTP server.
 - For iPXE, upload the **kernel** and **initramfs** files to your HTTP server.



IMPORTANT

If you plan to add more compute machines to your cluster after you finish installation, do not delete these files.

5. Configure the network boot infrastructure so that the machines boot from their local disks after RHCOS is installed on them.
6. Configure PXE installation for the RHCOS images.
Modify one of the following example menu entries for your environment and verify that the image and Ignition files are properly accessible:

- For PXE:

```

DEFAULT pxeboot
TIMEOUT 20
PROMPT 0
LABEL pxeboot
    KERNEL rhcos-<version>-<architecture>-installer-kernel-<architecture> 1
    APPEND ip=dhcp rd.neednet=1 initrd=rhcos-<version>-<architecture>-installer-
initramfs.<architecture>.img console=tty0 console=ttyS0 coreos.inst=yes
coreos.inst.install_dev=sda coreos.inst.image_url=http://<HTTP_server>/rhcos-
<version>-<architecture>-metal.<architecture>.raw.gz
coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign 2 3
  
```

- 1** Specify the location of the **kernel** file available on your TFTP server.
- 2** If you use multiple NICs, specify a single interface in the **ip** option. For example, to use

- 3 Specify locations of the RHCOS files that you uploaded to your HTTP or TFTP server. The **initrd** parameter value is the location of the **initramfs** file on your TFTP server.

7. If you use UEFI, perform the following actions:

- a. Provide the EFI binaries and **grub.cfg** file that are required for booting the system. You need the **shim.efi** binary and the **grubx64.efi** binary.
 - Extract the necessary EFI binaries by mounting the RHCOS ISO on your host and then mounting the **images/efiboot.img** file to your host. From the **efiboot.img** mount point, you then copy the **EFI/redhat/shimx64.efi** and **EFI/redhat/grubx64.efi** files to your TFTP server.

```
# mkdir -p /mnt/{iso,efiboot}
# mount -o loop rhcos-installer.x86_64.iso /mnt/iso
# mount -o loop,ro /mnt/iso/images/efiboot.img /mnt/efiboot
# cp /mnt/efiboot/EFI/redhat/{shimx64.efi,grubx64.efi} .
# umount /mnt/{efiboot,iso}
```

- b. Copy the **EFI/redhat/grub.cfg** file that is included in the RHCOS ISO to your TFTP server.
- c. Edit the **grub.cfg** file to include the following arguments:

```
menuentry 'Install Red Hat Enterprise Linux CoreOS' --class fedora --class gnu-linux --
class gnu --class os {
  linux rhcos-<version>-<architecture>-installer-kernel-<architecture> nomodeset
  rd.neednet=1 coreos.inst=yes coreos.inst.install_dev=sda
  coreos.inst.image_url=http://<HTTP_server>/rhcos-<version>-<architecture>-metal.
<architecture>.raw.gz coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign 1
  initrd rhcos-<version>-<architecture>-installer-initramfs.<architecture>.img 2
}
```

- 1 The first argument to the **linux** line item is the location of the **kernel** file that you uploaded to your TFTP server. For the **coreos.inst.image_url** parameter value, specify the location of the compressed metal RAW image that you uploaded to your HTTP server. For the **coreos.inst.ignition_url** parameter, specify the location of the bootstrap Ignition config file that you uploaded to your HTTP server.

- 2 Specify the location of the **initramfs** file that you uploaded to your TFTP server.

8. Continue to create the machines for your cluster.



IMPORTANT

You must create the bootstrap and control plane machines at this time. Because some pods are deployed on compute machines by default, also create at least two compute machine before you install the cluster.

1.2.9. Creating the cluster

To create the OpenShift Container Platform cluster, you wait for the bootstrap process to complete on the machines that you provisioned by using the Ignition config files that you generated with the installation program.

Prerequisites

- Create the required infrastructure for the cluster.
- You obtained the installation program and generated the Ignition config files for your cluster.
- You used the Ignition config files to create RHCOS machines for your cluster.

Procedure

1. Monitor the bootstrap process:

```
$ ./openshift-install --dir=<installation_directory> wait-for bootstrap-complete \ 1
--log-level=info 2
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com...
INFO API v1.17.1 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
```

- 1 For **<installation_directory>**, specify the path to the directory that you stored the installation files in.
- 2 To view different installation details, specify **warn**, **debug**, or **error** instead of **info**.

The command succeeds when the Kubernetes API server signals that it has been bootstrapped on the control plane machines.

2. After bootstrap process is complete, remove the bootstrap machine from the load balancer.



IMPORTANT

You must remove the bootstrap machine from the load balancer at this point. You can also remove or reformat the machine itself.

1.2.10. Logging in to the cluster

You can log in to your cluster as a default system user by exporting the cluster **kubeconfig** file. The **kubeconfig** file contains information about the cluster that is used by the CLI to connect a client to the correct cluster and API server. The file is specific to a cluster and is created during OpenShift Container Platform installation.

Prerequisites

- Deploy an OpenShift Container Platform cluster.
- Install the **oc** CLI.

Procedure

1. Export the **kubeadmin** credentials:

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

- 1 For `<installation_directory>`, specify the path to the directory that you stored the installation files in.

2. Verify you can run `oc` commands successfully using the exported configuration:

```
$ oc whoami
system:admin
```

1.2.11. Approving the certificate signing requests for your machines

When you add machines to a cluster, two pending certificate signing requests (CSRs) are generated for each machine that you added. You must confirm that these CSRs are approved or, if necessary, approve them yourself. The client requests must be approved first, followed by the server requests.

Prerequisites

- You added machines to your cluster.

Procedure

1. Confirm that the cluster recognizes the machines:

```
# oc get nodes

NAME                STATUS  ROLES  AGE  VERSION
master-01.example.com Ready  master  40d  v1.17.1
master-02.example.com Ready  master  40d  v1.17.1
master-03.example.com Ready  master  40d  v1.17.1
worker-01.example.com Ready  worker  40d  v1.17.1
worker-02.example.com Ready  worker  40d  v1.17.1
```

The output lists all of the machines that you created.

2. Review the pending CSRs and ensure that you see the client requests with the **Pending** or **Approved** status for each machine that you added to the cluster:

```
$ oc get csr

NAME      AGE  REQUESTOR                                     CONDITION
csr-8b2br 15m  system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending
csr-8vnps 15m  system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending
...
```

In this example, two machines are joining the cluster. You might see more approved CSRs in the list.

3. If the CSRs were not approved, after all of the pending CSRs for the machines you added are in **Pending** status, approve the CSRs for your cluster machines:

**NOTE**

Because the CSRs rotate automatically, approve your CSRs within an hour of adding the machines to the cluster. If you do not approve them within an hour, the certificates will rotate, and more than two certificates will be present for each node. You must approve all of these certificates. After you approve the initial CSRs, the subsequent node client CSRs are automatically approved by the cluster **kube-controller-manager**. You must implement a method of automatically approving the kubelet serving certificate requests.

- To approve them individually, run the following command for each valid CSR:

```
$ oc adm certificate approve <csr_name> 1
```

- 1** **<csr_name>** is the name of a CSR from the list of current CSRs.

- To approve all pending CSRs, run the following command:

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{"\n"}{{end}}{{end}}' | xargs oc adm certificate approve
```

- Now that your client requests are approved, you must review the server requests for each machine that you added to the cluster:

```
$ oc get csr
```

Example output

```
NAME      AGE   REQUESTOR                                     CONDITION
csr-bfd72 5m26s system:node:ip-10-0-50-126.us-east-2.compute.internal
Pending
csr-c57lv 5m26s system:node:ip-10-0-95-157.us-east-2.compute.internal
Pending
...
```

- If the remaining CSRs are not approved, and are in the **Pending** status, approve the CSRs for your cluster machines:

- To approve them individually, run the following command for each valid CSR:

```
$ oc adm certificate approve <csr_name> 1
```

- 1** **<csr_name>** is the name of a CSR from the list of current CSRs.

- To approve all pending CSRs, run the following command:

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{"\n"}{{end}}{{end}}' | xargs oc adm certificate approve
```

- After all client and server CSRs have been approved, the machines have the **Ready** status. Verify this by running the following command:

■

```
$ oc get nodes
```

Example output

```
NAME      STATUS    ROLES    AGE   VERSION
master-0  Ready    master   73m   v1.20.0
master-1  Ready    master   73m   v1.20.0
master-2  Ready    master   74m   v1.20.0
worker-0  Ready    worker   11m   v1.20.0
worker-1  Ready    worker   11m   v1.20.0
```



NOTE

It can take a few minutes after approval of the server CSRs for the machines to transition to the **Ready** status.

Additional information

- For more information on CSRs, see [Certificate Signing Requests](#).

1.2.12. Initial Operator configuration

After the control plane initializes, you must immediately configure some Operators so that they all become available.

Prerequisites

- Your control plane has initialized.

Procedure

1. Watch the cluster components come online:

```
$ watch -n5 oc get clusteroperators
```

```
NAME              VERSION AVAILABLE PROGRESSING DEGRADED
SINCE
authentication    4.4.0 True      False      False      69s
cloud-credential  4.4.0 True      False      False      12m
cluster-autoscaler 4.4.0 True      False      False      11m
console           4.4.0 True      False      False      46s
dns               4.4.0 True      False      False      11m
image-registry    4.4.0 True      False      False      5m26s
ingress           4.4.0 True      False      False      5m36s
kube-apiserver    4.4.0 True      False      False      8m53s
kube-controller-manager 4.4.0 True      False      False      7m24s
kube-scheduler    4.4.0 True      False      False      12m
machine-api       4.4.0 True      False      False      12m
machine-config    4.4.0 True      False      False      7m36s
marketplace       4.4.0 True      False      False      7m54m
monitoring        4.4.0 True      False      False      7h54s
network           4.4.0 True      False      False      5m9s
node-tuning       4.4.0 True      False      False      11m
openshift-apiserver 4.4.0 True      False      False      11m
```

| | | | | | |
|------------------------------------|-------|------|-------|-------|--------|
| openshift-controller-manager | 4.4.0 | True | False | False | 5m943s |
| openshift-samples | 4.4.0 | True | False | False | 3m55s |
| operator-lifecycle-manager | 4.4.0 | True | False | False | 11m |
| operator-lifecycle-manager-catalog | 4.4.0 | True | False | False | 11m |
| service-ca | 4.4.0 | True | False | False | 11m |
| service-catalog-apiserver | 4.4.0 | True | False | False | 5m26s |
| service-catalog-controller-manager | 4.4.0 | True | False | False | 5m25s |
| storage | 4.4.0 | True | False | False | 5m30s |

2. Configure the Operators that are not available.

1.2.12.1. Image registry storage configuration

The Image Registry Operator is not initially available for platforms that do not provide default storage. After installation, you must configure your registry to use storage so the Registry Operator is made available.

Instructions for both configuring a persistent volume, which is required for production clusters, and for configuring an empty directory as the storage location, which is available for only non-production clusters, are shown.

1.2.12.1.1. Configuring registry storage for bare metal

As a cluster administrator, following installation you must configure your registry to use storage.

Prerequisites

- Cluster administrator permissions.
- A cluster on bare metal.
- Persistent storage provisioned for your cluster, such as Red Hat OpenShift Container Storage.



IMPORTANT

OpenShift Container Platform supports **ReadWriteOnce** access for image registry storage when you have only one replica. To deploy an image registry that supports high availability with two or more replicas, **ReadWriteMany** access is required.

- Must have 100Gi capacity.

Procedure

1. To configure your registry to use storage, change the **spec.storage.pvc** in the **configs.imageregistry/cluster** resource.

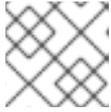


NOTE

When using shared storage, review your security settings to prevent outside access.

2. Verify that you do not have a registry pod:
 -

```
$ oc get pod -n openshift-image-registry
```

**NOTE**

If the storage type is **emptyDIR**, the replica number cannot be greater than **1**.

3. Check the registry configuration:

```
$ oc edit configs.imageregistry.operator.openshift.io
```

Example output

```
storage:
  pvc:
    claim:
```

Leave the **claim** field blank to allow the automatic creation of an **image-registry-storage** PVC.

4. Check the **clusteroperator** status:

```
$ oc get clusteroperator image-registry
```

1.2.12.1.2. Configuring storage for the image registry in non-production clusters

You must configure storage for the Image Registry Operator. For non-production clusters, you can set the image registry to an empty directory. If you do so, all images are lost if you restart the registry.

Procedure

- To set the image registry storage to an empty directory:

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec": {"storage":{"emptyDir":{}}}'
```

**WARNING**

Configure this option for only non-production clusters.

If you run this command before the Image Registry Operator initializes its components, the **oc patch** command fails with the following error:

```
Error from server (NotFound): configs.imageregistry.operator.openshift.io "cluster" not found
```

Wait a few minutes and run the command again.

1.2.13. Completing installation on user-provisioned infrastructure

After you complete the Operator configuration, you can finish installing the cluster on infrastructure that you provide.

Prerequisites

- Your control plane has initialized.
- You have completed the initial Operator configuration.

Procedure

1. Confirm that all the cluster components are online:

```
$ watch -n5 oc get clusteroperators
```

| NAME | VERSION | AVAILABLE | PROGRESSING | DEGRADED | SINCE |
|--|---------|-----------|-------------|----------|-------|
| authentication | 4.4.3 | True | False | False | 7m56s |
| cloud-credential | 4.4.3 | True | False | False | 31m |
| cluster-autoscaler | 4.4.3 | True | False | False | 16m |
| console | 4.4.3 | True | False | False | 10m |
| csi-snapshot-controller | 4.4.3 | True | False | False | 16m |
| dns | 4.4.3 | True | False | False | 22m |
| etcd | 4.4.3 | False | False | False | 25s |
| image-registry | 4.4.3 | True | False | False | 16m |
| ingress | 4.4.3 | True | False | False | 16m |
| insights | 4.4.3 | True | False | False | 17m |
| kube-apiserver | 4.4.3 | True | False | False | 19m |
| kube-controller-manager | 4.4.3 | True | False | False | 20m |
| kube-scheduler | 4.4.3 | True | False | False | 20m |
| kube-storage-version-migrator | 4.4.3 | True | False | False | 16m |
| machine-api | 4.4.3 | True | False | False | 22m |
| machine-config | 4.4.3 | True | False | False | 22m |
| marketplace | 4.4.3 | True | False | False | 16m |
| monitoring | 4.4.3 | True | False | False | 10m |
| network | 4.4.3 | True | False | False | 23m |
| node-tuning | 4.4.3 | True | False | False | 23m |
| openshift-apiserver | 4.4.3 | True | False | False | 17m |
| openshift-controller-manager | 4.4.3 | True | False | False | 15m |
| openshift-samples | 4.4.3 | True | False | False | 16m |
| operator-lifecycle-manager | 4.4.3 | True | False | False | 22m |
| operator-lifecycle-manager-catalog | 4.4.3 | True | False | False | 22m |
| operator-lifecycle-manager-packageserver | 4.4.3 | True | False | False | 18m |
| service-ca | 4.4.3 | True | False | False | 23m |
| service-catalog-apiserver | 4.4.3 | True | False | False | 23m |
| service-catalog-controller-manager | 4.4.3 | True | False | False | 23m |
| storage | 4.4.3 | True | False | False | 17m |

When all of the cluster Operators are **AVAILABLE**, you can complete the installation.

2. Monitor for cluster completion:

```
$ ./openshift-install --dir=<installation_directory> wait-for install-complete 1
INFO Waiting up to 30m0s for the cluster to initialize...
```

- 1 For `<installation_directory>`, specify the path to the directory that you stored the installation files in.

The command succeeds when the Cluster Version Operator finishes deploying the OpenShift Container Platform cluster from Kubernetes API server.



IMPORTANT

The Ignition config files that the installation program generates contain certificates that expire after 24 hours, which are then renewed at that time. If the cluster is shut down before renewing the certificates and the cluster is later restarted after the 24 hours have elapsed, the cluster automatically recovers the expired certificates. The exception is that you must manually approve the pending **node-bootstrapper** certificate signing requests (CSRs) to recover kubelet certificates. See the documentation for *Recovering from expired control plane certificates* for more information.

3. Confirm that the Kubernetes API server is communicating with the pods.
 - a. To view a list of all pods, use the following command:

```
$ oc get pods --all-namespaces
```

| NAMESPACE | NAME | READY | STATUS |
|-----------------------------------|---|-------|---------|
| RESTARTS | AGE | | |
| openshift-apiserver-operator | openshift-apiserver-operator-85cb746d55-zqhs8 | 1/1 | Running |
| 1 | 9m | | |
| openshift-apiserver | apiserver-67b9g | 1/1 | Running |
| 3m | | | |
| openshift-apiserver | apiserver-ljcmx | 1/1 | Running |
| 1m | | | |
| openshift-apiserver | apiserver-z25h4 | 1/1 | Running |
| 2m | | | |
| openshift-authentication-operator | authentication-operator-69d5d8bf84-vh2n8 | 1/1 | Running |
| 0 | 5m | | |
| ... | | | |

- b. View the logs for a pod that is listed in the output of the previous command by using the following command:

```
$ oc logs <pod_name> -n <namespace> 1
```

- 1 Specify the pod name and namespace, as shown in the output of the previous command.

If the pod logs display, the Kubernetes API server can communicate with the cluster machines.

1.2.14. Next steps

- [Customize your cluster.](#)

