OpenShift Container Platform 4.20

Installing on IBM Cloud

Installing OpenShift Container Platform on IBM Cloud Bare Metal (Classic)

Last Updated: 2026-01-22

OpenShift Container Platform 4.20 Installing on IBM Cloud

Installing OpenShift Container Platform on IBM Cloud Bare Metal (Classic)

Legal Notice

Abstract

This document describes how to install OpenShift Container Platform on IBM Cloud Bare Metal.

Table of Contents

CHAPTER 1. INSTALLATION METHODS

You can install OpenShift Container Platform on IBM Cloud® using installer-provisioned infrastructure. This process involves using an installation program to provision the underlying infrastructure for your cluster. Installing OpenShift Container Platform on IBM Cloud® using user-provisioned infrastructure is not supported at this time.

See Installation process for more information about installer-provisioned installation processes.

1.1. INSTALLING A CLUSTER ON INSTALLER-PROVISIONED INFRASTRUCTURE

You can install a cluster on IBM Cloud® infrastructure that is provisioned by the OpenShift Container Platform installation program by using one of the following methods:

- **Installing a customized cluster on IBM Cloud®**: You can install a customized cluster on IBM Cloud® infrastructure that the installation program provisions. The installation program allows for some customization to be applied at the installation stage. Many other customization options are available post-installation.
- **Installing a cluster on IBM Cloud® with network customizations**: You can customize your OpenShift Container Platform network configuration during installation, so that your cluster can coexist with your existing IP address allocations and adhere to your network requirements.
- **Installing a cluster on IBM Cloud® into an existing VPC**: You can install OpenShift Container Platform on an existing IBM Cloud®. You can use this installation method if you have constraints set by the guidelines of your company, such as limits when creating new accounts or infrastructure.
- **Installing a private cluster on an existing VPC**: You can install a private cluster on an existing Virtual Private Cloud (VPC). You can use this method to deploy OpenShift Container Platform on an internal network that is not visible to the internet.
- **Installing a cluster on IBM Cloud in a restricted network**: You can install OpenShift Container Platform on IBM Cloud on installer-provisioned infrastructure by using an internal mirror of the installation release content. You can use this method to install a cluster that does not require an active internet connection to obtain the software components.

1.2. NEXT STEPS

- Configuring an IBM Cloud® account

CHAPTER 2. CONFIGURING AN IBM CLOUD ACCOUNT

Before you can install OpenShift Container Platform, you must configure an IBM Cloud® account.

2.1. PREREQUISITES

- You have an IBM Cloud® account with a subscription. You cannot install OpenShift Container Platform on a free or trial IBM Cloud® account.

2.2. QUOTAS AND LIMITS ON IBM CLOUD

The OpenShift Container Platform cluster uses a number of IBM Cloud® components, and the default quotas and limits affect your ability to install OpenShift Container Platform clusters. If you use certain cluster configurations, deploy your cluster in certain regions, or run multiple clusters from your account, you might need to request additional resources for your IBM Cloud® account.

For a comprehensive list of the default IBM Cloud® quotas and service limits, see IBM Cloud®'s documentation for Quotas and service limits.

2.2.1. Virtual Private Cloud (VPC)

Each OpenShift Container Platform cluster creates its own VPC. The default quota of VPCs per region is 10 and will allow 10 clusters. To have more than 10 clusters in a single region, you must increase this quota.

2.2.2. Application load balancer

By default, each cluster creates three application load balancers (ALBs):

- Internal load balancer for the master API server
- External load balancer for the master API server
- Load balancer for the router

You can create additional **LoadBalancer** service objects to create additional ALBs. The default quota of VPC ALBs are 50 per region. To have more than 50 ALBs, you must increase this quota.

VPC ALBs are supported. Classic ALBs are not supported for IBM Cloud®.

2.2.3. Floating IP address

By default, the installation program distributes control plane and compute machines across all availability zones within a region to provision the cluster in a highly available configuration. In each availability zone, a public gateway is created and requires a separate floating IP address.

The default quota for a floating IP address is 20 addresses per availability zone. The default cluster configuration yields three floating IP addresses:

- Two floating IP addresses in the **us-east-1** primary zone. The IP address associated with the bootstrap node is removed after installation.
- One floating IP address in the **us-east-2** secondary zone.

- One floating IP address in the **us-east-3** secondary zone.

IBM Cloud® can support up to 19 clusters per region in an account. If you plan to have more than 19 default clusters, you must increase this quota.

2.2.4. Virtual Server Instances (VSI)

By default, a cluster creates VSIs using **bx2-4x16** profiles, which includes the following resources by default:

- 4 vCPUs
- 16 GB RAM

The following nodes are created:

- One **bx2-4x16** bootstrap machine, which is removed after the installation is complete
- Three **bx2-4x16** control plane nodes
- Three **bx2-4x16** compute nodes

For more information, see IBM Cloud®'s documentation on supported profiles.

Table 2.1. VSI component quotas and limits

VSI component	Default IBM Cloud® quota	Default cluster configuration	Maximum number of clusters
vCPU	200 vCPUs per region	28 vCPUs, or 24 vCPUs after bootstrap removal	8 per region
RAM	1600 GB per region	112 GB, or 96 GB after bootstrap removal	16 per region
Storage	18 TB per region	1050 GB, or 900 GB after bootstrap removal	19 per region

If you plan to exceed the resources stated in the table, you must increase your IBM Cloud® account quota.

2.2.5. Block Storage Volumes

For each VPC machine, a block storage device is attached for its boot volume. The default cluster configuration creates seven VPC machines, resulting in seven block storage volumes. Additional Kubernetes persistent volume claims (PVCs) of the IBM Cloud® storage class create additional block storage volumes. The default quota of VPC block storage volumes are 300 per region. To have more than 300 volumes, you must increase this quota.

2.3. CONFIGURING DNS RESOLUTION

How you configure DNS resolution depends on the type of OpenShift Container Platform cluster you are installing:

- If you are installing a public cluster, you use IBM Cloud Internet Services (CIS).
- If you are installing a private cluster, you use IBM Cloud® DNS Services (DNS Services)

2.3.1. Using IBM Cloud Internet Services for DNS resolution

The installation program uses IBM Cloud® Internet Services (CIS) to configure cluster DNS resolution and provide name lookup for a public cluster.

NOTE

This offering does not support IPv6, so dual stack or IPv6 environments are not possible.

You must create a domain zone in CIS in the same account as your cluster. You must also ensure the zone is authoritative for the domain. You can do this using a root domain or subdomain.

Prerequisites

- You have installed the IBM Cloud® CLI.
- You have an existing domain and registrar. For more information, see the IBM® documentation.

Procedure

1. Create a CIS instance to use with your cluster:
   a. Install the CIS plugin:

```
$ ibmcloud plugin install cis
```

   b. Create the CIS instance:

```
$ ibmcloud cis instance-create <instance_name> standard-next 1
```

   1 At a minimum, you require a **Standard Next** plan for CIS to manage the cluster subdomain and its DNS records.

NOTE

After you have configured your registrar or DNS provider, it can take up to 24 hours for the changes to take effect.

2. Connect an existing domain to your CIS instance:
   a. Set the context instance for CIS:

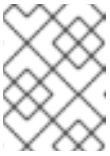
```
$ ibmcloud cis instance-set <instance_name> 1
```

   1 The instance cloud resource name.

   b. Add the domain for CIS:

```
$ ibmcloud cis domain-add <domain_name> 1
```

1 The fully qualified domain name. You can use either the root domain or subdomain value as the domain name, depending on which you plan to configure.



NOTE

A root domain uses the form **openshiftcorp.com**. A subdomain uses the form **clusters.openshiftcorp.com**.

3. Open the CIS web console, navigate to the **Overview** page, and note your CIS name servers. These name servers will be used in the next step.

4. Configure the name servers for your domains or subdomains at the domain's registrar or DNS provider. For more information, see the IBM Cloud® documentation.

2.3.2. Using IBM Cloud DNS Services for DNS resolution

The installation program uses IBM Cloud® DNS Services to configure cluster DNS resolution and provide name lookup for a private cluster.

You configure DNS resolution by creating a DNS services instance for the cluster, and then adding a DNS zone to the DNS Services instance. Ensure that the zone is authoritative for the domain. You can do this using a root domain or subdomain.

NOTE

IBM Cloud® does not support IPv6, so dual stack or IPv6 environments are not possible.

Prerequisites

- You have installed the IBM Cloud® CLI.
- You have an existing domain and registrar. For more information, see the IBM® documentation.

Procedure

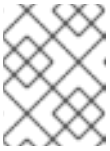
1. Create a DNS Services instance to use with your cluster:
   a. Install the DNS Services plugin by running the following command:

```
$ ibmcloud plugin install cloud-dns-services
```

   b. Create the DNS Services instance by running the following command:

```
$ ibmcloud dns instance-create <instance-name> standard-dns 1
```

1 At a minimum, you require a **Standard DNS** plan for DNS Services to manage the cluster subdomain and its DNS records.



NOTE

After you have configured your registrar or DNS provider, it can take up to 24 hours for the changes to take effect.

2. Create a DNS zone for the DNS Services instance:

   a. Set the target operating DNS Services instance by running the following command:

   ```
   $ ibmcloud dns instance-target <instance-name>
   ```

   b. Add the DNS zone to the DNS Services instance by running the following command:

   ```
   $ ibmcloud dns zone-create <zone-name> 1
   ```

   1 The fully qualified zone name. You can use either the root domain or subdomain value as the zone name, depending on which you plan to configure. A root domain uses the form **openshiftcorp.com**. A subdomain uses the form **clusters.openshiftcorp.com**.

3. Record the name of the DNS zone you have created. As part of the installation process, you must update the **install-config.yaml** file before deploying the cluster. Use the name of the DNS zone as the value for the **baseDomain** parameter.

NOTE

You do not have to manage permitted networks or configure an "A" DNS resource record. As required, the installation program configures these resources automatically.

2.4. IBM CLOUD IAM POLICIES AND API KEY

To install OpenShift Container Platform into your IBM Cloud® account, the installation program requires an IAM API key, which provides authentication and authorization to access IBM Cloud® service APIs. You can use an existing IAM API key that contains the required policies or create a new one.

For an IBM Cloud® IAM overview, see the IBM Cloud® documentation.

2.4.1. Required access policies

You must assign the required access policies to your IBM Cloud® account.

Table 2.2. Required access policies

Service type	Service	Access policy scope	Platform access	Service access
Account management	IAM Identity Service	All resources or a subset of resources [1]	Editor, Operator, Viewer, Administrator	Service ID creator

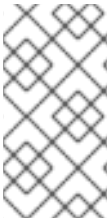
Service type	Service	Access policy scope	Platform access	Service access
Account management [2]	Identity and Access Management	All resources	Editor, Operator, Viewer, Administrator	
Account management	Resource group only	All resource groups in the account	Administrator	
IAM services	Cloud Object Storage	All resources or a subset of resources [1]	Editor, Operator, Viewer, Administrator	Reader, Writer, Manager, Content Reader, Object Reader, Object Writer
IAM services	Internet Services	All resources or a subset of resources [1]	Editor, Operator, Viewer, Administrator	Reader, Writer, Manager
IAM services	DNS Services	All resources or a subset of resources [1]	Editor, Operator, Viewer, Administrator	Reader, Writer, Manager
IAM services	VPC Infrastructure Services	All resources or a subset of resources [1]	Editor, Operator, Viewer, Administrator	Reader, Writer, Manager

1. The policy access scope should be set based on how granular you want to assign access. The scope can be set to **All resources** or **Resources based on selected attributes**.
2. Optional: This access policy is only required if you want the installation program to create a resource group. For more information about resource groups, see the IBM® documentation.

2.4.2. Access policy assignment

In IBM Cloud® IAM, access policies can be attached to different subjects:

- Access group (Recommended)
- Service ID
- User



NOTE

The recommended method is to define IAM access policies in an access group. This helps organize all the access required for OpenShift Container Platform and enables you to onboard users and service IDs to this group. You can also assign access to users and service IDs directly, if desired.

2.4.3. Creating an API key

You must create a user API key or a service ID API key for your IBM Cloud® account.

Prerequisites

- You have assigned the required access policies to your IBM Cloud® account.
- You have attached you IAM access policies to an access group, or other appropriate resource.

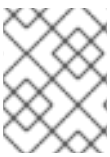
Procedure

- Create an API key, depending on how you defined your IAM access policies.
  For example, if you assigned your access policies to a user, you must create a user API key. If you assigned your access policies to a service ID, you must create a service ID API key . If your access policies are assigned to an access group, you can use either API key type. For more information on IBM Cloud® API keys, see Understanding API keys.

2.5. SUPPORTED IBM CLOUD REGIONS

You can deploy an OpenShift Container Platform cluster to the following regions:

- **au-syd** (Sydney, Australia)
- **br-sao** (Sao Paulo, Brazil)
- **ca-tor** (Toronto, Canada)
- **eu-de** (Frankfurt, Germany)
- **eu-gb** (London, United Kingdom)
- **eu-es** (Madrid, Spain)
- **jp-osa** (Osaka, Japan)
- **jp-tok** (Tokyo, Japan)
- **us-east** (Washington DC, United States)
- **us-south** (Dallas, United States)



NOTE

Deploying your cluster in the **eu-es** (Madrid, Spain) region is not supported for OpenShift Container Platform 4.14.6 and earlier versions.

2.6. NEXT STEPS

- Configuring IAM for IBM Cloud®

CHAPTER 3. CONFIGURING IAM FOR IBM CLOUD

In environments where the cloud identity and access management (IAM) APIs are not reachable, you must put the Cloud Credential Operator (CCO) into manual mode before you install the cluster.

3.1. ALTERNATIVES TO STORING ADMINISTRATOR-LEVEL SECRETS IN THE KUBE-SYSTEM PROJECT

The Cloud Credential Operator (CCO) manages cloud provider credentials as Kubernetes custom resource definitions (CRDs). You can configure the CCO to suit the security requirements of your organization by setting different values for the **credentialsMode** parameter in the **install-config.yaml** file.

Storing an administrator-level credential secret in the cluster **kube-system** project is not supported for IBM Cloud®; therefore, you must set the **credentialsMode** parameter for the CCO to **Manual** when installing OpenShift Container Platform and manage your cloud credentials manually.

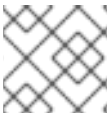
Using manual mode allows each cluster component to have only the permissions it requires, without storing an administrator-level credential in the cluster. You can also use this mode if your environment does not have connectivity to the cloud provider public IAM endpoint. However, you must manually reconcile permissions with new release images for every upgrade. You must also manually supply credentials for every component that requests them.

Additional resources

- About the Cloud Credential Operator

3.2. CONFIGURING THE CLOUD CREDENTIAL OPERATOR UTILITY

To create and manage cloud credentials from outside of the cluster when the Cloud Credential Operator (CCO) is operating in manual mode, extract and prepare the CCO utility (**ccoctl**) binary.



NOTE

The **ccoctl** utility is a Linux binary that must run in a Linux environment.

Prerequisites

- You have access to an OpenShift Container Platform account with cluster administrator access.
- You have installed the OpenShift CLI (**oc**).

Procedure

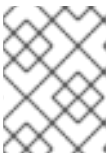
1. Set a variable for the OpenShift Container Platform release image by running the following command:

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. Obtain the CCO container image from the OpenShift Container Platform release image by running the following command:

```
$ CCO_IMAGE=$(oc adm release info --image-for='cloud-credential-operator'
```

```
$RELEASE_IMAGE -a ~/.pull-secret)
```



NOTE

Ensure that the architecture of the **$RELEASE_IMAGE** matches the architecture of the environment in which you will use the **ccoctl** tool.

3. Extract the **ccoctl** binary from the CCO container image within the OpenShift Container Platform release image by running the following command:

```
$ oc image extract $CCO_IMAGE \
  --file="/usr/bin/ccoctl.<rhel_version>" \ 1
  -a ~/.pull-secret
```

1 For **<rhel_version>**, specify the value that corresponds to the version of Red Hat Enterprise Linux (RHEL) that the host uses. If no value is specified, **ccoctl.rhel8** is used by default. The following values are valid:

- **rhel8**: Specify this value for hosts that use RHEL 8.
- **rhel9**: Specify this value for hosts that use RHEL 9.

NOTE

The **ccoctl** binary is created in the directory from where you executed the command and not in **/usr/bin/**. You must rename the directory or move the **ccoctl.<rhel_version>** binary to **ccoctl**.

4. Change the permissions to make **ccoctl** executable by running the following command:

```
$ chmod 775 ccoctl
```

Verification

- To verify that **ccoctl** is ready to use, display the help file. Use a relative file name when you run the command, for example:

```
$ ./ccoctl
```

Example output

```
OpenShift credentials provisioning tool

Usage:
  ccoctl [command]

Available Commands:
  aws         Manage credentials objects for AWS cloud
  azure       Manage credentials objects for Azure
  gcp         Manage credentials objects for Google cloud
  help        Help about any command
```

```
  ibmcloud    Manage credentials objects for {ibm-cloud-title}
  nutanix     Manage credentials objects for Nutanix

Flags:
  -h, --help   help for ccoctl

Use "ccoctl [command] --help" for more information about a command.
```

Additional resources

- Rotating API keys for IBM Cloud®

3.3. NEXT STEPS

- Installing a cluster on IBM Cloud® with customizations

3.4. ADDITIONAL RESOURCES

- Preparing to update a cluster with manually maintained credentials

CHAPTER 4. USER-MANAGED ENCRYPTION FOR IBM CLOUD

By default, provider-managed encryption is used to secure the following when you deploy an OpenShift Container Platform cluster:

- The root (boot) volume of control plane and compute machines
- Persistent volumes (data volumes) that are provisioned after the cluster is deployed

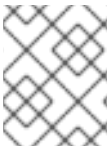
You can override the default behavior by specifying an IBM® Key Protect for IBM Cloud® (Key Protect) root key as part of the installation process.

When you bring our own root key, you modify the installation configuration file (**install-config.yaml**) to specify the Cloud Resource Name (CRN) of the root key by using the **encryptionKey** parameter.

You can specify that:

- The same root key be used be used for all cluster machines. You do so by specifying the key as part of the cluster's default machine configuration.
  When specified as part of the default machine configuration, all managed storage classes are updated with this key. As such, data volumes that are provisioned after the installation are also encrypted using this key.
- Separate root keys be used for the control plane and compute machine pools.

For more information about the **encryptionKey** parameter, see Additional IBM Cloud configuration parameters.



NOTE

Make sure you have integrated Key Protect with your IBM Cloud Block Storage service. For more information, see the Key Protect documentation.

4.1. NEXT STEPS

Install an OpenShift Container Platform cluster:

- Installing a cluster on IBM Cloud with customizations
- Installing a cluster on IBM Cloud with network customizations
- Installing a cluster on IBM Cloud into an existing VPC
- Installing a private cluster on IBM Cloud

CHAPTER 5. INSTALLING A CLUSTER ON IBM CLOUD WITH CUSTOMIZATIONS

In OpenShift Container Platform version 4.20, you can install a customized cluster on IBM Cloud® by using installer-provisioned infrastructure with customizations, including network configuration options. In each, you modify parameters in the **install-config.yaml** file before you install the cluster.

By customizing your network configuration, your cluster can coexist with existing IP address allocations in your environment and integrate with existing MTU and VXLAN configurations.

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

5.1. PREREQUISITES

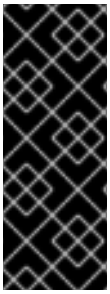
- You reviewed details about the OpenShift Container Platform installation and update processes.
- You read the documentation on selecting a cluster installation method and preparing it for users.
- You configured an IBM Cloud® account to host the cluster.
- If you use a firewall, you configured it to allow the sites that your cluster requires access to.
- You configured the **ccoctl** utility before you installed the cluster. For more information, see Configuring IAM for IBM Cloud®.

5.2. INTERNET ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.20, you require access to the internet to install your cluster.

You must have internet access to perform the following actions:

- 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 required content and use it to populate a mirror registry with the installation packages. 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.

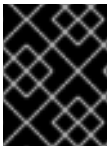
5.3. GENERATING A KEY PAIR FOR CLUSTER NODE SSH ACCESS

To enable secure, passwordless SSH access to your cluster nodes, provide an SSH public key during the

OpenShift Container Platform installation. This ensures that the installation program automatically configures the Red Hat Enterprise Linux CoreOS (RHCOS) nodes for remote authentication through the **core** user.

The SSH public key gets added to the **~/.ssh/authorized_keys** list for the **core** user on each node. After the key is passed to the Red Hat Enterprise Linux CoreOS (RHCOS) nodes through their Ignition config files, you can use the key pair to SSH in to the RHCOS nodes as the user **core**. To access the nodes through SSH, the private key identity must be managed by SSH for your local user.

If you want to SSH in to your cluster nodes to perform installation debugging or disaster recovery, you must provide the SSH public key during the installation process. The **./openshift-install gather** command also requires the SSH public key to be in place on the cluster nodes.



IMPORTANT

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

NOTE

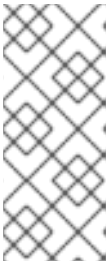
You must use a local key, not one that you configured with platform-specific approaches.

Procedure

1. If you do not have an existing SSH key pair on your local machine to use for authentication onto your cluster nodes, 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>
```

   Specifies the path and file name, such as **~/.ssh/id_ed25519**, of the new SSH key. If you have an existing key pair, ensure your public key is in the your **~/.ssh** directory.



   NOTE

   If you plan to install an OpenShift Container Platform cluster that uses the RHEL cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 Validation on only the **x86_64**, **ppc64le**, and **s390x** architectures, do not create a key that uses the **ed25519** algorithm. Instead, create a key that uses the **rsa** or **ecdsa** algorithm.

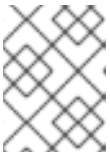
2. View the public SSH key:

```
$ cat <path>/<file_name>.pub
```

   For example, run the following to view the **~/.ssh/id_ed25519.pub** public key:

```
$ cat ~/.ssh/id_ed25519.pub
```

3. Add the SSH private key identity to the SSH agent for your local user, if it has not already been added. SSH agent management of the key is required for password-less SSH authentication onto your cluster nodes, or if you want to use the **./openshift-install gather** command.



NOTE

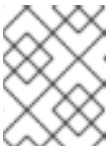
On some distributions, default SSH private key identities such as **~/.ssh/id_rsa** and **~/.ssh/id_dsa** are managed automatically.

a. If the **ssh-agent** process is not already running for your local user, start it 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.

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

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

Specifies the path and file name for your SSH private key, such as **~/.ssh/id_ed25519**

Example output

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

Next steps

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

5.4. OBTAINING THE INSTALLATION PROGRAM

Before you install OpenShift Container Platform, download the installation file on the host you are using for installation.

Prerequisites

- You have a computer that runs Linux or macOS, with 500 MB of local disk space.

Procedure

1. Go to the Cluster Type page on the Red Hat Hybrid Cloud Console. If you have a Red Hat account, log in with your credentials. If you do not, create an account.

TIP

You can also download the binaries for a specific OpenShift Container Platform release .

2. Select your infrastructure provider from the **Run it yourself** section of the page.
3. Select your host operating system and architecture from the dropdown menus under **OpenShift Installer** and click **Download Installer**.
4. Place the downloaded file in the directory where you want to 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 of the files are required to delete the cluster.
- 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.

5. 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
```

6. Download your installation pull secret from 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.

TIP

Alternatively, you can retrieve the installation program from the Red Hat Customer Portal, where you can specify a version of the installation program to download. However, you must have an active subscription to access this page.

5.5. EXPORTING THE API KEY

You must set the API key you created as a global variable; the installation program ingests the variable during startup to set the API key.

Prerequisites

- You have created either a user API key or service ID API key for your IBM Cloud® account.

Procedure

- Export your API key for your account as a global variable:

```
$ export IC_API_KEY=<api_key>
```

IMPORTANT

You must set the variable name exactly as specified; the installation program expects the variable name to be present during startup.

5.6. CREATING THE INSTALLATION CONFIGURATION FILE

You can customize the OpenShift Container Platform cluster you install on

IBM Cloud®.

Prerequisites

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

Procedure

1. Create the **install-config.yaml** file.

   a. Change to the directory that contains the installation program and run the following command:

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

   - **<installation_directory>**: For **<installation_directory>**, specify the directory name to store the files that the installation program creates.
     When specifying the directory:

   - Verify that the directory has the **execute** permission. This permission is required to run Terraform binaries under the installation directory.

   - Use an empty directory. Some installation assets, such as bootstrap X.509 certificates, have short expiration intervals, therefore 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.

   b. At the prompts, provide the configuration details for your cloud:

      i. Optional: Select an SSH key to use to 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.

      ii. Select **ibmcloud** as the platform to target.

      iii. Select the region to deploy the cluster to.

      iv. Select the base domain to deploy the cluster to. The base domain corresponds to the public DNS zone that you created for your cluster.

v. Enter a descriptive name for your cluster.

2. Modify the **install-config.yaml** file. You can find more information about the available parameters in the "Installation configuration parameters" section.

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 installation process. If you want to reuse the file, you must back it up now.

Additional resources

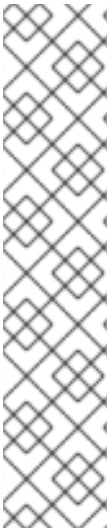
- Installation configuration parameters for IBM Cloud®

5.6.1. Minimum resource requirements for cluster installation

Each created cluster must meet minimum requirements so that the cluster runs as expected.

Table 5.1. Minimum resource requirements

Machine	Operating System	vCPU	Virtual RAM	Storage	Input/Output Per Second (IOPS)
Bootstrap	RHCOS	4	16 GB	100 GB	300
Control plane	RHCOS	4	16 GB	100 GB	300
Compute	RHCOS	2	8 GB	100 GB	300



NOTE

For OpenShift Container Platform version 4.19, RHCOS is based on RHEL version 9.6, which updates the micro-architecture requirements. The following list contains the minimum instruction set architectures (ISA) that each architecture requires:

- x86-64 architecture requires x86-64-v2 ISA
- ARM64 architecture requires ARMv8.0-A ISA
- IBM Power architecture requires Power 9 ISA
- s390x architecture requires z14 ISA

For more information, see Architectures (RHEL documentation).

If an instance type for your platform meets the minimum requirements for cluster machines, it is supported to use in OpenShift Container Platform.

Additional resources

- Optimizing storage

5.6.2. Tested instance types for IBM Cloud

The following IBM Cloud® instance types have been tested with OpenShift Container Platform.

Example 5.1. Machine series

- **bx2-8x32**
- **bx2d-4x16**
- **bx3d-4x20**
- **bx3dc-8x40**
- **cx2-8x16**
- **cx2d-4x8**
- **cx3d-8x20**
- **cx3dc-4x10**
- **gx2-8x64x1v100**
- **gx3-16x80x1l4**
- **gx3d-160x1792x8h100**
- **mx2-8x64**
- **mx2d-4x32**
- **mx3d-4x40**
- **ox2-8x64**
- **ux2d-2x56**
- **vx2d-4x56**

5.6.3. Sample customized install-config.yaml file for IBM Cloud

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.

IMPORTANT

This sample YAML file is provided for reference only. You must obtain your **install-config.yaml** file by using the installation program and then modify it.

```
apiVersion: v1
```

```
baseDomain: example.com (1)
controlPlane: (2) (3)
  hyperthreading: Enabled (4)
  name: master
  platform:
    ibmcloud: {}
  replicas: 3
compute: (5) (6)
- hyperthreading: Enabled (7)
  name: worker
  platform:
    ibmcloud: {}
  replicas: 3
metadata:
  name: test-cluster (8)
networking:
networking: (9)
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OVNKubernetes (10)
  serviceNetwork:
  - 172.30.0.0/16
platform:
  ibmcloud:
    region: us-south (11)
credentialsMode: Manual
publish: External
pullSecret: '{"auths": ...}' (12)
fips: false (13)
sshKey: ssh-ed25519 AAAA... (14)
```

(1) (8) (11) (12) Required. The installation program prompts you for this value.

(2) (5) (9) If you do not provide these parameters and values, the installation program provides the default value.

(3) (6) 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.

(4) (7) Enables or disables simultaneous multithreading, also known as Hyper-Threading. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores. You can disable it by setting the parameter value to **Disabled**. If you disable simultaneous multithreading in some cluster machines, you must disable it in all cluster machines.

IMPORTANT

If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance. Use larger machine types, such as **n1-standard-8**, for your machines if you disable simultaneous multithreading.

10 The cluster network plugin to install. The default value **OVNKubernetes** is the only supported value.

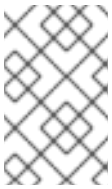
13 Enables or disables FIPS mode. By default, FIPS mode is not enabled. If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS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

IMPORTANT

To enable FIPS mode for your cluster, you must run the installation program from a Red Hat Enterprise Linux (RHEL) computer configured to operate in FIPS mode. For more information about configuring FIPS mode on RHEL, see Installing the system in FIPS mode.

When running Red Hat Enterprise Linux (RHEL) or Red Hat Enterprise Linux CoreOS (RHCOS) booted in FIPS mode, OpenShift Container Platform core components use the RHEL cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 Validation on only the x86_64, ppc64le, and s390x architectures.

14 Optional: provide the **sshKey** value that you use to access the machines in your cluster.



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.

5.6.4. Configuring the cluster-wide proxy during installation

To enable internet access in environments that deny direct connections, configure a cluster-wide proxy in the **install-config.yaml** file. This configuration ensures that the new OpenShift Container Platform cluster routes traffic through the specified HTTP or HTTPS proxy.

Prerequisites

- You have an existing **install-config.yaml** file.
- You have 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, Microsoft Azure, and Red Hat OpenStack Platform (RHOSP), the **Proxy** object **status.noProxy** field is also populated with the instance metadata endpoint (**169.254.169.254**).

Procedure

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

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port>
  httpsProxy: https://<username>:<pswd>@<ip>:<port>
  noProxy: example.com
additionalTrustBundle: |
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle>
# ...
```

where:

proxy.httpProxy

Specifies a proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**.

proxy.httpsProxy

Specifies a proxy URL to use for creating HTTPS connections outside the cluster.

proxy.noProxy

Specifies 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.

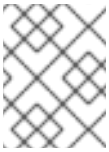
additionalTrustBundle

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.

additionalTrustBundlePolicy

Specifies the policy that determines the configuration of the **Proxy** object to reference the

user-ca-bundle config map in the **trustedCA** field. The allowed values are **Proxyonly** and **Always**. Use **Proxyonly** to reference the **user-ca-bundle** config map only when **http/https** proxy is configured. Use **Always** to always reference the **user-ca-bundle** config map. The default value is **Proxyonly**. Optional parameter.



NOTE

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

NOTE

If the installer times out, restart and then complete the deployment by using the **wait-for** command of the installer. For example:

+

```
$ ./openshift-install wait-for install-complete --log-level debug
```

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.

5.7. MANUALLY CREATING IAM

Installing the cluster requires that the Cloud Credential Operator (CCO) operate in manual mode. While the installation program configures the CCO for manual mode, you must specify the identity and access management secrets for you cloud provider.

You can use the Cloud Credential Operator (CCO) utility (**ccoctl**) to create the required IBM Cloud® resources.

Prerequisites

- You have configured the **ccoctl** binary.
- You have an existing **install-config.yaml** file.

Procedure

1. Edit the **install-config.yaml** configuration file so that the file includes the **credentialsMode** parameter set to **Manual**.

   Example install-config.yaml configuration file

```
apiVersion: v1
baseDomain: cluster1.example.com
credentialsMode: Manual
compute:
- architecture: amd64
  hyperthreading: Enabled
```

- **credentialsMode**: Set the **credentialsMode** parameter to **Manual**.

2. To generate the manifests, run the following command from the directory that includes the installation program:

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

3. From the directory that includes the installation program, set a **$RELEASE_IMAGE** variable with the release image from your installation file by running the following command:

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

4. Extract the list of **CredentialsRequest** custom resources (CRs) from the OpenShift Container Platform release image by running the following command:

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \
  --to=<path_to_directory_for_credentials_requests>
```

- **--included**: Includes only the manifests that your specific cluster configuration requires.
- **<path_to_directory_with_installation_configuration>**: Specify the location of the **install-config.yaml** file.
- **<path_to_directory_for_credentials_requests>**: Specify the path to the directory where you want to store the **CredentialsRequest** objects. If the specified directory does not exist, this command creates it.
  This command creates a YAML file for each **CredentialsRequest** object.

  Sample CredentialsRequest object

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: openshift-image-registry-ibmcos
  namespace: openshift-cloud-credential-operator
spec:
  secretRef:
    name: installer-cloud-credentials
    namespace: openshift-image-registry
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
```

```
kind: IBMCloudProviderSpec
policies:
- attributes:
  - name: serviceName
    value: cloud-object-storage
  roles:
  - crn:v1:bluemix:public:iam::::role:Viewer
  - crn:v1:bluemix:public:iam::::role:Operator
  - crn:v1:bluemix:public:iam::::role:Editor
  - crn:v1:bluemix:public:iam::::serviceRole:Reader
  - crn:v1:bluemix:public:iam::::serviceRole:Writer
- attributes:
  - name: resourceType
    value: resource-group
  roles:
  - crn:v1:bluemix:public:iam::::role:Viewer
```

5. Create the service ID for each credential request, assign the policies defined, create an API key, and generate the secret:

```
$ ccoctl ibmcloud create-service-id \
  --credentials-requests-dir=<path_to_credential_requests_directory> \
  --name=<cluster_name> \
  --output-dir=<installation_directory> \
  --resource-group-name=<resource_group_name>
```

- **<path_to_credential_requests_directory>**: Specify the directory containing the files for the **CredentialsRequest** objects.
- **<cluster_name>**: Specify the name of the OpenShift Container Platform cluster.
- **<installation_directory>**: Optional parameter. Specify the directory in which you want the **ccoctl** utility to create objects. By default, the utility creates objects in the directory in which you run the commands.
- **<resource_group_name>**: Optional parameter. Specify the name of the resource group used for scoping the access policies.

NOTE

If you enabled Technology Preview features by using the **TechPreviewNoUpgrade** feature set for your cluster, you must include the **--enable-tech-preview** parameter in the configuration for the **CredentialsRequest** object.

If you provided a wrong resource group name, the installation fails during the bootstrap phase. To find the correct resource group name, run the following command:

```
$ grep resourceGroupName <installation_directory>/manifests/cluster-infrastructure-02-config.yml
```

Verification

- Check that the appropriate secrets exist in the **manifests** directory of your cluster.

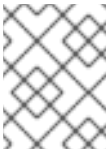
5.8. NETWORK CONFIGURATION PHASES

There are two phases prior to OpenShift Container Platform installation where you can customize the network configuration.

Phase 1

You can customize the following network-related fields in the **install-config.yaml** file before you create the manifest files:

- **networking.networkType**
- **networking.clusterNetwork**
- **networking.serviceNetwork**
- **networking.machineNetwork**
- **nodeNetworking**
  For more information, see "Installation configuration parameters".



NOTE

Set the **networking.machineNetwork** to match the Classless Inter-Domain Routing (CIDR) where the preferred subnet is located.

IMPORTANT

The CIDR range **172.17.0.0/16** is reserved by **libVirt**. You cannot use any other CIDR range that overlaps with the **172.17.0.0/16** CIDR range for networks in your cluster.

Phase 2

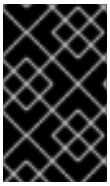
After creating the manifest files by running **openshift-install create manifests**, you can define a customized Cluster Network Operator manifest with only the fields you want to modify. You can use the manifest to specify an advanced network configuration.

During phase 2, you cannot override the values that you specified in phase 1 in the **install-config.yaml** file. However, you can customize the network plugin during phase 2.

5.9. SPECIFYING ADVANCED NETWORK CONFIGURATION

You can use advanced network configuration for your network plugin to integrate your cluster into your existing network environment.

You can specify advanced network configuration only before you install the cluster.



IMPORTANT

Customizing your network configuration by modifying the OpenShift Container Platform manifest files created by the installation program is not supported. Applying a manifest file that you create, as in the following procedure, is supported.

Prerequisites

- You have created the **install-config.yaml** file and completed any modifications to it.

Procedure

1. Change to the directory that contains the installation program and create the manifests:

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

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

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

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
```

3. Specify the advanced network configuration for your cluster in the **cluster-network-03-config.yml** file, such as in the following example:

Enable IPsec for the OVN-Kubernetes network provider

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  defaultNetwork:
    ovnKubernetesConfig:
      ipsecConfig:
        mode: Full
```

4. Optional: Back up the **manifests/cluster-network-03-config.yml** file. The installation program consumes the **manifests/** directory when you create the Ignition config files.

5. Remove the Kubernetes manifest files that define the control plane machines and compute **MachineSets**:

```
$ rm -f openshift/99_openshift-cluster-api_master-machines-*.yaml openshift/99_openshift-cluster-api_worker-machineset-*.yaml
```

Because you create and manage these resources yourself, you do not have to initialize them.

- You can preserve the **MachineSet** files to create compute machines by using the machine API, but you must update references to them to match your environment.

5.10. CLUSTER NETWORK OPERATOR CONFIGURATION

The configuration for the cluster network is specified as part of the Cluster Network Operator (CNO) configuration and stored in a custom resource (CR) object that is named **cluster**. The CR specifies the fields for the **Network** API in the **operator.openshift.io** API group.

The CNO configuration inherits the following fields during cluster installation from the **Network** API in the **Network.config.openshift.io** API group:

clusterNetwork

IP address pools from which pod IP addresses are allocated.

serviceNetwork

IP address pool for services.

defaultNetwork.type

Cluster network plugin. **OVNKubernetes** is the only supported plugin during installation.

You can specify the cluster network plugin configuration for your cluster by setting the fields for the **defaultNetwork** object in the CNO object named **cluster**.

5.10.1. Cluster Network Operator configuration object

The fields for the Cluster Network Operator (CNO) are described in the following table:

Table 5.2. Cluster Network Operator configuration object

Field	Type	Description
metadata.name	**string**	The name of the CNO object. This name is always **cluster**.

spec.clusterNetwork	**array**	A list specifying the blocks of IP addresses from which pod IP addresses are allocated and the subnet prefix length assigned to each individual node in the cluster. For example:
		`spec:`
		`  clusterNetwork:`
		`  - cidr: 10.128.0.0/19`
		`  hostPrefix: 23`
		`  - cidr: 10.128.32.0/19`
		`  hostPrefix: 23`

Field	Type	Description

spec.serviceNetwork	**array**	A block of IP addresses for services. The OVN-Kubernetes network plugin supports only a single IP address block for the service network. For example:
		`spec:`
		`  serviceNetwork:`
		`  - 172.30.0.0/14`
		You can customize this field only in the **install-config.yaml** file before you create the manifests. The value is read-only in the manifest file.

spec.defaultNetwork	**object**	Configures the network plugin for the cluster network.

spec.additionalRoutingCapabilities.providers	**array**	This setting enables a dynamic routing provider. The FRR routing capability provider is required for the route advertisement feature. The only supported value is **FRR**.
		- **FRR**: The FRR routing provider
		`spec:`
		`  additionalRoutingCapabilities:`
		`  providers:`
		`  - FRR`

IMPORTANT

For a cluster that needs to deploy objects across multiple networks, ensure that you specify the same value for the **clusterNetwork.hostPrefix** parameter for each network type that is defined in the **install-config.yaml** file. Setting a different value for each **clusterNetwork.hostPrefix** parameter can impact the OVN-Kubernetes network plugin, where the plugin cannot effectively route object traffic among different nodes.

5.10.1.1. defaultNetwork object configuration

The values for the **defaultNetwork** object are defined in the following table:

Table 5.3. defaultNetwork object

Field	Type	Description

Field	Type	Description
type	**string**	**OVNKubernetes**. The Red Hat OpenShift Networking network plugin is selected during installation. This value cannot be changed after cluster installation.  **NOTE** OpenShift Container Platform uses the OVN-Kubernetes network plugin by default.
ovnKubernetesConfig	**object**	This object is only valid for the OVN-Kubernetes network plugin.

5.10.1.1.1. Configuration for the OVN-Kubernetes network plugin

The following table describes the configuration fields for the OVN-Kubernetes network plugin:

Table 5.4. ovnKubernetesConfig object

Field	Type	Description
mtu	**integer**	The maximum transmission unit (MTU) for the Geneve (Generic Network Virtualization Encapsulation) overlay network. This is detected automatically based on the MTU of the primary network interface. You do not normally need to override the detected MTU. If the auto-detected value is not what you expect it to be, confirm that the MTU on the primary network interface on your nodes is correct. You cannot use this option to change the MTU value of the primary network interface on the nodes. If your cluster requires different MTU values for different nodes, you must set this value to **100** less than the lowest MTU value in your cluster. For example, if some nodes in your cluster have an MTU of **9001**, and some have an MTU of**1500**, you must set this value to **1400**.
genevePort	**integer**	The port to use for all Geneve packets. The default value is **6081**. This value cannot be changed after cluster installation.
ipsecConfig	**object**	Specify a configuration object for customizing the IPsec configuration.
ipv4	**object**	Specifies a configuration object for IPv4 settings.

Field	Type	Description
ipv6	**object**	Specifies a configuration object for IPv6 settings.
policyAuditConfig	**object**	Specify a configuration object for customizing network policy audit logging. If unset, the defaults audit log settings are used.
routeAdvertisements	**string**	Specifies whether to advertise cluster network routes. The default value is **Disabled**. • **Enabled**: Import routes to the cluster network and advertise cluster network routes as configured in **RouteAdvertisements** objects. • **Disabled**: Do not import routes to the cluster network or advertise cluster network routes.
gatewayConfig	**object**	Optional: Specify a configuration object for customizing how egress traffic is sent to the node gateway. Valid values are **Shared** and **Local**. The default value is **Shared**. In the default setting, the Open vSwitch (OVS) outputs traffic directly to the node IP interface. In the **Local** setting, it traverses the host network; consequently, it gets applied to the routing table of the host.  NOTE While migrating egress traffic, you can expect some disruption to workloads and service traffic until the Cluster Network Operator (CNO) successfully rolls out the changes.

Table 5.5. **ovnKubernetesConfig.ipv4** object

Field	Type	Description
internalTransitSwitchSubnet	string	If your existing network infrastructure overlaps with the **100.88.0.0/16** IPv4 subnet, you can specify a different IP address range for internal use by OVN-Kubernetes. The subnet for the distributed transit switch that enables east-west traffic. This subnet cannot overlap with any other subnets used by OVN-Kubernetes or on the host itself. It must be large enough to accommodate one IP address per node in your cluster. The default value is **100.88.0.0/16**.

Field	Type	Description
internalJoinSubnet	string	If your existing network infrastructure overlaps with the **100.64.0.0/16** IPv4 subnet, you can specify a different IP address range for internal use by OVN-Kubernetes. You must ensure that the IP address range does not overlap with any other subnet used by your OpenShift Container Platform installation. The IP address range must be larger than the maximum number of nodes that can be added to the cluster. For example, if the **clusterNetwork.cidr** value is **10.128.0.0/14** and the **clusterNetwork.hostPrefix** value is **/23**, then the maximum number of nodes is **2^(23-14)=512**. The default value is **100.64.0.0/16**.

Table 5.6. **ovnKubernetesConfig.ipv6** object

Field	Type	Description
internalTransitSwitchSubnet	string	If your existing network infrastructure overlaps with the **fd97::/64** IPv6 subnet, you can specify a different IP address range for internal use by OVN-Kubernetes. The subnet for the distributed transit switch that enables east-west traffic. This subnet cannot overlap with any other subnets used by OVN-Kubernetes or on the host itself. It must be large enough to accommodate one IP address per node in your cluster. The default value is **fd97::/64**.
internalJoinSubnet	string	If your existing network infrastructure overlaps with the **fd98::/64** IPv6 subnet, you can specify a different IP address range for internal use by OVN-Kubernetes. You must ensure that the IP address range does not overlap with any other subnet used by your OpenShift Container Platform installation. The IP address range must be larger than the maximum number of nodes that can be added to the cluster. The default value is **fd98::/64**.

Table 5.7. **policyAuditConfig** object

Field	Type	Description
rateLimit	integer	The maximum number of messages to generate every second per node. The default value is **20** messages per second.
maxFileSize	integer	The maximum size for the audit log in bytes. The default value is **50000000** or 50 MB.

Field	Type	Description
maxLogFiles	integer	The maximum number of log files that are retained.
destination	string	One of the following additional audit log targets: **libc** The libc **syslog()** function of the journald process on the host. **udp:<host>:<port>** A syslog server. Replace **<host>:<port>** with the host and port of the syslog server. **unix:<file>** A Unix Domain Socket file specified by **<file>**. **null** Do not send the audit logs to any additional target.
syslogFacility	string	The syslog facility, such as **kern**, as defined by RFC5424. The default value is **local0**.

Table 5.8. **gatewayConfig** object

Field	Type	Description
routingViaHost	**boolean**	Set this field to **true** to send egress traffic from pods to the host networking stack. For highly-specialized installations and applications that rely on manually configured routes in the kernel routing table, you might want to route egress traffic to the host networking stack. By default, egress traffic is processed in OVN to exit the cluster and is not affected by specialized routes in the kernel routing table. The default value is **false**. This field has an interaction with the Open vSwitch hardware offloading feature. If you set this field to **true**, you do not receive the performance benefits of the offloading because egress traffic is processed by the host networking stack.
ipForwarding	**object**	You can control IP forwarding for all traffic on OVN-Kubernetes managed interfaces by using the **ipForwarding** specification in the **Network** resource. Specify **Restricted** to only allow IP forwarding for Kubernetes related traffic. Specify **Global** to allow forwarding of all IP traffic. For new installations, the default is **Restricted**. For updates to OpenShift Container Platform 4.14 or later, the default is **Global**.  NOTE The default value of **Restricted** sets the IP forwarding to drop.

Field	Type	Description
ipv4	**object**	Optional: Specify an object to configure the internal OVN-Kubernetes masquerade address for host to service traffic for IPv4 addresses.
ipv6	**object**	Optional: Specify an object to configure the internal OVN-Kubernetes masquerade address for host to service traffic for IPv6 addresses.

Table 5.9. **gatewayConfig.ipv4** object

Field	Type	Description
internalMasqueradeSubnet	**string**	The masquerade IPv4 addresses that are used internally to enable host to service traffic. The host is configured with these IP addresses as well as the shared gateway bridge interface. The default value is **169.254.169.0/29**.  IMPORTANT For OpenShift Container Platform 4.17 and later versions, clusters use **169.254.0.0/17** as the default masquerade subnet. For upgraded clusters, there is no change to the default masquerade subnet.

Table 5.10. **gatewayConfig.ipv6** object

Field	Type	Description
internalMasqueradeSubnet	**string**	The masquerade IPv6 addresses that are used internally to enable host to service traffic. The host is configured with these IP addresses as well as the shared gateway bridge interface. The default value is **fd69::/125**.  IMPORTANT For OpenShift Container Platform 4.17 and later versions, clusters use **fd69::/112** as the default masquerade subnet. For upgraded clusters, there is no change to the default masquerade subnet.

Table 5.11. **ipsecConfig** object

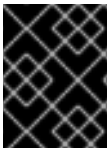
Field	Type	Description
mode	**string**	Specifies the behavior of the IPsec implementation. Must be one of the following values: • **Disabled**: IPsec is not enabled on cluster nodes. • **External**: IPsec is enabled for network traffic with external hosts. • **Full**: IPsec is enabled for pod traffic and network traffic with external hosts.

Example OVN-Kubernetes configuration with IPSec enabled

```
defaultNetwork:
  type: OVNKubernetes
  ovnKubernetesConfig:
    mtu: 1400
    genevePort: 6081
    ipsecConfig:
      mode: Full
```

5.11. DEPLOYING THE CLUSTER

You can install OpenShift Container Platform on a compatible cloud platform.



IMPORTANT

You can run the **create cluster** command of the installation program only once, during initial installation.

Prerequisites

- You have configured an account with the cloud platform that hosts your cluster.
- You have the OpenShift Container Platform installation program and the pull secret for your cluster.
- You have verified that the cloud provider account on your host has the correct permissions to deploy the cluster. An account with incorrect permissions causes the installation process to fail with an error message that displays the missing permissions.

Procedure

- In the directory that contains the installation program, initialize the cluster deployment by running the following command:

```
$ ./openshift-install create cluster --dir <installation_directory> \ 1
    --log-level=info 2
```

1. For **<installation_directory>**, specify the location of your customized **./install-config.yaml** file.
2. To view different installation details, specify **warn**, **debug**, or **error** instead of **info**.

Verification

When the cluster deployment completes successfully:

- The terminal displays directions for accessing your cluster, including a link to the web console and credentials for the **kubeadmin** user.
- Credential information also outputs to **<installation_directory>/.openshift_install.log**.

IMPORTANT

Do not delete the installation program or the files that the installation program creates. Both are required to delete the cluster.

Example output

```
...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/myuser/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.mycluster.example.com
INFO Login to the console with user: "kubeadmin", and password: "password"
INFO Time elapsed: 36m22s
```

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.

5.12. INSTALLING THE OPENSHIFT CLI ON LINUX

To manage your cluster and deploy applications from the command line, install the OpenShift CLI (**oc**) binary on Linux.

IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the Download OpenShift Container Platform page on the Red Hat Customer Portal.
2. Select the architecture from the **Product Variant** list.
3. Select the appropriate version from the **Version** list.
4. Click **Download Now** next to the **OpenShift v4.20 Linux Clients** entry and save the file.
5. Unpack the archive:

```
$ tar xvf <file>
```

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

```
$ echo $PATH
```

Verification

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

```
$ oc <command>
```

5.13. INSTALLING THE OPENSHIFT CLI ON WINDOWS

To manage your cluster and deploy applications from the command line, install OpenShift CLI (**oc**) binary on Windows.

IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the Download OpenShift Container Platform page on the Red Hat Customer Portal.
2. Select the appropriate version from the **Version** list.
3. Click **Download Now** next to the **OpenShift v4.20 Windows Client** entry and save the file.
4. Extract the archive with a ZIP program.

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

```
C:\> path
```

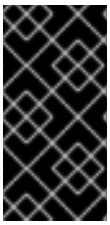
Verification

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

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

5.14. INSTALLING THE OPENSHIFT CLI ON MACOS

To manage your cluster and deploy applications from the command line, install the OpenShift CLI (**oc**) binary on macOS.



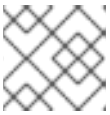
IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the Download OpenShift Container Platform page on the Red Hat Customer Portal.
2. Select the architecture from the **Product Variant** list.
3. Select the appropriate version from the **Version** list.
4. Click **Download Now** next to the **OpenShift v4.20 macOS Clients** entry and save the file.



NOTE

For macOS arm64, choose the **OpenShift v4.20 macOS arm64 Client** entry.

5. Unpack and unzip the archive.
6. Move the **oc** binary to a directory on your **PATH** variable.
   To check your **PATH** variable, open a terminal and execute the following command:

```
$ echo $PATH
```

Verification

- Verify your installation by using an **oc** command:

```
$ oc <command>
```

5.15. 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 OpenShift CLI (**oc**).

Procedure

1. Export the **kubeadmin** credentials by running the following command:

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

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

2. Verify you can run **oc** commands successfully using the exported configuration by running the following command:

```
$ oc whoami
```

Example output

```
system:admin
```

Additional resources

- Accessing the web console

5.16. TELEMETRY ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.20, 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

- About remote health monitoring

5.17. NEXT STEPS

- Customize your cluster.
- If necessary, you can Remote health reporting.

CHAPTER 6. INSTALLING A CLUSTER ON IBM CLOUD INTO AN EXISTING VPC

In OpenShift Container Platform version 4.20, you can install a cluster into an existing Virtual Private Cloud (VPC) on IBM Cloud®. The installation program provisions the rest of the required infrastructure, which you can then further customize. To customize the installation, you modify parameters in the **install-config.yaml** file before you install the cluster.

6.1. PREREQUISITES

- You reviewed details about the OpenShift Container Platform installation and update processes.
- You read the documentation on selecting a cluster installation method and preparing it for users.
- You configured an IBM Cloud® account to host the cluster.
- If you use a firewall, you configured it to allow the sites that your cluster requires access to.
- You configured the **ccoctl** utility before you installed the cluster. For more information, see Configuring IAM for IBM Cloud®.

6.2. ABOUT USING A CUSTOM VPC

In OpenShift Container Platform 4.20, you can deploy a cluster into the subnets of an existing IBM® Virtual Private Cloud (VPC). Deploying OpenShift Container Platform into an existing VPC can help you avoid limit constraints in new accounts or more easily abide by the operational constraints that your company's guidelines set. If you cannot obtain the infrastructure creation permissions that are required to create the VPC yourself, use this installation option.

Because the installation program cannot know what other components are in your existing subnets, it cannot choose subnet CIDRs and so forth. You must configure networking for the subnets to which you will install the cluster.

6.2.1. Requirements for using your VPC

You must correctly configure the existing VPC and its subnets before you install the cluster. The installation program does not create the following components:

- NAT gateways
- Subnets
- Route tables
- VPC network

The installation program cannot:

- Subdivide network ranges for the cluster to use
- Set route tables for the subnets

- Set VPC options like DHCP

NOTE

The installation program requires that you use the cloud-provided DNS server. Using a custom DNS server is not supported and causes the installation to fail.

6.2.2. VPC validation

The VPC and all of the subnets must be in an existing resource group. The cluster is deployed to the existing VPC.

As part of the installation, specify the following in the **install-config.yaml** file:

- The name of the existing resource group that contains the VPC and subnets (**networkResourceGroupName**)
- The name of the existing VPC (**vpcName**)
- The subnets that were created for control plane machines and compute machines (**controlPlaneSubnets** and **computeSubnets**)

NOTE

Additional installer-provisioned cluster resources are deployed to a separate resource group (**resourceGroupName**). You can specify this resource group before installing the cluster. If undefined, a new resource group is created for the cluster.

To ensure that the subnets that you provide are suitable, the installation program confirms the following:

- All of the subnets that you specify exist.
- For each availability zone in the region, you specify:
  - One subnet for control plane machines.
  - One subnet for compute machines.
- The machine CIDR that you specified contains the subnets for the compute machines and control plane machines.

NOTE

Subnet IDs are not supported.

6.2.3. Isolation between clusters

If you deploy OpenShift Container Platform to an existing network, the isolation of cluster services is reduced in the following ways:

- You can install multiple OpenShift Container Platform clusters in the same VPC.
- ICMP ingress is allowed to the entire network.
- TCP port 22 ingress (SSH) is allowed to the entire network.

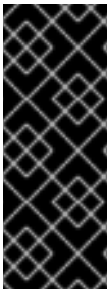
- Control plane TCP 6443 ingress (Kubernetes API) is allowed to the entire network.
- Control plane TCP 22623 ingress (MCS) is allowed to the entire network.

6.3. INTERNET ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.20, you require access to the internet to install your cluster.

You must have internet access to perform the following actions:

- 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 required content and use it to populate a mirror registry with the installation packages. 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.

6.4. GENERATING A KEY PAIR FOR CLUSTER NODE SSH ACCESS

To enable secure, passwordless SSH access to your cluster nodes, provide an SSH public key during the OpenShift Container Platform installation. This ensures that the installation program automatically configures the Red Hat Enterprise Linux CoreOS (RHCOS) nodes for remote authentication through the **core** user.

The SSH public key gets added to the **~/.ssh/authorized_keys** list for the **core** user on each node. After the key is passed to the Red Hat Enterprise Linux CoreOS (RHCOS) nodes through their Ignition config files, you can use the key pair to SSH in to the RHCOS nodes as the user **core**. To access the nodes through SSH, the private key identity must be managed by SSH for your local user.

If you want to SSH in to your cluster nodes to perform installation debugging or disaster recovery, you must provide the SSH public key during the installation process. The **./openshift-install gather** command also requires the SSH public key to be in place on the cluster nodes.

IMPORTANT

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

NOTE

You must use a local key, not one that you configured with platform-specific approaches.

Procedure

1. If you do not have an existing SSH key pair on your local machine to use for authentication onto your cluster nodes, 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>
```

Specifies the path and file name, such as **~/.ssh/id_ed25519**, of the new SSH key. If you have an existing key pair, ensure your public key is in the your **~/.ssh** directory.

NOTE

If you plan to install an OpenShift Container Platform cluster that uses the RHEL cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 Validation on only the **x86_64**, **ppc64le**, and **s390x** architectures, do not create a key that uses the **ed25519** algorithm. Instead, create a key that uses the **rsa** or **ecdsa** algorithm.

2. View the public SSH key:

```
$ cat <path>/<file_name>.pub
```

For example, run the following to view the **~/.ssh/id_ed25519.pub** public key:

```
$ cat ~/.ssh/id_ed25519.pub
```

3. Add the SSH private key identity to the SSH agent for your local user, if it has not already been added. SSH agent management of the key is required for password-less SSH authentication onto your cluster nodes, or if you want to use the **./openshift-install gather** command.

NOTE

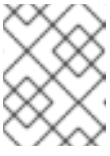
On some distributions, default SSH private key identities such as **~/.ssh/id_rsa** and **~/.ssh/id_dsa** are managed automatically.

a. If the **ssh-agent** process is not already running for your local user, start it 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.

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

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

Specifies the path and file name for your SSH private key, such as **~/.ssh/id_ed25519**

Example output

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

Next steps

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

6.5. OBTAINING THE INSTALLATION PROGRAM

Before you install OpenShift Container Platform, download the installation file on the host you are using for installation.

Prerequisites

- You have a computer that runs Linux or macOS, with 500 MB of local disk space.

Procedure

1. Go to the Cluster Type page on the Red Hat Hybrid Cloud Console. If you have a Red Hat account, log in with your credentials. If you do not, create an account.

   TIP

   You can also download the binaries for a specific OpenShift Container Platform release .

2. Select your infrastructure provider from the **Run it yourself** section of the page.
3. Select your host operating system and architecture from the dropdown menus under **OpenShift Installer** and click **Download Installer**.
4. Place the downloaded file in the directory where you want to 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 of the files are required to delete the cluster.
   - 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.

5. 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
```

6. Download your installation pull secret from 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.

TIP

Alternatively, you can retrieve the installation program from the Red Hat Customer Portal, where you can specify a version of the installation program to download. However, you must have an active subscription to access this page.

6.6. EXPORTING THE API KEY

You must set the API key you created as a global variable; the installation program ingests the variable during startup to set the API key.

Prerequisites

- You have created either a user API key or service ID API key for your IBM Cloud® account.

Procedure

- Export your API key for your account as a global variable:

```
$ export IC_API_KEY=<api_key>
```

IMPORTANT

You must set the variable name exactly as specified; the installation program expects the variable name to be present during startup.

6.7. CREATING THE INSTALLATION CONFIGURATION FILE

You can customize the OpenShift Container Platform cluster you install on

IBM Cloud®.

Prerequisites

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

Procedure

1. Create the **install-config.yaml** file.

   a. Change to the directory that contains the installation program and run the following command:

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

- **<installation_directory>**: For **<installation_directory>**, specify the directory name to store the files that the installation program creates.
  When specifying the directory:

- Verify that the directory has the **execute** permission. This permission is required to run Terraform binaries under the installation directory.

- Use an empty directory. Some installation assets, such as bootstrap X.509 certificates, have short expiration intervals, therefore 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.

b. At the prompts, provide the configuration details for your cloud:

i. Optional: Select an SSH key to use to 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.

ii. Select **ibmcloud** as the platform to target.

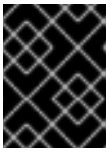
iii. Select the region to deploy the cluster to.

iv. Select the base domain to deploy the cluster to. The base domain corresponds to the public DNS zone that you created for your cluster.

v. Enter a descriptive name for your cluster.

2. Modify the **install-config.yaml** file. You can find more information about the available parameters in the "Installation configuration parameters" section.

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 installation process. If you want to reuse the file, you must back it up now.

Additional resources

- Installation configuration parameters for IBM Cloud®

6.7.1. Minimum resource requirements for cluster installation

Each created cluster must meet minimum requirements so that the cluster runs as expected.

Table 6.1. Minimum resource requirements

Machine	Operating System	vCPU	Virtual RAM	Storage	Input/Output Per Second (IOPS)
Bootstrap	RHCOS	4	16 GB	100 GB	300
Control plane	RHCOS	4	16 GB	100 GB	300
Compute	RHCOS	2	8 GB	100 GB	300

NOTE

For OpenShift Container Platform version 4.19, RHCOS is based on RHEL version 9.6, which updates the micro-architecture requirements. The following list contains the minimum instruction set architectures (ISA) that each architecture requires:

- x86-64 architecture requires x86-64-v2 ISA
- ARM64 architecture requires ARMv8.0-A ISA
- IBM Power architecture requires Power 9 ISA
- s390x architecture requires z14 ISA

For more information, see Architectures (RHEL documentation).

If an instance type for your platform meets the minimum requirements for cluster machines, it is supported to use in OpenShift Container Platform.

6.7.2. Tested instance types for IBM Cloud

The following IBM Cloud® instance types have been tested with OpenShift Container Platform.

Example 6.1. Machine series

- **bx2-8x32**
- **bx2d-4x16**
- **bx3d-4x20**
- **bx3dc-8x40**
- **cx2-8x16**
- **cx2d-4x8**
- **cx3d-8x20**
- **cx3dc-4x10**
- **gx2-8x64x1v100**

- **gx3-16x80x1l4**
- **gx3d-160x1792x8h100**
- **mx2-8x64**
- **mx2d-4x32**
- **mx3d-4x40**
- **ox2-8x64**
- **ux2d-2x56**
- **vx2d-4x56**

Additional resources

- Optimizing storage

6.7.3. Sample customized install-config.yaml file for IBM Cloud

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.

IMPORTANT

This sample YAML file is provided for reference only. You must obtain your **install-config.yaml** file by using the installation program and then modify it.

```
apiVersion: v1
baseDomain: example.com (1)
controlPlane: (2) (3)
  hyperthreading: Enabled (4)
  name: master
  platform:
    ibmcloud: {}
  replicas: 3
compute: (5) (6)
- hyperthreading: Enabled (7)
  name: worker
  platform:
    ibmcloud: {}
  replicas: 3
metadata:
  name: test-cluster (8)
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14 (9)
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
```

```
  networkType: OVNKubernetes 10
  serviceNetwork:
  - 172.30.0.0/16
platform:
  ibmcloud:
    region: eu-gb 11
    resourceGroupName: eu-gb-example-cluster-rg 12
    networkResourceGroupName: eu-gb-example-existing-network-rg 13
    vpcName: eu-gb-example-network-1 14
    controlPlaneSubnets: 15
      - eu-gb-example-network-1-cp-eu-gb-1
      - eu-gb-example-network-1-cp-eu-gb-2
      - eu-gb-example-network-1-cp-eu-gb-3
    computeSubnets: 16
      - eu-gb-example-network-1-compute-eu-gb-1
      - eu-gb-example-network-1-compute-eu-gb-2
      - eu-gb-example-network-1-compute-eu-gb-3
credentialsMode: Manual
publish: External
pullSecret: '{"auths": ...}' 17
fips: false 18
sshKey: ssh-ed25519 AAAA... 19
```

1 8 11 17 Required. The installation program prompts you for this value.

2 5 If you do not provide these parameters and values, the installation program provides the default value.

3 6 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.

4 7 Enables or disables simultaneous multithreading, also known as Hyper-Threading. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores. You can disable it by setting the parameter value to **Disabled**. If you disable simultaneous multithreading in some cluster machines, you must disable it in all cluster machines.

IMPORTANT

If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance. Use larger machine types, such as **n1-standard-8**, for your machines if you disable simultaneous multithreading.

9 The machine CIDR must contain the subnets for the compute machines and control plane machines.

10 The cluster network plugin to install. The default value **OVNKubernetes** is the only supported value.

12 The name of an existing resource group. All installer-provisioned cluster resources are deployed to this resource group. If undefined, a new resource group is created for the cluster.

13 Specify the name of the resource group that contains the existing virtual private cloud (VPC). The existing VPC and subnets should be in this resource group. The cluster will be installed to this VPC.

14 Specify the name of an existing VPC.

15 Specify the name of the existing subnets to which to deploy the control plane machines. The subnets must belong to the VPC that you specified. Specify a subnet for each availability zone in the region.

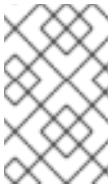
16 Specify the name of the existing subnets to which to deploy the compute machines. The subnets must belong to the VPC that you specified. Specify a subnet for each availability zone in the region.

18 Enables or disables 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

When running Red Hat Enterprise Linux (RHEL) or Red Hat Enterprise Linux CoreOS (RHCOS) booted in FIPS mode, OpenShift Container Platform core components use the RHEL cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 Validation on only the x86_64, ppc64le, and s390x architectures.

19 Optional: provide the **sshKey** value that you use to access the machines in your cluster.



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.

6.7.4. Configuring the cluster-wide proxy during installation

To enable internet access in environments that deny direct connections, configure a cluster-wide proxy in the **install-config.yaml** file. This configuration ensures that the new OpenShift Container Platform cluster routes traffic through the specified HTTP or HTTPS proxy.

Prerequisites

- You have an existing **install-config.yaml** file.
- You have 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, Microsoft Azure, and Red Hat OpenStack Platform (RHOSP), the **Proxy** object **status.noProxy** field is also populated with the instance metadata endpoint (**169.254.169.254**).

Procedure

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

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port>
  httpsProxy: https://<username>:<pswd>@<ip>:<port>
  noProxy: example.com
additionalTrustBundle: |
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle>
# ...
```

where:

proxy.httpProxy

Specifies a proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**.

proxy.httpsProxy

Specifies a proxy URL to use for creating HTTPS connections outside the cluster.

proxy.noProxy

Specifies 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.

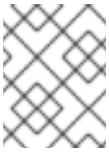
additionalTrustBundle

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.

additionalTrustBundlePolicy

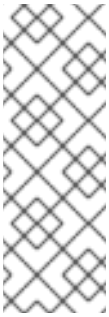
Specifies the policy that determines the configuration of the **Proxy** object to reference the

user-ca-bundle config map in the **trustedCA** field. The allowed values are **Proxyonly** and **Always**. Use **Proxyonly** to reference the **user-ca-bundle** config map only when **http/https** proxy is configured. Use **Always** to always reference the **user-ca-bundle** config map. The default value is **Proxyonly**. Optional parameter.



NOTE

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



NOTE

If the installer times out, restart and then complete the deployment by using the **wait-for** command of the installer. For example:

+

```
$ ./openshift-install wait-for install-complete --log-level debug
```

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.

6.8. MANUALLY CREATING IAM

Installing the cluster requires that the Cloud Credential Operator (CCO) operate in manual mode. While the installation program configures the CCO for manual mode, you must specify the identity and access management secrets for you cloud provider.

You can use the Cloud Credential Operator (CCO) utility (**ccoctl**) to create the required IBM Cloud® resources.

Prerequisites

- You have configured the **ccoctl** binary.
- You have an existing **install-config.yaml** file.

Procedure

1. Edit the **install-config.yaml** configuration file so that the file includes the **credentialsMode** parameter set to **Manual**.

   Example install-config.yaml configuration file

```
apiVersion: v1
baseDomain: cluster1.example.com
credentialsMode: Manual
compute:
- architecture: amd64
  hyperthreading: Enabled
```

- **credentialsMode**: Set the **credentialsMode** parameter to **Manual**.

2. To generate the manifests, run the following command from the directory that includes the installation program:

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

3. From the directory that includes the installation program, set a **$RELEASE_IMAGE** variable with the release image from your installation file by running the following command:

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

4. Extract the list of **CredentialsRequest** custom resources (CRs) from the OpenShift Container Platform release image by running the following command:

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \
  --to=<path_to_directory_for_credentials_requests>
```

- **--included**: Includes only the manifests that your specific cluster configuration requires.
- **<path_to_directory_with_installation_configuration>**: Specify the location of the **install-config.yaml** file.
- **<path_to_directory_for_credentials_requests>**: Specify the path to the directory where you want to store the **CredentialsRequest** objects. If the specified directory does not exist, this command creates it.
  This command creates a YAML file for each **CredentialsRequest** object.

  Sample CredentialsRequest object

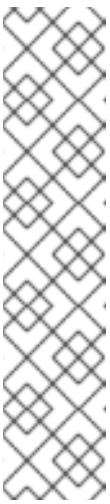
```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: openshift-image-registry-ibmcos
  namespace: openshift-cloud-credential-operator
spec:
  secretRef:
    name: installer-cloud-credentials
    namespace: openshift-image-registry
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
```

```
kind: IBMCloudProviderSpec
policies:
- attributes:
  - name: serviceName
    value: cloud-object-storage
  roles:
  - crn:v1:bluemix:public:iam::::role:Viewer
  - crn:v1:bluemix:public:iam::::role:Operator
  - crn:v1:bluemix:public:iam::::role:Editor
  - crn:v1:bluemix:public:iam::::serviceRole:Reader
  - crn:v1:bluemix:public:iam::::serviceRole:Writer
- attributes:
  - name: resourceType
    value: resource-group
  roles:
  - crn:v1:bluemix:public:iam::::role:Viewer
```

5. Create the service ID for each credential request, assign the policies defined, create an API key, and generate the secret:

```
$ ccoctl ibmcloud create-service-id \
  --credentials-requests-dir=<path_to_credential_requests_directory> \
  --name=<cluster_name> \
  --output-dir=<installation_directory> \
  --resource-group-name=<resource_group_name>
```

- **<path_to_credential_requests_directory>**: Specify the directory containing the files for the **CredentialsRequest** objects.
- **<cluster_name>**: Specify the name of the OpenShift Container Platform cluster.
- **<installation_directory>**: Optional parameter. Specify the directory in which you want the **ccoctl** utility to create objects. By default, the utility creates objects in the directory in which you run the commands.
- **<resource_group_name>**: Optional parameter. Specify the name of the resource group used for scoping the access policies.



NOTE

If you enabled Technology Preview features by using the **TechPreviewNoUpgrade** feature set for your cluster, you must include the **--enable-tech-preview** parameter in the configuration for the **CredentialsRequest** object.

If you provided a wrong resource group name, the installation fails during the bootstrap phase. To find the correct resource group name, run the following command:

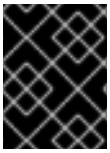
```
$ grep resourceGroupName <installation_directory>/manifests/cluster-infrastructure-02-config.yml
```

Verification

- Check that the appropriate secrets exist in the **manifests** directory of your cluster.

6.9. DEPLOYING THE CLUSTER

You can install OpenShift Container Platform on a compatible cloud platform.



IMPORTANT

You can run the **create cluster** command of the installation program only once, during initial installation.

Prerequisites

- You have configured an account with the cloud platform that hosts your cluster.
- You have the OpenShift Container Platform installation program and the pull secret for your cluster.
- You have verified that the cloud provider account on your host has the correct permissions to deploy the cluster. An account with incorrect permissions causes the installation process to fail with an error message that displays the missing permissions.

Procedure

- In the directory that contains the installation program, initialize the cluster deployment by running the following command:

```
$ ./openshift-install create cluster --dir <installation_directory> \ 1
    --log-level=info 2
```

1 For **<installation_directory>**, specify the location of your customized **./install-config.yaml** file.

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

Verification

When the cluster deployment completes successfully:

- The terminal displays directions for accessing your cluster, including a link to the web console and credentials for the **kubeadmin** user.
- Credential information also outputs to **<installation_directory>/.openshift_install.log**.

IMPORTANT

Do not delete the installation program or the files that the installation program creates. Both are required to delete the cluster.

Example output

```
...
INFO Install complete!
```

```
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/myuser/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.mycluster.example.com
INFO Login to the console with user: "kubeadmin", and password: "password"
INFO Time elapsed: 36m22s
```

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.

6.10. INSTALLING THE OPENSHIFT CLI ON LINUX

To manage your cluster and deploy applications from the command line, install the OpenShift CLI (**oc**) binary on Linux.

IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the Download OpenShift Container Platform page on the Red Hat Customer Portal.
2. Select the architecture from the **Product Variant** list.
3. Select the appropriate version from the **Version** list.
4. Click **Download Now** next to the **OpenShift v4.20 Linux Clients** entry and save the file.
5. Unpack the archive:

   ```
   $ tar xvf <file>
   ```

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

   ```
   $ echo $PATH
   ```

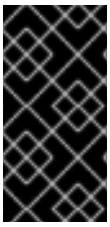
Verification

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

```
$ oc <command>
```

6.11. INSTALLING THE OPENSHIFT CLI ON WINDOWS

To manage your cluster and deploy applications from the command line, install OpenShift CLI (**oc**) binary on Windows.



IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the Download OpenShift Container Platform page on the Red Hat Customer Portal.
2. Select the appropriate version from the **Version** list.
3. Click **Download Now** next to the **OpenShift v4.20 Windows Client** entry and save the file.
4. Extract the archive with a ZIP program.
5. Move the **oc** binary to a directory that is on your **PATH** variable.
   To check your **PATH** variable, open the command prompt and execute the following command:

```
C:\> path
```

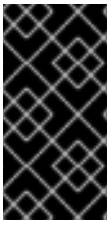
Verification

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

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

6.12. INSTALLING THE OPENSHIFT CLI ON MACOS

To manage your cluster and deploy applications from the command line, install the OpenShift CLI (**oc**) binary on macOS.



IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the Download OpenShift Container Platform page on the Red Hat Customer Portal.
2. Select the architecture from the **Product Variant** list.
3. Select the appropriate version from the **Version** list.
4. Click **Download Now** next to the **OpenShift v4.20 macOS Clients** entry and save the file.

NOTE

For macOS arm64, choose the **OpenShift v4.20 macOS arm64 Client** entry.

5. Unpack and unzip the archive.
6. Move the **oc** binary to a directory on your **PATH** variable.
   To check your **PATH** variable, open a terminal and execute the following command:

```
$ echo $PATH
```

Verification

- Verify your installation by using an **oc** command:

```
$ oc <command>
```

6.13. 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 OpenShift CLI (**oc**).

Procedure

1. Export the **kubeadmin** credentials by running the following command:

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

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

2. Verify you can run **oc** commands successfully using the exported configuration by running the following command:

```
$ oc whoami
```

Example output

```
system:admin
```

Additional resources

- Accessing the web console

6.14. TELEMETRY ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.20, 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

- About remote health monitoring

6.15. NEXT STEPS

- Customize your cluster.
- Optional: Remote health reporting.

CHAPTER 7. INSTALLING A PRIVATE CLUSTER ON IBM CLOUD

In OpenShift Container Platform version 4.20, you can install a private cluster into an existing VPC. The installation program provisions the rest of the required infrastructure, which you can further customize. To customize the installation, you modify parameters in the **install-config.yaml** file before you install the cluster.

7.1. PREREQUISITES

- You reviewed details about the OpenShift Container Platform installation and update processes.
- You read the documentation on selecting a cluster installation method and preparing it for users.
- You configured an IBM Cloud® account to host the cluster.
- If you use a firewall, you configured it to allow the sites that your cluster requires access to.
- You configured the **ccoctl** utility before you installed the cluster. For more information, see Configuring IAM for IBM Cloud®.

7.2. PRIVATE CLUSTERS

You can deploy a private OpenShift Container Platform cluster that does not expose external endpoints. Private clusters are accessible from only an internal network and are not visible to the internet.

By default, OpenShift Container Platform is provisioned to use publicly-accessible DNS and endpoints. A private cluster sets the DNS, Ingress Controller, and API server to private when you deploy your cluster. This means that the cluster resources are only accessible from your internal network and are not visible to the internet.

IMPORTANT

If the cluster has any public subnets, load balancer services created by administrators might be publicly accessible. To ensure cluster security, verify that these services are explicitly annotated as private.

To deploy a private cluster, you must:

- Use existing networking that meets your requirements. Your cluster resources might be shared between other clusters on the network.
- Create a DNS zone using IBM Cloud® DNS Services and specify it as the base domain of the cluster. For more information, see "Using IBM Cloud® DNS Services to configure DNS resolution".
- Deploy from a machine that has access to:
  - The API services for the cloud to which you provision.
  - The hosts on the network that you provision.

- The internet to obtain installation media.

7.3. PRIVATE CLUSTERS IN IBM CLOUD

To create a private cluster on IBM Cloud®, you must provide an existing private VPC and subnets to host the cluster. The installation program must also be able to resolve the DNS records that the cluster requires. The installation program configures the Ingress Operator and API server for only internal traffic.

The cluster still requires access to internet to access the IBM Cloud® APIs.

The following items are not required or created when you install a private cluster:

- Public subnets
- Public network load balancers, which support public ingress
- A public DNS zone that matches the **baseDomain** for the cluster

The installation program does use the **baseDomain** that you specify to create a private DNS zone and the required records for the cluster. The cluster is configured so that the Operators do not create public records for the cluster and all cluster machines are placed in the private subnets that you specify.

7.3.1. Limitations

Private clusters on IBM Cloud® are subject only to the limitations associated with the existing VPC that was used for cluster deployment.

7.4. ABOUT USING A CUSTOM VPC

In OpenShift Container Platform 4.20, you can deploy a cluster into the subnets of an existing IBM® Virtual Private Cloud (VPC). Deploying OpenShift Container Platform into an existing VPC can help you avoid limit constraints in new accounts or more easily abide by the operational constraints that your company's guidelines set. If you cannot obtain the infrastructure creation permissions that are required to create the VPC yourself, use this installation option.

Because the installation program cannot know what other components are in your existing subnets, it cannot choose subnet CIDRs and so forth. You must configure networking for the subnets to which you will install the cluster.

7.4.1. Requirements for using your VPC

You must correctly configure the existing VPC and its subnets before you install the cluster. The installation program does not create the following components:

- NAT gateways
- Subnets
- Route tables
- VPC network

The installation program cannot:

- Subdivide network ranges for the cluster to use
- Set route tables for the subnets
- Set VPC options like DHCP

NOTE

The installation program requires that you use the cloud-provided DNS server. Using a custom DNS server is not supported and causes the installation to fail.

7.4.2. VPC validation

The VPC and all of the subnets must be in an existing resource group. The cluster is deployed to the existing VPC.

As part of the installation, specify the following in the **install-config.yaml** file:

- The name of the existing resource group that contains the VPC and subnets (**networkResourceGroupName**)
- The name of the existing VPC (**vpcName**)
- The subnets that were created for control plane machines and compute machines (**controlPlaneSubnets** and **computeSubnets**)

NOTE

Additional installer-provisioned cluster resources are deployed to a separate resource group (**resourceGroupName**). You can specify this resource group before installing the cluster. If undefined, a new resource group is created for the cluster.

To ensure that the subnets that you provide are suitable, the installation program confirms the following:

- All of the subnets that you specify exist.
- For each availability zone in the region, you specify:
  - One subnet for control plane machines.
  - One subnet for compute machines.
- The machine CIDR that you specified contains the subnets for the compute machines and control plane machines.

NOTE

Subnet IDs are not supported.

7.4.3. Isolation between clusters

If you deploy OpenShift Container Platform to an existing network, the isolation of cluster services is reduced in the following ways:

- You can install multiple OpenShift Container Platform clusters in the same VPC.

- ICMP ingress is allowed to the entire network.
- TCP port 22 ingress (SSH) is allowed to the entire network.
- Control plane TCP 6443 ingress (Kubernetes API) is allowed to the entire network.
- Control plane TCP 22623 ingress (MCS) is allowed to the entire network.

7.5. INTERNET ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.20, you require access to the internet to install your cluster.

You must have internet access to perform the following actions:

- 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 required content and use it to populate a mirror registry with the installation packages. 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.

7.6. GENERATING A KEY PAIR FOR CLUSTER NODE SSH ACCESS

To enable secure, passwordless SSH access to your cluster nodes, provide an SSH public key during the OpenShift Container Platform installation. This ensures that the installation program automatically configures the Red Hat Enterprise Linux CoreOS (RHCOS) nodes for remote authentication through the **core** user.

The SSH public key gets added to the **~/.ssh/authorized_keys** list for the **core** user on each node. After the key is passed to the Red Hat Enterprise Linux CoreOS (RHCOS) nodes through their Ignition config files, you can use the key pair to SSH in to the RHCOS nodes as the user **core**. To access the nodes through SSH, the private key identity must be managed by SSH for your local user.

If you want to SSH in to your cluster nodes to perform installation debugging or disaster recovery, you must provide the SSH public key during the installation process. The **./openshift-install gather** command also requires the SSH public key to be in place on the cluster nodes.

IMPORTANT

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

NOTE

You must use a local key, not one that you configured with platform-specific approaches.

Procedure

1. If you do not have an existing SSH key pair on your local machine to use for authentication onto your cluster nodes, 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>
```

Specifies the path and file name, such as **~/.ssh/id_ed25519**, of the new SSH key. If you have an existing key pair, ensure your public key is in the your **~/.ssh** directory.

NOTE

If you plan to install an OpenShift Container Platform cluster that uses the RHEL cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 Validation on only the **x86_64**, **ppc64le**, and **s390x** architectures, do not create a key that uses the **ed25519** algorithm. Instead, create a key that uses the **rsa** or **ecdsa** algorithm.

2. View the public SSH key:

```
$ cat <path>/<file_name>.pub
```

For example, run the following to view the **~/.ssh/id_ed25519.pub** public key:

```
$ cat ~/.ssh/id_ed25519.pub
```

3. Add the SSH private key identity to the SSH agent for your local user, if it has not already been added. SSH agent management of the key is required for password-less SSH authentication onto your cluster nodes, or if you want to use the **./openshift-install gather** command.

NOTE

On some distributions, default SSH private key identities such as **~/.ssh/id_rsa** and **~/.ssh/id_dsa** are managed automatically.

a. If the **ssh-agent** process is not already running for your local user, start it 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.

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

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

Specifies the path and file name for your SSH private key, such as **~/.ssh/id_ed25519**

Example output

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

Next steps

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

7.7. OBTAINING THE INSTALLATION PROGRAM

Before you install OpenShift Container Platform, download the installation file on a bastion host on your cloud network or a machine that has access to the to the network through a VPN. This ensures that installation assets exist for deployment in your environment.

For more information about private cluster installation requirements, see "Private clusters".

Prerequisites

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

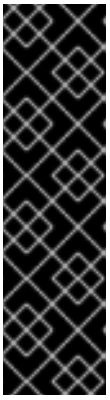
Procedure

1. Go to the Cluster Type page on the Red Hat Hybrid Cloud Console. If you have a Red Hat account, log in with your credentials. If you do not, create an account.

   TIP

   You can also download the binaries for a specific OpenShift Container Platform release .

2. Select your infrastructure provider from the **Run it yourself** section of the page.
3. Select your host operating system and architecture from the dropdown menus under **OpenShift Installer** and click **Download Installer**.
4. Place the downloaded file in the directory where you want to 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 of the files are required to delete the cluster.
- 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.

5. 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
```

6. Download your installation pull secret from 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.

TIP

Alternatively, you can retrieve the installation program from the Red Hat Customer Portal, where you can specify a version of the installation program to download. However, you must have an active subscription to access this page.

7.8. EXPORTING THE API KEY

You must set the API key you created as a global variable; the installation program ingests the variable during startup to set the API key.

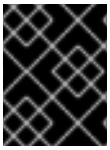
Prerequisites

- You have created either a user API key or service ID API key for your IBM Cloud® account.

Procedure

- Export your API key for your account as a global variable:

```
$ export IC_API_KEY=<api_key>
```



IMPORTANT

You must set the variable name exactly as specified; the installation program expects the variable name to be present during startup.

7.9. MANUALLY CREATING THE INSTALLATION CONFIGURATION FILE

To customise your OpenShift Container Platform deployment and meet specific network requirements, manually create the installation configuration file. This ensures that the installation program uses your tailored settings rather than default values during the setup process.

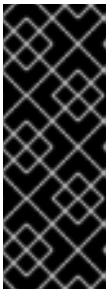
Prerequisites

- You have an SSH public key on your local machine for use with the installation program. You can use the key for SSH authentication onto your cluster nodes for debugging and disaster recovery.
- You have obtained the OpenShift Container Platform installation program and the pull secret for your cluster.

Procedure

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

```
$ mkdir <installation_directory>
```



IMPORTANT

You must create a directory. Some installation assets, such as 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 provided sample **install-config.yaml** file template and save the file 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 many clusters.

IMPORTANT

Back up the **install-config.yaml** file now, because the installation process consumes the file in the next step.

Additional resources

- Installation configuration parameters for IBM Cloud®

7.9.1. Minimum resource requirements for cluster installation

Each created cluster must meet minimum requirements so that the cluster runs as expected.

Table 7.1. Minimum resource requirements

Machine	Operating System	vCPU	Virtual RAM	Storage	Input/Output Per Second (IOPS)
Bootstrap	RHCOS	4	16 GB	100 GB	300
Control plane	RHCOS	4	16 GB	100 GB	300
Compute	RHCOS	2	8 GB	100 GB	300

NOTE

For OpenShift Container Platform version 4.19, RHCOS is based on RHEL version 9.6, which updates the micro-architecture requirements. The following list contains the minimum instruction set architectures (ISA) that each architecture requires:

- x86-64 architecture requires x86-64-v2 ISA
- ARM64 architecture requires ARMv8.0-A ISA
- IBM Power architecture requires Power 9 ISA
- s390x architecture requires z14 ISA

For more information, see Architectures (RHEL documentation).

If an instance type for your platform meets the minimum requirements for cluster machines, it is supported to use in OpenShift Container Platform.

Additional resources

- Optimizing storage

7.9.2. Tested instance types for IBM Cloud

The following IBM Cloud® instance types have been tested with OpenShift Container Platform.

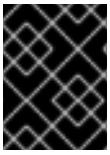
Example 7.1. Machine series

- **bx2-8x32**
- **bx2d-4x16**
- **bx3d-4x20**
- **bx3dc-8x40**
- **cx2-8x16**
- **cx2d-4x8**
- **cx3d-8x20**

- **cx3dc-4x10**
- **gx2-8x64x1v100**
- **gx3-16x80x1l4**
- **gx3d-160x1792x8h100**
- **mx2-8x64**
- **mx2d-4x32**
- **mx3d-4x40**
- **ox2-8x64**
- **ux2d-2x56**
- **vx2d-4x56**

7.9.3. Sample customized install-config.yaml file for IBM Cloud

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.



IMPORTANT

This sample YAML file is provided for reference only. You must obtain your **install-config.yaml** file by using the installation program and then modify it.

```
apiVersion: v1
baseDomain: example.com (1)
controlPlane: (2) (3)
  hyperthreading: Enabled (4)
  name: master
  platform:
    ibmcloud: {}
  replicas: 3
compute: (5) (6)
- hyperthreading: Enabled (7)
  name: worker
  platform:
    ibmcloud: {}
  replicas: 3
metadata:
  name: test-cluster (8)
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14 (9)
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16 (10)
```

```
  networkType: OVNKubernetes (11)
  serviceNetwork:
  - 172.30.0.0/16
platform:
  ibmcloud:
    region: eu-gb (12)
    resourceGroupName: eu-gb-example-cluster-rg (13)
    networkResourceGroupName: eu-gb-example-existing-network-rg (14)
    vpcName: eu-gb-example-network-1 (15)
    controlPlaneSubnets: (16)
      - eu-gb-example-network-1-cp-eu-gb-1
      - eu-gb-example-network-1-cp-eu-gb-2
      - eu-gb-example-network-1-cp-eu-gb-3
    computeSubnets: (17)
      - eu-gb-example-network-1-compute-eu-gb-1
      - eu-gb-example-network-1-compute-eu-gb-2
      - eu-gb-example-network-1-compute-eu-gb-3
credentialsMode: Manual
publish: Internal (18)
pullSecret: '{"auths": ...}' (19)
fips: false (20)
sshKey: ssh-ed25519 AAAA... (21)
```

(1) (8) (12) (19) Required.

(2) (5) If you do not provide these parameters and values, the installation program provides the default value.

(3) (6) 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.

(4) (7) Enables or disables simultaneous multithreading, also known as Hyper-Threading. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores. You can disable it by setting the parameter value to **Disabled**. If you disable simultaneous multithreading in some cluster machines, you must disable it in all cluster machines.

IMPORTANT

If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance. Use larger machine types, such as **n1-standard-8**, for your machines if you disable simultaneous multithreading.

(9) The machine CIDR must contain the subnets for the compute machines and control plane machines.

(10) The CIDR must contain the subnets defined in **platform.ibmcloud.controlPlaneSubnets** and **platform.ibmcloud.computeSubnets**.

(11) The cluster network plugin to install. The default value **OVNKubernetes** is the only supported value.

13 The name of an existing resource group. All installer-provisioned cluster resources are deployed to this resource group. If undefined, a new resource group is created for the cluster.

14 Specify the name of the resource group that contains the existing virtual private cloud (VPC). The existing VPC and subnets should be in this resource group. The cluster will be installed to this VPC.

15 Specify the name of an existing VPC.

16 Specify the name of the existing subnets to which to deploy the control plane machines. The subnets must belong to the VPC that you specified. Specify a subnet for each availability zone in the region.

17 Specify the name of the existing subnets to which to deploy the compute machines. The subnets must belong to the VPC that you specified. Specify a subnet for each availability zone in the region.

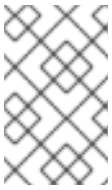
18 How to publish the user-facing endpoints of your cluster. Set **publish** to **Internal** to deploy a private cluster. The default value is **External**.

20 Enables or disables 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

When running Red Hat Enterprise Linux (RHEL) or Red Hat Enterprise Linux CoreOS (RHCOS) booted in FIPS mode, OpenShift Container Platform core components use the RHEL cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 Validation on only the x86_64, ppc64le, and s390x architectures.

21 Optional: provide the **sshKey** value that you use to access the machines in your cluster.



NOTE

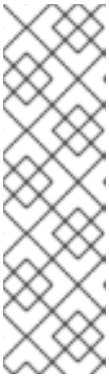
For production OpenShift Container Platform clusters on which you want to perform installation debugging or disaster recovery, specify an SSH key that your **ssh-agent** process uses.

7.9.4. Configuring the cluster-wide proxy during installation

To enable internet access in environments that deny direct connections, configure a cluster-wide proxy in the **install-config.yaml** file. This configuration ensures that the new OpenShift Container Platform cluster routes traffic through the specified HTTP or HTTPS proxy.

Prerequisites

- You have an existing **install-config.yaml** file.
- You have 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, Microsoft Azure, and Red Hat OpenStack Platform (RHOSP), the **Proxy** object **status.noProxy** field is also populated with the instance metadata endpoint (**169.254.169.254**).

Procedure

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

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port>
  httpsProxy: https://<username>:<pswd>@<ip>:<port>
  noProxy: example.com
additionalTrustBundle: |
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle>
# ...
```

where:

proxy.httpProxy

Specifies a proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**.

proxy.httpsProxy

Specifies a proxy URL to use for creating HTTPS connections outside the cluster.

proxy.noProxy

Specifies 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.

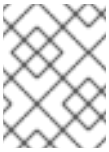
additionalTrustBundle

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.

additionalTrustBundlePolicy

Specifies the policy that determines the configuration of the **Proxy** object to reference the

user-ca-bundle config map in the **trustedCA** field. The allowed values are **Proxyonly** and **Always**. Use **Proxyonly** to reference the **user-ca-bundle** config map only when **http/https** proxy is configured. Use **Always** to always reference the **user-ca-bundle** config map. The default value is **Proxyonly**. Optional parameter.



NOTE

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

NOTE

If the installer times out, restart and then complete the deployment by using the **wait-for** command of the installer. For example:

+

```
$ ./openshift-install wait-for install-complete --log-level debug
```

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.

7.10. MANUALLY CREATING IAM

Installing the cluster requires that the Cloud Credential Operator (CCO) operate in manual mode. While the installation program configures the CCO for manual mode, you must specify the identity and access management secrets for you cloud provider.

You can use the Cloud Credential Operator (CCO) utility (**ccoctl**) to create the required IBM Cloud® resources.

Prerequisites

- You have configured the **ccoctl** binary.
- You have an existing **install-config.yaml** file.

Procedure

1. Edit the **install-config.yaml** configuration file so that the file includes the **credentialsMode** parameter set to **Manual**.

   Example install-config.yaml configuration file

```
apiVersion: v1
baseDomain: cluster1.example.com
credentialsMode: Manual
compute:
- architecture: amd64
  hyperthreading: Enabled
```

- **credentialsMode**: Set the **credentialsMode** parameter to **Manual**.

2. To generate the manifests, run the following command from the directory that includes the installation program:

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

3. From the directory that includes the installation program, set a **$RELEASE_IMAGE** variable with the release image from your installation file by running the following command:

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

4. Extract the list of **CredentialsRequest** custom resources (CRs) from the OpenShift Container Platform release image by running the following command:

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \
  --to=<path_to_directory_for_credentials_requests>
```

- **--included**: Includes only the manifests that your specific cluster configuration requires.
- **<path_to_directory_with_installation_configuration>**: Specify the location of the **install-config.yaml** file.
- **<path_to_directory_for_credentials_requests>**: Specify the path to the directory where you want to store the **CredentialsRequest** objects. If the specified directory does not exist, this command creates it.
  This command creates a YAML file for each **CredentialsRequest** object.

  Sample CredentialsRequest object

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: openshift-image-registry-ibmcos
  namespace: openshift-cloud-credential-operator
spec:
  secretRef:
    name: installer-cloud-credentials
    namespace: openshift-image-registry
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
```

```
kind: IBMCloudProviderSpec
policies:
- attributes:
  - name: serviceName
    value: cloud-object-storage
  roles:
  - crn:v1:bluemix:public:iam::::role:Viewer
  - crn:v1:bluemix:public:iam::::role:Operator
  - crn:v1:bluemix:public:iam::::role:Editor
  - crn:v1:bluemix:public:iam::::serviceRole:Reader
  - crn:v1:bluemix:public:iam::::serviceRole:Writer
- attributes:
  - name: resourceType
    value: resource-group
  roles:
  - crn:v1:bluemix:public:iam::::role:Viewer
```

5. Create the service ID for each credential request, assign the policies defined, create an API key, and generate the secret:

```
$ ccoctl ibmcloud create-service-id \
  --credentials-requests-dir=<path_to_credential_requests_directory> \
  --name=<cluster_name> \
  --output-dir=<installation_directory> \
  --resource-group-name=<resource_group_name>
```

- **<path_to_credential_requests_directory>**: Specify the directory containing the files for the **CredentialsRequest** objects.
- **<cluster_name>**: Specify the name of the OpenShift Container Platform cluster.
- **<installation_directory>**: Optional parameter. Specify the directory in which you want the **ccoctl** utility to create objects. By default, the utility creates objects in the directory in which you run the commands.
- **<resource_group_name>**: Optional parameter. Specify the name of the resource group used for scoping the access policies.

NOTE

If you enabled Technology Preview features by using the **TechPreviewNoUpgrade** feature set for your cluster, you must include the **--enable-tech-preview** parameter in the configuration for the **CredentialsRequest** object.

If you provided a wrong resource group name, the installation fails during the bootstrap phase. To find the correct resource group name, run the following command:

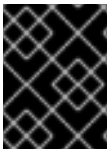
```
$ grep resourceGroupName <installation_directory>/manifests/cluster-infrastructure-02-config.yml
```

Verification

- Check that the appropriate secrets exist in the **manifests** directory of your cluster.

7.11. DEPLOYING THE CLUSTER

You can install OpenShift Container Platform on a compatible cloud platform.



IMPORTANT

You can run the **create cluster** command of the installation program only once, during initial installation.

Prerequisites

- You have configured an account with the cloud platform that hosts your cluster.
- You have the OpenShift Container Platform installation program and the pull secret for your cluster.
- You have verified that the cloud provider account on your host has the correct permissions to deploy the cluster. An account with incorrect permissions causes the installation process to fail with an error message that displays the missing permissions.

Procedure

- In the directory that contains the installation program, initialize the cluster deployment by running the following command:

```
$ ./openshift-install create cluster --dir <installation_directory> \ 1
    --log-level=info 2
```

1 For **<installation_directory>**, specify the location of your customized **./install-config.yaml** file.

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

Verification

When the cluster deployment completes successfully:

- The terminal displays directions for accessing your cluster, including a link to the web console and credentials for the **kubeadmin** user.
- Credential information also outputs to **<installation_directory>/.openshift_install.log**.

IMPORTANT

Do not delete the installation program or the files that the installation program creates. Both are required to delete the cluster.

Example output

```
...
INFO Install complete!
```

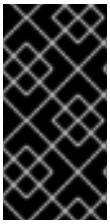
```
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/myuser/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.mycluster.example.com
INFO Login to the console with user: "kubeadmin", and password: "password"
INFO Time elapsed: 36m22s
```

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.

7.12. INSTALLING THE OPENSHIFT CLI ON LINUX

To manage your cluster and deploy applications from the command line, install the OpenShift CLI (**oc**) binary on Linux.



IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the Download OpenShift Container Platform page on the Red Hat Customer Portal.
2. Select the architecture from the **Product Variant** list.
3. Select the appropriate version from the **Version** list.
4. Click **Download Now** next to the **OpenShift v4.20 Linux Clients** entry and save the file.
5. Unpack the archive:

```
$ tar xvf <file>
```

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

```
$ echo $PATH
```

Verification

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

```
$ oc <command>
```

7.13. INSTALLING THE OPENSHIFT CLI ON WINDOWS

To manage your cluster and deploy applications from the command line, install OpenShift CLI (**oc**) binary on Windows.

IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the Download OpenShift Container Platform page on the Red Hat Customer Portal.
2. Select the appropriate version from the **Version** list.
3. Click **Download Now** next to the **OpenShift v4.20 Windows Client** entry and save the file.
4. Extract the archive with a ZIP program.
5. Move the **oc** binary to a directory that is on your **PATH** variable.
   To check your **PATH** variable, open the command prompt and execute the following command:

```
C:\> path
```

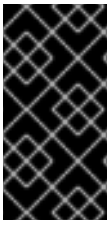
Verification

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

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

7.14. INSTALLING THE OPENSHIFT CLI ON MACOS

To manage your cluster and deploy applications from the command line, install the OpenShift CLI (**oc**) binary on macOS.



IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the Download OpenShift Container Platform page on the Red Hat Customer Portal.
2. Select the architecture from the **Product Variant** list.
3. Select the appropriate version from the **Version** list.
4. Click **Download Now** next to the **OpenShift v4.20 macOS Clients** entry and save the file.

NOTE

For macOS arm64, choose the **OpenShift v4.20 macOS arm64 Client** entry.

5. Unpack and unzip the archive.
6. Move the **oc** binary to a directory on your **PATH** variable.
   To check your **PATH** variable, open a terminal and execute the following command:

```
$ echo $PATH
```

Verification

- Verify your installation by using an **oc** command:

```
$ oc <command>
```

7.15. 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 OpenShift CLI (**oc**).

Procedure

1. Export the **kubeadmin** credentials by running the following command:

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

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

2. Verify you can run **oc** commands successfully using the exported configuration by running the following command:

```
$ oc whoami
```

Example output

```
system:admin
```

Additional resources

- Accessing the web console

7.16. TELEMETRY ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.20, 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

- About remote health monitoring

7.17. NEXT STEPS

- Customize your cluster.
- If necessary, you can Remote health reporting.

CHAPTER 8. INSTALLING A CLUSTER ON IBM CLOUD IN A DISCONNECTED ENVIRONMENT

In OpenShift Container Platform 4.20, you can install a cluster in a restricted network by creating an internal mirror of the installation release content that is accessible to an existing Virtual Private Cloud (VPC) on IBM Cloud®.

8.1. PREREQUISITES

- You reviewed details about the OpenShift Container Platform installation and update processes.
- You configured an IBM Cloud account to host the cluster.
- You have a container image registry that is accessible to the internet and your restricted network. The container image registry should mirror the contents of the OpenShift image registry and contain the installation media. For more information, see Mirroring images for a disconnected installation by using the oc-mirror plugin v2.
- You have an existing VPC on IBM Cloud® that meets the following requirements:
    - The VPC contains the mirror registry or has firewall rules or a peering connection to access the mirror registry that is hosted elsewhere.
    - The VPC can access IBM Cloud® service endpoints using a public endpoint. If network restrictions limit access to public service endpoints, evaluate those services for alternate endpoints that might be available. For more information see Access to IBM service endpoints.

    You cannot use the VPC that the installation program provisions by default.
- If you plan on configuring endpoint gateways to use IBM Cloud® Virtual Private Endpoints, consider the following requirements:
    - Endpoint gateway support is currently limited to the **us-east** and **us-south** regions.
    - The VPC must allow traffic to and from the endpoint gateways. You can use the VPC's default security group, or a new security group, to allow traffic on port 443. For more information, see Allowing endpoint gateway traffic.
- If you use a firewall, you configured it to allow the sites that your cluster requires access to.
- You configured the **ccoctl** utility before you installed the cluster. For more information, see Configuring IAM for IBM Cloud.

8.2. ABOUT INSTALLATIONS IN RESTRICTED NETWORKS

In OpenShift Container Platform 4.20, 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.

8.2.1. Required internet access and an installation host

You complete the installation using a bastion host or portable device that can access both the internet and your closed network. You must use a host with internet access to:

- Download the installation program, the OpenShift CLI (**oc**), and the CCO utility (**ccoctl**).
- Use the installation program to locate the Red Hat Enterprise Linux CoreOS (RHCOS) image and create the installation configuration file.
- Use **oc** to extract **ccoctl** from the CCO container image.
- Use **oc** and **ccoctl** to configure IAM for IBM Cloud®.

8.2.2. Access to a mirror registry

To complete a restricted network installation, you must create a registry that mirrors the contents of the OpenShift image registry and contains the installation media.

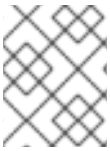
You can create this registry on a mirror host, which can access both the internet and your restricted network, or by using other methods that meet your organization's security restrictions.

For more information on mirroring images for a disconnected installation, see "Additional resources".

8.2.3. Access to IBM service endpoints

The installation program requires access to the following IBM Cloud® service endpoints:

- Cloud Object Storage
- DNS Services
- Global Search
- Global Tagging
- Identity Services
- Resource Controller
- Resource Manager
- VPC



NOTE

If you are specifying an IBM® Key Protect for IBM Cloud® root key as part of the installation process, the service endpoint for Key Protect is also required.

By default, the public endpoint is used to access the service. If network restrictions limit access to public service endpoints, you can override the default behavior.

Before deploying the cluster, you can update the installation configuration file (**install-config.yaml**) to specify the URI of an alternate service endpoint. For more information on usage, see "Additional resources".

8.2.4. 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.

Additional resources

- Mirroring images for a disconnected installation by using the oc-mirror plugin v2
- Additional IBM Cloud configuration parameters

8.3. ABOUT USING A CUSTOM VPC

In OpenShift Container Platform 4.20, you can deploy a cluster into the subnets of an existing IBM® Virtual Private Cloud (VPC). Deploying OpenShift Container Platform into an existing VPC can help you avoid limit constraints in new accounts or more easily abide by the operational constraints that your company's guidelines set. If you cannot obtain the infrastructure creation permissions that are required to create the VPC yourself, use this installation option.

Because the installation program cannot know what other components are in your existing subnets, it cannot choose subnet CIDRs and so forth. You must configure networking for the subnets to which you will install the cluster.

8.3.1. Requirements for using your VPC

You must correctly configure the existing VPC and its subnets before you install the cluster. The installation program does not create the following components:

- NAT gateways
- Subnets
- Route tables
- VPC network

The installation program cannot:

- Subdivide network ranges for the cluster to use
- Set route tables for the subnets
- Set VPC options like DHCP

NOTE

The installation program requires that you use the cloud-provided DNS server. Using a custom DNS server is not supported and causes the installation to fail.

8.3.2. VPC validation

The VPC and all of the subnets must be in an existing resource group. The cluster is deployed to the existing VPC.

As part of the installation, specify the following in the **install-config.yaml** file:

- The name of the existing resource group that contains the VPC and subnets (**networkResourceGroupName**)
- The name of the existing VPC (**vpcName**)
- The subnets that were created for control plane machines and compute machines (**controlPlaneSubnets** and **computeSubnets**)

NOTE

Additional installer-provisioned cluster resources are deployed to a separate resource group (**resourceGroupName**). You can specify this resource group before installing the cluster. If undefined, a new resource group is created for the cluster.

To ensure that the subnets that you provide are suitable, the installation program confirms the following:

- All of the subnets that you specify exist.
- For each availability zone in the region, you specify:
  - One subnet for control plane machines.
  - One subnet for compute machines.
- The machine CIDR that you specified contains the subnets for the compute machines and control plane machines.

NOTE

Subnet IDs are not supported.

8.3.3. Isolation between clusters

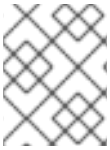
If you deploy OpenShift Container Platform to an existing network, the isolation of cluster services is reduced in the following ways:

- You can install multiple OpenShift Container Platform clusters in the same VPC.
- ICMP ingress is allowed to the entire network.
- TCP port 22 ingress (SSH) is allowed to the entire network.
- Control plane TCP 6443 ingress (Kubernetes API) is allowed to the entire network.
- Control plane TCP 22623 ingress (MCS) is allowed to the entire network.

8.3.4. Allowing endpoint gateway traffic

If you are using IBM Cloud® Virtual Private endpoints, your Virtual Private Cloud (VPC) must be configured to allow traffic to and from the endpoint gateways.

A VPC's default security group is configured to allow all outbound traffic to endpoint gateways. Therefore, the simplest way to allow traffic between your VPC and endpoint gateways is to modify the default security group to allow inbound traffic on port 443.



NOTE

If you choose to configure a new security group, the security group must be configured to allow both inbound and outbound traffic.

Prerequisites

- You have installed the IBM Cloud® Command Line Interface utility (**ibmcloud**).

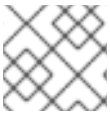
Procedure

1. Obtain the identifier for the default security group by running the following command:

```
$ DEFAULT_SG=$(ibmcloud is vpc <your_vpc_name> --output JSON | jq -r
'.default_security_group.id')
```

2. Add a rule that allows inbound traffic on port 443 by running the following command:

```
$ ibmcloud is security-group-rule-add $DEFAULT_SG inbound tcp --remote 0.0.0.0/0 --port-
min 443 --port-max 443
```



NOTE

Be sure that your endpoint gateways are configured to use this security group.

8.4. GENERATING A KEY PAIR FOR CLUSTER NODE SSH ACCESS

To enable secure, passwordless SSH access to your cluster nodes, provide an SSH public key during the OpenShift Container Platform installation. This ensures that the installation program automatically configures the Red Hat Enterprise Linux CoreOS (RHCOS) nodes for remote authentication through the **core** user.

The SSH public key gets added to the **~/.ssh/authorized_keys** list for the **core** user on each node. After the key is passed to the Red Hat Enterprise Linux CoreOS (RHCOS) nodes through their Ignition config files, you can use the key pair to SSH in to the RHCOS nodes as the user **core**. To access the nodes through SSH, the private key identity must be managed by SSH for your local user.

If you want to SSH in to your cluster nodes to perform installation debugging or disaster recovery, you must provide the SSH public key during the installation process. The **./openshift-install gather** command also requires the SSH public key to be in place on the cluster nodes.

IMPORTANT

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

NOTE

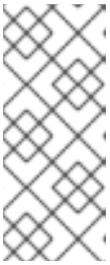
You must use a local key, not one that you configured with platform-specific approaches.

Procedure

1. If you do not have an existing SSH key pair on your local machine to use for authentication onto your cluster nodes, 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>
```

Specifies the path and file name, such as **~/.ssh/id_ed25519**, of the new SSH key. If you have an existing key pair, ensure your public key is in the your **~/.ssh** directory.



NOTE

If you plan to install an OpenShift Container Platform cluster that uses the RHEL cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 Validation on only the **x86_64**, **ppc64le**, and **s390x** architectures, do not create a key that uses the **ed25519** algorithm. Instead, create a key that uses the **rsa** or **ecdsa** algorithm.

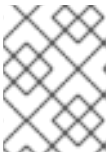
2. View the public SSH key:

```
$ cat <path>/<file_name>.pub
```

For example, run the following to view the **~/.ssh/id_ed25519.pub** public key:

```
$ cat ~/.ssh/id_ed25519.pub
```

3. Add the SSH private key identity to the SSH agent for your local user, if it has not already been added. SSH agent management of the key is required for password-less SSH authentication onto your cluster nodes, or if you want to use the **./openshift-install gather** command.



NOTE

On some distributions, default SSH private key identities such as **~/.ssh/id_rsa** and **~/.ssh/id_dsa** are managed automatically.

a. If the **ssh-agent** process is not already running for your local user, start it 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.

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

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

Specifies the path and file name for your SSH private key, such as **~/.ssh/id_ed25519**

Example output

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

Next steps

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

8.5. EXPORTING THE API KEY

You must set the API key you created as a global variable; the installation program ingests the variable during startup to set the API key.

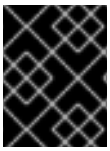
Prerequisites

- You have created either a user API key or service ID API key for your IBM Cloud® account.

Procedure

- Export your API key for your account as a global variable:

```
$ export IC_API_KEY=<api_key>
```



IMPORTANT

You must set the variable name exactly as specified; the installation program expects the variable name to be present during startup.

8.6. DOWNLOADING THE RHCOS CLUSTER IMAGE

The installation program requires the Red Hat Enterprise Linux CoreOS (RHCOS) image to install the cluster. While optional, downloading the Red Hat Enterprise Linux CoreOS (RHCOS) before deploying removes the need for internet access when creating the cluster.

Use the installation program to locate and download the Red Hat Enterprise Linux CoreOS (RHCOS) image.

Prerequisites

- The host running the installation program has internet access.

Procedure

1. Change to the directory that contains the installation program and run the following command:

```
$ ./openshift-install coreos print-stream-json
```

2. Use the output of the command to find the location of the IBM Cloud® image.

```
.Example output
----
  "release": "415.92.202311241643-0",
  "formats": {
    "qcow2.gz": {
      "disk": {
        "location": "https://rhcos.mirror.openshift.com/art/storage/prod/streams/4.15-
9.2/builds/415.92.202311241643-0/x86_64/rhcos-415.92.202311241643-0-
ibmcloud.x86_64.qcow2.gz",
        "sha256":
"6b562dee8431bec3b93adeac1cfefcd5e812d41e3b7d78d3e28319870ffc9eae",
        "uncompressed-sha256":
"5a0f9479505e525a30367b6a6a6547c86a8f03136f453c1da035f3aa5daa8bc9"
----
```

3. Download and extract the image archive. Make the image available on the host that the installation program uses to create the cluster.

8.7. MANUALLY CREATING THE INSTALLATION CONFIGURATION FILE

To customise your OpenShift Container Platform deployment and meet specific network requirements, manually create the installation configuration file. This ensures that the installation program uses your tailored settings rather than default values during the setup process.

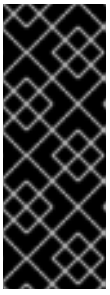
Prerequisites

- You have obtained the OpenShift Container Platform installation program and the pull secret for your cluster.
- You have the **imageContentSourcePolicy.yaml** file that was created when you mirrored your registry.
- You have obtained 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, such as 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 provided sample **install-config.yaml** file template and save the file in the **<installation_directory>**.

NOTE

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

When customizing the sample template, be sure to provide the information that is required for an installation in a restricted network:

a. Update the **pullSecret** value to contain the authentication information for your registry:

```
pullSecret: '{"auths":{"<mirror_host_name>:5000": {"auth": "<credentials>","email":
"you@example.com"}}}'
```

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

b. Add the **additionalTrustBundle** parameter and value.

```
additionalTrustBundle: |
  -----BEGIN CERTIFICATE-----

  ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
  -----END CERTIFICATE-----
```

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

c. Define the network and subnets for the VPC to install the cluster in under the parent **platform.ibmcloud** field:

```
vpcName: <existing_vpc>
controlPlaneSubnets: <control_plane_subnet>
computeSubnets: <compute_subnet>
```

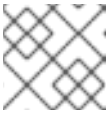
For **platform.ibmcloud.vpcName**, specify the name for the existing IBM Cloud Virtual Private Cloud (VPC) network. For **platform.ibmcloud.controlPlaneSubnets** and **platform.ibmcloud.computeSubnets**, specify the existing subnets to deploy the control plane machines and compute machines, respectively.

d. Add the image content resources, which resemble the following YAML excerpt:

```
imageContentSources:
- mirrors:
  - <mirror_host_name>:5000/<repo_name>/release
  source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - <mirror_host_name>:5000/<repo_name>/release
  source: registry.redhat.io/ocp/release
```

For these values, use the **imageContentSourcePolicy.yaml** file that was created when you mirrored the registry.

e. If network restrictions limit the use of public endpoints to access the required IBM Cloud® services, add the **serviceEndpoints** stanza to **platform.ibmcloud** to specify an alternate service endpoint.



NOTE

You can specify only one alternate service endpoint for each service.

Example of using alternate services endpoints

```
# ...
serviceEndpoints:
  - name: IAM
    url: <iam_alternate_endpoint_url>
  - name: VPC
    url: <vpc_alternate_endpoint_url>
  - name: ResourceController
    url: <resource_controller_alternate_endpoint_url>
  - name: ResourceManager
    url: <resource_manager_alternate_endpoint_url>
  - name: DNSServices
    url: <dns_services_alternate_endpoint_url>
  - name: COS
    url: <cos_alternate_endpoint_url>
  - name: GlobalSearch
    url: <global_search_alternate_endpoint_url>
  - name: GlobalTagging
    url: <global_tagging_alternate_endpoint_url>
# ...
```

f. Optional: Set the publishing strategy to **Internal**:

```
publish: Internal
```

By setting this option, you create an internal Ingress Controller and a private load balancer.

NOTE

If you use the default value of **External**, your network must be able to access the public endpoint for IBM Cloud® Internet Services (CIS). CIS is not enabled for Virtual Private Endpoints.

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

IMPORTANT

Back up the **install-config.yaml** file now, because the installation process consumes the file in the next step.

8.7.1. Configuring the cluster-wide proxy during installation

To enable internet access in environments that deny direct connections, configure a cluster-wide proxy in the **install-config.yaml** file. This configuration ensures that the new OpenShift Container Platform cluster routes traffic through the specified HTTP or HTTPS proxy.

Prerequisites

- You have an existing **install-config.yaml** file.
- You have 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, Microsoft Azure, and Red Hat OpenStack Platform (RHOSP), the **Proxy** object **status.noProxy** field is also populated with the instance metadata endpoint (**169.254.169.254**).

Procedure

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

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port>
  httpsProxy: https://<username>:<pswd>@<ip>:<port>
  noProxy: example.com
additionalTrustBundle: |
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle>
# ...
```

where:

proxy.httpProxy

Specifies a proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**.

proxy.httpsProxy

Specifies a proxy URL to use for creating HTTPS connections outside the cluster.

proxy.noProxy

Specifies 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.

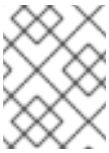
For example, **.y.com** matches **x.y.com**, but not **y.com**. Use * to bypass the proxy for all destinations.

additionalTrustBundle

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.

additionalTrustBundlePolicy

Specifies the policy that determines the configuration of the **Proxy** object to reference the **user-ca-bundle** config map in the **trustedCA** field. The allowed values are **Proxyonly** and **Always**. Use **Proxyonly** to reference the **user-ca-bundle** config map only when **http/https** proxy is configured. Use **Always** to always reference the **user-ca-bundle** config map. The default value is **Proxyonly**. Optional parameter.



NOTE

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

NOTE

If the installer times out, restart and then complete the deployment by using the **wait-for** command of the installer. For example:

+

```
$ ./openshift-install wait-for install-complete --log-level debug
```

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.

Additional resources

- Installation configuration parameters for IBM Cloud®

8.7.2. Minimum resource requirements for cluster installation

Each created cluster must meet minimum requirements so that the cluster runs as expected.

Table 8.1. Minimum resource requirements

Machine	Operating System	vCPU	Virtual RAM	Storage	Input/Output Per Second (IOPS)
Bootstrap	RHCOS	4	16 GB	100 GB	300
Control plane	RHCOS	4	16 GB	100 GB	300
Compute	RHCOS	2	8 GB	100 GB	300

NOTE

For OpenShift Container Platform version 4.19, RHCOS is based on RHEL version 9.6, which updates the micro-architecture requirements. The following list contains the minimum instruction set architectures (ISA) that each architecture requires:

- x86-64 architecture requires x86-64-v2 ISA
- ARM64 architecture requires ARMv8.0-A ISA
- IBM Power architecture requires Power 9 ISA
- s390x architecture requires z14 ISA

For more information, see Architectures (RHEL documentation).

If an instance type for your platform meets the minimum requirements for cluster machines, it is supported to use in OpenShift Container Platform.

8.7.3. Tested instance types for IBM Cloud

The following IBM Cloud® instance types have been tested with OpenShift Container Platform.

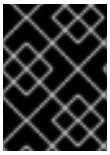
Example 8.1. Machine series

- **bx2-8x32**
- **bx2d-4x16**
- **bx3d-4x20**
- **bx3dc-8x40**
- **cx2-8x16**
- **cx2d-4x8**
- **cx3d-8x20**
- **cx3dc-4x10**

- **gx2-8x64x1v100**
- **gx3-16x80x1l4**
- **gx3d-160x1792x8h100**
- **mx2-8x64**
- **mx2d-4x32**
- **mx3d-4x40**
- **ox2-8x64**
- **ux2d-2x56**
- **vx2d-4x56**

8.7.4. Sample customized install-config.yaml file for IBM Cloud

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.



IMPORTANT

This sample YAML file is provided for reference only. You must obtain your **install-config.yaml** file by using the installation program and then modify it.

```
apiVersion: v1
baseDomain: example.com (1)
controlPlane: (2) (3)
  hyperthreading: Enabled (4)
  name: master
  platform:
    ibm-cloud: {}
  replicas: 3
compute: (5) (6)
- hyperthreading: Enabled (7)
  name: worker
  platform:
    ibmcloud: {}
  replicas: 3
metadata:
  name: test-cluster (8)
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14 (9)
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16 (10)
  networkType: OVNKubernetes (11)
  serviceNetwork:
```

```
  - 172.30.0.0/16
platform:
  ibmcloud:
    region: us-east (12)
    resourceGroupName: us-east-example-cluster-rg (13)
    serviceEndpoints: (14)
      - name: IAM
        url: https://private.us-east.iam.cloud.ibm.com
      - name: VPC
        url: https://us-east.private.iaas.cloud.ibm.com/v1
      - name: ResourceController
        url: https://private.us-east.resource-controller.cloud.ibm.com
      - name: ResourceManager
        url: https://private.us-east.resource-controller.cloud.ibm.com
      - name: DNSServices
        url: https://api.private.dns-svcs.cloud.ibm.com/v1
      - name: COS
        url: https://s3.direct.us-east.cloud-object-storage.appdomain.cloud
      - name: GlobalSearch
        url: https://api.private.global-search-tagging.cloud.ibm.com
      - name: GlobalTagging
        url: https://tags.private.global-search-tagging.cloud.ibm.com
    networkResourceGroupName: us-east-example-existing-network-rg (15)
    vpcName: us-east-example-network-1 (16)
    controlPlaneSubnets: (17)
      - us-east-example-network-1-cp-us-east-1
      - us-east-example-network-1-cp-us-east-2
      - us-east-example-network-1-cp-us-east-3
    computeSubnets: (18)
      - us-east-example-network-1-compute-us-east-1
      - us-east-example-network-1-compute-us-east-2
      - us-east-example-network-1-compute-us-east-3
credentialsMode: Manual
pullSecret: '{"auths":{"<local_registry>": {"auth": "<credentials>","email": "you@example.com"}}}' (19)
fips: false (20)
sshKey: ssh-ed25519 AAAA... (21)
additionalTrustBundle: | (22)
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
imageContentSources: (23)
- mirrors:
  - <local_registry>/<local_repository_name>/release
  source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - <local_registry>/<local_repository_name>/release
  source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
```

(1) (8) (12) Required.

(2) (5) If you do not provide these parameters and values, the installation program provides the default value.

(3) (6)

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.

4 7 Enables or disables simultaneous multithreading, also known as Hyper-Threading. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores. You can disable it by setting the parameter value to **Disabled**. If you disable simultaneous multithreading in some cluster machines, you must disable it in all cluster machines.

IMPORTANT

If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance. Use larger machine types, such as **n1-standard-8**, for your machines if you disable simultaneous multithreading.

9 The machine CIDR must contain the subnets for the compute machines and control plane machines.

10 The CIDR must contain the subnets defined in **platform.ibmcloud.controlPlaneSubnets** and **platform.ibmcloud.computeSubnets**.

11 The cluster network plugin to install. The default value **OVNKubernetes** is the only supported value.

13 The name of an existing resource group. All installer-provisioned cluster resources are deployed to this resource group. If undefined, a new resource group is created for the cluster.

14 Based on the network restrictions of the VPC, specify alternate service endpoints as needed. This overrides the default public endpoint for the service.

15 Specify the name of the resource group that contains the existing virtual private cloud (VPC). The existing VPC and subnets should be in this resource group. The cluster will be installed to this VPC.

16 Specify the name of an existing VPC.

17 Specify the name of the existing subnets to which to deploy the control plane machines. The subnets must belong to the VPC that you specified. Specify a subnet for each availability zone in the region.

18 Specify the name of the existing subnets to which to deploy the compute machines. The subnets must belong to the VPC that you specified. Specify a subnet for each availability zone in the region.

19 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.

20 Enables or disables 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.

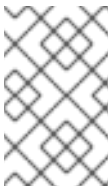
IMPORTANT

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

21 Optional: provide the **sshKey** value that you use to access the machines in your cluster.

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

23 Provide these values from the **metadata.name: release-0** section of the **imageContentSourcePolicy.yaml** file that was created when you mirrored the registry.



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.

8.8. INSTALLING THE OPENSHIFT CLI ON LINUX

To manage your cluster and deploy applications from the command line, install the OpenShift CLI (**oc**) binary on Linux.

IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the Download OpenShift Container Platform page on the Red Hat Customer Portal.
2. Select the architecture from the **Product Variant** list.
3. Select the appropriate version from the **Version** list.
4. Click **Download Now** next to the **OpenShift v4.20 Linux Clients** entry and save the file.
5. Unpack the archive:

```
$ tar xvf <file>
```

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

```
$ echo $PATH
```

Verification

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

```
$ oc <command>
```

8.9. INSTALLING THE OPENSHIFT CLI ON WINDOWS

To manage your cluster and deploy applications from the command line, install OpenShift CLI (**oc**) binary on Windows.

IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the Download OpenShift Container Platform page on the Red Hat Customer Portal.
2. Select the appropriate version from the **Version** list.
3. Click **Download Now** next to the **OpenShift v4.20 Windows Client** entry and save the file.
4. Extract the archive with a ZIP program.
5. Move the **oc** binary to a directory that is on your **PATH** variable.
   To check your **PATH** variable, open the command prompt and execute the following command:

```
C:\> path
```

Verification

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

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

8.10. INSTALLING THE OPENSHIFT CLI ON MACOS

To manage your cluster and deploy applications from the command line, install the OpenShift CLI (**oc**) binary on macOS.

IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the Download OpenShift Container Platform page on the Red Hat Customer Portal.

2. Select the architecture from the **Product Variant** list.
3. Select the appropriate version from the **Version** list.
4. Click **Download Now** next to the **OpenShift v4.20 macOS Clients** entry and save the file.

NOTE

For macOS arm64, choose the **OpenShift v4.20 macOS arm64 Client** entry.

5. Unpack and unzip the archive.
6. Move the **oc** binary to a directory on your **PATH** variable.
   To check your **PATH** variable, open a terminal and execute the following command:

```
$ echo $PATH
```

Verification

- Verify your installation by using an **oc** command:

```
$ oc <command>
```

8.11. MANUALLY CREATING IAM

Installing the cluster requires that the Cloud Credential Operator (CCO) operate in manual mode. While the installation program configures the CCO for manual mode, you must specify the identity and access management secrets for you cloud provider.

You can use the Cloud Credential Operator (CCO) utility (**ccoctl**) to create the required IBM Cloud® resources.

Prerequisites

- You have configured the **ccoctl** binary.
- You have an existing **install-config.yaml** file.

Procedure

1. Edit the **install-config.yaml** configuration file so that the file includes the **credentialsMode** parameter set to **Manual**.

   Example install-config.yaml configuration file

```
apiVersion: v1
baseDomain: cluster1.example.com
credentialsMode: Manual
compute:
- architecture: amd64
  hyperthreading: Enabled
```

- **credentialsMode**: Set the **credentialsMode** parameter to **Manual**.

2. To generate the manifests, run the following command from the directory that includes the installation program:

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

3. From the directory that includes the installation program, set a **$RELEASE_IMAGE** variable with the release image from your installation file by running the following command:

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

4. Extract the list of **CredentialsRequest** custom resources (CRs) from the OpenShift Container Platform release image by running the following command:

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --included \
  --install-config=<path_to_directory_with_installation_configuration>/install-config.yaml \
  --to=<path_to_directory_for_credentials_requests>
```

- **--included**: Includes only the manifests that your specific cluster configuration requires.
- **<path_to_directory_with_installation_configuration>**: Specify the location of the **install-config.yaml** file.
- **<path_to_directory_for_credentials_requests>**: Specify the path to the directory where you want to store the **CredentialsRequest** objects. If the specified directory does not exist, this command creates it.
  This command creates a YAML file for each **CredentialsRequest** object.

  Sample CredentialsRequest object

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: openshift-image-registry-ibmcos
  namespace: openshift-cloud-credential-operator
spec:
  secretRef:
    name: installer-cloud-credentials
    namespace: openshift-image-registry
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: IBMCloudProviderSpec
    policies:
    - attributes:
      - name: serviceName
        value: cloud-object-storage
      roles:
      - crn:v1:bluemix:public:iam::::role:Viewer
```

```
    - crn:v1:bluemix:public:iam::::role:Operator
    - crn:v1:bluemix:public:iam::::role:Editor
    - crn:v1:bluemix:public:iam::::serviceRole:Reader
    - crn:v1:bluemix:public:iam::::serviceRole:Writer
  - attributes:
    - name: resourceType
      value: resource-group
    roles:
    - crn:v1:bluemix:public:iam::::role:Viewer
```

5. Create the service ID for each credential request, assign the policies defined, create an API key, and generate the secret:

```
$ ccoctl ibmcloud create-service-id \
  --credentials-requests-dir=<path_to_credential_requests_directory> \
  --name=<cluster_name> \
  --output-dir=<installation_directory> \
  --resource-group-name=<resource_group_name>
```

- **<path_to_credential_requests_directory>**: Specify the directory containing the files for the **CredentialsRequest** objects.
- **<cluster_name>**: Specify the name of the OpenShift Container Platform cluster.
- **<installation_directory>**: Optional parameter. Specify the directory in which you want the **ccoctl** utility to create objects. By default, the utility creates objects in the directory in which you run the commands.
- **<resource_group_name>**: Optional parameter. Specify the name of the resource group used for scoping the access policies.

NOTE

If you enabled Technology Preview features by using the **TechPreviewNoUpgrade** feature set for your cluster, you must include the **--enable-tech-preview** parameter in the configuration for the **CredentialsRequest** object.

If you provided a wrong resource group name, the installation fails during the bootstrap phase. To find the correct resource group name, run the following command:

```
$ grep resourceGroupName <installation_directory>/manifests/cluster-
infrastructure-02-config.yml
```

Verification

- Check that the appropriate secrets exist in the **manifests** directory of your cluster.

8.12. DEPLOYING THE CLUSTER

You can install OpenShift Container Platform on a compatible cloud platform.

IMPORTANT

You can run the **create cluster** command of the installation program only once, during initial installation.

Prerequisites

- You have configured an account with the cloud platform that hosts your cluster.
- You have the OpenShift Container Platform installation program and the pull secret for your cluster.
  If the Red Hat Enterprise Linux CoreOS (RHCOS) image is available locally, the host running the installation program does not require internet access.
- You have verified that the cloud provider account on your host has the correct permissions to deploy the cluster. An account with incorrect permissions causes the installation process to fail with an error message that displays the missing permissions.

Procedure

1. Export the **OPENSHIFT_INSTALL_OS_IMAGE_OVERRIDE** variable to specify the location of the Red Hat Enterprise Linux CoreOS (RHCOS) image by running the following command:

```
$ export OPENSHIFT_INSTALL_OS_IMAGE_OVERRIDE="<path_to_image>/rhcos-<image_version>-ibmcloud.x86_64.qcow2.gz"
```

2. In the directory that contains the installation program, initialize the cluster deployment by running the following command:

```
$ ./openshift-install create cluster --dir <installation_directory> \ 1
    --log-level=info 2
```

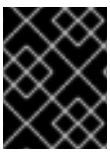
1 For **<installation_directory>**, specify the location of your customized **./install-config.yaml** file.

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

Verification

When the cluster deployment completes successfully:

- The terminal displays directions for accessing your cluster, including a link to the web console and credentials for the **kubeadmin** user.
- Credential information also outputs to **<installation_directory>/.openshift_install.log**.



IMPORTANT

Do not delete the installation program or the files that the installation program creates. Both are required to delete the cluster.

Example output

```
...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/myuser/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.mycluster.example.com
INFO Login to the console with user: "kubeadmin", and password: "password"
INFO Time elapsed: 36m22s
```

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.

8.13. 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 OpenShift CLI (**oc**).

Procedure

1. Export the **kubeadmin** credentials by running the following command:

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

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

2. Verify you can run **oc** commands successfully using the exported configuration by running the following command:

```
$ oc whoami
```

Example output

Example output

```
system:admin
```

Additional resources

- Accessing the web console

8.14. POST INSTALLATION

Complete the following steps to complete the configuration of your cluster.

8.14.1. Disabling the default software catalog sources

Operator catalogs that source content provided by Red Hat and community projects are configured for the software catalog 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:

```
$ oc patch OperatorHub cluster --type json \
    -p '[{"op": "add", "path": "/spec/disableAllDefaultSources", "value": true}]'
```

TIP

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

8.14.2. Installing the policy resources into the cluster

Mirroring the OpenShift Container Platform content using the oc-mirror OpenShift CLI (oc) plugin creates resources, which include **catalogSource-certified-operator-index.yaml** and **imageContentSourcePolicy.yaml**.

- The **ImageContentSourcePolicy** resource associates the mirror registry with the source registry and redirects image pull requests from the online registries to the mirror registry.
- The **CatalogSource** resource is used by Operator Lifecycle Manager (OLM) Classic to retrieve information about the available Operators in the mirror registry, which lets users discover and install Operators.

NOTE

OLM v1 uses the **ClusterCatalog** resource to retrieve information about the available cluster extensions in the mirror registry.

The oc-mirror plugin v1 does not generate **ClusterCatalog** resources automatically; you must manually create them. The oc-mirror plugin v2 does, however, generate **ClusterCatalog** resources automatically.

For more information on creating and applying **ClusterCatalog** resources, see "Adding a catalog to a cluster" in "Extensions".

After you install the cluster, you must install these resources into the cluster.

Prerequisites

- You have mirrored the image set to the registry mirror in the disconnected environment.
- You have access to the cluster as a user with the **cluster-admin** role.

Procedure

1. Log in to the OpenShift CLI as a user with the **cluster-admin** role.
2. Apply the YAML files from the results directory to the cluster:

```
$ oc apply -f ./oc-mirror-workspace/results-<id>/
```

Verification

1. Verify that the **ImageContentSourcePolicy** resources were successfully installed:

```
$ oc get imagecontentsourcepolicy
```

2. Verify that the **CatalogSource** resources were successfully installed:

```
$ oc get catalogsource --all-namespaces
```

8.15. TELEMETRY ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.20, 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

- About remote health monitoring

8.16. NEXT STEPS

- Customize your cluster.
- Optional: Remote health reporting.

CHAPTER 9. INSTALLATION CONFIGURATION PARAMETERS FOR IBM CLOUD

Before you deploy an OpenShift Container Platform cluster on IBM Cloud®, you provide parameters to customize your cluster and the platform that hosts it. When you create the **install-config.yaml** file, you provide values for the required parameters through the command line. You can then modify the **install-config.yaml** file to customize your cluster further.

9.1. AVAILABLE INSTALLATION CONFIGURATION PARAMETERS FOR IBM CLOUD

The following tables specify the required, optional, and IBM Cloud-specific installation configuration parameters that you can set as part of the installation process.

IMPORTANT

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

9.1.1. Required configuration parameters

Required installation configuration parameters are described in the following table:

Table 9.1. Required parameters

Parameter	Description
apiVersion:	The API version for the **install-config.yaml** content. The current version is **v1**. The installation program might also support older API versions. **Value:** String
baseDomain:	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 **<metadata.name>.<baseDomain>** format. **Value:** A fully-qualified domain or subdomain name, such as **example.com**.
metadata:	Kubernetes resource **ObjectMeta**, from which only the **name** parameter is consumed. **Value:** Object

Parameter	Description
metadata: name:	The name of the cluster. DNS records for the cluster are all subdomains of **{{.metadata.name}}.{{.baseDomain}}**. **Value:** String of lowercase letters, hyphens (-), and periods (.), such as **dev**.
platform:	The configuration for the specific platform upon which to perform the installation: **aws**, **baremetal**, **azure**, **gcp**, **ibmcloud**, **nutanix**, **openstack**, **powervs**, **vsphere**, or **{}**. For additional information about **platform.<platform>** parameters, consult the table for your specific platform that follows. **Value:** Object

pullSecret:	Get a pull secret from Red Hat OpenShift Cluster Manager to authenticate downloading container images for OpenShift Container Platform components from services such as Quay.io.
	
	Value:
	
	{
	  "auths":{
	  "cloud.openshift.com":{
	  "auth":"b3Blb=",
	  "email":"you@example.com"
	  },
	  "quay.io":{
	  "auth":"b3Blb=",
	  "email":"you@example.com"
	  }
	  }
	}

9.1.2. 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 configure different IP address blocks than the defaults.

Only IPv4 addresses are supported.

Table 9.2. Network parameters

Parameter	Description
`networking:`	The configuration for the cluster network. **Value:** Object **NOTE** You cannot change parameters specified by the **networking** object after installation.
`networking:` `  networkType:`	The Red Hat OpenShift Networking network plugin to install. Value:**OVNKubernetes**. **OVNKubernetes** is a Container Network Interface (CNI) plugin for Linux networks and hybrid networks that contain both Linux and Windows servers. The default value is **OVNKubernetes**.

`networking:`
`  clusterNetwork:`
	The IP address blocks for pods.
	
	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.
	
	Value: An array of objects. For example:
	
	`networking:`
	`  clusterNetwork:`
	`  - cidr: 10.128.0.0/14`
	`  hostPrefix: 23`

`networking:`
`  clusterNetwork:`
`  cidr:`
	Required if you use **networking.clusterNetwork**. An IP address block.
	
	An IPv4 network.
	
	Value: An IP address block in Classless Inter-Domain Routing (CIDR) notation. The prefix length for an IPv4 block is between **0** and **32**.

Parameter	Description

`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 - 23) - 2) pod IP addresses.
	
	Value: 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**.
	
	The OVN-Kubernetes network plugins supports only a single IP address block for the service network.
	
	Value: An array with an IP address block in CIDR format. For example:
	
	`networking:`
	`  serviceNetwork:`
	`  - 172.30.0.0/16`

`networking:`
`  machineNetwork:`
	The IP address blocks for machines.
	
	If you specify multiple IP address blocks, the blocks must not overlap.
	
	Value: An array of objects. For example:
	
	`networking:`
	`  machineNetwork:`
	`  - cidr: 10.0.0.0/16`

Parameter	Description
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 and IBM Power® Virtual Server. For libvirt, the default value is **192.168.126.0/24**. For IBM Power® Virtual Server, the default value is **192.168.0.0/24**. If you are deploying the cluster to an existing Virtual Private Cloud (VPC), the CIDR must contain the subnets defined in **platform.ibmcloud.controlPlaneSubnets** and **platform.ibmcloud.computeSubnets**. **Value:** 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.
networking: ovnKubernetesConfig: ipv4: internalJoinSubnet:	Configures the IPv4 join subnet that is used internally by **ovn-kubernetes**. This subnet must not overlap with any other subnet that OpenShift Container Platform is using, including the node network. The size of the subnet must be larger than the number of nodes. You cannot change the value after installation. **Value:** An IP network block in CIDR notation. The default value is **100.64.0.0/16**.

9.1.3. Optional configuration parameters

Optional installation configuration parameters are described in the following table:

Table 9.3. Optional parameters

Parameter	Description

Parameter	Description
additionalTrustBundle:	A PEM-encoded X.509 certificate bundle that is added to the nodes' trusted certificate store. This trust bundle might also be used when a proxy has been configured. **Value:** String
capabilities:	Controls the installation of optional core cluster components. You can reduce the footprint of your OpenShift Container Platform cluster by disabling optional components. For more information, see the "Cluster capabilities" page in *Installing*. **Value:** String array
capabilities: baselineCapabilitySet:	Selects an initial set of optional capabilities to enable. Valid values are **None**, **v4.11**, **v4.12** and **vCurrent**. The default value is **vCurrent**. **Value:** String
capabilities: additionalEnabledCapabilities:	Extends the set of optional capabilities beyond what you specify in **baselineCapabilitySet**. You can specify multiple capabilities in this parameter. **Value:** String array
cpuPartitioningMode:	Enables workload partitioning, which isolates OpenShift Container Platform services, cluster management workloads, and infrastructure pods to run on a reserved set of CPUs. You can only enable workload partitioning during installation. You cannot disable it after installation. While this field enables workload partitioning, it does not configure workloads to use specific CPUs. For more information, see the *Workload partitioning* page in the *Scalability and Performance* section. **Value:** **None** or **AllNodes**. **None** is the default value.
compute:	The configuration for the machines that comprise the compute nodes. **Value:** Array of **MachinePool** objects.

Parameter	Description
compute: architecture:	Determines the instruction set architecture of the machines in the pool. Currently, clusters with varied architectures are not supported. All pools must specify the same architecture. Valid values are **amd64** (the default). **Value:** String
compute: hyperthreading:	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. Value: **Enabled** or **Disabled**
compute: name:	Required if you use **compute**. The name of the machine pool. Value: **worker**
compute: platform:	Required if you use **compute**. Use this parameter to specify the cloud provider to host the worker machines. This parameter value must match the **controlPlane.platform** parameter value. Value:**aws**, **azure**, **gcp**, **ibmcloud**, **nutanix**, **openstack**, **powervs**, **vsphere**, or **{}**
compute: replicas:	The number of compute machines, which are also known as worker machines, to provision. **Value:** A positive integer greater than or equal to **2**. The default value is **3**.

Parameter	Description
featureSet:	Enables the cluster for a feature set. A feature set is a collection of OpenShift Container Platform features that are not enabled by default. For more information about enabling a feature set during installation, see "Enabling features using feature gates". **Value:** String. The name of the feature set to enable, such as **TechPreviewNoUpgrade**.
controlPlane:	The configuration for the machines that form the control plane. **Value:** Array of **MachinePool** objects.
controlPlane: architecture:	Determines the instruction set architecture of the machines in the pool. Currently, clusters with varied architectures are not supported. All pools must specify the same architecture. Valid values are **amd64** (the default). **Value:** String
controlPlane: hyperthreading:	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.  IMPORTANT If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance. **Value:** **Enabled** or **Disabled**
controlPlane: name:	Required if you use **controlPlane**. The name of the machine pool. **Value:** **master**

Parameter	Description
controlPlane: platform:	Required if you use **controlPlane**. Use this parameter to specify the cloud provider that hosts the control plane machines. This parameter value must match the **compute.platform** parameter value. Value:**aws**, **azure**, **gcp**, **ibmcloud**, **nutanix**, **openstack**, **powervs**, **vsphere**, or **{}**
controlPlane: replicas:	The number of control plane machines to provision. **Value:** Supported values are **3**, or **1** when deploying single-node OpenShift.
arbiter: name: arbiter	The OpenShift Container Platform cluster requires a name for arbiter nodes. For example, **arbiter**.
arbiter: replicas: 1	The **replicas** parameter sets the number of arbiter nodes for the OpenShift Container Platform cluster. You cannot set this field to a value that is greater than 1.
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 about CCO modes, see the "Managing cloud provider credentials" entry in the *Authentication and authorization* content. Value: **Mint**, **Passthrough**, **Manual** or an empty string (**""**).

Parameter	Description
fips:	Enable or disable FIPS mode. The default is **false** (disabled). If you enable FIPS mode, 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 RHCOS provides instead.  **IMPORTANT** To enable FIPS mode for your cluster, you must run the installation program from a Red Hat Enterprise Linux (RHEL) computer configured to operate in FIPS mode. For more information about configuring FIPS mode on RHEL, see Switching RHEL to FIPS mode. When running Red Hat Enterprise Linux (RHEL) or Red Hat Enterprise Linux CoreOS (RHCOS) booted in FIPS mode, OpenShift Container Platform core components use the RHEL cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 Validation on only the x86_64, ppc64le, and s390x architectures.  **IMPORTANT** If you are using Azure File storage, you cannot enable FIPS mode. **Value:** **false** or **true**
imageContentSources:	Sources and repositories for the release-image content. **Value:** 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. **Value:** String
imageContentSources: mirrors:	Specify one or more repositories that might also contain the same images. **Value:** Array of strings

Parameter	Description
publish:	How to publish or expose the user-facing endpoints of your cluster, such as the Kubernetes API, OpenShift routes. Value:**Internal** or **External**. To deploy a private cluster that cannot be accessed from the internet, set the **publish** parameter to **Internal**. The default value is **External**.
sshKey:	The SSH key to authenticate access to 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. **Value:** For example, **sshKey: ssh-ed25519 AAAA..**.

9.1.4. Additional IBM Cloud configuration parameters

Additional IBM Cloud® configuration parameters are described in the following table:

Table 9.4. Additional IBM Cloud(R) parameters

Parameter	Description

controlPlane:
  platform:
  ibmcloud:
  bootVolume:
  encryptionKey:
	An IBM® Key Protect for IBM Cloud® (Key Protect) root key that should be used to encrypt the root (boot) volume of only control plane machines.
	
	Value: The Cloud Resource Name (CRN) of the root key.
	
	The CRN must be enclosed in quotes ("").

compute:
  platform:
  ibmcloud:
  bootVolume:
  encryptionKey:
	A Key Protect root key that should be used to encrypt the root (boot) volume of only compute machines.
	
	Value: The CRN of the root key.
	
	The CRN must be enclosed in quotes ("").

Parameter	Description

`platform:`
`  ibmcloud:`
`  defaultMachinePlatform:`
`  bootvolume:`
`  encryptionKey:`
	A Key Protect root key that should be used to encrypt the root (boot) volume of all of the cluster's machines.
	
	When specified as part of the default machine configuration, all managed storage classes are updated with this key. Data volumes that are provisioned after the installation are also encrypted using this key.
	
	Value: The CRN of the root key.
	
	The CRN must be enclosed in quotes ("").

`platform:`
`  ibmcloud:`
`  resourceGroupName:`
	The name of an existing resource group. By default, an installer-provisioned VPC and cluster resources are created and placed in this resource group. The installation program creates the resource group for the cluster if you do not specify these parameters.
	
	If you are deploying the cluster into an existing VPC, the installation-program-provisioned cluster resources are placed in this resource group. The installation program creates the resource group for the cluster if you do not specify these parameters. The VPC resources that you have provisioned must exist in a resource group that you specify using the **networkResourceGroupName** parameter.
	
	In either case, this resource group must only be used for a single cluster installation, as the cluster components assume ownership of all of the resources in the resource group. [1]
	
	Value: String, for example **existing_resource_group**.

Parameter	Description

`platform:`
`  ibmcloud:`
`  serviceEndpoints:`
`  - name:`
`  url:`
	A list of service endpoint names and URIs.
	
	By default, the installation program and cluster components use public service endpoints to access the required IBM Cloud® services.
	
	If network restrictions limit access to public service endpoints, you can specify an alternate service endpoint to override the default behavior.
	
	You can specify only one alternate service endpoint for each of the following services:
	
	● Cloud Object Storage
	● DNS Services
	● Global Search
	● Global Tagging
	● Identity Services
	● Key Protect
	● Resource Controller
	● Resource Manager
	● VPC
	
	Value: A valid service endpoint name and fully qualified URI.
	
	Valid names include:
	
	● **COS**
	● **DNSServices**
	● **GlobalServices**
	● **GlobalTagging**
	● **IAM**
	● **KeyProtect**
	● **ResourceController**
	● **ResourceManager**
	● **VPC**

Parameter	Description

`platform:`
`  ibmcloud:`
`  networkResourceGroupName:`
	The name of an existing resource group. This resource contains the existing VPC and subnets to which the cluster is deployed. This parameter is required when deploying the cluster to a VPC that you have provisioned.
	
	Value: String, for example **existing_network_resource_group**.

`platform:`
`  ibmcloud:`
`  dedicatedHosts:`
`  profile:`
	The new dedicated host to create. If you specify a value for **platform.ibmcloud.dedicatedHosts.name**, this parameter is not required.
	
	Value: Valid IBM Cloud® dedicated host profile, such as **cx2-host-152x304**. [2]

`platform:`
`  ibmcloud:`
`  dedicatedHosts:`
`  name:`
	An existing dedicated host. If you specify a value for **platform.ibmcloud.dedicatedHosts.profile**, this parameter is not required.
	
	Value: String, for example **my-dedicated-host-name**.

`platform:`
`  ibmcloud:`
`  type:`
	The instance type for all IBM Cloud® machines.
	
	Value: Valid IBM Cloud® instance type, such as **bx2-8x32**. [2]

`platform:`
`  ibmcloud:`
`  vpcName:`
	The name of the existing VPC that you want to deploy your cluster to.
	
	Value: String.

`platform:`
`  ibmcloud:`
`  controlPlaneSubnets:`
	The name(s) of the existing subnet(s) in your VPC that you want to deploy your control plane machines to. Specify a subnet for each availability zone.
	
	Value: String array

`platform:`
`  ibmcloud:`
`  computeSubnets:`
	The name(s) of the existing subnet(s) in your VPC that you want to deploy your compute machines to. Specify a subnet for each availability zone. Subnet IDs are not supported.
	
	Value: String array

1. Whether you define an existing resource group, or if the installation program creates one, determines how the resource group is treated when the cluster is uninstalled. If you define a resource group, the installation program removes all of the installer-provisioned resources, but leaves the resource group alone; if a resource group is created as part of the installation, the installation program removes all of the installer-provisioned resources and the resource group.

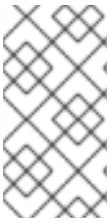
2. To determine which profile best meets your needs, see Instance Profiles in the IBM® documentation.

CHAPTER 10. UNINSTALLING A CLUSTER ON IBM CLOUD

You can remove a cluster that you deployed to IBM Cloud®.

10.1. REMOVING A CLUSTER THAT USES INSTALLER-PROVISIONED INFRASTRUCTURE

You can remove a cluster that uses installer-provisioned infrastructure that you provisioned from your cloud platform.



NOTE

After uninstallation, check your cloud provider for any resources that were not removed properly, especially with user-provisioned infrastructure clusters. Some resources might exist because either the installation program did not create the resource or could not access the resource.

Prerequisites

- You have a copy of the installation program that you used to deploy the cluster.
- You have the files that the installation program generated when you created your cluster.
- You have configured the **ccoctl** binary.
- You have installed the IBM Cloud® CLI and installed or updated the VPC infrastructure service plugin. For more information see "Prerequisites" in the IBM Cloud® CLI documentation.

Procedure

1. If the following conditions are met, this step is required:
   - The installer created a resource group as part of the installation process.
   - You or one of your applications created persistent volume claims (PVCs) after the cluster was deployed.
     In which case, the PVCs are not removed when uninstalling the cluster, which might prevent the resource group from being successfully removed. To prevent a failure:
     a. Log in to the IBM Cloud® using the CLI.
     b. To list the PVCs, run the following command:

     ```
     $ ibmcloud is volumes --resource-group-name <infrastructure_id>
     ```

     For more information about listing volumes, see the IBM Cloud® CLI documentation.
     c. To delete the PVCs, run the following command:

     ```
     $ ibmcloud is volume-delete --force <volume_id>
     ```

     For more information about deleting volumes, see the IBM Cloud® CLI documentation.
2. Export the API key that was created as part of the installation process.

```
$ export IC_API_KEY=<api_key>
```

NOTE

You must set the variable name exactly as specified. The installation program expects the variable name to be present to remove the service IDs that were created when the cluster was installed.

3. From the directory that has the installation program on the computer that you used to install the cluster, run the following command:

```
$ ./openshift-install destroy cluster \
--dir <installation_directory> --log-level info
```

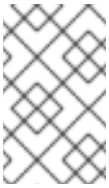
where:

<installation_directory>

Specify the path to the directory that you stored the installation files in.

--log-level info

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



NOTE

You must specify the directory that includes the cluster definition files for your cluster. The installation program requires the **metadata.json** file in this directory to delete the cluster.

4. Remove the manual CCO credentials that were created for the cluster:

```
$ ccoctl ibmcloud delete-service-id \
   --credentials-requests-dir <path_to_credential_requests_directory> \
   --name <cluster_name>
```

NOTE

If your cluster uses Technology Preview features that are enabled by the **TechPreviewNoUpgrade** feature set, you must include the **--enable-tech-preview** parameter.

5. Optional: Delete the **<installation_directory>** directory and the OpenShift Container Platform installation program.