OpenShift Container Platform 4.3

Installing on bare metal

Installing OpenShift Container Platform 4.3 bare metal clusters

Last Updated: 2020-01-17

OpenShift Container Platform 4.3 Installing on bare metal

Installing OpenShift Container Platform 4.3 bare metal clusters

Legal Notice

Abstract

This document provides instructions for installing OpenShift Container Platform 4.3 clusters on bare metal infrastructure.

Table of Contents

CHAPTER 1. INSTALLING ON BARE METAL

1.1. INSTALLING A CLUSTER ON BARE METAL

In OpenShift Container Platform version 4.3, you can install a cluster on bare metal infrastructure that you provision.

IMPORTANT

While you might be able to follow this procedure to deploy a cluster on virtualized or cloud environments, you must be aware of additional considerations for non-bare metal platforms. Review the information in the guidelines for deploying OpenShift Container Platform on non-tested platforms before you attempt to install an OpenShift Container Platform cluster in such an environment.

Prerequisites

- 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.

1.1.1. Internet and Telemetry access for OpenShift Container Platform

In OpenShift Container Platform 4.3, you require access to the internet to install and entitle 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 . From there, you can allocate entitlements to your cluster.

You must have internet access to:

- Access the Red Hat OpenShift Cluster Manager page to download the installation program and perform subscription management and entitlement. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster. If the Telemetry service cannot entitle your cluster, you must manually entitle it on the Cluster registration page.
- 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 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.

1.1.2. 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.

1.1.2.1. 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, or 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 .

1.1.2.2. 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 a DHCP server in order to establish a network connection to download their Ignition config files.

1.1.2.3. Minimum resource requirements

Each cluster machine must meet the following minimum requirements:

Machine	Operating System	vCPU	RAM	Storage
Bootstrap	RHCOS	4	16 GB	120 GB
Control plane	RHCOS	4	16 GB	120 GB
Compute	RHCOS or RHEL 7.6	2	8 GB	120 GB

1.1.2.4. 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.

1.1.3. Creating the user-provisioned infrastructure

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

Prerequistes

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

Procedure

1. Configure DHCP.
2. Provision the required load balancers.
3. Configure the ports for your machines.
4. Configure DNS.
5. Ensure network connectivity.

1.1.3.1. 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 a DHCP server 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 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
TCP	**2379-2380**	etcd server, peer, and metrics ports

Protocol	Port	Description
	6443	Kubernetes API
	9000-**9999**	Host level services, including the node exporter on ports **9100**-**9101** and the Cluster Version Operator on port**9099**.
	10249-**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**.
	30000-**32767**	Kubernetes NodePort

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 layer-4 load balancers. The API requires one load balancer and the default Ingress Controller needs the second load balancer to provide ingress to applications.

Port	Machines	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.	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

Port	Machines	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

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.

1.1.3.2. User-provisioned DNS requirements

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.

Table 1.2. Required DNS records

Component	Record	Description
Kubernetes API	**api.<cluster_name>.<base_domain>**	This DNS record must point to the load balancer for the control plane machines. This record must be resolvable by both clients external to the cluster and from all the nodes within the cluster.
	api-int.<cluster_name>.<base_domain>	This DNS record must point to the load balancer for the control plane machines. This record 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 it cannot resolve the node names, proxied API calls can fail, and you cannot retrieve logs from Pods.

Component	Record	Description
Routes	***.apps.<cluster_name>.<base_domain>**	A wildcard DNS record that points to the load balancer that targets the machines that run the Ingress router pods, which are the worker nodes by default. This record must be resolvable by both clients external to the cluster and from all the nodes within the cluster.
etcd	**etcd-<index>.<cluster_name>.<base_domain>**	OpenShift Container Platform requires DNS records for each etcd instance to point to the control plane machines that host the instances. The etcd instances are differentiated by **<index>** values, which start with **0** and end with **n-1**, where **n** is the number of control plane machines in the cluster. The DNS record must resolve to an unicast IPv4 address for the control plane machine, and the records must be resolvable from all the nodes in the cluster.

Component	Record	Description
	_etcd-server-ssl._tcp.<cluster_name>.<base_domain>	For each control plane machine, OpenShift Container Platform also requires a SRV DNS record for etcd server on that machine with priority **0**, weight **10** and port **2380**. A cluster that uses three control plane machines requires the following records: # _service._proto.name. TTL class SRV priority weight port target. _etcd-server-ssl._tcp.<cluster_name>.<base_domain> 86400 IN SRV 0 10 2380 etcd-0.<cluster_name>.<base_domain>. _etcd-server-ssl._tcp.<cluster_name>.<base_domain> 86400 IN SRV 0 10 2380 etcd-1.<cluster_name>.<base_domain>. _etcd-server-ssl._tcp.<cluster_name>.<base_domain> 86400 IN SRV 0 10 2380 etcd-2.<cluster_name>.<base_domain>.

```
# _service._proto.name.                            TTL    class SRV priority weight port target.
_etcd-server-ssl._tcp.<cluster_name>.<base_domain>  86400 IN    SRV 0        10     2380 etcd-0.
<cluster_name>.<base_domain>.
_etcd-server-ssl._tcp.<cluster_name>.<base_domain>  86400 IN    SRV 0        10     2380 etcd-1.
<cluster_name>.<base_domain>.
_etcd-server-ssl._tcp.<cluster_name>.<base_domain>  86400 IN    SRV 0        10     2380 etcd-2.
<cluster_name>.<base_domain>.
```

1.1.4. 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 to the installation program.

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 rsa -b 4096 -N '' \
    -f <path>/<file_name> 1
```

1 Specify the path and file name, such as **~/.ssh/id_rsa**, of the SSH key.

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. If you install a cluster on infrastructure that you provision, you must provide this key to your cluster's machines.

1.1.5. 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.

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.

1.1.6. Installing the CLI

You can install the CLI in order to interact with OpenShift Container Platform using a command-line interface.

IMPORTANT

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.3. Download and install the new version of **oc**.

Procedure

1. From the Infrastructure Provider page on the Red Hat OpenShift Cluster Manager site, navigate to the page for your installation type and click **Download Command-line Tools**.

2. Click the folder for your operating system and architecture and click the compressed file.

   NOTE

   You can install **oc** on Linux, Windows, or macOS.

3. Save the file to your file system.

4. Extract the compressed file.

5. Place it in a directory that is on your **PATH**.

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

```
$ oc <command>
```

1.1.7. 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.

1.1.7.1. Sample **install-config.yaml** file for bare metal

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, and if you need to access the Pods from an external network, 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 bare metal 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 FIPS validated 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 on, specify an SSH key that your **ssh-agent** process uses.

1.1.7.2. 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.

NOTE

For bare metal installations, if you do not assign node IP addresses from the range that is specified in the **networking.machineCIDR** field in the **install-config.yaml** file, you must include them in the **proxy.noProxy** field.

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's **status.noProxy** field is populated by default with the instance metadata endpoint (**169.254.169.254**) and with the values of the **networking.machineCIDR**, **networking.clusterNetwork.cidr**, and **networking.serviceNetwork** fields from your installation configuration.

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**.

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. The URL scheme must be **http**; **https** is currently not supported.

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 ConfigMap that is named **user-ca-bundle** in the **openshift-config** namespace that contains one or more additional CA certificates that are required for proxying HTTPS connections. The Cluster Network Operator then creates a **trusted-ca-bundle** ConfigMap that merges these contents with the Red Hat Enterprise Linux CoreOS (RHCOS) trust bundle, and this ConfigMap is referenced in the Proxy object's **trustedCA** field. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust 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.

1.1.8. 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. You must complete your cluster installation and keep the cluster running for 24 hours in a non-degraded state to ensure that the first certificate rotation has finished.

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

WARNING There are no compute nodes specified. The cluster will not fully initialize without
compute nodes.
INFO Consuming "Install Config" from target directory
```

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 **manifests/cluster-scheduler-02-config.yml** Kubernetes manifest file to prevent Pods from being scheduled on the control plane machines:

   a. Open the **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
```

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

Before you install a cluster on bare metal 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.

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

Before you install a cluster on bare metal 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.
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.

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.

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=<bare_metal_image_URL> 2
coreos.inst.ignition_url=http://example.com/config.ign 3
```

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.

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.

1.1.9.2. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines by PXE or iPXE booting

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

Prerequisites

- Obtain the Ignition config files for your cluster.
- Configure suitable PXE or iPXE infrastructure.
- Have access to an HTTP 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.

2. Obtain the RHCOS ISO image, 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.

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

   - ISO: **rhcos-<version>-installer.<architecture>.iso**
   - Compressed metal RAW image: **rhcos-<version>metal.<architecture>.raw.gz**
   - **kernel**: **rhcos-<version>-installer-kernel-<architecture>**
   - **initframs**: **rhcos-<version>-installer-initramfs.<architecture>.img**

3. Upload the compressed metal RAW image and the **kernel** and **initramfs** files to your HTTP server.

4. Configure the network boot infrastructure so that the machines boot from their local disks after RHCOS is installed on them.

5. Configure PXE or iPXE 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 http://<HTTP_server>/rhcos-<version>-installer-kernel-<architecture> 1
    APPEND ip=dhcp rd.neednet=1 initrd=http://<HTTP_server>/rhcos-<version>-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>-metal.<architecture>.raw.gz coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign 2 3
```

1 Specify the location of the **kernel** file that you uploaded to your HTTP server.

2 If you use multiple NICs, specify a single interface in the **ip** option. For example, to use DHCP on a NIC that is named **eno1**, set **ip=eno1:dhcp**.

3 Specify locations of the RHCOS files that you uploaded to your HTTP server. The **initrd** parameter value is the location of the **initramfs** file, the **coreos.inst.image_url**

- For iPXE:

```
kernel  http://<HTTP_server>/rhcos-<version>-installer-kernel-<architecture> ip=dhcp
rd.neednet=1 initrd=http://<HTTP_server>/rhcos-<version>-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>-metal.<arhcitectutre>.raw.gz
coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign 1 2
initrd http://<HTTP_server>/rhcos-<version>-installer-initramfs.<architecture>.img 3
boot
```

1 Specify locations of the RHCOS files that you uploaded to your HTTP server. The **kernel** parameter value is the location of the **kernel** file, the **initrd** parameter value is the location of the **initramfs** file, the **coreos.inst.image_url** parameter value is the location of the compressed metal RAW image, and the **coreos.inst.ignition_url** parameter value is the location of the bootstrap Ignition config file.

2 If you use multiple NICs, specify a single interface in the **ip** option. For example, to use DHCP on a NIC that is named **eno1**, set **ip=eno1:dhcp**.

3 Specify the location of the **initramfs** file that you uploaded to your HTTP server.

6. 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.

1.1.10. 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.

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:6443...
INFO API v1.14.6+c4799753c up
INFO Waiting up to 30m0s for the bootstrap-complete event...
```

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.

1.1.11. 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
```

1.1.12. Approving the CSRs for your machines

When you add machines to a cluster, two pending certificates signing request (CSRs) are generated for each machine that you added. You must confirm that these CSRs are approved or, if necessary, approve them yourself.

Prerequisites

- You added machines to your cluster.
- Install the **jq** package.

Procedure

1. Confirm that the cluster recognizes the machines:

   ```
   $ oc get nodes

   NAME      STATUS    ROLES   AGE  VERSION
   master-0  Ready     master  63m  v1.16.2
   master-1  Ready     master  63m  v1.16.2
   master-2  Ready     master  64m  v1.16.2
   worker-0  NotReady  worker  76s  v1.16.2
   worker-1  NotReady  worker  70s  v1.16.2
   ```

   The output lists all of the machines that you created.

2. Review the pending certificate signing requests (CSRs) and ensure that the you see a client and server request with **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 ①
   csr-8vnps   15m     system:serviceaccount:openshift-machine-config-operator:node-
   bootstrapper   Pending
   csr-bfd72   5m26s   system:node:ip-10-0-50-126.us-east-2.compute.internal
   Pending ②
   csr-c57lv   5m26s   system:node:ip-10-0-95-157.us-east-2.compute.internal
   Pending
   ...
   ```

   ① A client request CSR.

   ② A server request CSR.

   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.

- If all the CSRs are valid, approve them all by running the following command:

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

1.1.13. 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                 4.3.0     True        False         False     69s
cloud-credential                4.3.0     True        False         False     12m
cluster-autoscaler               4.3.0     True        False         False     11m
console                       4.3.0     True        False         False     46s
dns                          4.3.0     True        False         False     11m
image-registry                 4.3.0     True        False         False     5m26s
ingress                       4.3.0     True        False         False     5m36s
kube-apiserver                  4.3.0     True        False         False     8m53s
kube-controller-manager             4.3.0     True        False         False     7m24s
kube-scheduler                   4.3.0     True        False         False     12m
machine-api                    4.3.0     True        False         False     12m
machine-config                   4.3.0     True        False         False     7m36s
marketplace                    4.3.0     True        False         False     7m54m
monitoring                    4.3.0     True        False         False     7h54s
network                       4.3.0     True        False         False     5m9s
```

```
node-tuning                          4.3.0     True        False         False      11m
openshift-apiserver                  4.3.0     True        False         False      11m
openshift-controller-manager         4.3.0     True        False         False      5m943s
openshift-samples                    4.3.0     True        False         False      3m55s
operator-lifecycle-manager           4.3.0     True        False         False      11m
operator-lifecycle-manager-catalog   4.3.0     True        False         False      11m
service-ca                           4.3.0     True        False         False      11m
service-catalog-apiserver            4.3.0     True        False         False      5m26s
service-catalog-controller-manager   4.3.0     True        False         False      5m25s
storage                              4.3.0     True        False         False      5m30s
```

2. Configure the Operators that are not available.

1.1.13.1. Image registry removed during installation

On platforms that do not provide shareable object storage, the OpenShift Image Registry Operator bootstraps itself as **Removed**. This allows **openshift-installer** to complete installations on these platform types.

After installation, you must edit the Image Registry Operator configuration to switch the **ManagementState** from **Removed** to **Managed**.

NOTE

The Prometheus console provides an **ImageRegistryRemoved** alert, for example:

"Image Registry has been removed. **ImageStreamTags**, **BuildConfigs** and **DeploymentConfigs** which reference **ImageStreamTags** may not work as expected. Please configure storage and update the config to **Managed** state by editing configs.imageregistry.operator.openshift.io."

1.1.13.2. Image registry storage configuration

If the **image-registry** Operator is not available, you must configure storage for it. Instructions for both configuring a PersistentVolume, 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. 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                          4.3.0     True        False         False      10m
cloud-credential                        4.3.0     True        False         False      22m
cluster-autoscaler                      4.3.0     True        False         False      21m
console                                 4.3.0     True        False         False      10m
dns                                     4.3.0     True        False         False      21m
image-registry                          4.3.0     True        False         False      16m
ingress                                 4.3.0     True        False         False      16m
kube-apiserver                          4.3.0     True        False         False      19m
kube-controller-manager                 4.3.0     True        False         False      18m
kube-scheduler                          4.3.0     True        False         False      22m
machine-api                             4.3.0     True        False         False      22m
machine-config                          4.3.0     True        False         False      18m
marketplace                             4.3.0     True        False         False      18m
monitoring                              4.3.0     True        False         False      18m
network                                 4.3.0     True        False         False      16m
node-tuning                             4.3.0     True        False         False      21m
openshift-apiserver                     4.3.0     True        False         False      21m
openshift-controller-manager            4.3.0     True        False         False      17m
openshift-samples                       4.3.0     True        False         False      14m
operator-lifecycle-manager              4.3.0     True        False         False      21m
operator-lifecycle-manager-catalog      4.3.0     True        False         False      21m
service-ca                              4.3.0     True        False         False      21m
service-catalog-apiserver               4.3.0     True        False         False      16m
service-catalog-controller-manager      4.3.0     True        False         False      16m
storage                                 4.3.0     True        False         False      16m
```

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. You must keep the cluster running for 24 hours in a non-degraded state to ensure that the first certificate rotation has finished.

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     0
3m
openshift-apiserver          apiserver-ljcmx                         1/1     Running     0
1m
openshift-apiserver          apiserver-z25h4                           1/1     Running     0
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.

Next steps

- Customize your cluster.
- If necessary, you can opt out of remote health reporting .
- Set up your registry and configure registry storage .

1.2. INSTALLING A CLUSTER ON BARE METAL WITH NETWORK CUSTOMIZATIONS

In OpenShift Container Platform version 4.3, you can install a cluster on bare metal infrastructure that you provision with customized network configuration options. By customizing your network configuration, your cluster can coexist with existing IP address allocations in your environment and integrate with existing MTU and VXLAN configurations.

You must set most of the network configuration parameters during installation, and you can modify only **kubeProxy** configuration parameters in a running cluster.

Prerequisites

- Review details about the OpenShift Container Platform installation and update processes.
- If you use a firewall, you must configure it to access Red Hat Insights .

1.2.1. Internet and Telemetry access for OpenShift Container Platform

In OpenShift Container Platform 4.3, you require access to the internet to install and entitle 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 . From there, you can allocate entitlements to your cluster.

You must have internet access to:

- Access the Red Hat OpenShift Cluster Manager page to download the installation program and perform subscription management and entitlement. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster. If the Telemetry service cannot entitle your cluster, you must manually entitle it on the Cluster registration page.
- 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 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.

1.2.2. 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.

1.2.2.1. 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, or 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 .

1.2.2.2. 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 a DHCP server in order to establish a network connection to download their Ignition config files.

1.2.2.3. Minimum resource requirements

Each cluster machine must meet the following minimum requirements:

Machine	Operating System	vCPU	RAM	Storage
Bootstrap	RHCOS	4	16 GB	120 GB
Control plane	RHCOS	4	16 GB	120 GB
Compute	RHCOS or RHEL 7.6	2	8 GB	120 GB

1.2.2.4. 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.

1.2.3. Creating the user-provisioned infrastructure

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

Prerequistes

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

Procedure

1. Configure DHCP.
2. Provision the required load balancers.
3. Configure the ports for your machines.
4. Configure DNS.
5. Ensure network connectivity.

1.2.3.1. 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 a DHCP server 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 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 1.3. All machines to all machines

Protocol	Port	Description
TCP	**2379-2380**	etcd server, peer, and metrics ports
	6443	Kubernetes API
	9000-9999	Host level services, including the node exporter on ports **9100-9101** and the Cluster Version Operator on port**9099**.
	10249-10259	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**.
	30000-32767	Kubernetes NodePort

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 layer-4 load balancers. The API requires one load balancer and the default Ingress Controller needs the second load balancer to provide ingress to applications.

Port	Machines	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.	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
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

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.

1.2.3.2. User-provisioned DNS requirements

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.

Table 1.4. Required DNS records

Component	Record	Description
Kubernetes API	**api.<cluster_name>.<base_domain>**	This DNS record must point to the load balancer for the control plane machines. This record must be resolvable by both clients external to the cluster and from all the nodes within the cluster.

Component	Record	Description
	api-int.<cluster_name>.<base_domain>	This DNS record must point to the load balancer for the control plane machines. This record 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 it cannot resolve the node names, proxied API calls can fail, and you cannot retrieve logs from Pods.
Routes	***.apps.<cluster_name>.<base_domain>**	A wildcard DNS record that points to the load balancer that targets the machines that run the Ingress router pods, which are the worker nodes by default. This record must be resolvable by both clients external to the cluster and from all the nodes within the cluster.
etcd	**etcd-<index>.<cluster_name>.<base_domain>**	OpenShift Container Platform requires DNS records for each etcd instance to point to the control plane machines that host the instances. The etcd instances are differentiated by **<index>** values, which start with **0** and end with **n-1**, where **n** is the number of control plane machines in the cluster. The DNS record must resolve to an unicast IPv4 address for the control plane machine, and the records must be resolvable from all the nodes in the cluster.

Component	Record	Description
	_etcd-server-ssl._tcp.<cluster_name>.<base_domain>	For each control plane machine, OpenShift Container Platform also requires a SRV DNS record for etcd server on that machine with priority **0**, weight **10** and port **2380**. A cluster that uses three control plane machines requires the following records: # _service._proto.name. TTL class SRV priority weight port target. _etcd-server-ssl._tcp.<cluster_name>.<base_domain> 86400 IN SRV 0 10 2380 etcd-0.<cluster_name>.<base_domain>. _etcd-server-ssl._tcp.<cluster_name>.<base_domain> 86400 IN SRV 0 10 2380 etcd-1.<cluster_name>.<base_domain>. _etcd-server-ssl._tcp.<cluster_name>.<base_domain> 86400 IN SRV 0 10 2380 etcd-2.<cluster_name>.<base_domain>.

```
# _service._proto.name.                            TTL    class SRV priority weight port target.
_etcd-server-ssl._tcp.<cluster_name>.<base_domain>  86400 IN    SRV 0        10     2380 etcd-0.
<cluster_name>.<base_domain>.
_etcd-server-ssl._tcp.<cluster_name>.<base_domain>  86400 IN    SRV 0        10     2380 etcd-1.
<cluster_name>.<base_domain>.
_etcd-server-ssl._tcp.<cluster_name>.<base_domain>  86400 IN    SRV 0        10     2380 etcd-2.
<cluster_name>.<base_domain>.
```

1.2.4. 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 to the installation program.

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 rsa -b 4096 -N '' \
       -f <path>/<file_name> 1
   ```

   1 Specify the path and file name, such as **~/.ssh/id_rsa**, of the SSH key.

   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.

1.2.5. 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.

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.

1.2.6. Installing the CLI

You can install the CLI in order to interact with OpenShift Container Platform using a command-line interface.

IMPORTANT

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.3. Download and install the new version of **oc**.

Procedure

1. From the Infrastructure Provider page on the Red Hat OpenShift Cluster Manager site, navigate to the page for your installation type and click **Download Command-line Tools**.

2. Click the folder for your operating system and architecture and click the compressed file.

   NOTE

   You can install **oc** on Linux, Windows, or macOS.

3. Save the file to your file system.

4. Extract the compressed file.

5. Place it in a directory that is on your **PATH**.

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

```
$ oc <command>
```

1.2.7. 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.

1.2.7.1. Sample install-config.yaml file for bare metal

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, and if you need to access the Pods from an external network, 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) 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 bare metal 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 FIPS validated 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 on, specify an SSH key that your **ssh-agent** process uses.

1.2.7.2. Network configuration parameters

You can modify your cluster network configuration parameters in the **install-config.yaml** configuration file. The following table describes the parameters.

NOTE

You cannot modify these parameters in the **install-config.yaml** file after installation.

Table 1.5. Required network parameters

Parameter	Description	Value
networking.networkType	The network plug-in to deploy. The **OpenShiftSDN** plug-in is the only plug-in supported in OpenShift Container Platform 4.3.	The default value is **OpenShiftSDN**.
networking.clusterNetwork.cidr	A block of IP addresses from which Pod IP addresses are allocated. The **OpenShiftSDN** network plug-in supports multiple cluster networks. The address blocks for multiple cluster networks must not overlap. Select address pools large enough to fit your anticipated workload.	An IP address allocation in CIDR format. The default value is **10.128.0.0/14**.

Parameter	Description	Value
networking.clusterNetwork.hostPrefix	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**, allowing for 510 (2^(32 - 23) - 2) Pod IP addresses.	A subnet prefix. The default value is **23**.
networking.serviceNetwork	A block of IP addresses for services. **OpenShiftSDN** allows only one **serviceNetwork** block. The address block must not overlap with any other network block.	An IP address allocation in CIDR format. The default value is **172.30.0.0/16**.
networking.machineCIDR	A block of IP addresses used by the OpenShift Container Platform installation program while installing the cluster. The address block must not overlap with any other network block.	An IP address allocation in CIDR format. The default value is **10.0.0.0/16**.

1.2.8. Creating the Ignition config files

Because you must manually start the cluster machines, you must generate the 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. You must complete your cluster installation and keep the cluster running for 24 hours in a non-degraded state to ensure that the first certificate rotation has finished.

Prerequisites

- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.

Procedure

1. Obtain the Ignition config files:

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

1 For **<installation_directory>**, specify the directory name to store the files that the installation program creates.

IMPORTANT

If you created an **install-config.yaml** file, specify the directory that contains it. Otherwise, specify an empty 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.

The following files are generated in the directory:

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

1.2.9. Modifying advanced network configuration parameters

You can modify the advanced network configuration parameters only before you install the cluster. Advanced configuration customization lets you integrate your cluster into your existing network environment by specifying an MTU or VXLAN port, by allowing customization of kube-proxy settings, and by specifying a different **mode** for the **openshiftSDNConfig** parameter.

IMPORTANT

Modifying the OpenShift Container Platform manifest files directly is not supported.

Prerequisites

- Create the **install-config.yaml** file and complete any modifications to it.
- Create the Ignition config files for your cluster.

Procedure

1. Use the following command to create manifests:

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

1 For **<installation_directory>**, specify the name of the directory that contains the **install-config.yaml** file for your cluster.

2. Create a file that is named **cluster-network-03-config.yml** in the **<installation_directory>/manifests/** directory:

```
$ touch <installation_directory>/manifests/cluster-network-03-config.yml 1
```

1 For **<installation_directory>**, specify the directory name that contains the **manifests/** directory for your cluster.

After creating the file, several network configuration files are in the **manifests/** directory, as shown:

```
$ ls <installation_directory>/manifests/cluster-network-*
cluster-network-01-crd.yml
cluster-network-02-config.yml
cluster-network-03-config.yml
```

3. Open the **cluster-network-03-config.yml** file in an editor and enter a CR that describes the Operator configuration you want:

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec: 1
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  serviceNetwork:
  - 172.30.0.0/16
  defaultNetwork:
    type: OpenShiftSDN
    openshiftSDNConfig:
      mode: NetworkPolicy
      mtu: 1450
      vxlanPort: 4789
```

1 The parameters for the **spec** parameter are only an example. Specify your configuration for the Cluster Network Operator in the CR.

The CNO provides default values for the parameters in the CR, so you must specify only the parameters that you want to change.

4. Save the **cluster-network-03-config.yml** file and quit the text editor.

5. Optional: Back up the **manifests/cluster-network-03-config.yml** file. The installation program deletes the **manifests/** directory when creating the cluster.

1.2.10. Cluster Network Operator custom resource (CR)

The cluster network configuration in the **Network.operator.openshift.io** custom resource (CR) stores the configuration settings for the Cluster Network Operator (CNO). The Operator manages the cluster network.

You can specify the cluster network configuration for your OpenShift Container Platform cluster by setting the parameters for the **defaultNetwork** parameter in the CNO CR. The following CR displays the default configuration for the CNO and explains both the parameters you can configure and valid parameter values:

Cluster Network Operator CR

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  clusterNetwork: ①
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  serviceNetwork: ②
  - 172.30.0.0/16
  defaultNetwork: ③
    ...
  kubeProxyConfig: ④
    iptablesSyncPeriod: 30s ⑤
    proxyArguments:
      iptables-min-sync-period: ⑥
      - 30s
```

① ② Specified in the **install-config.yaml** file.

③ Configures the software-defined networking (SDN) for the cluster network.

④ The parameters for this object specify the **kube-proxy** configuration. If you do not specify the parameter values, the Network Operator applies the displayed default parameter values.

⑤ The refresh period for **iptables** rules. The default value is **30s**. Valid suffixes include **s**, **m**, and **h** and are described in the Go time package documentation.

⑥ The minimum duration before refreshing **iptables** rules. This parameter ensures that the refresh does not happen too frequently. Valid suffixes include **s**, **m**, and **h** and are described in the Go time package

1.2.10.1. Configuration parameters for OpenShift SDN

The following YAML object describes the configuration parameters for OpenShift SDN:

```
defaultNetwork:
  type: OpenShiftSDN ①
  openshiftSDNConfig: ②
    mode: NetworkPolicy ③
    mtu: 1450 ④
    vxlanPort: 4789 ⑤
```

① Specified in the **install-config.yaml** file.

② Specify only if you want to override part of the OpenShift SDN configuration.

③ Configures the network isolation mode for **OpenShiftSDN**. The allowed values are **Multitenant**, **Subnet**, or **NetworkPolicy**. The default value is **NetworkPolicy**.

④ MTU for the VXLAN overlay network. This value is normally configured automatically, but if the nodes in your cluster do not all use the same MTU, then you must set this explicitly to 50 less than the smallest node MTU value.

5 The port to use for all VXLAN packets. The default value is **4789**. If you are running in a virtualized environment with existing nodes that are part of another VXLAN network, then you might be

On Amazon Web Services (AWS), you can select an alternate port for the VXLAN between port **9000** and port **9999**.

1.2.10.2. Cluster Network Operator example CR

A complete CR for the CNO is displayed in the following example:

Cluster Network Operator example CR

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  serviceNetwork:
  - 172.30.0.0/16
  defaultNetwork:
    type: OpenShiftSDN
    openshiftSDNConfig:
      mode: NetworkPolicy
      mtu: 1450
      vxlanPort: 4789
  kubeProxyConfig:
    iptablesSyncPeriod: 30s
    proxyArguments:
      iptables-min-sync-period:
      - 30s
```

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

Before you install a cluster on bare metal 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.

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

Before you install a cluster on bare metal 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.

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.

   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.

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=<bare_metal_image_URL> 2
   coreos.inst.ignition_url=http://example.com/config.ign 3
   ```

   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.

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.

1.2.11.2. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines by PXE or iPXE booting

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

Prerequisites

- Obtain the Ignition config files for your cluster.
- Configure suitable PXE or iPXE infrastructure.
- Have access to an HTTP 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.
2. Obtain the RHCOS ISO image, 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.

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

   - ISO: **rhcos-<version>-installer.<architecture>.iso**
   - Compressed metal RAW image: **rhcos-<version>metal.<architecture>.raw.gz**
   - **kernel**: **rhcos-<version>-installer-kernel-<architecture>**
   - **initframs**: **rhcos-<version>-installer-initramfs.<architecture>.img**
3. Upload the compressed metal RAW image and the **kernel** and **initramfs** files to your HTTP server.
4. Configure the network boot infrastructure so that the machines boot from their local disks after RHCOS is installed on them.
5. Configure PXE or iPXE 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 http://<HTTP_server>/rhcos-<version>-installer-kernel-<architecture> ❶
    APPEND ip=dhcp rd.neednet=1 initrd=http://<HTTP_server>/rhcos-<version>-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>-metal.<architecture>.raw.gz
coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign ❷ ❸
```

❶ Specify the location of the **kernel** file that you uploaded to your HTTP server.

❷ If you use multiple NICs, specify a single interface in the **ip** option. For example, to use DHCP on a NIC that is named **eno1**, set **ip=eno1:dhcp**.

❸ Specify locations of the RHCOS files that you uploaded to your HTTP server. The **initrd** parameter value is the location of the **initramfs** file, the **coreos.inst.image_url** parameter value is the location of the compressed metal RAW image, and the **coreos.inst.ignition_url** parameter value is the location of the bootstrap Ignition config file.

- For iPXE:

```
kernel  http://<HTTP_server>/rhcos-<version>-installer-kernel-<architecture> ip=dhcp
rd.neednet=1 initrd=http://<HTTP_server>/rhcos-<version>-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>-metal.<arhcitectutre>.raw.gz
coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign ❶ ❷
initrd http://<HTTP_server>/rhcos-<version>-installer-initramfs.<architecture>.img ❸
boot
```

❶ Specify locations of the RHCOS files that you uploaded to your HTTP server. The **kernel** parameter value is the location of the **kernel** file, the **initrd** parameter value is the location of the **initramfs** file, the **coreos.inst.image_url** parameter value is the location of the compressed metal RAW image, and the **coreos.inst.ignition_url** parameter value is the location of the bootstrap Ignition config file.

❷ If you use multiple NICs, specify a single interface in the **ip** option. For example, to use DHCP on a NIC that is named **eno1**, set **ip=eno1:dhcp**.

❸ Specify the location of the **initramfs** file that you uploaded to your HTTP server.

6. 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.

1.2.12. 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.

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:6443...
INFO API v1.14.6+c4799753c up
INFO Waiting up to 30m0s for the bootstrap-complete event...
```

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.

1.2.13. 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
   ```

1.2.14. Approving the CSRs for your machines

When you add machines to a cluster, two pending certificates signing request (CSRs) are generated for each machine that you added. You must confirm that these CSRs are approved or, if necessary, approve them yourself.

Prerequisites

- You added machines to your cluster.
- Install the **jq** package.

Procedure

1. Confirm that the cluster recognizes the machines:

   ```
   $ oc get nodes

   NAME      STATUS    ROLES   AGE  VERSION
   master-0  Ready     master  63m  v1.16.2
   master-1  Ready     master  63m  v1.16.2
   master-2  Ready     master  64m  v1.16.2
   worker-0  NotReady  worker  76s  v1.16.2
   worker-1  NotReady  worker  70s  v1.16.2
   ```

   The output lists all of the machines that you created.

2. Review the pending certificate signing requests (CSRs) and ensure that the you see a client and server request with **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 1
csr-8vnps   15m     system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper   Pending
csr-bfd72   5m26s   system:node:ip-10-0-50-126.us-east-2.compute.internal
Pending 2
csr-c57lv   5m26s   system:node:ip-10-0-95-157.us-east-2.compute.internal
Pending
...
```

1 A client request CSR.

2 A server request CSR.

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.

- If all the CSRs are valid, approve them all by running the following command:

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

1.2.15. 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                   4.3.0     True        False         False     69s
cloud-credential                  4.3.0     True        False         False     12m
cluster-autoscaler                 4.3.0     True        False         False     11m
console                         4.3.0     True        False         False     46s
dns                           4.3.0     True        False         False     11m
image-registry                   4.3.0     True        False         False     5m26s
ingress                         4.3.0     True        False         False     5m36s
kube-apiserver                    4.3.0     True        False         False     8m53s
kube-controller-manager              4.3.0     True        False         False     7m24s
kube-scheduler                    4.3.0     True        False         False     12m
machine-api                     4.3.0     True        False         False     12m
machine-config                    4.3.0     True        False         False     7m36s
marketplace                     4.3.0     True        False         False     7m54m
monitoring                      4.3.0     True        False         False     7h54s
network                        4.3.0     True        False         False     5m9s
node-tuning                     4.3.0     True        False         False     11m
openshift-apiserver                 4.3.0     True        False         False     11m
openshift-controller-manager          4.3.0     True        False         False     5m943s
openshift-samples                  4.3.0     True        False         False     3m55s
operator-lifecycle-manager           4.3.0     True        False         False     11m
operator-lifecycle-manager-catalog  4.3.0     True        False         False     11m
service-ca                      4.3.0     True        False         False     11m
service-catalog-apiserver            4.3.0     True        False         False     5m26s
service-catalog-controller-manager  4.3.0     True        False         False     5m25s
storage                         4.3.0     True        False         False     5m30s
```

2. Configure the Operators that are not available.

1.2.15.1. Image registry removed during installation

On platforms that do not provide shareable object storage, the OpenShift Image Registry Operator bootstraps itself as **Removed**. This allows **openshift-installer** to complete installations on these platform types.

After installation, you must edit the Image Registry Operator configuration to switch the **ManagementState** from **Removed** to **Managed**.

NOTE

The Prometheus console provides an **ImageRegistryRemoved** alert, for example:

"Image Registry has been removed. **ImageStreamTags**, **BuildConfigs** and **DeploymentConfigs** which reference **ImageStreamTags** may not work as expected. Please configure storage and update the config to **Managed** state by editing configs.imageregistry.operator.openshift.io."

1.2.15.2. Image registry storage configuration

If the **image-registry** Operator is not available, you must configure storage for it. Instructions for both configuring a PersistentVolume, 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.

1.2.16. 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                       4.3.0     True        False         False      10m
cloud-credential                     4.3.0     True        False         False      22m
cluster-autoscaler                   4.3.0     True        False         False      21m
console                              4.3.0     True        False         False      10m
dns                                  4.3.0     True        False         False      21m
image-registry                       4.3.0     True        False         False      16m
ingress                              4.3.0     True        False         False      16m
kube-apiserver                       4.3.0     True        False         False      19m
kube-controller-manager              4.3.0     True        False         False      18m
kube-scheduler                       4.3.0     True        False         False      22m
machine-api                          4.3.0     True        False         False      22m
machine-config                       4.3.0     True        False         False      18m
marketplace                          4.3.0     True        False         False      18m
monitoring                           4.3.0     True        False         False      18m
network                              4.3.0     True        False         False      16m
node-tuning                          4.3.0     True        False         False      21m
openshift-apiserver                  4.3.0     True        False         False      21m
openshift-controller-manager         4.3.0     True        False         False      17m
openshift-samples                    4.3.0     True        False         False      14m
operator-lifecycle-manager           4.3.0     True        False         False      21m
operator-lifecycle-manager-catalog   4.3.0     True        False         False      21m
service-ca                           4.3.0     True        False         False      21m
service-catalog-apiserver            4.3.0     True        False         False      16m
service-catalog-controller-manager   4.3.0     True        False         False      16m
storage                              4.3.0     True        False         False      16m
```

   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. You must keep the cluster running for 24 hours in a non-degraded state to ensure that the first certificate rotation has finished.

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     0
3m
openshift-apiserver               apiserver-ljcmx                                1/1     Running     0
1m
openshift-apiserver               apiserver-z25h4                                1/1     Running     0
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.

Next steps

- Customize your cluster.
- If necessary, you can opt out of remote health reporting .
- Set up your registry and configure registry storage .

1.3. INSTALLING A CLUSTER ON BARE METAL IN A RESTRICTED NETWORK

In OpenShift Container Platform version 4.3, you can install a cluster on bare metal infrastructure that you provision in a restricted network.

IMPORTANT

While you might be able to follow this procedure to deploy a cluster on virtualized or cloud environments, you must be aware of additional considerations for non-bare metal platforms. Review the information in the guidelines for deploying OpenShift Container Platform on non-tested platforms before you attempt to install an OpenShift Container Platform cluster in such an environment.

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 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.

1.3.1. About installations in restricted networks

In OpenShift Container Platform 4.3, 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.

If you choose to perform a restricted network installation on a cloud platform, you still require access to its cloud APIs. Some cloud functions, like Amazon Web Service's IAM service, require internet access, so you might still require internet access. Depending on your network, you might require less internet access for an installation on bare metal hardware or on VMware vSphere.

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 mirror on a bastion 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.

1.3.1.1. 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 ImageStreamTags.

1.3.2. Internet and Telemetry access for OpenShift Container Platform

In OpenShift Container Platform 4.3, you require access to the internet to install and entitle 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 . From there, you can allocate entitlements to your cluster.

You must have internet access to:

- Access the Red Hat OpenShift Cluster Manager page to download the installation program and perform subscription management and entitlement. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster. If the Telemetry service cannot entitle your cluster, you must manually entitle it on the Cluster registration page.
- 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 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.

1.3.3. 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.

1.3.3.1. 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, or 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 .

1.3.3.2. 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 a DHCP server in order to establish a network connection to download their Ignition config files.

1.3.3.3. Minimum resource requirements

Each cluster machine must meet the following minimum requirements:

Machine	Operating System	vCPU	RAM	Storage
Bootstrap	RHCOS	4	16 GB	120 GB
Control plane	RHCOS	4	16 GB	120 GB
Compute	RHCOS or RHEL 7.6	2	8 GB	120 GB

1.3.3.4. 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.

1.3.4. Creating the user-provisioned infrastructure

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

Prerequistes

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

Procedure

1. Configure DHCP.
2. Provision the required load balancers.
3. Configure the ports for your machines.
4. Configure DNS.
5. Ensure network connectivity.

1.3.4.1. 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 a DHCP server 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 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
TCP	**2379**-**2380**	etcd server, peer, and metrics ports
	6443	Kubernetes API
	9000-**9999**	Host level services, including the node exporter on ports **9100**-**9101** and the Cluster Version Operator on port**9099**.
	10249-**10259**	The default ports that Kubernetes reserves

Protocol	Port	Description
	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**.
	30000-**32767**	Kubernetes NodePort

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 layer-4 load balancers. The API requires one load balancer and the default Ingress Controller needs the second load balancer to provide ingress to applications.

Port	Machines	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.	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
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

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.

1.3.4.2. User-provisioned DNS requirements

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.

Table 1.7. Required DNS records

Component	Record	Description
Kubernetes API	**api.<cluster_name>.<base_domain>**	This DNS record must point to the load balancer for the control plane machines. This record must be resolvable by both clients external to the cluster and from all the nodes within the cluster.
	api-int.<cluster_name>.<base_domain>	This DNS record must point to the load balancer for the control plane machines. This record 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 it cannot resolve the node names, proxied API calls can fail, and you cannot retrieve logs from Pods.
Routes	***.apps.<cluster_name>.<base_domain>**	A wildcard DNS record that points to the load balancer that targets the machines that run the Ingress router pods, which are the worker nodes by default. This record must be resolvable by both clients external to the cluster and from all the nodes within the cluster.

Component	Record	Description
etcd	**etcd-<index>.<cluster_name>.<base_domain>**	OpenShift Container Platform requires DNS records for each etcd instance to point to the control plane machines that host the instances. The etcd instances are differentiated by **<index>** values, which start with **0** and end with **n-1**, where **n** is the number of control plane machines in the cluster. The DNS record must resolve to an unicast IPv4 address for the control plane machine, and the records must be resolvable from all the nodes in the cluster.
	_etcd-server-ssl._tcp.<cluster_name>.<base_domain>	For each control plane machine, OpenShift Container Platform also requires a SRV DNS record for etcd server on that machine with priority **0**, weight **10** and port **2380**. A cluster that uses three control plane machines requires the following records: # _service._proto.name. TTL class SRV priority weight port target. _etcd-server-ssl._tcp.<cluster_name>.<base_domain> 86400 IN SRV 0 10 2380 etcd-0.<cluster_name>.<base_domain>. _etcd-server-ssl._tcp.<cluster_name>.<base_domain> 86400 IN SRV 0 10 2380 etcd-1.<cluster_name>.<base_domain>. _etcd-server-ssl._tcp.<cluster_name>.<base_domain> 86400 IN SRV 0 10 2380 etcd-2.<cluster_name>.<base_domain>.

```
# _service._proto.name.                            TTL    class SRV priority weight port target.
_etcd-server-ssl._tcp.<cluster_name>.<base_domain>  86400 IN    SRV 0        10     2380 etcd-0.
```

```
<cluster_name>.<base_domain>.
_etcd-server-ssl._tcp.<cluster_name>.<base_domain>  86400 IN   SRV 0      10    2380 etcd-1.
<cluster_name>.<base_domain>.
_etcd-server-ssl._tcp.<cluster_name>.<base_domain>  86400 IN   SRV 0      10    2380 etcd-2.
<cluster_name>.<base_domain>.
```

1.3.5. 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 to the installation program.

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 rsa -b 4096 -N '' \
       -f <path>/<file_name> 1
   ```

   1 Specify the path and file name, such as **~/.ssh/id_rsa**, of the SSH key.

   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. If you install a cluster on infrastructure that you provision, you must provide this key to your cluster's machines.

1.3.6. 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.
- Obtain the **imageContentSources** section from the output of the command to mirror the repository.
- Obtain the contents of the certificate for your mirror registry.

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**.

- Unless you use a registry that RHCOS trusts by default, such as **docker.io**, you must provide the contents of the certificate for your mirror repository in the **additionalTrustBundle** section. In most cases, you must provide the certificate for your mirror.
- You must include the **imageContentSources** section from the output of the command to mirror the repository.

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.

1.3.6.1. Sample install-config.yaml file for bare metal

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":{"<bastion_host_name>:5000": {"auth": "<credentials>","email":
"you@example.com"}}}' 15
sshKey: 'ssh-ed25519 AAAA...' 16
additionalTrustBundle: | 17
  -----BEGIN CERTIFICATE-----
  ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
  -----END CERTIFICATE-----
imageContentSources: 18
- mirrors:
  - <bastion_host_name>:5000/<repo_name>/release
  source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - <bastion_host_name>:5000/<repo_name>/release
  source: registry.svc.ci.openshift.org/ocp/release
```

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, and if you need to access the Pods from an external network, 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 bare metal 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 FIPS validated cryptography modules that are provided with RHCOS instead.

15 For **bastion_host_name**, specify the registry domain name that you specified in the certificate for your mirror registry, and for **<credentials>**, specify the base64-encoded user name and password for your mirror registry.

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 on, specify an SSH key that your **ssh-agent** process uses.

17 Provide the contents of the certificate file that you used for your mirror registry.

18 Provide the **imageContentSources** section from the output of the command to mirror the repository.

1.3.6.2. 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's **status.noProxy** field is populated by default with the instance metadata endpoint (**169.254.169.254**) and with the values of the **networking.machineCIDR**, **networking.clusterNetwork.cidr**, and **networking.serviceNetwork** fields from your installation configuration.

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**.

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. The URL scheme must be **http**; **https** is currently not supported.

scheme must be **http**; **https** is currently not supported.

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 ConfigMap that is named **user-ca-bundle** in the **openshift-config** namespace that contains one or more additional CA certificates that are required for proxying HTTPS connections. The Cluster Network Operator then creates a **trusted-ca-bundle** ConfigMap that merges these contents with the Red Hat Enterprise Linux CoreOS (RHCOS) trust bundle, and this ConfigMap is referenced in the Proxy object's **trustedCA** field. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust 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.

1.3.7. 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. You must complete your cluster installation and keep the cluster running for 24 hours in a non-degraded state to ensure that the first certificate rotation has finished.

Prerequisites

- Obtain the OpenShift Container Platform installation program. For a restricted network installation, these files are on your bastion host.

- 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
```

```
WARNING There are no compute nodes specified. The cluster will not fully initialize without
compute nodes.
INFO Consuming "Install Config" from target directory
```

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 **manifests/cluster-scheduler-02-config.yml** Kubernetes manifest file to prevent Pods from being scheduled on the control plane machines:

   a. Open the **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
```

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

Before you install a cluster on bare metal 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.

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

Before you install a cluster on bare metal 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.
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.

   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.
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=<bare_metal_image_URL> ②
   coreos.inst.ignition_url=http://example.com/config.ign ③
   ```

   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.

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.

1.3.8.2. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines by PXE or iPXE booting

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

Prerequisites

- Obtain the Ignition config files for your cluster.
- Configure suitable PXE or iPXE infrastructure.
- Have access to an HTTP 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.

2. Obtain the RHCOS ISO image, 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.

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

   - ISO: **rhcos-<version>-installer.<architecture>.iso**
   - Compressed metal RAW image: **rhcos-<version>metal.<architecture>.raw.gz**
   - **kernel**: **rhcos-<version>-installer-kernel-<architecture>**
   - **initframs**: **rhcos-<version>-installer-initramfs.<architecture>.img**

3. Upload the compressed metal RAW image and the **kernel** and **initramfs** files to your HTTP server.

4. Configure the network boot infrastructure so that the machines boot from their local disks after RHCOS is installed on them.

5. Configure PXE or iPXE 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 http://<HTTP_server>/rhcos-<version>-installer-kernel-<architecture> ❶
         APPEND ip=dhcp rd.neednet=1 initrd=http://<HTTP_server>/rhcos-<version>-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>-metal.<architecture>.raw.gz
     coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign ❷ ❸
     ```

     ❶ Specify the location of the **kernel** file that you uploaded to your HTTP server.

     ❷ If you use multiple NICs, specify a single interface in the **ip** option. For example, to use DHCP on a NIC that is named **eno1**, set **ip=eno1:dhcp**.

     ❸ Specify locations of the RHCOS files that you uploaded to your HTTP server. The **initrd** parameter value is the location of the **initramfs** file, the **coreos.inst.image_url** parameter value is the location of the compressed metal RAW image, and the **coreos.inst.ignition_url** parameter value is the location of the bootstrap Ignition config file.

   - For iPXE:

     ```
     kernel  http://<HTTP_server>/rhcos-<version>-installer-kernel-<architecture> ip=dhcp
     rd.neednet=1 initrd=http://<HTTP_server>/rhcos-<version>-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>-metal.<arhcitectutre>.raw.gz
     coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign ❶ ❷
     initrd http://<HTTP_server>/rhcos-<version>-installer-initramfs.<architecture>.img ❸
     boot
     ```

     ❶ Specify locations of the RHCOS files that you uploaded to your HTTP server. The **kernel** parameter value is the location of the **kernel** file, the **initrd** parameter value is the location of the **initramfs** file, the **coreos.inst.image_url** parameter value is the location of the compressed metal RAW image, and the **coreos.inst.ignition_url** parameter value is the location of the bootstrap Ignition config file.

     ❷ If you use multiple NICs, specify a single interface in the **ip** option. For example, to use DHCP on a NIC that is named **eno1**, set **ip=eno1:dhcp**.

     ❸ Specify the location of the **initramfs** file that you uploaded to your HTTP server.

6. 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.

1.3.9. 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.
- Your machines have direct internet access.

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:6443...
INFO API v1.14.6+c4799753c up
INFO Waiting up to 30m0s for the bootstrap-complete event...
```

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.

1.3.10. 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
   ```

1.3.11. Approving the CSRs for your machines

When you add machines to a cluster, two pending certificates signing request (CSRs) are generated for each machine that you added. You must confirm that these CSRs are approved or, if necessary, approve them yourself.

Prerequisites

- You added machines to your cluster.
- Install the **jq** package.

Procedure

1. Confirm that the cluster recognizes the machines:

   ```
   $ oc get nodes

   NAME      STATUS    ROLES   AGE  VERSION
   master-0  Ready     master  63m  v1.16.2
   master-1  Ready     master  63m  v1.16.2
   master-2  Ready     master  64m  v1.16.2
   worker-0  NotReady  worker  76s  v1.16.2
   worker-1  NotReady  worker  70s  v1.16.2
   ```

   The output lists all of the machines that you created.

2. Review the pending certificate signing requests (CSRs) and ensure that the you see a client and server request with **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 1
csr-8vnps  15m    system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper  Pending
csr-bfd72  5m26s  system:node:ip-10-0-50-126.us-east-2.compute.internal
Pending 2
csr-c57lv  5m26s  system:node:ip-10-0-95-157.us-east-2.compute.internal
Pending
...
```

1 A client request CSR.

2 A server request CSR.

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.

- If all the CSRs are valid, approve them all by running the following command:

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

1.3.12. 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                       4.3.0     True        False         False      69s
cloud-credential                     4.3.0     True        False         False      12m
cluster-autoscaler                   4.3.0     True        False         False      11m
console                              4.3.0     True        False         False      46s
dns                                  4.3.0     True        False         False      11m
image-registry                       4.3.0     True        False         False      5m26s
ingress                              4.3.0     True        False         False      5m36s
kube-apiserver                       4.3.0     True        False         False      8m53s
kube-controller-manager              4.3.0     True        False         False      7m24s
kube-scheduler                       4.3.0     True        False         False      12m
machine-api                          4.3.0     True        False         False      12m
machine-config                       4.3.0     True        False         False      7m36s
marketplace                          4.3.0     True        False         False      7m54m
monitoring                           4.3.0     True        False         False      7h54s
network                              4.3.0     True        False         False      5m9s
node-tuning                          4.3.0     True        False         False      11m
openshift-apiserver                  4.3.0     True        False         False      11m
openshift-controller-manager         4.3.0     True        False         False      5m943s
openshift-samples                    4.3.0     True        False         False      3m55s
operator-lifecycle-manager           4.3.0     True        False         False      11m
operator-lifecycle-manager-catalog   4.3.0     True        False         False      11m
service-ca                           4.3.0     True        False         False      11m
service-catalog-apiserver            4.3.0     True        False         False      5m26s
service-catalog-controller-manager   4.3.0     True        False         False      5m25s
storage                              4.3.0     True        False         False      5m30s
```

2. Configure the Operators that are not available.

1.3.12.1. Image registry storage configuration

If the **image-registry** Operator is not available, you must configure storage for it. Instructions for both configuring a PersistentVolume, 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.

1.3.12.1.1. 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.
- Provision persistent storage for your cluster. To deploy a private image registry, your storage must provide ReadWriteMany access mode.
- Must have "100Gi" capacity.

Procedure

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

2. Verify 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**. If the storage type is **NFS**, and you want to scale up the registry Pod by setting **replica>1** you must enable the **no_wdelay** mount option. For example:

```
# cat /etc/exports
/mnt/data *(rw,sync,no_wdelay,no_root_squash,insecure,fsid=0)
sh-4.3# exportfs -rv
exporting *:/mnt/data
```

3. Check the registry configuration:

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

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
```

1.3.12.1.2. 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.

1.3.13. 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                       4.3.0     True        False         False      10m
cloud-credential                     4.3.0     True        False         False      22m
cluster-autoscaler                   4.3.0     True        False         False      21m
console                              4.3.0     True        False         False      10m
dns                                  4.3.0     True        False         False      21m
image-registry                       4.3.0     True        False         False      16m
ingress                              4.3.0     True        False         False      16m
kube-apiserver                       4.3.0     True        False         False      19m
kube-controller-manager              4.3.0     True        False         False      18m
kube-scheduler                       4.3.0     True        False         False      22m
machine-api                          4.3.0     True        False         False      22m
machine-config                       4.3.0     True        False         False      18m
marketplace                          4.3.0     True        False         False      18m
monitoring                           4.3.0     True        False         False      18m
network                              4.3.0     True        False         False      16m
node-tuning                          4.3.0     True        False         False      21m
openshift-apiserver                  4.3.0     True        False         False      21m
openshift-controller-manager         4.3.0     True        False         False      17m
openshift-samples                    4.3.0     True        False         False      14m
operator-lifecycle-manager           4.3.0     True        False         False      21m
```

```
operator-lifecycle-manager-catalog   4.3.0     True        False         False      21m
service-ca                           4.3.0     True        False         False      21m
service-catalog-apiserver            4.3.0     True        False         False      16m
service-catalog-controller-manager   4.3.0     True        False         False      16m
storage                              4.3.0     True        False         False      16m
```

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. You must keep the cluster running for 24 hours in a non-degraded state to ensure that the first certificate rotation has finished.

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     0
3m
openshift-apiserver               apiserver-ljcmx                                 1/1     Running     0
1m
openshift-apiserver               apiserver-z25h4                                 1/1     Running     0
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.

4. Register your cluster on the Cluster registration page.

Next steps

- Customize your cluster.
- If necessary, you can opt out of remote health reporting .