OpenShift Container Platform 4.6

Installing on IBM Z and LinuxONE

Installing OpenShift Container Platform IBM Z clusters
Abstract

This document provides instructions for installing OpenShift Container Platform clusters on IBM Z.
1.1.15.1.2. Configuring storage for the image registry in non-production clusters
1.1.16. Completing installation on user-provisioned infrastructure
1.1.17. Telemetry access for OpenShift Container Platform
1.1.18. Collecting debugging information
1.1.19. Next steps
1.2. INSTALLING A CLUSTER ON IBM Z AND LINUXONE IN A RESTRICTED NETWORK
1.2.1. About installations in restricted networks
1.2.1.1. Additional limits
1.2.2. Internet access for OpenShift Container Platform
1.2.3. Machine requirements for a cluster with user-provisioned infrastructure
1.2.3.1. Required machines
1.2.3.2. Network connectivity requirements
1.2.3.3. IBM Z network connectivity requirements
1.2.3.4. Minimum resource requirements
1.2.3.5. Minimum IBM Z system environment
   Hardware requirements
   Operating system requirements
   IBM Z network connectivity requirements
   Disk storage for the z/VM guest virtual machines
   Storage / Main Memory
1.2.3.6. Preferred IBM Z system environment
   Hardware requirements
   Operating system requirements
   IBM Z network connectivity requirements
   Disk storage for the z/VM guest virtual machines
   Storage / Main Memory
1.2.3.7. Certificate signing requests management
1.2.4. Creating the user-provisioned infrastructure
1.2.4.1. Networking requirements for user-provisioned infrastructure
   Network topology requirements
   Load balancers
   NTP configuration
1.2.4.2. User-provisioned DNS requirements
1.2.5. Generating an SSH private key and adding it to the agent
1.2.6. Manually creating the installation configuration file
1.2.6.1. Installation configuration parameters
   1.2.6.1.1. Required configuration parameters
   1.2.6.1.2. Network configuration parameters
   1.2.6.1.3. Optional configuration parameters
   1.2.6.2. Sample install-config.yaml file for IBM Z
   1.2.6.3. Configuring the cluster-wide proxy during installation
1.2.7. Creating the Kubernetes manifest and Ignition config files
1.2.8. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines
   1.2.8.1. Advanced RHCOS installation reference
   Routing and bonding options at RHCOS boot prompt
1.2.9. Creating the cluster
1.2.10. Logging in to the cluster by using the CLI
1.2.11. Approving the certificate signing requests for your machines
1.2.12. Initial Operator configuration
   1.2.12.1. Disabling the default OperatorHub sources
   1.2.12.2. Image registry storage configuration
      1.2.12.2.1. Configuring registry storage for IBM Z
      1.2.12.2.2. Configuring storage for the image registry in non-production clusters
1.2.13. Completing installation on user-provisioned infrastructure 90
1.2.14. Telemetry access for OpenShift Container Platform 92
1.2.15. Collecting debugging information 92
1.2.16. Next steps 93
1.1. INSTALLING A CLUSTER ON IBM Z AND LINUXONE

In OpenShift Container Platform version 4.6, you can install a cluster on IBM Z or LinuxONE infrastructure that you provision.

**NOTE**

While this document refers only to IBM Z, all information in it also applies to LinuxONE.

**IMPORTANT**

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

1.1.1. Prerequisites

- Before you begin the installation process, you must clean the installation directory. This ensures that the required installation files are created and updated during the installation process.

- Provision persistent storage using NFS for your cluster. To deploy a private image registry, your storage must provide **ReadWriteMany** access modes.

- Review details about the OpenShift Container Platform installation and update processes.

- If you use a firewall, you must configure it to allow the sites that your cluster requires access to.

**NOTE**

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

1.1.2. Internet access for OpenShift Container Platform

In OpenShift Container Platform 4.6, you require access to the Internet to install your cluster.

You must have Internet access to:

- Access **OpenShift Cluster Manager** to download the installation program and perform subscription management. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster.

- Access **Quay.io** to obtain the packages that are required to install your cluster.

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

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

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

1.1.3.1. Required machines

The smallest OpenShift Container Platform clusters require the following hosts:

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

NOTE

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

IMPORTANT

To improve high availability of your cluster, distribute the control plane machines over different z/VM instances on at least two physical machines.

The bootstrap and control plane machines must use Red Hat Enterprise Linux CoreOS (RHCOS) as the operating system. However, the compute machines can choose between Red Hat Enterprise Linux CoreOS (RHCOS) or Red Hat Enterprise Linux (RHEL) 7.9.

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

1.1.3.2. Network connectivity requirements

All the Red Hat Enterprise Linux CoreOS (RHCOS) machines require network in initramfs during boot to fetch Ignition config files from the Machine Config Server. The machines are configured with static IP addresses. No DHCP server is required. Additionally, each OpenShift Container Platform node in the cluster must have access to a Network Time Protocol (NTP) server.

1.1.3.3. IBM Z network connectivity requirements

To install on IBM Z under z/VM, you require a single z/VM virtual NIC in layer 2 mode. You also need:

- A direct-attached OSA or RoCE network adapter
• A z/VM VSwitch set up. For a preferred setup, use OSA link aggregation.

1.1.3.4. Minimum resource requirements

Each cluster machine must meet the following minimum requirements:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bootstrap</td>
<td>RHCOS</td>
<td>4</td>
<td>16 GB</td>
<td>100 GB</td>
<td>N/A</td>
</tr>
<tr>
<td>Control plane</td>
<td>RHCOS</td>
<td>4</td>
<td>16 GB</td>
<td>100 GB</td>
<td>N/A</td>
</tr>
<tr>
<td>Compute</td>
<td>RHCOS</td>
<td>2</td>
<td>8 GB</td>
<td>100 GB</td>
<td>N/A</td>
</tr>
</tbody>
</table>

1. One vCPU is equivalent to one physical core when simultaneous multithreading (SMT), or hyperthreading, is not enabled. When enabled, use the following formula to calculate the corresponding ratio: (threads per core × cores) × sockets = vCPUs.

1.1.3.5. Minimum IBM Z system environment

You can install OpenShift Container Platform version 4.6 on the following IBM hardware:

• IBM z15 (all models), IBM z14 (all models), IBM z13, and IBM z13s
• LinuxONE, any version

Hardware requirements

• The equivalent of 6 IFLs, which are SMT2 enabled, for each cluster.

• At least one network connection to both connect to the LoadBalancer service and to serve data for traffic outside the cluster.

**NOTE**

You can use dedicated or shared IFLs to assign sufficient compute resources. Resource sharing is one of the key strengths of IBM Z. However, you must adjust capacity correctly on each hypervisor layer and ensure sufficient resources for every OpenShift Container Platform cluster.

**IMPORTANT**

Since the overall performance of the cluster can be impacted, the LPARs that are used to setup the OpenShift Container Platform clusters must provide sufficient compute capacity. In this context, LPAR weight management, entitlements, and CPU shares on the hypervisor level play an important role.

Operating system requirements

• One instance of z/VM 7.1 or later

On your z/VM instance, set up:
- 3 guest virtual machines for OpenShift Container Platform control plane machines
- 2 guest virtual machines for OpenShift Container Platform compute machines
- 1 guest virtual machine for the temporary OpenShift Container Platform bootstrap machine

**IBM Z network connectivity requirements**
To install on IBM Z under z/VM, you require a single z/VM virtual NIC in layer 2 mode. You also need:
- A direct-attached OSA or RoCE network adapter
- A z/VM VSwitch set up. For a preferred setup, use OSA link aggregation.

**Disk storage for the z/VM guest virtual machines**
- FICON attached disk storage (DASDs). These can be z/VM minidisks, fullpack minidisks, or dedicated DASDs, all of which must be formatted as CDL, which is the default. To reach the minimum required DASD size for Red Hat Enterprise Linux CoreOS (RHCOS) installations, you need extended address volumes (EAV). If available, use HyperPAV to ensure optimal performance.
- FCP attached disk storage

**Storage / Main Memory**
- 16 GB for OpenShift Container Platform control plane machines
- 8 GB for OpenShift Container Platform compute machines
- 16 GB for the temporary OpenShift Container Platform bootstrap machine

1.1.3.6. Preferred IBM Z system environment

**Hardware requirements**
- 3 LPARS that each have the equivalent of 6 IFLs, which are SMT2 enabled, for each cluster.
- Two network connections to connect to both connect to the LoadBalancer service and to serve data for traffic outside the cluster.
- HiperSockets, which are attached to a node either directly as a device or by bridging with one z/VM VSWITCH to be transparent to the z/VM guest. To directly connect HiperSockets to a node, you must set up a gateway to the external network via a RHEL 8 guest to bridge to the HiperSockets network.

**Operating system requirements**
- 2 or 3 instances of z/VM 7.1 or later for high availability

On your z/VM instances, set up:
- 3 guest virtual machines for OpenShift Container Platform control plane machines, one per z/VM instance.
- At least 6 guest virtual machines for OpenShift Container Platform compute machines, distributed across the z/VM instances.
- 1 guest virtual machine for the temporary OpenShift Container Platform bootstrap machine.
To ensure the availability of integral components in an overcommitted environment, increase the priority of the control plane by using the CP command `SET SHARE`. Do the same for infrastructure nodes, if they exist. See `SET SHARE` in IBM Documentation.

**IBM Z network connectivity requirements**

To install on IBM Z under z/VM, you require a single z/VM virtual NIC in layer 2 mode. You also need:

- A direct-attached OSA or RoCE network adapter
- A z/VM VSwitch set up. For a preferred setup, use OSA link aggregation.

**Disk storage for the z/VM guest virtual machines**

- FICON attached disk storage (DASDs). These can be z/VM minidisks, fullpack minidisks, or dedicated DASDs, all of which must be formatted as CDL, which is the default. To reach the minimum required DASD size for Red Hat Enterprise Linux CoreOS (RHCOS) installations, you need extended address volumes (EAV). If available, use HyperPAV and High Performance FICON (zHPF) to ensure optimal performance.
- FCP attached disk storage

**Storage / Main Memory**

- 16 GB for OpenShift Container Platform control plane machines
- 8 GB for OpenShift Container Platform compute machines
- 16 GB for the temporary OpenShift Container Platform bootstrap machine

**1.1.3.7. 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.

**Additional resources**

- See Bridging a HiperSockets LAN with a z/VM Virtual Switch in IBM Documentation.
- See Scaling HyperPAV alias devices on Linux guests on z/VM for performance optimization.
- See Topics in LPAR performance for LPAR weight management and entitlements.

**1.1.4. Creating the user-provisioned infrastructure**

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

**Prerequisites**

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

1. Set up static IP addresses.
2. Set up an FTP server.
3. Provision the required load balancers.
4. Configure the ports for your machines.
5. Configure DNS.

1.1.4.1. Networking requirements for user-provisioned infrastructure

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

During the initial boot, the machines require an FTP server in order to establish a network connection to download their Ignition config files.

Ensure that the machines have persistent IP addresses and host names.

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

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMP</td>
<td>N/A</td>
<td>Network reachability tests</td>
</tr>
<tr>
<td>TCP</td>
<td>1936</td>
<td>Metrics</td>
</tr>
<tr>
<td></td>
<td>9000-9999</td>
<td>Host level services, including the node exporter on ports 9100-9101 and the Cluster Version Operator on port 9099.</td>
</tr>
<tr>
<td></td>
<td>10250-10259</td>
<td>The default ports that Kubernetes reserves</td>
</tr>
<tr>
<td></td>
<td>10256</td>
<td>openshift-sdn</td>
</tr>
<tr>
<td>UDP</td>
<td>4789</td>
<td>VXLAN and Geneve</td>
</tr>
<tr>
<td></td>
<td>6081</td>
<td>VXLAN and Geneve</td>
</tr>
<tr>
<td></td>
<td>9000-9999</td>
<td>Host level services, including the node exporter on ports 9100-9101.</td>
</tr>
</tbody>
</table>
TCP/UDP | 30000-32767 | Kubernetes node port

Table 1.2. All machines to control plane

TCP | 6443 | Kubernetes API

Table 1.3. Control plane machines to control plane machines

TCP | 2379-2380 | etcd server and peer ports

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

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

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

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

   **IMPORTANT**
   Do not configure session persistence for an API load balancer.

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

Table 1.4. API load balancer
<table>
<thead>
<tr>
<th>Port</th>
<th>Back-end machines (pool members)</th>
<th>Internal</th>
<th>External</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>6443</strong></td>
<td>Bootstrap and control plane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane. You must configure the /readyz endpoint for the API server health check probe.</td>
<td>X</td>
<td>X</td>
<td>Kubernetes API server</td>
</tr>
<tr>
<td><strong>22623</strong></td>
<td>Bootstrap and control plane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane.</td>
<td>X</td>
<td></td>
<td>Machine config server</td>
</tr>
</tbody>
</table>

**NOTE**

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

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

- Layer 4 load balancing only. This can be referred to as Raw TCP, SSL Passthrough, or SSL Bridge mode. If you use SSL Bridge mode, you must enable Server Name Indication (SNI) for the Ingress routes.

- A connection-based or session-based persistence is recommended, based on the options available and types of applications that will be hosted on the platform.

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

**Table 1.5. Application Ingress load balancer**

<table>
<thead>
<tr>
<th>Port</th>
<th>Back-end machines (pool members)</th>
<th>Internal</th>
<th>External</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>443</strong></td>
<td>The machines that run the Ingress router pods, compute, or worker, by default.</td>
<td>X</td>
<td>X</td>
<td>HTTPS traffic</td>
</tr>
<tr>
<td><strong>80</strong></td>
<td>The machines that run the Ingress router pods, compute, or worker, by default.</td>
<td>X</td>
<td>X</td>
<td>HTTP traffic</td>
</tr>
</tbody>
</table>
TIP

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

NOTE

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

NTP configuration

OpenShift Container Platform clusters are configured to use a public Network Time Protocol (NTP) server by default. If you want to use a local enterprise NTP server, or if your cluster is being deployed in a disconnected network, you can configure the cluster to use a specific time server. For more information, see the documentation for Configuring chrony time service.

Additional resources

- Configuring chrony time service

1.1.4.2. User-provisioned DNS requirements

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

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

Table 1.6. Required DNS records

<table>
<thead>
<tr>
<th>Component</th>
<th>Record</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kubernetes API</td>
<td>api.&lt;cluster_name&gt;.&lt;base_domain&gt;.</td>
<td>Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the load balancer for the control plane machines. These records must be resolvable by both clients external to the cluster and from all the nodes within the cluster.</td>
</tr>
</tbody>
</table>
Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the load balancer for the control plane machines. These records must be resolvable from all the nodes within the cluster.

**IMPORTANT**

The API server must be able to resolve the worker nodes by the host names that are recorded in Kubernetes. If the API server cannot resolve the node names, then proxied API calls can fail, and you cannot retrieve logs from pods.

---

**Routes**

<table>
<thead>
<tr>
<th>Component</th>
<th>Record</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routes</td>
<td>*.apps.&lt;cluster_name&gt;.&lt;base_domain&gt;.</td>
<td>Add a wildcard DNS A/AAAA or CNAME record that refers to the load balancer that targets the machines that run the Ingress router pods, which are the worker nodes by default. These records must be resolvable by both clients external to the cluster and from all the nodes within the cluster.</td>
</tr>
</tbody>
</table>

**Bootstrap**

<table>
<thead>
<tr>
<th>Component</th>
<th>Record</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bootstrap</td>
<td>bootstrap.&lt;cluster_name&gt;.&lt;base_domain&gt;.</td>
<td>Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the bootstrap machine. These records must be resolvable by the nodes within the cluster.</td>
</tr>
</tbody>
</table>

**Master hosts**

<table>
<thead>
<tr>
<th>Component</th>
<th>Record</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master hosts</td>
<td>&lt;master&gt;&lt;n&gt;.&lt;cluster_name&gt;.&lt;base_domain&gt;.</td>
<td>DNS A/AAAA or CNAME records and DNS PTR records to identify each machine for the control plane nodes (also known as the master nodes). These records must be resolvable by the nodes within the cluster.</td>
</tr>
</tbody>
</table>

**Worker hosts**

<table>
<thead>
<tr>
<th>Component</th>
<th>Record</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worker hosts</td>
<td>&lt;worker&gt;&lt;n&gt;.&lt;cluster_name&gt;.&lt;base_domain&gt;.</td>
<td>Add DNS A/AAAA or CNAME records and DNS PTR records to identify each machine for the worker nodes. These records must be resolvable by the nodes within the cluster.</td>
</tr>
</tbody>
</table>

**TIP**

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

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

---

**Example 1.1. Sample DNS zone database**

<table>
<thead>
<tr>
<th>Component</th>
<th>Record</th>
<th>Description</th>
</tr>
</thead>
</table>
The following example BIND zone file shows sample PTR records for reverse name resolution.

Example 1.2. Sample DNS zone database for reverse records

```
$TTL 1W
@ IN SOA ns1.example.com. root ( 2019070700 ; serial 3H ; refresh (3 hours) 30M ; retry (30 minutes) 2W ; expiry (2 weeks) 1W ) ; minimum (1 week)
IN NS ns1.example.com.
IN MX 10 smtp.example.com.
;
ns1 IN A 192.168.1.5
smtp IN A 192.168.1.5
;
helper IN A 192.168.1.5
helper.ocp4 IN A 192.168.1.5
;
; The api identifies the IP of your load balancer.
api.ocp4 IN A 192.168.1.5
api-int.ocp4 IN A 192.168.1.5
;
; The wildcard also identifies the load balancer.
*.apps.ocp4 IN A 192.168.1.5
;
; Create an entry for the bootstrap host.
bootstrap.ocp4 IN A 192.168.1.96
;
; Create entries for the master hosts.
master0.ocp4 IN A 192.168.1.97
master1.ocp4 IN A 192.168.1.98
master2.ocp4 IN A 192.168.1.99
;
; Create entries for the worker hosts.
worker0.ocp4 IN A 192.168.1.11
worker1.ocp4 IN A 192.168.1.17
;
;EOF
```

The syntax is "last octet" and the host must have an FQDN with a trailing dot.
1.1.5. Generating an SSH private key and adding it to the agent

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

NOTE

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

IMPORTANT

Do not skip this procedure in production environments where disaster recovery and debugging is required.

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.

Procedure

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

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

   Specify the path and file name, such as ~/.ssh/id_rsa, of the new SSH key. If you have an existing key pair, ensure your public key is in the your ~/.ssh directory.

   Running this command generates an SSH key that does not require a password in the location that you specified.
If you plan to install an OpenShift Container Platform cluster that uses FIPS Validated / Modules in Process cryptographic libraries on the x86_64 architecture, do not create a key that uses the ed25519 algorithm. Instead, create a key that uses the rsa or ecdsa algorithm.

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

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

Example output

```
Agent pid 31874
```

**NOTE**

If your cluster is in FIPS mode, only use FIPS-compliant algorithms to generate the SSH key. The key must be either RSA or ECDSA.

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

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

Example output

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

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

Next steps

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

### 1.1.6. Obtaining the installation program

Before you install OpenShift Container Platform, download the installation file on your provisioning machine.

**Prerequisites**

- You have a machine that runs Linux, for example Red Hat Enterprise Linux 8, with 500 MB of local disk space

**Procedure**

1. Access the Infrastructure Provider page on the OpenShift Cluster Manager site. If you have a Red Hat account, log in with your credentials. If you do not, create an account.

2. Select your infrastructure provider.
3. 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 the installation program and the files that the installation program creates after you finish installing the cluster. Both files are required to delete the cluster.

**IMPORTANT**

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

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

```
$ tar xvf openshift-install-linux.tar.gz
```

5. Download your installation pull secret from the Red Hat OpenShift Cluster Manager. This pull secret allows you to authenticate with the services that are provided by the included authorities, including Quay.io, which serves the container images for OpenShift Container Platform components.

### 1.1.7. Installing the OpenShift CLI by downloading the binary

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

**IMPORTANT**

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.6. Download and install the new version of **oc**.

#### 1.1.7.1. Installing the OpenShift CLI on Linux

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

**Procedure**


2. Select the appropriate version in the **Version** drop-down menu.

3. Click **Download Now** next to the **OpenShift v4.6 Linux Client** entry and save the file.

4. Unpack the archive:
Place the `oc` binary in a directory that is on your `PATH`. To check your `PATH`, execute the following command:

```
$ echo $PATH
```

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

```
$ oc <command>
```

### 1.1.7.2. Installing the OpenShift CLI on Windows

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

**Procedure**


2. Select the appropriate version in the `Version` drop-down menu.

3. Click **Download Now** next to the **OpenShift v4.6 Windows Client** entry and save the file.

4. Unzip the archive with a ZIP program.

5. Move the `oc` binary to a directory that is on your `PATH`. To check your `PATH`, open the command prompt and execute the following command:

```
C:\> path
```

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

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

### 1.1.7.3. Installing the OpenShift CLI on macOS

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

**Procedure**


2. Select the appropriate version in the `Version` drop-down menu.

3. Click **Download Now** next to the **OpenShift v4.6 MacOSX Client** entry and save the file.

4. Unpack and unzip the archive.

5. Move the `oc` binary to a directory on your `PATH`. To check your `PATH`, open a terminal and execute the following command:

```
$ tar xzvf <file>
$ echo $PATH
$ oc <command>
```

```
C:\> path
C:\> oc <command>
```
After you install the OpenShift CLI, it is available using the `oc` command:

```bash
$ echo $PATH
```

$ oc <command>

### 1.1.8. Manually creating the installation configuration file

For installations of OpenShift Container Platform that use user-provisioned infrastructure, you 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:

   ```bash
   $ mkdir <installation_directory>
   ```

   **IMPORTANT**

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

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

   **NOTE**

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

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

   **IMPORTANT**

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

### 1.1.8.1. Installation configuration parameters

Before you deploy an OpenShift Container Platform cluster, you provide parameter values to describe your account on the cloud platform that hosts your cluster and optionally customize your cluster’s platform. When you create the `install-config.yaml` installation configuration file, you provide values for
the required parameters through the command line. If you customize your cluster, you can modify the install-config.yaml file to provide more details about the platform.

**NOTE**

After installation, you cannot modify these parameters in the install-config.yaml file.

**IMPORTANT**

The openshift-install command does not validate field names for parameters. If an incorrect name is specified, the related file or object is not created, and no error is reported. Ensure that the field names for any parameters that are specified are correct.

1.1.8.1.1. Required configuration parameters

Required installation configuration parameters are described in the following table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiVersion</td>
<td>The API version for the install-config.yaml content. The current version is v1. The installer may also support older API versions.</td>
<td>String</td>
</tr>
<tr>
<td>baseDomain</td>
<td>The base domain of your cloud provider. The base domain is used to create routes to your OpenShift Container Platform cluster components. The full DNS name for your cluster is a combination of the baseDomain and metadata.name parameter values that uses the &lt;metadata.name&gt;.&lt;baseDomain&gt; format.</td>
<td>A fully-qualified domain or subdomain name, such as example.com.</td>
</tr>
<tr>
<td>metadata</td>
<td>Kubernetes resource ObjectMeta, from which only the name parameter is consumed.</td>
<td>Object</td>
</tr>
<tr>
<td>metadata.name</td>
<td>The name of the cluster. DNS records for the cluster are all subdomains of {{.metadata.name}}.{{.baseDomain}}.</td>
<td>String of lowercase letters, hyphens (-), and periods (.), such as dev.</td>
</tr>
</tbody>
</table>
The configuration for the specific platform upon which to perform the installation: **aws**, **baremetal**, **azure**, **openstack**, **ovirt**, **vsphere**. For additional information about `platform.<platform>` parameters, consult the following table for your specific platform.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>platform</strong></td>
<td>The configuration for the specific platform upon which to perform the installation: <strong>aws</strong>, <strong>baremetal</strong>, <strong>azure</strong>, <strong>openstack</strong>, <strong>ovirt</strong>, <strong>vsphere</strong>.</td>
<td>Object</td>
</tr>
<tr>
<td><strong>pullSecret</strong></td>
<td>Get a pull secret from the Red Hat OpenShift Cluster Manager to authenticate downloading container images for OpenShift Container Platform components from services such as Quay.io.</td>
<td><code>{  &quot;auths&quot;:{  &quot;cloud.openshift.com&quot;:{  &quot;auth&quot;:&quot;b3Blb=&quot;,  &quot;email&quot;:&quot;you@example.com&quot;  },  &quot;quay.io&quot;:{  &quot;auth&quot;:&quot;b3Blb=&quot;,  &quot;email&quot;:&quot;you@example.com&quot;  }  }}</code></td>
</tr>
</tbody>
</table>

1.1.8.1.2. Network configuration parameters

You can customize your installation configuration based on the requirements of your existing network infrastructure. For example, you can expand the IP address block for the cluster network or provide different IP address blocks than the defaults.

Only IPv4 addresses are supported.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>networking</strong></td>
<td>The configuration for the cluster network.</td>
<td>Object</td>
</tr>
</tbody>
</table>

**NOTE**
You cannot modify parameters specified by the `networking` object after installation.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>networking.networkType</td>
<td>The cluster network provider Container Network Interface (CNI) plug-in to install.</td>
<td>Either OpenShiftSDN or OVNKubernetes. The default value is OpenShiftSDN.</td>
</tr>
<tr>
<td>networking.clusterNetwork</td>
<td>The IP address blocks for pods.</td>
<td>An array of objects. For example:</td>
</tr>
<tr>
<td></td>
<td>The default value is 10.128.0.0/14 with a host prefix of /23.</td>
<td>networking:</td>
</tr>
<tr>
<td></td>
<td>If you specify multiple IP address blocks, the blocks must not overlap.</td>
<td>clusterNetwork:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- cidr: 10.128.0.0/14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- hostPrefix: 23</td>
</tr>
<tr>
<td>networking.clusterNetwork.cidr</td>
<td>Required if you use networking.clusterNetwork. An IP address block.</td>
<td>An IP address block in Classless Inter-Domain Routing (CIDR) notation. The prefix length for an IPv4 block is between 0 and 32.</td>
</tr>
<tr>
<td></td>
<td>An IPv4 network.</td>
<td></td>
</tr>
<tr>
<td>networking.clusterNetwork.hostPrefix</td>
<td>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. A hostPrefix value of 23 provides 510 (2^(32 - 23) - 2) pod IP addresses.</td>
<td>A subnet prefix.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value is 23.</td>
</tr>
<tr>
<td>networking.serviceNetwork</td>
<td>The IP address block for services. The default value is 172.30.0.0/16.</td>
<td>An array with an IP address block in CIDR format. For example:</td>
</tr>
<tr>
<td></td>
<td>The OpenShift SDN and OVN-Kubernetes network providers support only a single IP address block for the service network.</td>
<td>networking:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>serviceNetwork:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 172.30.0.0/16</td>
</tr>
<tr>
<td>networking.machineNetwork</td>
<td>The IP address blocks for machines.</td>
<td>An array of objects. For example:</td>
</tr>
<tr>
<td></td>
<td>If you specify multiple IP address blocks, the blocks must not overlap.</td>
<td>networking:</td>
</tr>
<tr>
<td></td>
<td>If you specify multiple IP kernel arguments, the machineNetwork.cidr value must be the CIDR of the primary network.</td>
<td>machineNetwork:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- cidr: 10.0.0.0/16</td>
</tr>
</tbody>
</table>
networking.machineNetwork.cidr

Required if you use networking.machineNetwork. An IP address block. The default value is 10.0.0.0/16 for all platforms other than libvirt. For libvirt, the default value is 192.168.126.0/24.

An IP network block in CIDR notation. For example, 10.0.0.0/16.

NOTE

Set the networking.machineNetwork to match the CIDR that the preferred NIC resides in.

1.1.8.13. Optional configuration parameters

Optional installation configuration parameters are described in the following table:

Table 1.9. Optional parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>additionalTrustBundle</td>
<td>A PEM-encoded X.509 certificate bundle that is added to the nodes’ trusted certificate store. This trust bundle may also be used when a proxy has been configured.</td>
<td>String</td>
</tr>
<tr>
<td>compute</td>
<td>The configuration for the machines that comprise the compute nodes.</td>
<td>Array of machine-pool objects. For details, see the following &quot;Machine-pool&quot; table.</td>
</tr>
<tr>
<td>compute.architecture</td>
<td>Determines the instruction set architecture of the machines in the pool. Currently, heterogeneous clusters are not supported, so all pools must specify the same architecture. Valid values are <code>amd64</code> (the default).</td>
<td>String</td>
</tr>
</tbody>
</table>
Whether to enable or disable simultaneous multithreading, or hyperthreading, on compute machines. By default, simultaneous multithreading is enabled to increase the performance of your machines’ cores.

**IMPORTANT**

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>compute.hyperthreading</strong></td>
<td>Whether to enable or disable simultaneous multithreading, or hyperthreading, on compute machines. By default, simultaneous multithreading is enabled to increase the performance of your machines’ cores.</td>
<td>Enabled or Disabled</td>
</tr>
<tr>
<td><strong>compute.name</strong></td>
<td>Required if you use <code>compute</code>. The name of the machine pool.</td>
<td>worker</td>
</tr>
<tr>
<td><strong>compute.platform</strong></td>
<td>Required if you use <code>compute</code>. Use this parameter to specify the cloud provider to host the worker machines. This parameter value must match the <code>controlPlane.platform</code> parameter value.</td>
<td><code>aws</code>, <code>azure</code>, <code>gcp</code>, <code>openstack</code>, <code>ovirt</code>, <code>vsphere</code>, or <code>{}</code></td>
</tr>
<tr>
<td><strong>compute.replicas</strong></td>
<td>The number of compute machines, which are also known as worker machines, to provision.</td>
<td>A positive integer greater than or equal to 2. The default value is 3.</td>
</tr>
<tr>
<td><strong>controlPlane</strong></td>
<td>The configuration for the machines that comprise the control plane.</td>
<td>Array of <code>MachinePool</code> objects. For details, see the following &quot;Machine-pool&quot; table.</td>
</tr>
<tr>
<td><strong>controlPlane.architecture</strong></td>
<td>Determines the instruction set architecture of the machines in the pool. Currently, heterogeneous clusters are not supported, so all pools must specify the same architecture. Valid values are <code>amd64</code> (the default).</td>
<td>String</td>
</tr>
</tbody>
</table>

---

CHAPTER 1. INSTALLING ON IBM Z
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>controlPlane.hypertreading</code></td>
<td>Whether to enable or disable simultaneous multithreading, or hyperthreading, on control plane machines. By default, simultaneous multithreading is enabled to increase the performance of your machines’ cores.</td>
<td>Enabled or Disabled</td>
</tr>
<tr>
<td></td>
<td><strong>IMPORTANT</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.</td>
<td></td>
</tr>
<tr>
<td><code>controlPlane.name</code></td>
<td>Required if you use <code>controlPlane</code>. The name of the machine pool.</td>
<td>master</td>
</tr>
<tr>
<td><code>controlPlane.platform</code></td>
<td>Required if you use <code>controlPlane</code>. Use this parameter to specify the cloud provider that hosts the control plane machines. This parameter value must match the <code>compute.platform</code> parameter value.</td>
<td><code>aws, azure, gcp, openstack, ovirt, vsphere, or {}</code></td>
</tr>
<tr>
<td><code>controlPlane.replicas</code></td>
<td>The number of control plane machines to provision.</td>
<td>The only supported value is 3, which is the default value.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Values</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>credentialsMode</td>
<td>The Cloud Credential Operator (CCO) mode. If no mode is specified, the CCO dynamically tries to determine the capabilities of the provided credentials, with a preference for mint mode on the platforms where multiple modes are supported.</td>
<td>Mint, Passthrough, Manual, or an empty string (&quot;&quot;&quot;).</td>
</tr>
</tbody>
</table>

**NOTE**

Not all CCO modes are supported for all cloud providers. For more information on CCO modes, see the Cloud Credential Operator entry in the Red Hat Operators reference content.
Enable or disable FIPS mode. The default is `false` (disabled). If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.

**IMPORTANT**

The use of FIPS Validated / Modules in Process cryptographic libraries is only supported on OpenShift Container Platform deployments on the x86_64 architecture.

**NOTE**

If you are using Azure File storage, you cannot enable FIPS mode.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fips</code></td>
<td>Enable or disable FIPS mode. The default is <code>false</code> (disabled). If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.</td>
<td><code>false</code> or <code>true</code></td>
</tr>
<tr>
<td><code>imageContentSources</code></td>
<td>Sources and repositories for the release-image content.</td>
<td>Array of objects. Includes a <code>source</code> and, optionally, <code>mirrors</code>, as described in the following rows of this table.</td>
</tr>
<tr>
<td><code>imageContentSources.source</code></td>
<td>Required if you use <code>imageContentSources</code>. Specify the repository that users refer to, for example, in image pull specifications.</td>
<td>String</td>
</tr>
<tr>
<td><code>imageContentSources.mirrors</code></td>
<td>Specify one or more repositories that may also contain the same images.</td>
<td>Array of strings</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Values</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>--------</td>
</tr>
<tr>
<td>publish</td>
<td>How to publish or expose the user-facing endpoints of your cluster, such as the Kubernetes API, OpenShift routes.</td>
<td>Internal or External. The default value is External. Setting this field to Internal is not supported on non-cloud platforms.</td>
</tr>
<tr>
<td>sshKey</td>
<td>The SSH key or keys to authenticate access your cluster machines.</td>
<td>One or more keys. For example: sshKey: &lt;key1&gt; &lt;key2&gt; &lt;key3&gt;</td>
</tr>
</tbody>
</table>

**NOTE**

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

**IMPORTANT**

If the value of the field is set to Internal, the cluster will become non-functional. For more information, refer to BZ#1953035.

1.1.8.2. Sample install-config.yaml file for IBM Z

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.

```yaml
apiVersion: v1
baseDomain: example.com
compute:
  - hyperthreading: Enabled
    name: worker
    replicas: 0
    architecture: s390x
controlPlane:
  hyperthreading: Enabled
  name: master
  replicas: 3
  architecture: s390x
metadata:
  name: test
```
The base domain of the cluster. All DNS records must be sub-domains of this base and include the cluster name.

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. Only one control plane pool is used.

Whether to enable or disable simultaneous multithreading (SMT), or **hyperthreading**. By default, SMT is enabled to increase the performance of your machines' cores. You can disable it by setting the parameter value to **Disabled**. If you disable SMT, you must disable it in all cluster machines; this includes both control plane and compute machines.

**NOTE**

Simultaneous multithreading (SMT) is enabled by default. If SMT is not enabled in your BIOS settings, the **hyperthreading** parameter has no effect.

**IMPORTANT**

If you disable **hyperthreading**, whether in the BIOS or in the **install-config.yaml**, ensure that your capacity planning accounts for the dramatically decreased machine performance.

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.

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.

The cluster name that you specified in your DNS records.

A block of IP addresses from which pod IP addresses are allocated. This block must not overlap with existing physical networks. These IP addresses are used for the pod network. If you need to access the pods from an external network, you must configure load balancers and routers to manage the traffic.

```yaml
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
    hostPrefix: 23
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
  platform:
    none: {}
  fips: false
  pullSecret: {'"auths": ...'}
  sshKey: 'ssh-ed25519 AAAA...
```
NOTE

Class E CIDR range is reserved for a future use. To use the Class E CIDR range, you must ensure your networking environment accepts the IP addresses within the Class E CIDR range.

10 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.

11 The IP address pool to use for service IP addresses. You can enter only one IP address pool. This block must not overlap with existing physical networks. If you need to access the services from an external network, configure load balancers and routers to manage the traffic.

12 You must set the platform to `none`. You cannot provide additional platform configuration variables for IBM Z infrastructure.

13 Whether to enable or disable FIPS mode. By default, FIPS mode is not enabled. If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.

IMPORTANT

The use of FIPS Validated / Modules in Process cryptographic libraries is only supported on OpenShift Container Platform deployments on the x86_64 architecture.

14 The pull secret from the Red Hat OpenShift Cluster Manager. 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.

15 The public portion of the default SSH key for the `core` user in Red Hat Enterprise Linux CoreOS (RHCOS).

NOTE

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

1.1.9. 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

- You have an existing `install-config.yaml` file.
- You reviewed the sites that your cluster requires access to and determined whether any of
them need to bypass the proxy. By default, all cluster egress traffic is proxied, including calls to hosting cloud provider APIs. You added sites to the Proxy object’s spec.noProxy field to bypass the proxy if necessary.

NOTE

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

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

Procedure

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

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

3 A comma-separated list of destination domain names, IP addresses, or other network CIDRs to exclude from proxying. Preface a domain with . to match subdomains only. For example, .y.com matches x.y.com, but not y.com. Use * to bypass the proxy for all destinations.

4 If provided, the installation program generates a config map that is named user-ca-bundle in the openshift-config namespace to hold the additional CA certificates. If you provide additionalTrustBundle and at least one proxy setting, the Proxy object is configured to reference the user-ca-bundle config map in the trustedCA field. The Cluster Network Operator then creates a trusted-ca-bundle config map that merges the contents specified for the trustedCA parameter with the RH COS trust bundle. The additionalTrustBundle field is required unless the proxy’s identity certificate is signed by an authority from the RH COS 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.

### 1.1.10. 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.

The installation configuration file transforms into the Kubernetes manifests. The manifests wrap into the Ignition configuration files, which are later used to create the cluster.

**IMPORTANT**

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

- It is recommended that you use Ignition config files within 12 hours after they are generated because the 24-hour certificate rotates from 16 to 22 hours after the cluster is installed. By using the Ignition config files within 12 hours, you can avoid installation failure if the certificate update runs during installation.

**Prerequisites**

- You obtained the OpenShift Container Platform installation program.
- You created the install-config.yaml installation configuration file.

**Procedure**

1. Change to the directory that contains the installation program and generate the Kubernetes manifests for the cluster:

   ```
   $ ./openshift-install create manifests --dir <installation_directory>
   
   For `<installation_directory>`, specify the installation directory that contains the install-config.yaml file you created.
   ```

2. Check that the mastersSchedulable parameter in the `<installation_directory>/manifests/cluster-scheduler-02-config.yml` Kubernetes manifest file is set to false. This setting prevents pods from being scheduled on the control plane.
machines:
   a. Open the `<installation_directory>/manifests/cluster-scheduler-02-config.yml` file.
   b. Locate the `mastersSchedulable` parameter and ensure that it is set to `false`.
   c. Save and exit the file.

3. To create the Ignition configuration files, run the following command from the directory that contains the installation program:

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

   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
   ```

### 1.1.11. Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines

Before you install a cluster on IBM Z infrastructure that you provision, you must install RHCOS on z/VM guest virtual machines for the cluster to use. Complete the following steps to create the machines.

**Prerequisites**

- An FTP server running on your provisioning machine that is accessible to the machines you create.

**Procedure**

1. Log in to Linux on your provisioning machine.

2. Obtain the Red Hat Enterprise Linux CoreOS (RHCOS) kernel, initramfs, and rootfs files from the RHCOS image mirror.

   **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. Only use the appropriate kernel, initramfs, and rootfs artifacts described in the following procedure.

   The file names contain the OpenShift Container Platform version number. They resemble the following examples:
3. Create parameter files. The following parameters are specific for a particular virtual machine:

- For `coreos.inst.install_dev=`, specify `dasda` for a DASD installation, or `sda` for FCP. Note that FCP requires `zfcp.allow_lun_scan=0`.
- For `rd.dasd=`, specify the DASD where RHCOS is to be installed.
- `rd.zfcp=<adapter>,<wwpn>,<lun>` specifies the FCP disk to install RHCOS on.
- For `ip=` specify the following seven entries:
  1. The IP address for the machine.
  2. An empty string.
  3. The gateway.
  4. The netmask.
  5. The machine host and domain name in the form `hostname.domainname`. Omit this value to let RHCOS decide.
  6. The network interface name. Omit this value to let RHCOS decide.
  7. If you use static IP addresses, an empty string.
- For `coreos.inst.ignition_url=`, specify the Ignition file for the machine role. Use `bootstrap.ign`, `master.ign`, or `worker.ign`. Only HTTP and HTTPS protocols are supported.
- For `coreos.live.rootfs_url=`, specify the matching rootfs artifact for the kernel and initramfs you are booting. Only HTTP and HTTPS protocols are supported.
- All other parameters can stay as they are.

Example parameter file, `bootstrap-0.parm`, for the bootstrap machine:

```
rd.neednet=1 \
console=ttyscp0 \
coreos.inst.install_dev=dasda \
coreos.live.rootfs_url=http://cl1.provide.example.com:8080/assets/rhcos-live-rootfs.s390x.img \
coreos.inst.ignition_url=http://cl1.provide.example.com:8080/ignition/bootstrap.ign \
ip=172.18.78.2::172.18.78.1:255.255.255.0:::none nameserver=172.18.78.1 \
rd.znet=qeth,0,0.bdf0,0,0.bdf1,0,0.bdf2,layer2=1,portno=0 \
zfcp.allow_lun_scan=0 \
rd.dasd=0.0.3490
```
Write all options in the parameter file as a single line and make sure you have no newline characters.

4. Transfer the initramfs, kernel, parameter files, and RHCOS images to z/VM, for example with FTP. For details about how to transfer the files with FTP and boot from the virtual reader, see Installing under Z/VM.

5. Punch the files to the virtual reader of the z/VM guest virtual machine that is to become your bootstrap node.
   See PUNCH in IBM Documentation.

   **TIP**
   You can use the CP PUNCH command or, if you use Linux, the `vmur` command to transfer files between two z/VM guest virtual machines.


7. IPL the bootstrap machine from the reader:

   ```
   $ ipl c
   ```
   See IPL in IBM Documentation.

8. Repeat this procedure for the other machines in the cluster.

### 1.1.11.1. Advanced RHCOS installation reference

This section illustrates the networking configuration and other advanced options that allow you to modify the Red Hat Enterprise Linux CoreOS (RHCOS) manual installation process. The following tables describe the kernel arguments and command-line options you can use with the RHCOS live installer and the `coreos-installer` command.

**Routing and bonding options at RHCOS boot prompt**
If you install RHCOS from an ISO image, you can add kernel arguments manually when you boot that image to configure the node’s networking. If no networking arguments are used, the installation defaults to using DHCP.

**IMPORTANT**
When adding networking arguments, you must also add the `rd.neednet=1` kernel argument.

The following table describes how to use `ip=`, `nameserver=`, and `bond=` kernel arguments for live ISO installs.

**NOTE**
Ordering is important when adding kernel arguments: `ip=`, `nameserver=`, and then `bond=`.

**Routing and bonding options for ISO**
The following table provides examples for configuring networking of your Red Hat Enterprise Linux
CoreOS (RHCOS) nodes. These are networking options that are passed to the `dracut` tool during system boot. For more information about the networking options supported by `dracut`, see the `dracut.cmdline` manual page.

<table>
<thead>
<tr>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>To configure an IP address, either use DHCP (<code>ip=dhcp</code>) or set an individual static IP address (<code>ip=&lt;host_ip&gt;</code>). Then identify the DNS server IP address (<code>nameserver=&lt;dns_ip&gt;</code>) on each node. This example sets:</td>
<td><code>ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none</code>&lt;br&gt;nameserver=4.4.4.41</td>
</tr>
<tr>
<td>• The node’s IP address to <strong>10.10.10.2</strong>&lt;br&gt;• The gateway address to <strong>10.10.10.254</strong>&lt;br&gt;• The netmask to <strong>255.255.255.0</strong>&lt;br&gt;• The hostname to <strong>core0.example.com</strong>&lt;br&gt;• The DNS server address to <strong>4.4.4.41</strong></td>
<td></td>
</tr>
<tr>
<td>Specify multiple network interfaces by specifying multiple <code>ip=</code> entries.</td>
<td><code>ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none</code>&lt;br&gt;<code>ip=10.10.10.3::10.10.10.254:255.255.255.0:core0.example.com:enp2s0:none</code></td>
</tr>
<tr>
<td>Optional: You can configure routes to additional networks by setting an <code>rd.route=</code> value.</td>
<td>To configure the default gateway:</td>
</tr>
<tr>
<td>If the additional network gateway is different from the primary network gateway, the default gateway must be the primary network gateway.</td>
<td>To configure the route for the additional network:</td>
</tr>
<tr>
<td>Disable DHCP on a single interface, such as when there are two or more network interfaces and only one interface is being used.</td>
<td></td>
</tr>
<tr>
<td>You can combine DHCP and static IP configurations on systems with multiple network interfaces.</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Optional: You can configure VLANs on individual interfaces by using the <code>vlan=</code> parameter.</td>
<td>To configure a VLAN on a network interface and use a static IP address:</td>
</tr>
<tr>
<td></td>
<td>ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp2s0.100:none</td>
</tr>
<tr>
<td></td>
<td>vlan=enp2s0.100:enp2s0</td>
</tr>
<tr>
<td></td>
<td>To configure a VLAN on a network interface and to use DHCP:</td>
</tr>
<tr>
<td></td>
<td>ip=enp2s0.100:dhcp</td>
</tr>
<tr>
<td></td>
<td>vlan=enp2s0.100:enp2s0</td>
</tr>
<tr>
<td>You can provide multiple DNS servers by adding a <code>nameserver=</code> entry for each server.</td>
<td>nameserver=1.1.1.1</td>
</tr>
<tr>
<td></td>
<td>nameserver=8.8.8.8</td>
</tr>
<tr>
<td>Optional: Bonding multiple network interfaces to a single interface is supported using the <code>bond=</code> option. In these two examples:</td>
<td>To configure the bonded interface to use DHCP, set the bond’s IP address to <code>dhcp</code>. For example:</td>
</tr>
<tr>
<td></td>
<td>- The syntax for configuring a bonded interface is:</td>
</tr>
<tr>
<td></td>
<td><code>bond=name[:network_interfaces][:options]</code></td>
</tr>
<tr>
<td></td>
<td>- <code>name</code> is the bonding device name (<code>bond0</code>), <code>network_interfaces</code> represents a comma-separated list of physical (ethernet) interfaces (<code>em1,em2</code>), and <code>options</code> is a comma-separated list of bonding options. Enter <code>modinfo bonding</code> to see available options.</td>
</tr>
</tbody>
</table>
|                                                                             |   - When you create a bonded interface using `bond=`, you must specify how the IP address is assigned and other information for the bonded interface.
|                                                                             | To configure the bonded interface to use a static IP address, enter the specific IP address you want and related information. For example: |
|                                                                             |   bond=bond0:em1,em2:mode=active-backup                                  |
|                                                                             |   ip=bond0:dhcp                                                          |
|                                                                             | To configure the bonded interface to use a static IP address, enter the specific IP address you want and related information. For example: |
|                                                                             |   bond=bond0:em1,em2:mode=active-backup                                  |
|                                                                             |   ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:bond0:none  |
Optional: You can configure VLANs on bonded interfaces by using the `vlan=` parameter.

To configure the bonded interface with a VLAN and to use DHCP:

```
ip=bond0.100:dhcp
bond=bond0:em1,em2:mode=active-backup
vlan=bond0.100:bond0
```

To configure the bonded interface with a VLAN and to use a static IP address:

```
ip=10.10.10.2::10.10.10.254:255.255.255.0:cor0.example.com::bond0.100:none
bond=bond0:em1,em2:mode=active-backup
vlan=bond0.100:bond0
```

Optional: Network teaming can be used as an alternative to bonding by using the `team=` parameter. In this example:

- The syntax for configuring a team interface is: `team=name[:network_interfaces]`
  - `name` is the team device name (team0) and `network_interfaces` represents a comma-separated list of physical (ethernet) interfaces (em1, em2).

**NOTE**

Teaming is planned to be deprecated when RHCOS switches to an upcoming version of RHEL. For more information, see this [Red Hat Knowledgebase Article](#).

To configure a network team:

```
team=team0:em1,em2
ip=team0:dhcp
```

### 1.1.12. 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 or have an HTTP or HTTPS proxy available.

**Procedure**
1. Monitor the bootstrap process:

   $ ./openshift-install --dir <installation_directory> wait-for bootstrap-complete \  
   --log-level=info

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

   To view different installation details, specify `warn`, `debug`, or `error` instead of `info`.

**Example output**

```
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com:6443...
INFO API v1.19.0 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
```

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.

### 1.1.13. Logging in to the cluster by using the CLI

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**

- You deployed an OpenShift Container Platform cluster.
- You installed the `oc` CLI.

**Procedure**

1. Export the `kubeadm` credentials:

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

   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:

   ```bash
   $ oc whoami
   ```
1.14. Approving the certificate signing requests for your machines

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

Prerequisites

- You added machines to your cluster.

Procedure

1. Confirm that the cluster recognizes the machines:

   $ oc get nodes

   Example output

   system:admin

   NAME      STATUS    ROLES   AGE   VERSION
   master-0  Ready     master  63m   v1.19.0
   master-1  Ready     master  63m   v1.19.0
   master-2  Ready     master  64m   v1.19.0

   The output lists all of the machines that you created.

   **NOTE**

   The preceding output might not include the compute nodes, also known as worker nodes, until some CSRs are approved.

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

   $ oc get csr

   Example output

   NAME        AGE     REQUESTOR                                   CONDITION
   csr-mddf5   20m     system:node:master-01.example.com   Approved,Issued
   csr-z5rln   16m     system:node:worker-21.example.com   Approved,Issued

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. Once the client CSR is approved, the Kubelet creates a secondary CSR for the serving certificate, which requires manual approval. Then, subsequent serving certificate renewal requests are automatically approved by the machine-approver if the Kubelet requests a new certificate with identical parameters.

NOTE

For clusters running on platforms that are not machine API enabled, such as bare metal and other user-provisioned infrastructure, you must implement a method of automatically approving the kubelet serving certificate requests (CSRs). If a request is not approved, then the oc exec, oc rsh, and oc logs commands cannot succeed, because a serving certificate is required when the API server connects to the kubelet. Any operation that contacts the Kubelet endpoint requires this certificate approval to be in place. The method must watch for new CSRs, confirm that the CSR was submitted by the node-bootstrapper service account in the system:node or system:admin groups, and confirm the identity of the node.

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

  $ oc adm certificate approve <csr_name> 1

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

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

  $ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{{"\n"}}{{end}}{{end}}' | xargs --no-run-if-empty oc adm certificate approve

NOTE

Some Operators might not become available until some CSRs are approved.

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

  $ oc get csr

Example output

<table>
<thead>
<tr>
<th>NAME</th>
<th>AGE</th>
<th>REQUESTOR</th>
<th>CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>csr-bfd72</td>
<td>5m26s</td>
<td>system:node:ip-10-0-50-126.us-east-2.compute.internal</td>
<td>Pending</td>
</tr>
<tr>
<td>csr-c57lv</td>
<td>5m26s</td>
<td>system:node:ip-10-0-95-157.us-east-2.compute.internal</td>
<td>Pending</td>
</tr>
</tbody>
</table>
5. If the remaining CSRs are not approved, and are in the **Pending** status, approve the CSRs for your cluster machines:

   - To approve them individually, run the following command for each valid CSR:
     
     ```
     $ oc adm certificate approve <csr_name>
     ```

   - To approve all pending CSRs, run the following command:
     
     ```
     $ oc get csr -o go-template="{{range .items}}{{if not .status}}{{.metadata.name}}{{\n}}{{end}}{{end}}" | xargs oc adm certificate approve
     ```

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

   ```
   $ oc get nodes
   ```

   **Example output**

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>ROLES</th>
<th>AGE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>master-0</td>
<td>Ready</td>
<td>master</td>
<td>73m</td>
<td>v1.20.0</td>
</tr>
<tr>
<td>master-1</td>
<td>Ready</td>
<td>master</td>
<td>73m</td>
<td>v1.20.0</td>
</tr>
<tr>
<td>master-2</td>
<td>Ready</td>
<td>master</td>
<td>74m</td>
<td>v1.20.0</td>
</tr>
<tr>
<td>worker-0</td>
<td>Ready</td>
<td>worker</td>
<td>11m</td>
<td>v1.20.0</td>
</tr>
<tr>
<td>worker-1</td>
<td>Ready</td>
<td>worker</td>
<td>11m</td>
<td>v1.20.0</td>
</tr>
</tbody>
</table>

   **NOTE**

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

**Additional information**

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

### 1.1.15. 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
   ```
Configure the Operators that are not available.

1.1.15.1. Image registry storage configuration

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

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

Additional instructions are provided for allowing the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

1.1.15.1.1. Configuring registry storage for IBM Z

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

**Prerequisites**
Cluster administrator permissions.

A cluster on IBM Z.

Persistent storage provisioned for your cluster.

IMPORTANT

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

Must have 100Gi capacity.

Procedure

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

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

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

   $ oc get pod -n openshift-image-registry

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

3. Check the registry configuration:

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

   Example output

   storage:
   pvc:
   claim:

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

4. Check the clusteroperator status:

   $ oc get clusteroperator image-registry

5. Ensure that your registry is set to managed to enable building and pushing of images.

   • Run:
$ oc edit configs.imageregistry/cluster

Then, change the line

managementState: Removed

to

managementState: Managed

1.15.12. 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.

1.16. 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 with the following command:
Alternatively, the following command notifies you when all of the clusters are available. It also retrieves and displays credentials:

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

**Example output**

<table>
<thead>
<tr>
<th>NAME</th>
<th>VERSION</th>
<th>AVAILABLE</th>
<th>PROGRESSING</th>
<th>DEGRADED</th>
<th>SINCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3h56m</td>
</tr>
<tr>
<td>cloud-credential</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>cluster-autoscaler</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>config-operator</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>6h39m</td>
</tr>
<tr>
<td>console</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3h59m</td>
</tr>
<tr>
<td>csi-snapshot-controller</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h12m</td>
</tr>
<tr>
<td>dns</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h15m</td>
</tr>
<tr>
<td>etcd</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>image-registry</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3h59m</td>
</tr>
<tr>
<td>ingress</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h30m</td>
</tr>
<tr>
<td>insights</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>kube-apiserver</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>kube-controller-manager</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>kube-scheduler</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>kube-storage-version-migrator</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h2m</td>
</tr>
<tr>
<td>machine-api</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>machine-approver</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>6h34m</td>
</tr>
<tr>
<td>machine-config</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3h56m</td>
</tr>
<tr>
<td>marketplace</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h2m</td>
</tr>
<tr>
<td>monitoring</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>6h31m</td>
</tr>
<tr>
<td>network</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>node-tuning</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h30m</td>
</tr>
<tr>
<td>openshift-apiserver</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3h56m</td>
</tr>
<tr>
<td>openshift-controller-manager</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h36m</td>
</tr>
<tr>
<td>openshift-samples</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h30m</td>
</tr>
<tr>
<td>operator-lifecycle-manager</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>operator-lifecycle-manager-catalog</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>operator-lifecycle-manager-packageserver</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3h59m</td>
</tr>
<tr>
<td>service-ca</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>storage</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h30m</td>
</tr>
</tbody>
</table>

```

$ ./openshift-install --dir <installation_directory> wait-for install-complete
```

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

**Example output**

```
INFO Waiting up to 30m0s for the cluster to initialize...
```

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

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

- It is recommended that you use Ignition config files within 12 hours after they are generated because the 24-hour certificate rotates from 16 to 22 hours after the cluster is installed. By using the Ignition config files within 12 hours, you can avoid installation failure if the certificate update runs during installation.

2. Confirm that the Kubernetes API server is communicating with the pods.

   a. To view a list of all pods, use the following command:

   ```bash
   $ oc get pods --all-namespaces
   
   Example output
   
<table>
<thead>
<tr>
<th>NAMESPACE</th>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>openshift-apiserver-operator</td>
<td>openshift-apiserver-operator-85cb746d55-zqh8</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td></td>
<td>Running</td>
<td>1</td>
<td>9m</td>
</tr>
<tr>
<td>openshift-apiserver</td>
<td>apiserver-67b9g</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td></td>
<td>Running</td>
<td>0</td>
<td>3m</td>
</tr>
<tr>
<td>openshift-apiserver</td>
<td>apiserver-ljcmx</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td></td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>openshift-apiserver</td>
<td>apiserver-z25h4</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td></td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>openshift-authentication-operator</td>
<td>authentication-operator-69d5d8bf84-vh2n8</td>
<td>1/1</td>
<td>Running</td>
</tr>
<tr>
<td></td>
<td>Running</td>
<td>0</td>
<td>5m</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
   ```

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

   ```bash
   $ oc logs <pod_name> -n <namespace>
   
   1 Specify the pod name and namespace, as shown in the output of the previous command.
   ```

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

1.1.17. Telemetry access for OpenShift Container Platform
In OpenShift Container Platform 4.6, the Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, requires internet access. If your cluster is connected to the internet, Telemetry runs automatically, and your cluster is registered to OpenShift Cluster Manager.

After you confirm that your OpenShift Cluster Manager inventory is correct, either maintained automatically by Telemetry or manually by using OpenShift Cluster Manager, use subscription watch to track your OpenShift Container Platform subscriptions at the account or multi-cluster level.

Additional resources
- See About remote health monitoring for more information about the Telemetry service

1.1.18. Collecting debugging information

You can gather debugging information that might help you to troubleshoot and debug certain issues with an OpenShift Container Platform installation on IBM Z.

Prerequisites
- The oc CLI tool installed.

Procedure
1. Log in to the cluster:
   $ oc login
2. On the node you want to gather hardware information about, start a debugging container:
   $ oc debug node/<nodename>
3. Change to the /host file system and start toolbox:
   $ chroot /host
   $ toolbox
4. Collect the dbginfo data:
   $ dbginfo.sh
5. You can then retrieve the data, for example, using scp.

Additional resources
- See How to generate SOSREPORT within OpenShift4 nodes without SSH.

1.1.19. Next steps
- Customize your cluster.
- If necessary, you can opt out of remote health reporting.
1.2. INSTALLING A CLUSTER ON IBM Z AND LINUXONE IN A RESTRICTED NETWORK

In OpenShift Container Platform version 4.6, you can install a cluster on IBM Z and LinuxONE infrastructure that you provision in a restricted network.

**NOTE**

While this document refers to only IBM Z, all information in it also applies to LinuxONE.

**IMPORTANT**

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

**Prerequisites**

- Create a mirror registry for installation in a restricted network and obtain the `imageContentSources` data for your version of OpenShift Container Platform.

- Before you begin the installation process, you must move or remove any existing installation files. This ensures that the required installation files are created and updated during the installation process.

**IMPORTANT**

Ensure that installation steps are done from a machine with access to the installation media.

- Provision persistent storage using NFS for your cluster. To deploy a private image registry, your storage must provide `ReadWriteMany` access modes.

- Review details about the [OpenShift Container Platform installation and update processes](#).

- If you use a firewall and plan to use telemetry, you must configure the firewall to allow the sites that your cluster requires access to.

**NOTE**

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

### 1.2.1. About installations in restricted networks

In OpenShift Container Platform 4.6, you can perform an installation that does not require an active connection to the Internet to obtain software components. Restricted network installations can be completed using installer-provisioned infrastructure or user-provisioned infrastructure, depending on the cloud platform to which you are installing the cluster.

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 Route 53 DNS and IAM services, 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 registry on a mirror host, which can access both the Internet and your closed network, or by using other methods that meet your restrictions.

**IMPORTANT**

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 using user-provisioned infrastructure. Completing this test installation might make it easier to isolate and troubleshoot any issues that might arise during your installation in a restricted network.

### 1.2.1.1. 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 image stream tags.

### 1.2.2. Internet access for OpenShift Container Platform

In OpenShift Container Platform 4.6, you require access to the Internet to obtain the images that are necessary to install your cluster.

You must have Internet access to:

- Access **OpenShift Cluster Manager** to download the installation program and perform subscription management. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster.
- Access **Quay.io** to obtain the packages that are required to install your cluster.
- Obtain the packages that are required to perform cluster updates.

**IMPORTANT**

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

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

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

#### 1.2.3.1. Required machines

The smallest OpenShift Container Platform clusters require the following hosts:
One temporary bootstrap machine

Three control plane, or master, machines

At least two compute machines, which are also known as worker machines.

**NOTE**

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

**IMPORTANT**

To improve high availability of your cluster, distribute the control plane machines over different z/VM instances on at least two physical machines.

The bootstrap and control plane machines must use Red Hat Enterprise Linux CoreOS (RHCOS) as the operating system. However, the compute machines can choose between Red Hat Enterprise Linux CoreOS (RHCOS) or Red Hat Enterprise Linux (RHEL) 7.9.

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

### 1.2.3.2. Network connectivity requirements

All the Red Hat Enterprise Linux CoreOS (RHCOS) machines require network in initramfs during boot to fetch Ignition config files from the Machine Config Server. The machines are configured with static IP addresses. No DHCP server is required. Additionally, each OpenShift Container Platform node in the cluster must have access to a Network Time Protocol (NTP) server.

### 1.2.3.3. IBM Z network connectivity requirements

To install on IBM Z under z/VM, you require a single z/VM virtual NIC in layer 2 mode. You also need:

- A direct-attached OSA or RoCE network adapter
- A z/VM VSwitch set up. For a preferred setup, use OSA link aggregation.

### 1.2.3.4. Minimum resource requirements

Each cluster machine must meet the following minimum requirements:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bootstrap</td>
<td>RHCOS</td>
<td>4</td>
<td>16 GB</td>
<td>100 GB</td>
<td>N/A</td>
</tr>
<tr>
<td>Control plane</td>
<td>RHCOS</td>
<td>4</td>
<td>16 GB</td>
<td>100 GB</td>
<td>N/A</td>
</tr>
<tr>
<td>Compute</td>
<td>RHCOS</td>
<td>2</td>
<td>8 GB</td>
<td>100 GB</td>
<td>N/A</td>
</tr>
</tbody>
</table>
One vCPU is equivalent to one physical core when simultaneous multithreading (SMT), or hyperthreading, is not enabled. When enabled, use the following formula to calculate the corresponding ratio: \((\text{threads per core} \times \text{cores}) \times \text{sockets} = \text{vCPUs}\).

1.2.3.5. Minimum IBM Z system environment

You can install OpenShift Container Platform version 4.6 on the following IBM hardware:

- IBM z15 (all models), IBM z14 (all models), IBM z13, and IBM z13s
- LinuxONE, any version

**Hardware requirements**

- The equivalent of 6 IFLs, which are SMT2 enabled, for each cluster.
- At least one network connection to both connect to the LoadBalancer service and to serve data for traffic outside the cluster.

**NOTE**

You can use dedicated or shared IFLs to assign sufficient compute resources. Resource sharing is one of the key strengths of IBM Z. However, you must adjust capacity correctly on each hypervisor layer and ensure sufficient resources for every OpenShift Container Platform cluster.

**IMPORTANT**

Since the overall performance of the cluster can be impacted, the LPARs that are used to setup the OpenShift Container Platform clusters must provide sufficient compute capacity. In this context, LPAR weight management, entitlements, and CPU shares on the hypervisor level play an important role.

**Operating system requirements**

- One instance of z/VM 7.1 or later

On your z/VM instance, set up:

- 3 guest virtual machines for OpenShift Container Platform control plane machines
- 2 guest virtual machines for OpenShift Container Platform compute machines
- 1 guest virtual machine for the temporary OpenShift Container Platform bootstrap machine

**IBM Z network connectivity requirements**

To install on IBM Z under z/VM, you require a single z/VM virtual NIC in layer 2 mode. You also need:

- A direct-attached OSA or RoCE network adapter
- A z/VM VSwitch set up. For a preferred setup, use OSA link aggregation.

**Disk storage for the z/VM guest virtual machines**

- FICON attached disk storage (DASDs). These can be z/VM minidisks, fullpack minidisks, or dedicated DASDs, all of which must be formatted as CDL, which is the default. To reach the
minimum required DASD size for Red Hat Enterprise Linux CoreOS (RHCOS) installations, you need extended address volumes (EAV). If available, use HyperPAV to ensure optimal performance.

- FCP attached disk storage

Storage / Main Memory

- 16 GB for OpenShift Container Platform control plane machines
- 8 GB for OpenShift Container Platform compute machines
- 16 GB for the temporary OpenShift Container Platform bootstrap machine

1.2.3.6. Preferred IBM Z system environment

Hardware requirements

- 3 LPARS that each have the equivalent of 6 IFLs, which are SMT2 enabled, for each cluster.
- Two network connections to connect to both connect to the LoadBalancer service and to serve data for traffic outside the cluster.
- HiperSockets, which are attached to a node either directly as a device or by bridging with one z/VM VSWITCH to be transparent to the z/VM guest. To directly connect HiperSockets to a node, you must set up a gateway to the external network via a RHEL 8 guest to bridge to the HiperSockets network.

Operating system requirements

- 2 or 3 instances of z/VM 7.1 or later for high availability

On your z/VM instances, set up:

- 3 guest virtual machines for OpenShift Container Platform control plane machines, one per z/VM instance.
- At least 6 guest virtual machines for OpenShift Container Platform compute machines, distributed across the z/VM instances.
- 1 guest virtual machine for the temporary OpenShift Container Platform bootstrap machine.
- To ensure the availability of integral components in an overcommitted environment, increase the priority of the control plane by using the CP command SET SHARE. Do the same for infrastructure nodes, if they exist. See SET SHARE in IBM Documentation.

IBM Z network connectivity requirements

To install on IBM Z under z/VM, you require a single z/VM virtual NIC in layer 2 mode. You also need:

- A direct-attached OSA or RoCE network adapter
- A z/VM VSwitch set up. For a preferred setup, use OSA link aggregation.

Disk storage for the z/VM guest virtual machines

- FICON attached disk storage (DASDs). These can be z/VM minidisks, fullpack minidisks, or dedicated DASDs, all of which must be formatted as CDL, which is the default. To reach the minimum required DASD size for Red Hat Enterprise Linux CoreOS (RHCOS) installations, you...
need extended address volumes (EAV). If available, use HyperPAV and High Performance FICON (zHPF) to ensure optimal performance.

- FCP attached disk storage

**Storage / Main Memory**

- 16 GB for OpenShift Container Platform control plane machines
- 8 GB for OpenShift Container Platform compute machines
- 16 GB for the temporary OpenShift Container Platform bootstrap machine

**1.2.3.7. 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.

**Additional resources**

- See Bridging a HiperSockets LAN with a z/VM Virtual Switch in IBM Documentation.
- See Scaling HyperPAV alias devices on Linux guests on z/VM for performance optimization.
- See Topics in LPAR performance for LPAR weight management and entitlements.

**1.2.4. Creating the user-provisioned infrastructure**

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

**Prerequisites**

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

**Procedure**

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

**1.2.4.1. Networking requirements for user-provisioned infrastructure**
All the Red Hat Enterprise Linux CoreOS (RHCOS) machines require network in initramfs during boot to fetch Ignition config from the machine config server.

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

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

The Kubernetes API server 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 1.10. All machines to all machines

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMP</td>
<td>N/A</td>
<td>Network reachability tests</td>
</tr>
<tr>
<td>TCP</td>
<td>1936</td>
<td>Metrics</td>
</tr>
<tr>
<td></td>
<td>9000-9999</td>
<td>Host level services, including the node exporter on ports</td>
</tr>
<tr>
<td></td>
<td>9100-9101</td>
<td>and the Cluster Version Operator on port 9099.</td>
</tr>
<tr>
<td></td>
<td>10250-10259</td>
<td>The default ports that Kubernetes reserves</td>
</tr>
<tr>
<td></td>
<td>10256</td>
<td>openshift-sdn</td>
</tr>
<tr>
<td>UDP</td>
<td>4789</td>
<td>VXLAN and Geneve</td>
</tr>
<tr>
<td></td>
<td>6081</td>
<td>VXLAN and Geneve</td>
</tr>
<tr>
<td></td>
<td>9000-9999</td>
<td>Host level services, including the node exporter on ports</td>
</tr>
<tr>
<td></td>
<td>9100-9101</td>
<td>9100-9101.</td>
</tr>
<tr>
<td>TCP/UDP</td>
<td>30000-32767</td>
<td>Kubernetes node port</td>
</tr>
</tbody>
</table>

Table 1.11. All machines to control plane

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>6443</td>
<td>Kubernetes API</td>
</tr>
</tbody>
</table>

Table 1.12. Control plane machines to control plane machines
Network topology requirements
The infrastructure that you provision for your cluster must meet the following network topology requirements.

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

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

   **IMPORTANT**
   Do not configure session persistence for an API load balancer.

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

### Table 1.13. API load balancer

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

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

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

- Layer 4 load balancing only. This can be referred to as Raw TCP, SSL Passthrough, or SSL Bridge mode. If you use SSL Bridge mode, you must enable Server Name Indication (SNI) for the Ingress routes.

- A connection-based or session-based persistence is recommended, based on the options available and types of applications that will be hosted on the platform.

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

<table>
<thead>
<tr>
<th>Port</th>
<th>Back-end machines (pool members)</th>
<th>Internal</th>
<th>External</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>443</td>
<td>The machines that run the Ingress router pods, compute, or worker, by default.</td>
<td>X</td>
<td>X</td>
<td>HTTPS traffic</td>
</tr>
<tr>
<td>80</td>
<td>The machines that run the Ingress router pods, compute, or worker, by default.</td>
<td>X</td>
<td>X</td>
<td>HTTP traffic</td>
</tr>
</tbody>
</table>

**TIP**

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

**NOTE**

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

**NTP configuration**

OpenShift Container Platform clusters are configured to use a public Network Time Protocol (NTP) server by default. If you want to use a local enterprise NTP server, or if your cluster is being deployed in a disconnected network, you can configure the cluster to use a specific time server. For more information, see the documentation for *Configuring chrony time service*.

If a DHCP server provides NTP server information, the chrony time service on the Red Hat Enterprise Linux CoreOS (RHCOS) machines read the information and can sync the clock with the NTP servers.

**Additional resources**
1.2.4.2. User-provisioned DNS requirements

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

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

### Table 1.15. Required DNS records

<table>
<thead>
<tr>
<th>Component</th>
<th>Record</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kubernetes API</td>
<td>api.&lt;cluster_name&gt;.&lt;base_domain&gt;.</td>
<td>Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the load balancer for the control plane machines. These records must be resolvable by both clients external to the cluster and from all the nodes within the cluster.</td>
</tr>
<tr>
<td></td>
<td>api-int.&lt;cluster_name&gt;.&lt;base_domain&gt;.</td>
<td>Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the load balancer for the control plane machines. These records must be resolvable from all the nodes within the cluster.</td>
</tr>
<tr>
<td>Routes</td>
<td>*.apps.&lt;cluster_name&gt;.&lt;base_domain&gt;.</td>
<td>Add a wildcard DNS A/AAAA or CNAME record that refers to the load balancer that targets the machines that run the Ingress router pods, which are the worker nodes by default. These records must be resolvable by both clients external to the cluster and from all the nodes within the cluster.</td>
</tr>
<tr>
<td>Bootstrap</td>
<td>bootstrap.&lt;cluster_name&gt;.&lt;base_domain&gt;.</td>
<td>Add a DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the bootstrap machine. These records must be resolvable by the nodes within the cluster.</td>
</tr>
</tbody>
</table>

**IMPORTANT**

The API server must be able to resolve the worker nodes by the host names that are recorded in Kubernetes. If the API server cannot resolve the node names, then proxied API calls can fail, and you cannot retrieve logs from pods.
TIP

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

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

**Example 1.3. Sample DNS zone database**

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

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

Example 1.4. Sample DNS zone database for reverse records

$TTL 1W
@ IN SOA ns1.example.com. root (2019070700 ; serial
3H ; refresh (3 hours)
30M ; retry (30 minutes)
2W ; expiry (2 weeks)
1W ) ; minimum (1 week)
IN NS ns1.example.com.
;
; The syntax is "last octet" and the host must have an FQDN with a trailing dot.
97 IN PTR master0.ocp4.example.com.
98 IN PTR master1.ocp4.example.com.
99 IN PTR master2.ocp4.example.com.
;
96 IN PTR bootstrap.ocp4.example.com.
;
5 IN PTR api.ocp4.example.com.
5 IN PTR api-int.ocp4.example.com.
;
11 IN PTR worker0.ocp4.example.com.
7 IN PTR worker1.ocp4.example.com.
;
;EOF

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

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

NOTE

In a production environment, you require disaster recovery and debugging.
IMPORTANT
Do not skip this procedure in production environments where disaster recovery and debugging is required.

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.

Procedure

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

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

   Specify the path and file name, such as ~/.ssh/id_rsa, of the new SSH key. If you have an existing key pair, ensure your public key is in the your ~/.ssh directory.

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

   NOTE
   If you plan to install an OpenShift Container Platform cluster that uses FIPS Validated / Modules in Process cryptographic libraries on the x86_64 architecture, do not create a key that uses the ed25519 algorithm. Instead, create a key that uses the rsa or ecdsa algorithm.

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

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

   Example output

   Agent pid 31874

   NOTE
   If your cluster is in FIPS mode, only use FIPS-compliant algorithms to generate the SSH key. The key must be either RSA or ECDSA.

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

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

   Example output

   Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
Specify the path and file name for your SSH private key, such as ~/.ssh/id_rsa

Next steps

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

1.2.6. Manually creating the installation configuration file

For installations of OpenShift Container Platform that use user-provisioned infrastructure, you 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.
The `install-config.yaml` file is consumed during the next step of the installation process. You must back it up now.

1.2.6.1. Installation configuration parameters

Before you deploy an OpenShift Container Platform cluster, you provide parameter values to describe your account on the cloud platform that hosts your cluster and optionally customize your cluster’s platform. When you create the `install-config.yaml` installation configuration file, you provide values for the required parameters through the command line. If you customize your cluster, you can modify the `install-config.yaml` file to provide more details about the platform.

**NOTE**

After installation, you cannot modify these parameters in the `install-config.yaml` file.

**IMPORTANT**

The `openshift-install` command does not validate field names for parameters. If an incorrect name is specified, the related file or object is not created, and no error is reported. Ensure that the field names for any parameters that are specified are correct.

1.2.6.1.1. Required configuration parameters

Required installation configuration parameters are described in the following table:

**Table 1.16. Required parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiVersion</td>
<td>The API version for the <code>install-config.yaml</code> content. The current version is <code>v1</code>. The installer may also support older API versions.</td>
<td>String</td>
</tr>
<tr>
<td>baseDomain</td>
<td>The base domain of your cloud provider. The base domain is used to create routes to your OpenShift Container Platform cluster components. The full DNS name for your cluster is a combination of the <code>baseDomain</code> and <code>metadata.name</code> parameter values that uses the <code>&lt;metadata.name&gt;</code>, <code>&lt;baseDomain&gt;</code> format.</td>
<td>A fully-qualified domain or subdomain name, such as <code>example.com</code>.</td>
</tr>
</tbody>
</table>
### Parameter: metadata

#### ObjectMeta

- **Description:** Kubernetes resource ObjectMeta, from which only the name parameter is consumed.

- **Values:** Object

### Parameter: metadata.name

- **Description:** The name of the cluster. DNS records for the cluster are all subdomains of 

  \{{.metadata.name}}. {{.baseDomain}}.

- **Values:** String of lowercase letters, hyphens (-), and periods (.), such as dev.

### Parameter: platform

- **Description:** The configuration for the specific platform upon which to perform the installation: aws, baremetal, azure, openstack, ovirt, vsphere.

- **Values:** Object

### Parameter: pullSecret

- **Description:** Get a pull secret from the Red Hat OpenShift Cluster Manager to authenticate downloading container images for OpenShift Container Platform components from services such as Quay.io.

```json
{
  "auths":{
    "cloud.openshift.com":{
      "auth":"b3Blb=",
      "email":"you@example.com"
    },
    "quay.io":{
      "auth":"b3Blb=",
      "email":"you@example.com"
    }
  }
}
```

### 1.2.6.1.2. Network configuration parameters

You can customize your installation configuration based on the requirements of your existing network infrastructure. For example, you can expand the IP address block for the cluster network or provide different IP address blocks than the defaults.

Only IPv4 addresses are supported.

### Table 1.17. Network parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
</table>


### Parameter | Description | Values
---|---|---
**networking** | The configuration for the cluster network. | Object

**NOTE**
You cannot modify parameters specified by the `networking` object after installation.

**networking.networkType** | The cluster network provider Container Network Interface (CNI) plug-in to install. | Either OpenShiftSDN or OVNKubernetes. The default value is OpenShiftSDN.

**networking.clusterNetwork** | The IP address blocks for pods. | An array of objects. For example:

```yaml
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
```

The default value is **10.128.0.0/14** with a host prefix of **/23**.

If you specify multiple IP address blocks, the blocks must not overlap.


**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`. A `hostPrefix` value of **23** provides **510** ($2^{32} - 2 - 2$) pod IP addresses. | A subnet prefix. The default value is **23**.

**networking.serviceNetwork** | The IP address block for services. The default value is **172.30.0.0/16**. | An array with an IP address block in CIDR format. For example:

```yaml
networking:
  serviceNetwork:
    - 172.30.0.0/16
```

The OpenShift SDN and OVN-Kubernetes network providers support only a single IP address block for the service network.
### Networking: machineNetwork

The IP address blocks for machines. If you specify multiple IP address blocks, the blocks must not overlap. If you specify multiple IP kernel arguments, the `machineNetwork.cidr` value must be the CIDR of the primary network.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>networking.machineNetwork</code></td>
<td>The IP address blocks for machines. If you specify multiple IP address blocks, the blocks must not overlap. If you specify multiple IP kernel arguments, the <code>machineNetwork.cidr</code> value must be the CIDR of the primary network.</td>
<td>An array of objects. For example:</td>
</tr>
<tr>
<td><code>networking.machineNetwork.cidr</code></td>
<td>Required if you use <code>networking.machineNetwork</code>. An IP address block. The default value is <code>10.0.0.0/16</code> for all platforms other than libvirt. For libvirt, the default value is <code>192.168.126.0/24</code>.</td>
<td>An IP network block in CIDR notation. For example, <code>10.0.0.0/16</code>.</td>
</tr>
</tbody>
</table>

**NOTE**

Set the `networking.machineNetwork` to match the CIDR that the preferred NIC resides in.

### 1.2.6.1.3. Optional configuration parameters

Optional installation configuration parameters are described in the following table:

**Table 1.18. Optional parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>additionalTrustBundle</code></td>
<td>A PEM-encoded X.509 certificate bundle that is added to the nodes’ trusted certificate store. This trust bundle may also be used when a proxy has been configured.</td>
<td>String</td>
</tr>
<tr>
<td><code>compute</code></td>
<td>The configuration for the machines that comprise the compute nodes.</td>
<td>Array of machine-pool objects. For details, see the following &quot;Machine-pool&quot; table.</td>
</tr>
<tr>
<td><code>compute.architecture</code></td>
<td>Determines the instruction set architecture of the machines in the pool. Currently, heterogeneous clusters are not supported, so all pools must specify the same architecture. Valid values are <code>amd64</code> (the default).</td>
<td>String</td>
</tr>
</tbody>
</table>
Whether to enable or disable simultaneous multithreading, or hyperthreading, on compute machines. By default, simultaneous multithreading is enabled to increase the performance of your machines’ cores.

**IMPORTANT**

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>compute.hyperthreading</code></td>
<td>Whether to enable or disable simultaneous multithreading, or hyperthreading, on compute machines. By default, simultaneous multithreading is enabled to increase the performance of your machines’ cores.</td>
<td>Enabled or Disabled</td>
</tr>
<tr>
<td><code>compute.name</code></td>
<td>Required if you use <code>compute</code>. The name of the machine pool.</td>
<td>worker</td>
</tr>
<tr>
<td><code>compute.platform</code></td>
<td>Required if you use <code>compute</code>. Use this parameter to specify the cloud provider to host the worker machines. This parameter value must match the <code>controlPlane.platform</code> parameter value.</td>
<td>aws, azure, gcp, openstack, ovirt, vsphere, or {}</td>
</tr>
<tr>
<td><code>compute.replicas</code></td>
<td>The number of compute machines, which are also known as worker machines, to provision.</td>
<td>A positive integer greater than or equal to 2. The default value is 3.</td>
</tr>
<tr>
<td><code>controlPlane</code></td>
<td>The configuration for the machines that comprise the control plane.</td>
<td>Array of <code>MachinePool</code> objects. For details, see the following &quot;Machine-pool&quot; table.</td>
</tr>
<tr>
<td><code>controlPlane.architecture</code></td>
<td>Determines the instruction set architecture of the machines in the pool. Currently, heterogeneous clusters are not supported, so all pools must specify the same architecture. Valid values are amd64 (the default).</td>
<td>String</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>controlPlane.hypertreading</td>
<td>Whether to enable or disable simultaneous multithreading, or <code>hyperthreading</code>, on control plane machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores.</td>
<td>Enabled or Disabled</td>
</tr>
<tr>
<td><strong>IMPORTANT</strong></td>
<td>If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.</td>
<td></td>
</tr>
<tr>
<td>controlPlane.name</td>
<td>Required if you use <code>controlPlane</code>. The name of the machine pool.</td>
<td>master</td>
</tr>
<tr>
<td>controlPlane.platform</td>
<td>Required if you use <code>controlPlane</code>. Use this parameter to specify the cloud provider that hosts the control plane machines. This parameter value must match the <code>compute.platform</code> parameter value.</td>
<td>aws, azure, gcp, openstack, ovirt, vsphere, or{}</td>
</tr>
<tr>
<td>controlPlane.replicas</td>
<td>The number of control plane machines to provision.</td>
<td>The only supported value is 3, which is the default value.</td>
</tr>
</tbody>
</table>
### credentialsMode

The Cloud Credential Operator (CCO) mode. If no mode is specified, the CCO dynamically tries to determine the capabilities of the provided credentials, with a preference for mint mode on the platforms where multiple modes are supported.

**NOTE**

Not all CCO modes are supported for all cloud providers. For more information on CCO modes, see the Cloud Credential Operator entry in the Red Hat Operators reference content.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>credentialsMode</td>
<td>The Cloud Credential Operator (CCO) mode. If no mode is specified, the CCO dynamically tries to determine the capabilities of the provided credentials, with a preference for mint mode on the platforms where multiple modes are supported.</td>
<td>Mint, Passthrough, Manual, or an empty string (“”).</td>
</tr>
</tbody>
</table>
Enable or disable FIPS mode. The default is `false` (disabled). If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.

**IMPORTANT**
The use of FIPS Validated / Modules in Process cryptographic libraries is only supported on OpenShift Container Platform deployments on the `x86_64` architecture.

**NOTE**
If you are using Azure File storage, you cannot enable FIPS mode.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>fips</td>
<td>Enable or disable FIPS mode. The default is <code>false</code> (disabled). If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.</td>
<td><code>false</code> or <code>true</code></td>
</tr>
</tbody>
</table>

### imageContentSources
Sources and repositories for the release-image content. Array of objects. Includes a `source` and, optionally, `mirrors`, as described in the following rows of this table.

**imageContentSources.source**
Required if you use `imageContentSources`. Specify the repository that users refer to, for example, in image pull specifications.

**imageContentSources.mirrors**
Specify one or more repositories that may also contain the same images. Array of strings
### publish

How to publish or expose the user-facing endpoints of your cluster, such as the Kubernetes API, OpenShift routes.

**Internal** or **External**. The default value is **External**.

Setting this field to **Internal** is not supported on non-cloud platforms.

**IMPORTANT**

If the value of the field is set to **Internal**, the cluster will become non-functional. For more information, refer to [BZ#1953035](#).

### sshKey

The SSH key or keys to authenticate access your cluster machines.

**NOTE**

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

One or more keys. For example:

```
sshKey:
- <key1>
- <key2>
- <key3>
```

---

#### 1.2.6.2. Sample `install-config.yaml` file for IBM Z

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.

```yaml
apiVersion: v1
baseDomain: example.com
compute:
- hyperthreading: Enabled
  name: worker
  replicas: 0
  architecture: s390x
controlPlane:
- hyperthreading: Enabled
  name: master
  replicas: 3
  architecture: s390x
metadata:
  name: test
```
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
    - hostPrefix: 23
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
platform:
  none: {}
  fips: false
pullSecret: "{"auths":{"<local_registry>": {"auth": "<credentials>","email": "you@example.com"}}}
sshKey: 'ssh-ed25519 AAAA...
additionalTrustBundle: |
  -----BEGIN CERTIFICATE-----
  ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
  -----END CERTIFICATE-----
imageContentSources:
  - mirrors:
    - <local_repository>/ocp4/openshift4
      source: quay.io/openshift-release-dev/ocp-release
    - mirrors:
      - <local_repository>/ocp4/openshift4
        source: quay.io/openshift-release-dev/ocp-v4.0-art-dev

1. The base domain of the cluster. All DNS records must be sub-domains of this base and include the cluster name.

2. 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. Only one control plane pool is used.

3. Whether to enable or disable simultaneous multithreading (SMT), or hyperthreading. By default, SMT is enabled to increase the performance of your machines’ cores. You can disable it by setting the parameter value to Disabled. If you disable SMT, you must disable it in all cluster machines; this includes both control plane and compute machines.

   **NOTE**
   
   Simultaneous multithreading (SMT) is enabled by default. If SMT is not enabled in your BIOS settings, the hyperthreading parameter has no effect.

   **IMPORTANT**
   
   If you disable hyperthreading, whether in the BIOS or in the install-config.yaml, 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.

7. The number of control plane machines that you add to the cluster. Because the cluster uses this
The cluster name that you specified in your DNS records.

A block of IP addresses from which pod IP addresses are allocated. This block must not overlap with existing physical networks. These IP addresses are used for the pod network. If you need to access the pods from an external network, you must configure load balancers and routers to manage the traffic.

**NOTE**

Class E CIDR range is reserved for a future use. To use the Class E CIDR range, you must ensure your networking environment accepts the IP addresses within the Class E CIDR range.

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.

The IP address pool to use for service IP addresses. You can enter only one IP address pool. This block must not overlap with existing physical networks. If you need to access the services from an external network, configure load balancers and routers to manage the traffic.

You must set the platform to `none`. You cannot provide additional platform configuration variables for IBM Z infrastructure.

Whether to enable or disable FIPS mode. By default, FIPS mode is not enabled. If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.

**IMPORTANT**

The use of FIPS Validated / Modules in Process cryptographic libraries is only supported on OpenShift Container Platform deployments on the `x86_64` architecture.

For `<local_registry>`, specify the registry domain name, and optionally the port, that your mirror registry uses to serve content. For example `registry.example.com` or `registry.example.com:5000`. For `<credentials>`, specify the base64-encoded user name and password for your mirror registry.

The public portion of the default SSH key for the `core` user in Red Hat Enterprise Linux CoreOS (RHCOS).

**NOTE**

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

Add the `additionalTrustBundle` parameter and value. The value must be the contents of the certificate file that you used for your mirror registry, which can be an existing, trusted certificate authority or the self-signed certificate that you generated for the mirror registry.
1.2.6.3. 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

- You have an existing `install-config.yaml` file.
- You reviewed the sites that your cluster requires access to and determined whether any of them need to bypass the proxy. By default, all cluster egress traffic is proxied, including calls to hosting cloud provider APIs. You added sites to the Proxy object’s `spec.noProxy` field to bypass the proxy if necessary.

**NOTE**

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

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

Procedure

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

   ```yaml
   apiVersion: v1
   baseDomain: my.domain.com
   proxy:
     httpProxy: http://<username>:<pswd>@<ip>:<port> 1
     httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
     noProxy: example.com 3
   additionalTrustBundle: |
     -----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.
   3 A comma-separated list of destination domain names, IP addresses, or other network CIDRs to exclude from proxying. Preface a domain with . to match subdomains only. For example, example.com matches x.example.com but not example.com.
example, `.y.com` matches `x.y.com`, but not `y.com`. Use `*` to bypass the proxy for all destinations.

If provided, the installation program generates a config map that is named `user-ca-bundle` in the `openshift-config` namespace to hold the additional CA certificates. If you provide `additionalTrustBundle` and at least one proxy setting, the `Proxy` object is configured to reference the `user-ca-bundle` config map in the `trustedCA` field. The Cluster Network Operator then creates a `trusted-ca-bundle` config map that merges the contents specified for the `trustedCA` parameter with the RHCOS trust bundle. 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.

### 1.2.7. Creating the Kubernetes manifest and Ignition config files

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

The installation configuration file transforms into the Kubernetes manifests. The manifests wrap into the Ignition configuration files, which are later used to create the cluster.

**IMPORTANT**

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

- It is recommended that you use Ignition config files within 12 hours after they are generated because the 24-hour certificate rotates from 16 to 22 hours after the cluster is installed. By using the Ignition config files within 12 hours, you can avoid installation failure if the certificate update runs during installation.

**Prerequisites**

- You obtained the OpenShift Container Platform installation program.
• You created the `install-config.yaml` installation configuration file.

Procedure

1. Change to the directory that contains the installation program and generate the Kubernetes manifests for the cluster:

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

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

2. Check that the `mastersSchedulable` parameter in the `<installation_directory>/manifests/cluster-scheduler-02-config.yml` Kubernetes manifest file is set to `false`. This setting prevents pods from being scheduled on the control plane machines:

   a. Open the `<installation_directory>/manifests/cluster-scheduler-02-config.yml` file.
   b. Locate the `mastersSchedulable` parameter and ensure that it is set to `false`.
   c. Save and exit the file.

3. To create the Ignition configuration files, run the following command from the directory that contains the installation program:

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

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

   The following files are generated in the directory:

   - auth
     - kubeadmin-password
     - kubeconfig
     - bootstrap.ign
     - master.ign
     - metadata.json
     - worker.ign

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

Before you install a cluster on IBM Z infrastructure that you provision, you must install RHCOS on z/VM guest virtual machines for the cluster to use. Complete the following steps to create the machines.

Prerequisites

• An FTP server running on your provisioning machine that is accessible to the machines you create.

Procedure
1. Log in to Linux on your provisioning machine.

2. Obtain the Red Hat Enterprise Linux CoreOS (RHCOS) kernel, initramfs, and rootfs files from the RHCOS image mirror.

   **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. Only use the appropriate kernel, initramfs, and rootfs artifacts described in the following procedure.

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

   - **kernel**: `rhcos-<version>-live-kernel-<architecture>`
   - **initramfs**: `rhcos-<version>-live-initramfs.<architecture>.img`
   - **rootfs**: `rhcos-<version>-live-rootfs.<architecture>.img`

   **NOTE**
   
   The rootfs image is the same for FCP and DASD.

3. Create parameter files. The following parameters are specific for a particular virtual machine:

   - For **coreos.inst.install_dev=**, specify **dasda** for a DASD installation, or **sda** for FCP. Note that FCP requires `zfcp.allow_lun_scan=0`.

   - For **rd.dasd=**, specifies the DASD where RHCOS is to be installed.

   - **rd.zfcp=**<adapter>,<wwpn>,<lun> specifies the FCP disk to install RHCOS on.

   - For **ip=**, specify the following seven entries:
     
     i. The IP address for the machine.

     ii. An empty string.

     iii. The gateway.

     iv. The netmask.

     v. The machine host and domain name in the form `hostname.domainname`. Omit this value to let RHCOS decide.

     vi. The network interface name. Omit this value to let RHCOS decide.

     vii. If you use static IP addresses, an empty string.

   - For **coreos.inst.ignition_url=**, specify the Ignition file for the machine role. Use `bootstrap.ign`, `master.ign`, or `worker.ign`. Only HTTP and HTTPS protocols are supported.

   - For **coreos.live.rootfs_url=**, specify the matching rootfs artifact for the kernel and...
- For `coreos.live.roots_url`, specify the matching rootfs artifact for the kernel and initramfs you are booting. Only HTTP and HTTPS protocols are supported.

- All other parameters can stay as they are.

Example parameter file, `bootstrap-0.parm`, for the bootstrap machine:

```
rd.neednet=1 \nconsole=ttyscp0 \ncoreos.inst.install_dev=dasda \ncoreos.live.roots_url=http://cl1.provide.example.com:8080/assets/rhcos-live-rootsfs.s390x.img \ncoreos.inst.ignition_url=http://cl1.provide.example.com:8080/ignition/bootstrap.ign \nip=172.18.78.2::172.18.78.1:255.255.255.0:::none nameserver=172.18.78.1 \nrznet=qeth,0.0.bdf0,0.0.bdf1,0.0.bdf2,layer2=1,portno=0 \nzfcp.allow_lun_scan=0 \nrd.dasd=0.0.3490
```

Write all options in the parameter file as a single line and make sure you have no newline characters.

4. Transfer the initramfs, kernel, parameter files, and RHCOS images to z/VM, for example with FTP. For details about how to transfer the files with FTP and boot from the virtual reader, see Installing under Z/VM.

5. Punch the files to the virtual reader of the z/VM guest virtual machine that is to become your bootstrap node.

See PUNCH in IBM Documentation.

**TIP**

You can use the CP PUNCH command or, if you use Linux, the `vmur` command to transfer files between two z/VM guest virtual machines.


7. IPL the bootstrap machine from the reader:

   ```
   $ ipl c
   ```

   See IPL in IBM Documentation.

8. Repeat this procedure for the other machines in the cluster.

### 1.2.8.1. Advanced RHCOS installation reference

This section illustrates the networking configuration and other advanced options that allow you to modify the Red Hat Enterprise Linux CoreOS (RHCOS) manual installation process. The following tables describe the kernel arguments and command-line options you can use with the RHCOS live installer and the `coreos-installer` command.

**Routing and bonding options at RHCOS boot prompt**

If you install RHCOS from an ISO image, you can add kernel arguments manually when you boot that image to configure the node’s networking. If no networking arguments are used, the installation defaults to using DHCP.
IMPORTANT

When adding networking arguments, you must also add the `rd.neednet=1` kernel argument.

The following table describes how to use `ip=` , `nameserver=` , and `bond=` kernel arguments for live ISO installs.

### NOTE

Ordering is important when adding kernel arguments: `ip=`, `nameserver=`, and then `bond=`.

### Routing and bonding options for ISO

The following table provides examples for configuring networking of your Red Hat Enterprise Linux CoreOS (RHCOS) nodes. These are networking options that are passed to the `dracut` tool during system boot. For more information about the networking options supported by `dracut`, see the `dracut.cmdline` manual page.

<table>
<thead>
<tr>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>To configure an IP address, either use DHCP (<code>ip=dhcp</code>) or set an individual static IP address (<code>ip=&lt;host_ip&gt;</code>). Then identify the DNS server IP address (<code>nameserver=&lt;dns_ip&gt;</code>) on each node. This example sets:</td>
<td>ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none nameserver=4.4.4.41</td>
</tr>
<tr>
<td>- The node’s IP address to <strong>10.10.10.2</strong></td>
<td></td>
</tr>
<tr>
<td>- The gateway address to <strong>10.10.10.254</strong></td>
<td></td>
</tr>
<tr>
<td>- The netmask to <strong>255.255.255.0</strong></td>
<td></td>
</tr>
<tr>
<td>- The hostname to <strong>core0.example.com</strong></td>
<td></td>
</tr>
<tr>
<td>- The DNS server address to <strong>4.4.4.41</strong></td>
<td></td>
</tr>
<tr>
<td>Specify multiple network interfaces by specifying multiple <code>ip=</code> entries.</td>
<td>ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none nameserver=4.4.4.41</td>
</tr>
<tr>
<td>Optional: You can configure routes to additional networks by setting an <code>rd.route=</code> value.</td>
<td>To configure the default gateway:</td>
</tr>
<tr>
<td>If the additional network gateway is different from the primary network gateway, the default gateway must be the primary network gateway.</td>
<td><code>ip=:10.10.10.254:::</code></td>
</tr>
<tr>
<td></td>
<td>To configure the route for the additional network:</td>
</tr>
<tr>
<td></td>
<td><code>rd.route=20.20.20.0/24:20.20.20.254:enp2s0</code></td>
</tr>
<tr>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>Disable DHCP on a single interface, such as when there are two or more network interfaces and only one interface is being used.</td>
<td>ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none ip=:::core0.example.com:enp2s0:none</td>
</tr>
<tr>
<td>You can combine DHCP and static IP configurations on systems with multiple network interfaces.</td>
<td>ip=enp1s0:dhcp ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp2s0:none</td>
</tr>
<tr>
<td>Optional: You can configure VLANs on individual interfaces by using the vlan= parameter.</td>
<td>To configure a VLAN on a network interface and use a static IP address: ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp2s0.100:none vlan=enp2s0.100:enp2s0 To configure a VLAN on a network interface and to use DHCP: ip=enp2s0.100:dhcp vlan=enp2s0.100:enp2s0</td>
</tr>
<tr>
<td>You can provide multiple DNS servers by adding a nameserver= entry for each server.</td>
<td>nameserver=1.1.1.1 nameserver=8.8.8.8</td>
</tr>
<tr>
<td>Optional: Bonding multiple network interfaces to a single interface is supported using the bond= option. In these two examples:</td>
<td>To configure the bonded interface to use DHCP, set the bond’s IP address to dhcp. For example: bond=bond0:em1,em2:mode=active-backup ip=bond0:dhcp To configure the bonded interface to use a static IP address, enter the specific IP address you want and related information. For example: bond=bond0:em1,em2:mode=active-backup ip=10.10.10.2::10.10.254:255.255.255.0:core0.example.com:bond0:none</td>
</tr>
<tr>
<td>• The syntax for configuring a bonded interface is: bond=name[:network_interfaces] [:options] name is the bonding device name (bond0), network_interfaces represents a comma-separated list of physical (ethernet) interfaces (em1,em2), and options is a comma-separated list of bonding options. Enter modinfo bonding to see available options. • When you create a bonded interface using bond=, you must specify how the IP address is assigned and other information for the bonded interface.</td>
<td></td>
</tr>
</tbody>
</table>
Optional: You can configure VLANs on bonded interfaces by using the `vlan=` parameter.

To configure the bonded interface with a VLAN and to use DHCP:

```
ip=bond0.100:dhcp
bond=bond0:em1,em2:mode=active-backup
vlan=bond0.100:bond0
```

To configure the bonded interface with a VLAN and to use a static IP address:

```
ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:bond0.100:none
bond=bond0:em1,em2:mode=active-backup
vlan=bond0.100:bond0
```

Optional: Network teaming can be used as an alternative to bonding by using the `team=` parameter. In this example:

- The syntax for configuring a team interface is `team=name[:network_interfaces]`
  - `name` is the team device name (`team0`) and `network_interfaces` represents a comma-separated list of physical (ethernet) interfaces (`em1`, `em2`).

**NOTE**
Teaming is planned to be deprecated when RHCOS switches to an upcoming version of RHEL. For more information, see this Red Hat Knowledgebase Article.

To configure a network team:

```
team=team0:em1,em2
ip=team0:dhcp
```

1.2.9. Creating the cluster

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

**Prerequisites**

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

   ```bash
   $ ./openshift-install --dir <installation_directory> wait-for bootstrap-complete \  
   --log-level=info
   ```

   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`.

**Example output**

```
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com:6443...
INFO API v1.19.0 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
```

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

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

**IMPORTANT**

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

### 1.2.10. Logging in to the cluster by using the CLI

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**

- You deployed an OpenShift Container Platform cluster.
- You installed the `oc` CLI.

**Procedure**

1. Export the `kubeadmin` credentials:

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

   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:
1.2.11. Approving the certificate signing requests for your machines

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

Prerequisites

- You added machines to your cluster.

Procedure

1. Confirm that the cluster recognizes the machines:

   `$ oc get nodes`

   **Example output**

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>ROLES</th>
<th>AGE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>master-0</td>
<td>Ready</td>
<td>master</td>
<td>63m</td>
<td>v1.19.0</td>
</tr>
<tr>
<td>master-1</td>
<td>Ready</td>
<td>master</td>
<td>63m</td>
<td>v1.19.0</td>
</tr>
<tr>
<td>master-2</td>
<td>Ready</td>
<td>master</td>
<td>64m</td>
<td>v1.19.0</td>
</tr>
</tbody>
</table>

   The output lists all of the machines that you created.

   **NOTE**

   The preceding output might not include the compute nodes, also known as worker nodes, until some CSRs are approved.

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

   `$ oc get csr`

   **Example output**

<table>
<thead>
<tr>
<th>NAME</th>
<th>AGE</th>
<th>REQUESTOR</th>
<th>CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>csr-8b2br</td>
<td>15m</td>
<td>system:serviceaccount:openshift-machine-config-operator:node-bootstrapper</td>
<td>Pending</td>
</tr>
<tr>
<td>csr-8vnps</td>
<td>15m</td>
<td>system:serviceaccount:openshift-machine-config-operator:node-bootstrapper</td>
<td>Pending</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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. Once the client CSR is approved, the Kubelet creates a secondary CSR for the serving certificate, which requires manual approval. Then, subsequent serving certificate renewal requests are automatically approved by the machine-approver if the Kubelet requests a new certificate with identical parameters.

   NOTE

   For clusters running on platforms that are not machine API enabled, such as bare metal and other user-provisioned infrastructure, you must implement a method of automatically approving the kubelet serving certificate requests (CSRs). If a request is not approved, then the `oc exec`, `oc rsh`, and `oc logs` commands cannot succeed, because a serving certificate is required when the API server connects to the kubelet. Any operation that contacts the Kubelet endpoint requires this certificate approval to be in place. The method must watch for new CSRs, confirm that the CSR was submitted by the `node-bootstrapper` service account in the `system:node` or `system:admin` groups, and confirm the identity of the node.

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

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

     `<csr_name>` is the name of a CSR from the list of current CSRs.

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

     ```
     $ oc get csr -o go-template="{{range .items}}{{if not .status}}{{.metadata.name}}{{"\n"}}{{end}}{{end}}" | xargs --no-run-if-empty oc adm certificate approve
     ```

     Some Operators might not become available until some CSRs are approved.

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

   ```
   $ oc get csr
   ```

   Example output
5. If the remaining CSRs are not approved, and are in the **Pending** status, approve the CSRs for your cluster machines:

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

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

     `<csr_name>` is the name of a CSR from the list of current CSRs.

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

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

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

   ```
   $ oc get nodes
   ```

### Example output

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>ROLES</th>
<th>AGE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>master-0</td>
<td>Ready</td>
<td>master</td>
<td>73m</td>
<td>v1.20.0</td>
</tr>
<tr>
<td>master-1</td>
<td>Ready</td>
<td>master</td>
<td>73m</td>
<td>v1.20.0</td>
</tr>
<tr>
<td>master-2</td>
<td>Ready</td>
<td>master</td>
<td>74m</td>
<td>v1.20.0</td>
</tr>
<tr>
<td>worker-0</td>
<td>Ready</td>
<td>worker</td>
<td>11m</td>
<td>v1.20.0</td>
</tr>
<tr>
<td>worker-1</td>
<td>Ready</td>
<td>worker</td>
<td>11m</td>
<td>v1.20.0</td>
</tr>
</tbody>
</table>

**NOTE**

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

### Additional information

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

**1.2.12. Initial Operator configuration**

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

**Prerequisites**

- Your control plane has initialized.
Procedure

1. Watch the cluster components come online:

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

**Example output**

<table>
<thead>
<tr>
<th>NAME</th>
<th>VERSION</th>
<th>AVAILABLE</th>
<th>PROGRESSING</th>
<th>DEGRADED</th>
<th>SINCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3h56m</td>
</tr>
<tr>
<td>cloud-credential</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>cluster-autoscaler</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>config-operator</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>6h39m</td>
</tr>
<tr>
<td>console</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3h59m</td>
</tr>
<tr>
<td>csi-snapshot-controller</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h12m</td>
</tr>
<tr>
<td>dns</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h15m</td>
</tr>
<tr>
<td>etcd</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>image-registry</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3h59m</td>
</tr>
<tr>
<td>ingress</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h30m</td>
</tr>
<tr>
<td>insights</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>kube-apiserver</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>kube-controller-manager</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>kube-scheduler</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>kube-storage-version-migrator</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h2m</td>
</tr>
<tr>
<td>machine-api</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>machine-approver</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>6h34m</td>
</tr>
<tr>
<td>machine-config</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3h56m</td>
</tr>
<tr>
<td>marketplace</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h2m</td>
</tr>
<tr>
<td>monitoring</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>6h31m</td>
</tr>
<tr>
<td>network</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>node-tuning</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h30m</td>
</tr>
<tr>
<td>openshift-apiserver</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3h56m</td>
</tr>
<tr>
<td>openshift-controller-manager</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h36m</td>
</tr>
<tr>
<td>openshift-samples</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h30m</td>
</tr>
<tr>
<td>operator-lifecycle-manager</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>operator-lifecycle-manager-catalog</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>operator-lifecycle-manager-packageserver</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>3h59m</td>
</tr>
<tr>
<td>service-ca</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>29h</td>
</tr>
<tr>
<td>storage</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>4h30m</td>
</tr>
</tbody>
</table>

2. Configure the Operators that are not available.

1.2.12.1. Disabling the default OperatorHub sources

Operator catalogs that source content provided by Red Hat and community projects are configured for OperatorHub by default during an OpenShift Container Platform installation. In a restricted network environment, you must disable the default catalogs as a cluster administrator.

**Procedure**

- Disable the sources for the default catalogs by adding `disableAllDefaultSources: true` to the `OperatorHub` object:
TIP

Alternatively, you can use the web console to manage catalog sources. From the Administration → Cluster Settings → Global Configuration → OperatorHub page, click the Sources tab, where you can create, delete, disable, and enable individual sources.

1.2.12.2. Image registry storage configuration

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

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

Additional instructions are provided for allowing the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

1.2.12.2.1. Configuring registry storage for IBM Z

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

**Prerequisites**

- Cluster administrator permissions.
- A cluster on IBM Z.
- Persistent storage provisioned for your cluster.

**IMPORTANT**

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

- Must have 100Gi capacity.

**Procedure**

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

**NOTE**

When using shared storage, review your security settings to prevent outside access.
2. Verify that you do not have a registry pod:

   $ oc get pod -n openshift-image-registry

   **NOTE**

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

3. Check the registry configuration:

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

   **Example output**

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

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

4. Check the *clusteroperator* status:

   $ oc get clusteroperator image-registry

5. Ensure that your registry is set to managed to enable building and pushing of images.

   - Run:

     $ oc edit configs.imageregistry/cluster

     Then, change the line

     ```
     managementState: Removed
     ```

     to

     ```
     managementState: Managed
     ```

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

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

**Procedure**

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

  $ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec":
  
  "storage":{"emptyDir":{}}}'}
WARNING
Configure this option for only non-production clusters.

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

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

Wait a few minutes and run the command again.

### 1.2.13. Completing installation on user-provisioned infrastructure

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

**Prerequisites**

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

**Procedure**

1. Confirm that all the cluster components are online with the following command:

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

**Example output**

<table>
<thead>
<tr>
<th>NAME</th>
<th>VERSION</th>
<th>AVAILABLE</th>
<th>PROGRESSING</th>
<th>DEGRADED</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>cloud-credential</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>cluster-autoscaler</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>config-operator</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>console</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>csi-snapshot-controller</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>dns</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>etcd</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>image-registry</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>ingress</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>insights</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>kube-apiserver</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>kube-controller-manager</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>kube-scheduler</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>kube-storage-version-migrator</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>machine-api</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>machine-approver</td>
<td>4.6.0</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>
Alternatively, the following command notifies you when all of the clusters are available. It also retrieves and displays credentials:

```
$ ./openshift-install --dir <installation_directory> wait-for install-complete
```

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

**Example output**

```
INFO Waiting up to 30m0s for the cluster to initialize...
```

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

**IMPORTANT**

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

- It is recommended that you use Ignition config files within 12 hours after they are generated because the 24-hour certificate rotates from 16 to 22 hours after the cluster is installed. By using the Ignition config files within 12 hours, you can avoid installation failure if the certificate update runs during installation.

2. 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
```

**Example output**
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>
```

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.

3. Register your cluster on the Cluster registration page.

### 1.2.14. Telemetry access for OpenShift Container Platform

In OpenShift Container Platform 4.6, the Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, requires internet access. If your cluster is connected to the internet, Telemetry runs automatically, and your cluster is registered to OpenShift Cluster Manager.

After you confirm that your OpenShift Cluster Manager inventory is correct, either maintained automatically by Telemetry or manually by using OpenShift Cluster Manager, use subscription watch to track your OpenShift Container Platform subscriptions at the account or multi-cluster level.

**Additional resources**

- See About remote health monitoring for more information about the Telemetry service

### 1.2.15. Collecting debugging information

You can gather debugging information that might help you to troubleshoot and debug certain issues with an OpenShift Container Platform installation on IBM Z.

**Prerequisites**

- The `oc` CLI tool installed.

**Procedure**

1. Log in to the cluster:
$ oc login

2. On the node you want to gather hardware information about, start a debugging container:

   $ oc debug node/<nodename>

3. Change to the /host file system and start toolbox:

   $ chroot /host
   $ toolbox

4. Collect the dbginfo data:

   $ dbginfo.sh

5. You can then retrieve the data, for example, using scp.

Additional resources

- See How to generate SOSREPORT within OpenShift Container Platform version 4 nodes without SSH.

1.2.16. Next steps

- Customize your cluster.

- If the mirror registry that you used to install your cluster has a trusted CA, add it to the cluster by configuring additional trust stores.