



# OpenShift Container Platform 4.5

## Installing on Azure

Installing OpenShift Container Platform Azure clusters



# OpenShift Container Platform 4.5 Installing on Azure

---

Installing OpenShift Container Platform Azure clusters

## Legal Notice

Copyright © 2021 Red Hat, Inc.

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

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

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

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

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

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

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

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

All other trademarks are the property of their respective owners.

## Abstract

This document provides instructions for installing and uninstalling OpenShift Container Platform clusters on Microsoft Azure.

# Table of Contents

<b>CHAPTER 1. INSTALLING ON AZURE .....</b>	<b>6</b>
1.1. CONFIGURING AN AZURE ACCOUNT	6
1.1.1. Azure account limits	6
1.1.2. Configuring a public DNS zone in Azure	8
1.1.3. Increasing Azure account limits	9
1.1.4. Required Azure roles	10
1.1.5. Creating a service principal	10
1.1.6. Supported Azure regions	13
1.1.7. Next steps	14
1.2. INSTALLING A CLUSTER QUICKLY ON AZURE	14
1.2.1. Prerequisites	14
1.2.2. Internet and Telemetry access for OpenShift Container Platform	14
1.2.3. Generating an SSH private key and adding it to the agent	15
1.2.4. Obtaining the installation program	16
1.2.5. Deploying the cluster	17
1.2.6. Installing the CLI by downloading the binary	19
1.2.6.1. Installing the CLI on Linux	19
1.2.6.2. Installing the CLI on Windows	20
1.2.6.3. Installing the CLI on macOS	20
1.2.7. Logging in to the cluster	21
1.2.8. Next steps	21
1.3. INSTALLING A CLUSTER ON AZURE WITH CUSTOMIZATIONS	21
1.3.1. Prerequisites	21
1.3.2. Internet and Telemetry access for OpenShift Container Platform	22
1.3.3. Generating an SSH private key and adding it to the agent	22
1.3.4. Obtaining the installation program	24
1.3.5. Creating the installation configuration file	24
1.3.5.1. Installation configuration parameters	26
1.3.5.1.1. Required configuration parameters	26
1.3.5.1.2. Network configuration parameters	28
1.3.5.1.3. Optional configuration parameters	29
1.3.5.1.4. Additional Azure configuration parameters	33
1.3.5.2. Sample customized install-config.yaml file for Azure	34
1.3.6. Deploying the cluster	36
1.3.7. Installing the CLI by downloading the binary	37
1.3.7.1. Installing the CLI on Linux	37
1.3.7.2. Installing the CLI on Windows	37
1.3.7.3. Installing the CLI on macOS	38
1.3.8. Logging in to the cluster	38
1.3.9. Next steps	39
1.4. INSTALLING A CLUSTER ON AZURE WITH NETWORK CUSTOMIZATIONS	39
1.4.1. Prerequisites	39
1.4.2. Internet and Telemetry access for OpenShift Container Platform	40
1.4.3. Generating an SSH private key and adding it to the agent	40
1.4.4. Obtaining the installation program	41
1.4.5. Creating the installation configuration file	42
1.4.5.1. Installation configuration parameters	44
1.4.5.1.1. Required configuration parameters	44
1.4.5.1.2. Network configuration parameters	45
1.4.5.1.3. Optional configuration parameters	47
1.4.5.1.4. Additional Azure configuration parameters	50

1.4.5.2. Network configuration parameters	51
1.4.5.3. Sample customized install-config.yaml file for Azure	52
1.4.6. Modifying advanced network configuration parameters	54
1.4.7. Cluster Network Operator configuration	56
1.4.7.1. Configuration parameters for the OpenShift SDN default CNI network provider	57
1.4.7.2. Configuration parameters for the OVN-Kubernetes default CNI network provider	57
1.4.7.3. Cluster Network Operator example configuration	58
1.4.8. Deploying the cluster	59
1.4.9. Installing the CLI by downloading the binary	60
1.4.9.1. Installing the CLI on Linux	60
1.4.9.2. Installing the CLI on Windows	60
1.4.9.3. Installing the CLI on macOS	61
1.4.10. Logging in to the cluster	61
1.4.11. Next steps	62
1.5. INSTALLING A CLUSTER ON AZURE INTO AN EXISTING VNET	62
1.5.1. Prerequisites	62
1.5.2. About reusing a VNet for your OpenShift Container Platform cluster	62
1.5.2.1. Requirements for using your VNet	62
1.5.2.1.1. Network security group requirements	63
1.5.2.2. Division of permissions	64
1.5.2.3. Isolation between clusters	64
1.5.3. Internet and Telemetry access for OpenShift Container Platform	64
1.5.4. Generating an SSH private key and adding it to the agent	65
1.5.5. Obtaining the installation program	66
1.5.6. Creating the installation configuration file	67
1.5.6.1. Installation configuration parameters	69
1.5.6.1.1. Required configuration parameters	69
1.5.6.1.2. Network configuration parameters	70
1.5.6.1.3. Optional configuration parameters	72
1.5.6.1.4. Additional Azure configuration parameters	76
1.5.6.2. Sample customized install-config.yaml file for Azure	76
1.5.6.3. Configuring the cluster-wide proxy during installation	78
1.5.7. Deploying the cluster	80
1.5.8. Installing the CLI by downloading the binary	81
1.5.8.1. Installing the CLI on Linux	81
1.5.8.2. Installing the CLI on Windows	82
1.5.8.3. Installing the CLI on macOS	82
1.5.9. Logging in to the cluster	83
1.5.10. Next steps	83
1.6. INSTALLING A PRIVATE CLUSTER ON AZURE	83
1.6.1. Prerequisites	83
1.6.2. Private clusters	84
1.6.2.1. Private clusters in Azure	84
1.6.2.1.1. Limitations	84
1.6.3. About reusing a VNet for your OpenShift Container Platform cluster	85
1.6.3.1. Requirements for using your VNet	85
1.6.3.1.1. Network security group requirements	86
1.6.3.2. Division of permissions	86
1.6.3.3. Isolation between clusters	87
1.6.4. Internet and Telemetry access for OpenShift Container Platform	87
1.6.5. Generating an SSH private key and adding it to the agent	87
1.6.6. Obtaining the installation program	89
1.6.7. Manually creating the installation configuration file	89

1.6.7.1. Installation configuration parameters	90
1.6.7.1.1. Required configuration parameters	91
1.6.7.1.2. Network configuration parameters	92
1.6.7.1.3. Optional configuration parameters	93
1.6.7.1.4. Additional Azure configuration parameters	97
1.6.7.2. Sample customized install-config.yaml file for Azure	98
1.6.7.3. Configuring the cluster-wide proxy during installation	100
1.6.8. Deploying the cluster	101
1.6.9. Installing the CLI by downloading the binary	102
1.6.9.1. Installing the CLI on Linux	103
1.6.9.2. Installing the CLI on Windows	103
1.6.9.3. Installing the CLI on macOS	104
1.6.10. Logging in to the cluster	104
1.6.11. Next steps	105
1.7. INSTALLING A CLUSTER ON AZURE USING ARM TEMPLATES	105
1.7.1. Prerequisites	105
1.7.2. Internet and Telemetry access for OpenShift Container Platform	105
1.7.3. Configuring your Azure project	106
1.7.3.1. Azure account limits	106
1.7.3.2. Configuring a public DNS zone in Azure	108
1.7.3.3. Increasing Azure account limits	109
1.7.3.4. Certificate signing requests management	110
1.7.3.5. Required Azure roles	110
1.7.3.6. Creating a service principal	110
1.7.3.7. Supported Azure regions	113
1.7.4. Obtaining the installation program	114
1.7.5. Generating an SSH private key and adding it to the agent	115
1.7.6. Creating the installation files for Azure	116
1.7.6.1. Creating the installation configuration file	116
1.7.6.2. Configuring the cluster-wide proxy during installation	118
1.7.6.3. Exporting common variables for ARM templates	119
1.7.6.4. Creating the Kubernetes manifest and Ignition config files	121
1.7.7. Creating the Azure resource group and identity	123
1.7.8. Uploading the RHCOS cluster image and bootstrap Ignition config file	124
1.7.9. Example for creating DNS zones	125
1.7.10. Creating a VNet in Azure	126
1.7.10.1. ARM template for the VNet	127
1.7.11. Deploying the RHCOS cluster image for the Azure infrastructure	128
1.7.11.1. ARM template for image storage	129
1.7.12. Networking requirements for user-provisioned infrastructure	130
Network topology requirements	131
Load balancers	131
1.7.13. Creating networking and load balancing components in Azure	133
1.7.13.1. ARM template for the network and load balancers	134
1.7.14. Creating the bootstrap machine in Azure	139
1.7.14.1. ARM template for the bootstrap machine	139
1.7.15. Creating the control plane machines in Azure	144
1.7.15.1. ARM template for control plane machines	145
1.7.16. Wait for bootstrap completion and remove bootstrap resources in Azure	151
1.7.17. Creating additional worker machines in Azure	152
1.7.17.1. ARM template for worker machines	153
1.7.18. Installing the CLI by downloading the binary	157
1.7.18.1. Installing the CLI on Linux	157

1.7.18.2. Installing the CLI on Windows	158
1.7.18.3. Installing the CLI on macOS	158
1.7.19. Logging in to the cluster	159
1.7.20. Approving the certificate signing requests for your machines	159
1.7.21. Adding the Ingress DNS records	162
1.7.22. Completing an Azure installation on user-provisioned infrastructure	163
1.8. UNINSTALLING A CLUSTER ON AZURE	164
1.8.1. Removing a cluster that uses installer-provisioned infrastructure	164

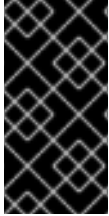




# CHAPTER 1. INSTALLING ON AZURE

## 1.1. CONFIGURING AN AZURE ACCOUNT

Before you can install OpenShift Container Platform, you must configure a Microsoft Azure account.

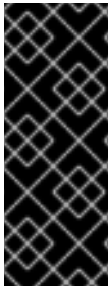


### IMPORTANT

All Azure resources that are available through public endpoints are subject to resource name restrictions, and you cannot create resources that use certain terms. For a list of terms that Azure restricts, see [Resolve reserved resource name errors](#) in the Azure documentation.

### 1.1.1. Azure account limits

The OpenShift Container Platform cluster uses a number of Microsoft Azure components, and the default [Azure subscription and service limits, quotas, and constraints](#) affect your ability to install OpenShift Container Platform clusters.



### IMPORTANT

Default limits vary by offer category types, such as Free Trial and Pay-As-You-Go, and by series, such as Dv2, F, and G. For example, the default for Enterprise Agreement subscriptions is 350 cores.

Check the limits for your subscription type and if necessary, increase quota limits for your account before you install a default cluster on Azure.

The following table summarizes the Azure components whose limits can impact your ability to install and run OpenShift Container Platform clusters.

Component	Number of components required by default	Default Azure limit	Description

Component	Number of components required by default	Default Azure limit	Description
vCPU	40	20 per region	<p>A default cluster requires 40 vCPUs, so you must increase the account limit.</p> <p>By default, each cluster creates the following instances:</p> <ul style="list-style-type: none"> <li>• One bootstrap machine, which is removed after installation</li> <li>• Three control plane machines</li> <li>• Three compute machines</li> </ul> <p>Because the bootstrap machine uses <b>Standard_D4s_v3</b> machines, which use 4 vCPUs, the control plane machines use <b>Standard_D8s_v3</b> virtual machines, which use 8 vCPUs, and the worker machines use <b>Standard_D4s_v3</b> virtual machines, which use 4 vCPUs, a default cluster requires 40 vCPUs. The bootstrap node VM, which uses 4 vCPUs, is used only during installation.</p> <p>To deploy more worker nodes, enable autoscaling, deploy large workloads, or use a different instance type, you must further increase the vCPU limit for your account to ensure that your cluster can deploy the machines that you require.</p> <p>By default, the installation program distributes control plane and compute machines across <a href="#">all availability zones</a> within a <a href="#">region</a>. To ensure high availability for your cluster, select a region with at least three availability zones. If your region contains fewer than three availability zones, the installation program places more than one control plane machine in the available zones.</p>
VNet	1	1000 per region	Each default cluster requires one Virtual Network (VNet), which contains two subnets.
Network interfaces	6	65,536 per region	Each default cluster requires six network interfaces. If you create more machines or your deployed workloads create load balancers, your cluster uses more network interfaces.

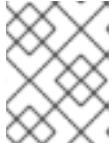
Component	Number of components required by default	Default Azure limit	Description						
Network security groups	2	5000	<p>Each default cluster creates network security groups for each subnet in the VNet. The default cluster creates network security groups for the control plane and for the compute node subnets:</p> <table border="1"> <tr> <td><b>control plane</b></td> <td>Allows the control plane machines to be reached on port 6443 from anywhere</td> </tr> <tr> <td><b>node</b></td> <td>Allows worker nodes to be reached from the Internet on ports 80 and 443</td> </tr> </table>	<b>control plane</b>	Allows the control plane machines to be reached on port 6443 from anywhere	<b>node</b>	Allows worker nodes to be reached from the Internet on ports 80 and 443		
<b>control plane</b>	Allows the control plane machines to be reached on port 6443 from anywhere								
<b>node</b>	Allows worker nodes to be reached from the Internet on ports 80 and 443								
Network load balancers	3	1000 per region	<p>Each cluster creates the following <a href="#">load balancers</a>:</p> <table border="1"> <tr> <td><b>default</b></td> <td>Public IP address that load balances requests to ports 80 and 443 across worker machines</td> </tr> <tr> <td><b>internal</b></td> <td>Private IP address that load balances requests to ports 6443 and 22623 across control plane machines</td> </tr> <tr> <td><b>external</b></td> <td>Public IP address that load balances requests to port 6443 across control plane machines</td> </tr> </table> <p>If your applications create more Kubernetes <b>LoadBalancer</b> service objects, your cluster uses more load balancers.</p>	<b>default</b>	Public IP address that load balances requests to ports 80 and 443 across worker machines	<b>internal</b>	Private IP address that load balances requests to ports 6443 and 22623 across control plane machines	<b>external</b>	Public IP address that load balances requests to port 6443 across control plane machines
<b>default</b>	Public IP address that load balances requests to ports 80 and 443 across worker machines								
<b>internal</b>	Private IP address that load balances requests to ports 6443 and 22623 across control plane machines								
<b>external</b>	Public IP address that load balances requests to port 6443 across control plane machines								
Public IP addresses	3		Each of the two public load balancers uses a public IP address. The bootstrap machine also uses a public IP address so that you can SSH into the machine to troubleshoot issues during installation. The IP address for the bootstrap node is used only during installation.						
Private IP addresses	7		The internal load balancer, each of the three control plane machines, and each of the three worker machines each use a private IP address.						

### 1.1.2. Configuring a public DNS zone in Azure

To install OpenShift Container Platform, the Microsoft Azure account you use must have a dedicated public hosted DNS zone in your account. This zone must be authoritative for the domain. This service provides cluster DNS resolution and name lookup for external connections to the cluster.

## Procedure

1. Identify your domain, or subdomain, and registrar. You can transfer an existing domain and registrar or obtain a new one through Azure or another source.



### NOTE

For more information about purchasing domains through Azure, see [Buy a custom domain name for Azure App Service](#) in the Azure documentation.

2. If you are using an existing domain and registrar, migrate its DNS to Azure. See [Migrate an active DNS name to Azure App Service](#) in the Azure documentation.
3. Configure DNS for your domain. Follow the steps in the [Tutorial: Host your domain in Azure DNS](#) in the Azure documentation to create a public hosted zone for your domain or subdomain, extract the new authoritative name servers, and update the registrar records for the name servers that your domain uses.  
Use an appropriate root domain, such as **openshiftcorp.com**, or subdomain, such as **clusters.openshiftcorp.com**.
4. If you use a subdomain, follow your company's procedures to add its delegation records to the parent domain.

## 1.1.3. Increasing Azure account limits

To increase an account limit, file a support request on the Azure portal.



### NOTE

You can increase only one type of quota per support request.

## Procedure

1. From the Azure portal, click **Help + support** in the lower left corner.
2. Click **New support request** and then select the required values:
  - a. From the **Issue type** list, select **Service and subscription limits (quotas)**
  - b. From the **Subscription** list, select the subscription to modify.
  - c. From the **Quota type** list, select the quota to increase. For example, select **Compute-VM (cores-vCPUs) subscription limit increases** to increase the number of vCPUs, which is required to install a cluster.
  - d. Click **Next: Solutions**.
3. On the **Problem Details** page, provide the required information for your quota increase:
  - a. Click **Provide details** and provide the required details in the **Quota details** window.

- b. In the SUPPORT METHOD and CONTACT INFO sections, provide the issue severity and your contact details.
4. Click **Next: Review + create** and then click **Create**.

### 1.1.4. Required Azure roles

Your Microsoft Azure account must have the following roles for the subscription that you use:

- **User Access Administrator**

To set roles on the Azure portal, see the [Manage access to Azure resources using RBAC and the Azure portal](#) in the Azure documentation.

### 1.1.5. Creating a service principal

Because OpenShift Container Platform and its installation program must create Microsoft Azure resources through Azure Resource Manager, you must create a service principal to represent it.

#### Prerequisites

- Install or update the [Azure CLI](#).
- Install the **jq** package.
- Your Azure account has the required roles for the subscription that you use.

#### Procedure

1. Log in to the Azure CLI:

```
$ az login
```

Log in to Azure in the web console by using your credentials.

2. If your Azure account uses subscriptions, ensure that you are using the right subscription.
  - a. View the list of available accounts and record the **tenantId** value for the subscription you want to use for your cluster:

```
$ az account list --refresh
```

#### Example output

```
[
  {
    "cloudName": "AzureCloud",
    "id": "9bab1460-96d5-40b3-a78e-17b15e978a80",
    "isDefault": true,
    "name": "Subscription Name",
    "state": "Enabled",
    "tenantId": "6057c7e9-b3ae-489d-a54e-de3f6bf6a8ee",
    "user": {
      "name": "you@example.com",
      "type": "user"
    }
  }
]
```

```

    }
  }
]

```

- b. View your active account details and confirm that the **tenantId** value matches the subscription you want to use:

```
$ az account show
```

### Example output

```

{
  "environmentName": "AzureCloud",
  "id": "9bab1460-96d5-40b3-a78e-17b15e978a80",
  "isDefault": true,
  "name": "Subscription Name",
  "state": "Enabled",
  "tenantId": "6057c7e9-b3ae-489d-a54e-de3f6bf6a8ee", ❶
  "user": {
    "name": "you@example.com",
    "type": "user"
  }
}

```

- ❶ Ensure that the value of the **tenantId** parameter is the UUID of the correct subscription.

- c. If you are not using the right subscription, change the active subscription:

```
$ az account set -s <id> ❶
```

- ❶ Substitute the value of the **id** for the subscription that you want to use for **<id>**.

- d. If you changed the active subscription, display your account information again:

```
$ az account show
```

### Example output

```

{
  "environmentName": "AzureCloud",
  "id": "33212d16-bdf6-45cb-b038-f6565b61edda",
  "isDefault": true,
  "name": "Subscription Name",
  "state": "Enabled",
  "tenantId": "8049c7e9-c3de-762d-a54e-dc3f6be6a7ee",
  "user": {
    "name": "you@example.com",
    "type": "user"
  }
}

```

3. Record the values of the **tenantId** and **id** parameters from the previous output. You need these values during OpenShift Container Platform installation.
4. Create the service principal for your account:

```
$ az ad sp create-for-rbac --role Contributor --name <service_principal> ❶
```

- ❶ Replace **<service\_principal>** with the name to assign to the service principal.

### Example output

```
Changing "<service_principal>" to a valid URI of "http://<service_principal>", which is the
required format used for service principal names
Retrying role assignment creation: 1/36
Retrying role assignment creation: 2/36
Retrying role assignment creation: 3/36
Retrying role assignment creation: 4/36
{
  "appId": "8bd0d04d-0ac2-43a8-928d-705c598c6956",
  "displayName": "<service_principal>",
  "name": "http://<service_principal>",
  "password": "ac461d78-bf4b-4387-ad16-7e32e328aec6",
  "tenant": "6048c7e9-b2ad-488d-a54e-dc3f6be6a7ee"
}
```

5. Record the values of the **appId** and **password** parameters from the previous output. You need these values during OpenShift Container Platform installation.
6. Grant additional permissions to the service principal. The service principal requires the legacy **Azure Active Directory Graph** → **Application.ReadWrite.OwnedBy** permission and the **User Access Administrator** role for the cluster to assign credentials for its components.
  - a. To assign the **User Access Administrator** role, run the following command:

```
$ az role assignment create --role "User Access Administrator" \
  --assignee-object-id $(az ad sp list --filter "appId eq '<appId>'" \
  | jq '[0].objectId' -r) ❶
```

- ❶ Replace **<appId>** with the **appId** parameter value for your service principal.

- b. To assign the **Azure Active Directory Graph** permission, run the following command:

```
$ az ad app permission add --id <appId> \ ❶
  --api 00000002-0000-0000-c000-000000000000 \
  --api-permissions 824c81eb-e3f8-4ee6-8f6d-de7f50d565b7=Role
```

- ❶ Replace **<appId>** with the **appId** parameter value for your service principal.

### Example output

```
Invoking "az ad app permission grant --id 46d33abc-b8a3-46d8-8c84-f0fd58177435 --api
00000002-0000-0000-c000-000000000000" is needed to make the change effective
```



■

For more information about the specific permissions that you grant with this command, see the [GUID Table for Windows Azure Active Directory Permissions](#) .

- c. Approve the permissions request. If your account does not have the Azure Active Directory tenant administrator role, follow the guidelines for your organization to request that the tenant administrator approve your permissions request.

```
$ az ad app permission grant --id <appld> \ 1
--api 00000002-0000-0000-c000-000000000000
```

- 1 Replace **<appld>** with the **appld** parameter value for your service principal.

### 1.1.6. Supported Azure regions

The installation program dynamically generates the list of available Microsoft Azure regions based on your subscription. The following Azure regions were tested and validated in OpenShift Container Platform version 4.5.4:

- **australiacentral** (Australia Central)
- **australiaeast** (Australia East)
- **australiasoutheast** (Australia South East)
- **brazilsouth** (Brazil South)
- **canadacentral** (Canada Central)
- **canadaeast** (Canada East)
- **centralindia** (Central India)
- **centralus** (Central US)
- **eastasia** (East Asia)
- **eastus** (East US)
- **eastus2** (East US 2)
- **francecentral** (France Central)
- **germanywestcentral** (Germany West Central)
- **japaneast** (Japan East)
- **japanwest** (Japan West)
- **koreacentral** (Korea Central)
- **koreasouth** (Korea South)
- **northcentralus** (North Central US)
- **northeurope** (North Europe)

- **norwayeast** (Norway East)
- **southafricanorth** (South Africa North)
- **southcentralus** (South Central US)
- **southeastasia** (Southeast Asia)
- **southindia** (South India)
- **switzerlandnorth** (Switzerland North)
- **uaenorth** (UAE North)
- **uksouth** (UK South)
- **ukwest** (UK West)
- **westcentralus** (West Central US)
- **westeurope** (West Europe)
- **westindia** (West India)
- **westus** (West US)
- **westus2** (West US 2)

### 1.1.7. Next steps

- Install an OpenShift Container Platform cluster on Azure. You can [install a customized cluster](#) or [quickly install a cluster](#) with default options.

## 1.2. INSTALLING A CLUSTER QUICKLY ON AZURE

In OpenShift Container Platform version 4.5, you can install a cluster on Microsoft Azure that uses the default configuration options.

### 1.2.1. Prerequisites

- Review details about the [OpenShift Container Platform installation and update](#) processes.
- [Configure an Azure account](#) to host the cluster and determine the tested and validated region to deploy the cluster to.
- If you use a firewall, you must [configure it to allow the sites](#) that your cluster requires access to.
- If you do not allow the system to manage identity and access management (IAM), then a cluster administrator can [manually create and maintain IAM credentials](#). Manual mode can also be used in environments where the cloud IAM APIs are not reachable.

### 1.2.2. Internet and Telemetry access for OpenShift Container Platform

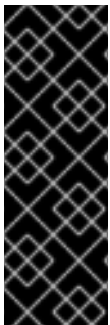
In OpenShift Container Platform 4.5, you require access to the Internet to install your cluster. The Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, also requires Internet access. If your cluster is connected to the Internet, Telemetry runs

automatically, and your cluster is registered to the [Red Hat OpenShift Cluster Manager \(OCM\)](#).

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

You must have Internet access to:

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



### IMPORTANT

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

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

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



### NOTE

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

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



### NOTE

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

### Procedure

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

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

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

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



#### NOTE

If you plan to install an OpenShift Container Platform cluster that uses FIPS Validated / Modules in Process cryptographic libraries on the **x86\_64** architecture, do not create a key that uses the **ed25519** algorithm. Instead, create a key that uses the **rsa** or **ecdsa** algorithm.

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

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

#### Example output

```
Agent pid 31874
```

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.

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

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

#### Example output

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

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

#### Next steps

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

### 1.2.4. Obtaining the installation program

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

#### Prerequisites

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

#### Procedure

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



### IMPORTANT

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



### IMPORTANT

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

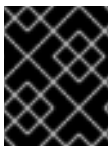
3. Extract the installation program. For example, on a computer that uses a Linux operating system, run the following command:

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

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

## 1.2.5. 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

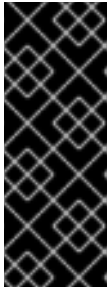
- Configure an account with the cloud platform that hosts your cluster.
- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.

### Procedure

1. Run the installation program:

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

- 1 For **<installation\_directory>**, specify the directory name to store the files that the installation program creates.
- 2 To view different installation details, specify **warn**, **debug**, or **error** instead of **info**.



### IMPORTANT

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

Provide values at the prompts:

- a. 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.

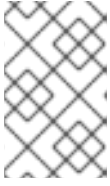
- b. Select **azure** as the platform to target.
- c. If you do not have a Microsoft Azure profile stored on your computer, specify the following Azure parameter values for your subscription and service principal:
  - **azure subscription id** The subscription ID to use for the cluster. Specify the **id** value in your account output.
  - **azure tenant id** The tenant ID. Specify the **tenantid** value in your account output.
  - **azure service principal client id** The value of the **appid** parameter for the service principal.
  - **azure service principal client secret** The value of the **password** parameter for the service principal.
- d. Select the region to deploy the cluster to.
- e. Select the base domain to deploy the cluster to. The base domain corresponds to the Azure DNS Zone that you created for your cluster.
- f. Enter a descriptive name for your cluster.



### IMPORTANT

All Azure resources that are available through public endpoints are subject to resource name restrictions, and you cannot create resources that use certain terms. For a list of terms that Azure restricts, see [Resolve reserved resource name errors](#) in the Azure documentation.

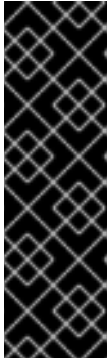
- g. Paste the pull secret that you obtained from the [Pull Secret](#) page on the Red Hat OpenShift Cluster Manager site.



### NOTE

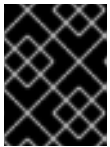
If the cloud provider account that you configured on your host does not have sufficient permissions to deploy the cluster, the installation process stops, and the missing permissions are displayed.

When the cluster deployment completes, directions for accessing your cluster, including a link to its web console and credentials for the **kubeadmin** user, display in your terminal.



### 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.



### IMPORTANT

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

## 1.2.6. Installing the CLI by downloading the binary

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



### IMPORTANT

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.5. Download and install the new version of **oc**.

### 1.2.6.1. Installing the CLI on Linux

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

#### Procedure

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

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

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

```
$ echo $PATH
```

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

```
$ oc <command>
```

### 1.2.6.2. Installing the CLI on Windows

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

#### Procedure

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

```
C:\> path
```

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

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

### 1.2.6.3. Installing the CLI on macOS

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

#### Procedure

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

-



```
$ echo $PATH
```

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

```
$ oc <command>
```

### 1.2.7. Logging in to the cluster

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

#### Prerequisites

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

#### Procedure

1. Export the **kubeadmin** credentials:

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

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

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

```
$ oc whoami
```

#### Example output

```
system:admin
```

### 1.2.8. Next steps

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

## 1.3. INSTALLING A CLUSTER ON AZURE WITH CUSTOMIZATIONS

In OpenShift Container Platform version 4.5, you can install a customized cluster on infrastructure that the installation program provisions on Microsoft Azure. To customize the installation, you modify parameters in the **install-config.yaml** file before you install the cluster.

### 1.3.1. Prerequisites

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

- [Configure an Azure account](#) to host the cluster and determine the tested and validated region to deploy the cluster to.
- If you use a firewall, you must [configure it to allow the sites](#) that your cluster requires access to.
- If you do not allow the system to manage identity and access management (IAM), then a cluster administrator can [manually create and maintain IAM credentials](#). Manual mode can also be used in environments where the cloud IAM APIs are not reachable.

### 1.3.2. Internet and Telemetry access for OpenShift Container Platform

In OpenShift Container Platform 4.5, you require access to the Internet to install your cluster. The Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, also requires Internet access. If your cluster is connected to the Internet, Telemetry runs automatically, and your cluster is registered to the [Red Hat OpenShift Cluster Manager \(OCM\)](#).

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

You must have Internet access to:

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



#### IMPORTANT

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

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

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



#### NOTE

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

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

**NOTE**

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

**Procedure**

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

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

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

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

**NOTE**

If you plan to install an OpenShift Container Platform cluster that uses FIPS Validated / Modules in Process cryptographic libraries on the **x86\_64** architecture, do not create a key that uses the **ed25519** algorithm. Instead, create a key that uses the **rsa** or **ecdsa** algorithm.

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

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

**Example output**

```
Agent pid 31874
```

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.

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

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

**Example output**

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

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

**Next steps**

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

### 1.3.4. Obtaining the installation program

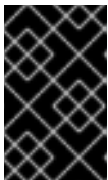
Before you install OpenShift Container Platform, download the installation file on a local computer.

#### Prerequisites

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

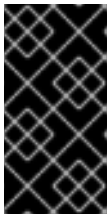
#### Procedure

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



#### IMPORTANT

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



#### IMPORTANT

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

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

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

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

### 1.3.5. Creating the installation configuration file

You can customize the OpenShift Container Platform cluster you install on Microsoft Azure.

#### Prerequisites

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

## Procedure

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

- a. Run the following command:

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

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



### IMPORTANT

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

- 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

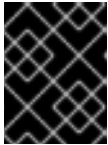
For production OpenShift Container Platform clusters on which you want to perform installation debugging or disaster recovery, specify an SSH key that your **ssh-agent** process uses.

- ii. Select **azure** as the platform to target.
- iii. If you do not have a Microsoft Azure profile stored on your computer, specify the following Azure parameter values for your subscription and service principal:
  - **azure subscription id** The subscription ID to use for the cluster. Specify the **id** value in your account output.
  - **azure tenant id** The tenant ID. Specify the **tenantId** value in your account output.
  - **azure service principal client id** The value of the **appId** parameter for the service principal.
  - **azure service principal client secret** The value of the **password** parameter for the service principal.
- iv. Select the region to deploy the cluster to.
- v. Select the base domain to deploy the cluster to. The base domain corresponds to the Azure DNS Zone that you created for your cluster.
- vi. Enter a descriptive name for your cluster.

**IMPORTANT**

All Azure resources that are available through public endpoints are subject to resource name restrictions, and you cannot create resources that use certain terms. For a list of terms that Azure restricts, see [Resolve reserved resource name errors](#) in the Azure documentation.

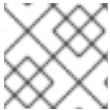
- vii. Paste the pull secret that you obtained from the [Pull Secret](#) page on the Red Hat OpenShift Cluster Manager site.
2. Modify the **install-config.yaml** file. You can find more information about the available parameters in the **Installation configuration parameters** section.
3. Back up the **install-config.yaml** file so that you can use it to install multiple clusters.

**IMPORTANT**

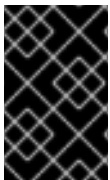
The **install-config.yaml** file is consumed during the installation process. If you want to reuse the file, you must back it up now.

**1.3.5.1. Installation configuration parameters**

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

**NOTE**

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

**IMPORTANT**

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

**1.3.5.1.1. Required configuration parameters**

Required installation configuration parameters are described in the following table:

**Table 1.1. Required parameters**

Parameter	Description	Values
<b>apiVersion</b>	The API version for the <b>install-config.yaml</b> content. The current version is <b>v1</b> . The installer may also support older API versions.	String

Parameter	Description	Values
<b>baseDomain</b>	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 <b>baseDomain</b> and <b>metadata.name</b> parameter values that uses the <b>&lt;metadata.name&gt;</b> . <b>&lt;baseDomain&gt;</b> format.	A fully-qualified domain or subdomain name, such as <b>example.com</b> .
<b>metadata</b>	Kubernetes resource <b>ObjectMeta</b> , from which only the <b>name</b> parameter is consumed.	Object
<b>metadata.name</b>	The name of the cluster. DNS records for the cluster are all subdomains of <b>{{.metadata.name}}</b> . <b>{{.baseDomain}}</b> .	String of lowercase letters, hyphens (-), and periods (.), such as <b>dev</b> .
<b>platform</b>	The configuration for the specific platform upon which to perform the installation: <b>aws, baremetal, azure, openstack, ovirt, vsphere</b> . For additional information about <b>platform.&lt;platform&gt;</b> parameters, consult the following table for your specific platform.	Object
<b>pullSecret</b>	Get a pull secret from <a href="https://cloud.redhat.com/openshift/install/pull-secret">https://cloud.redhat.com/openshift/install/pull-secret</a> to authenticate downloading container images for OpenShift Container Platform components from services such as Quay.io.	<pre>{   "auths":{     "cloud.openshift.com":{       "auth":"b3Blb=",       "email":"you@example.com"     },     "quay.io":{       "auth":"b3Blb=",       "email":"you@example.com"     }   } }</pre>


Parameter	Description	Values
-----------	-------------	--------

### 1.3.5.1.2. Network configuration parameters


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

Only IPv4 addresses are supported.

Table 1.2. Network parameters

Parameter	Description	Values
<b>networking</b>	The configuration for the cluster network.	Object   <b>NOTE</b> You cannot modify parameters specified by the <b>networking</b> object after installation.
<b>networking.networkType</b>	The cluster network provider Container Network Interface (CNI) plug-in to install.	Either <b>OpenShiftSDN</b> or <b>OVNKubernetes</b> . The default value is <b>OpenShiftSDN</b> .
<b>networking.clusterNetwork</b>	The IP address blocks for pods.  The default value is <b>10.128.0.0/14</b> with a host prefix of <b>/23</b> .  If you specify multiple IP address blocks, the blocks must not overlap.	An array of objects. For example:  <pre>networking:   clusterNetwork:   - cidr: 10.128.0.0/14     hostPrefix: 23</pre>





Parameter	Description	Values
<b>networking.clusterNetwork.cidr</b>	Required if you use <b>networking.clusterNetwork</b> . An IP address block.  An IPv4 network.	An IP address block in Classless Inter-Domain Routing (CIDR) notation. The prefix length for an IPv4 block is between <b>0</b> and <b>32</b> .
<b>networking.clusterNetwork.hostPrefix</b>	The subnet prefix length to assign to each individual node. For example, if <b>hostPrefix</b> is set to <b>23</b> then each node is assigned a <b>/23</b> subnet out of the given <b>cidr</b> . A <b>hostPrefix</b> value of <b>23</b> provides 510 ( $2^{(32 - 23)} - 2$ ) pod IP addresses.	A subnet prefix.  The default value is <b>23</b> .
<b>networking.serviceNetwork</b>	The IP address block for services. The default value is <b>172.30.0.0/16</b> .  The OpenShift SDN and OVN-Kubernetes network providers support only a single IP address block for the service network.	An array with an IP address block in CIDR format. For example:  <pre>networking:   serviceNetwork:     - 172.30.0.0/16</pre>
<b>networking.machineNetwork</b>	The IP address blocks for machines.  If you specify multiple IP address blocks, the blocks must not overlap.	An array of objects. For example:  <pre>networking:   machineNetwork:     - cidr: 10.0.0.0/16</pre>
<b>networking.machineNetwork.cidr</b>	Required if you use <b>networking.machineNetwork</b> . An IP address block. The default value is <b>10.0.0.0/16</b> for all platforms other than libvirt. For libvirt, the default value is <b>192.168.126.0/24</b> .	An IP network block in CIDR notation.  For example, <b>10.0.0.0/16</b> .   <b>NOTE</b>  Set the <b>networking.machineNetwork</b> to match the CIDR that the preferred NIC resides in.


### 1.3.5.1.3. Optional configuration parameters


Optional installation configuration parameters are described in the following table:

Table 1.3. Optional parameters

Parameter	Description	Values
<b>additionalTrustBundle</b>	A PEM-encoded X.509 certificate bundle that is added to the nodes' trusted certificate store. This trust bundle may also be used when a proxy has been configured.	String
<b>compute</b>	The configuration for the machines that comprise the compute nodes.	Array of machine-pool objects. For details, see the following "Machine-pool" table.
<b>compute.architecture</b>	Determines the instruction set architecture of the machines in the pool. Currently, heterogeneous clusters are not supported, so all pools must specify the same architecture. Valid values are <b>amd64</b> (the default).	String
<b>compute.hyperthreading</b>	Whether to enable or disable simultaneous multithreading, or <b>hyperthreading</b> , on compute machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores.   <b>IMPORTANT</b> If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.	<b>Enabled</b> or <b>Disabled</b>
<b>compute.name</b>	Required if you use <b>compute</b> . The name of the machine pool.	<b>worker</b>
<b>compute.platform</b>	Required if you use <b>compute</b> . Use this parameter to specify the cloud provider to host the worker machines. This parameter value must match the <b>controlPlane.platform</b> parameter value.	<b>aws, azure, gcp, openstack, ovirt, vsphere</b> , or <b>{}</b>
<b>compute.replicas</b>	The number of compute machines, which are also known as worker machines, to provision.	A positive integer greater than or equal to <b>2</b> . The default value is <b>3</b> .

Parameter	Description	Values
<b>controlPlane</b>	The configuration for the machines that comprise the control plane.	Array of <b>MachinePool</b> objects. For details, see the following "Machine-pool" table.
<b>controlPlane.architecture</b>	Determines the instruction set architecture of the machines in the pool. Currently, heterogeneous clusters are not supported, so all pools must specify the same architecture. Valid values are <b>amd64</b> (the default).	String
<b>controlPlane.hyperthreading</b>	<p>Whether to enable or disable simultaneous multithreading, or <b>hyperthreading</b>, on control plane machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores.</p> <div style="display: flex; align-items: center;">  <div> <p><b>IMPORTANT</b></p> <p>If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.</p> </div> </div>	<b>Enabled</b> or <b>Disabled</b>
<b>controlPlane.name</b>	Required if you use <b>controlPlane</b> . The name of the machine pool.	<b>master</b>
<b>controlPlane.platform</b>	Required if you use <b>controlPlane</b> . Use this parameter to specify the cloud provider that hosts the control plane machines. This parameter value must match the <b>compute.platform</b> parameter value.	<b>aws, azure, gcp, openstack, ovirt, vsphere, or {}</b>
<b>controlPlane.replicas</b>	The number of control plane machines to provision.	The only supported value is <b>3</b> , which is the default value.

Parameter	Description	Values
<b>fips</b>	<p>Enable or disable FIPS mode. The default is <b>false</b> (disabled). If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.</p> <div style="display: flex; align-items: flex-start;">  <div> <p><b>NOTE</b></p> <p>If you are using Azure File storage, you cannot enable FIPS mode.</p> </div> </div>	<b>false</b> or <b>true</b>
<b>imageContentSources</b>	Sources and repositories for the release-image content.	Array of objects. Includes a <b>source</b> and, optionally, <b>mirrors</b> , as described in the following rows of this table.
<b>imageContentSources.source</b>	Required if you use <b>imageContentSources</b> . Specify the repository that users refer to, for example, in image pull specifications.	String
<b>imageContentSources.mirrors</b>	Specify one or more repositories that may also contain the same images.	Array of strings
<b>publish</b>	How to publish or expose the user-facing endpoints of your cluster, such as the Kubernetes API, OpenShift routes.	<b>Internal</b> or <b>External</b> . To deploy a private cluster, which cannot be accessed from the internet, set <b>publish</b> to <b>Internal</b> . The default value is <b>External</b> .

Parameter	Description	Values
<b>sshKey</b>	<p>The SSH key to authenticate access to your cluster machines.</p>  <p><b>NOTE</b></p> <p>For production OpenShift Container Platform clusters on which you want to perform installation debugging or disaster recovery, specify an SSH key that your <b>ssh-agent</b> process uses.</p>	For example, <b>sshKey: ssh-ed25519 AAAA...</b>

#### 1.3.5.1.4. Additional Azure configuration parameters

Additional Azure configuration parameters are described in the following table:

Table 1.4. Additional Azure parameters

Parameter	Description	Values
<b>controlPlane.platform.azure.osDisk.diskSizeGB</b>	The Azure disk size for the VM.	Integer that represents the size of the disk in GB. The minimum supported disk size is <b>1024</b> .
<b>platform.azure.baseDomainResourceGroupName</b>	The name of the resource group that contains the DNS zone for your base domain.	String, for example <b>production_cluster</b> .
<b>platform.azure.region</b>	The name of the Azure region that hosts your cluster.	Any valid region name, such as <b>centralus</b> .
<b>platform.azure.zone</b>	List of availability zones to place machines in. For high availability, specify at least two zones.	List of zones, for example <b>["1", "2", "3"]</b> .
<b>platform.azure.networkResourceGroupName</b>	The name of the resource group that contains the existing VNet that you want to deploy your cluster to. This name cannot be the same as the <b>platform.azure.baseDomainResourceGroupName</b> .	String.
<b>platform.azure.virtualNetwork</b>	The name of the existing VNet that you want to deploy your cluster to.	String.

Parameter	Description	Values
<b>platform.azure.controlPlaneSubnet</b>	The name of the existing subnet in your VNet that you want to deploy your control plane machines to.	Valid CIDR, for example <b>10.0.0.0/16</b> .
<b>platform.azure.computeSubnet</b>	The name of the existing subnet in your VNet that you want to deploy your compute machines to.	Valid CIDR, for example <b>10.0.0.0/16</b> .

**NOTE**

You cannot customize [Azure Availability Zones](#) or [Use tags to organize your Azure resources](#) with an Azure cluster.

### 1.3.5.2. Sample customized `install-config.yaml` file for Azure

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 modify it.

```

apiVersion: v1
baseDomain: example.com 1
controlPlane: 2
  hyperthreading: Enabled 3 4
  name: master
  platform:
    azure:
      osDisk:
        diskSizeGB: 1024 5
        type: Standard_D8s_v3
      replicas: 3
compute: 6
- hyperthreading: Enabled 7
  name: worker
  platform:
    azure:
      type: Standard_D2s_v3
      osDisk:
        diskSizeGB: 512 8
      zones: 9
      - "1"
      - "2"
      - "3"
  replicas: 5
metadata:

```

```

name: test-cluster 10
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
platform:
  azure:
    region: centralus 11
    baseDomainResourceGroupName: resource_group 12
pullSecret: '{"auths": ...}' 13
ifndef::openshift-origin
fips: false 14
sshKey: ssh-ed25519 AAAA... 15
endif::openshift-origin
ifdef::openshift-origin
sshKey: ssh-ed25519 AAAA... 16
endif::openshift-origin

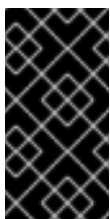
```

**1 10 11 13** Required. The installation program prompts you for this value.

**2 6** If you do not provide these parameters and values, the installation program provides the default value.

**3 7** The **controlPlane** section is a single mapping, but the compute section is a sequence of mappings. To meet the requirements of the different data structures, the first line of the **compute** section must begin with a hyphen, -, and the first line of the **controlPlane** section must not. Although both sections currently define a single machine pool, it is possible that future versions of OpenShift Container Platform will support defining multiple compute pools during installation. Only one control plane pool is used.

**4** Whether to enable or disable simultaneous multithreading, or **hypertreading**. 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 virtual machine types, such as **Standard\_D8s\_v3**, for your machines if you disable simultaneous multithreading.

**5 8** You can specify the size of the disk to use in GB. Minimum recommendation for master nodes is 1024 GB.

**9** Specify a list of zones to deploy your machines to. For high availability, specify at least two zones.

**12** Specify the name of the resource group that contains the DNS zone for your base domain.

- 14 16 Whether to enable or disable FIPS mode. By default, FIPS mode is not enabled. If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container
- 15 You can optionally 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.

### 1.3.6. 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

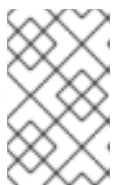
- Configure an account with the cloud platform that hosts your cluster.
- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.

#### Procedure

1. Run the installation program:

```
$ ./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**.

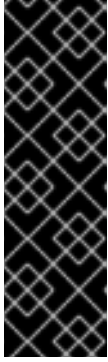


#### NOTE

If the cloud provider account that you configured on your host does not have sufficient permissions to deploy the cluster, the installation process stops, and the missing permissions are displayed.

When the cluster deployment completes, directions for accessing your cluster, including a link to its web console and credentials for the **kubeadmin** user, display in your terminal.



**IMPORTANT**

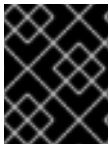
The Ignition config files that the installation program generates contain certificates that expire after 24 hours, which are then renewed at that time. If the cluster is shut down before renewing the certificates and the cluster is later restarted after the 24 hours have elapsed, the cluster automatically 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.

**IMPORTANT**

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

### 1.3.7. Installing the CLI by downloading the binary

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

**IMPORTANT**

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.5. Download and install the new version of **oc**.

#### 1.3.7.1. Installing the CLI on Linux

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

**Procedure**

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

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

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

```
$ echo $PATH
```

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

```
$ oc <command>
```

#### 1.3.7.2. Installing the CLI on Windows

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

### Procedure

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

```
C:\> path
```

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

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

### 1.3.7.3. Installing the CLI on macOS

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

### Procedure

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

```
$ echo $PATH
```

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

```
$ oc <command>
```

### 1.3.8. Logging in to the cluster

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

## Prerequisites

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

## Procedure

1. Export the **kubeadmin** credentials:

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

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

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

```
$ oc whoami
```

### Example output

```
system:admin
```

### 1.3.9. Next steps

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

## 1.4. INSTALLING A CLUSTER ON AZURE WITH NETWORK CUSTOMIZATIONS

In OpenShift Container Platform version 4.5, you can install a cluster with a customized network configuration on infrastructure that the installation program provisions on Microsoft Azure. 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.

### 1.4.1. Prerequisites

- Review details about the [OpenShift Container Platform installation and update](#) processes.
- [Configure an Azure account](#) to host the cluster and determine the tested and validated region to deploy the cluster to.
- If you use a firewall, you must [configure it to allow the sites](#) that your cluster requires access to.
- If you do not allow the system to manage identity and access management (IAM), then a cluster administrator can [manually create and maintain IAM credentials](#). Manual mode can also be used in environments where the cloud IAM APIs are not reachable.

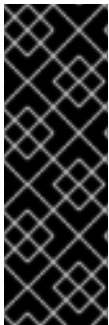
## 1.4.2. Internet and Telemetry access for OpenShift Container Platform

In OpenShift Container Platform 4.5, you require access to the Internet to install your cluster. The Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, also requires Internet access. If your cluster is connected to the Internet, Telemetry runs automatically, and your cluster is registered to the [Red Hat OpenShift Cluster Manager \(OCM\)](#).

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

You must have Internet access to:

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



### IMPORTANT

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

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

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



### NOTE

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

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



### NOTE

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

### Procedure

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

■

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

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

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



#### NOTE

If you plan to install an OpenShift Container Platform cluster that uses FIPS Validated / Modules in Process cryptographic libraries on the **x86\_64** architecture, do not create a key that uses the **ed25519** algorithm. Instead, create a key that uses the **rsa** or **ecdsa** algorithm.

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

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

#### Example output

```
Agent pid 31874
```

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.

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

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

#### Example output

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

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

#### Next steps

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

### 1.4.4. Obtaining the installation program

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

#### Prerequisites

- You must install the cluster from a computer that uses Linux or macOS.

- You need 500 MB of local disk space to download the installation program.

## Procedure

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



### IMPORTANT

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



### IMPORTANT

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

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

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

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

## 1.4.5. Creating the installation configuration file

You can customize the OpenShift Container Platform cluster you install on Microsoft Azure.

### Prerequisites

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

### Procedure

1. Create the **install-config.yaml** file.
  - a. Run the following command:

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

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



## IMPORTANT

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

- 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 **azure** as the platform to target.
- iii. If you do not have a Microsoft Azure profile stored on your computer, specify the following Azure parameter values for your subscription and service principal:
  - **azure subscription id** The subscription ID to use for the cluster. Specify the **id** value in your account output.
  - **azure tenant id** The tenant ID. Specify the **tenantId** value in your account output.
  - **azure service principal client id** The value of the **appId** parameter for the service principal.
  - **azure service principal client secret** The value of the **password** parameter for the service principal.
- iv. Select the region to deploy the cluster to.
- v. Select the base domain to deploy the cluster to. The base domain corresponds to the Azure DNS Zone that you created for your cluster.
- vi. Enter a descriptive name for your cluster.

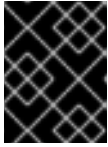


## IMPORTANT

All Azure resources that are available through public endpoints are subject to resource name restrictions, and you cannot create resources that use certain terms. For a list of terms that Azure restricts, see [Resolve reserved resource name errors](#) in the Azure documentation.

- vii. Paste the pull secret that you obtained from the [Pull Secret](#) page on the Red Hat OpenShift Cluster Manager site.
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.

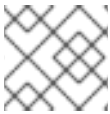


### IMPORTANT

The **install-config.yaml** file is consumed during the installation process. If you want to reuse the file, you must back it up now.

#### 1.4.5.1. Installation configuration parameters

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



### NOTE

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



### IMPORTANT

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

##### 1.4.5.1.1. Required configuration parameters

Required installation configuration parameters are described in the following table:

Table 1.5. Required parameters

Parameter	Description	Values
<b>apiVersion</b>	The API version for the <b>install-config.yaml</b> content. The current version is <b>v1</b> . The installer may also support older API versions.	String
<b>baseDomain</b>	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 <b>baseDomain</b> and <b>metadata.name</b> parameter values that uses the <b>&lt;metadata.name&gt;</b> . <b>&lt;baseDomain&gt;</b> format.	A fully-qualified domain or subdomain name, such as <b>example.com</b> .



Parameter	Description	Values
<b>metadata</b>	Kubernetes resource <b>ObjectMeta</b> , from which only the <b>name</b> parameter is consumed.	Object
<b>metadata.name</b>	The name of the cluster. DNS records for the cluster are all subdomains of <b>{{.metadata.name}}.{{.baseDomain}}</b> .	String of lowercase letters, hyphens (-), and periods (.), such as <b>dev</b> .
<b>platform</b>	The configuration for the specific platform upon which to perform the installation: <b>aws, baremetal, azure, openstack, ovirt, vsphere</b> . For additional information about <b>platform.&lt;platform&gt;</b> parameters, consult the following table for your specific platform.	Object
<b>pullSecret</b>	Get a pull secret from <a href="https://cloud.redhat.com/openshift/install/pull-secret">https://cloud.redhat.com/openshift/install/pull-secret</a> to authenticate downloading container images for OpenShift Container Platform components from services such as Quay.io.	<pre>{   "auths":{     "cloud.openshift.com":{       "auth":"b3Blb=",       "email":"you@example.com"     },     "quay.io":{       "auth":"b3Blb=",       "email":"you@example.com"     }   } }</pre>

#### 1.4.5.1.2. Network configuration parameters


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

Only IPv4 addresses are supported.

**Table 1.6. Network parameters**

Parameter	Description	Values
-----------	-------------	--------

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


Parameter	Description	Values
<b>networking.machineNetwork.cidr</b>	Required if you use <b>networking.machineNetwork</b> . An IP address block. The default value is <b>10.0.0.0/16</b> for all platforms other than libvirt. For libvirt, the default value is <b>192.168.126.0/24</b> .	An IP network block in CIDR notation. For example, <b>10.0.0.0/16</b> .  <b>NOTE</b> Set the <b>networking.machineNetwork</b> to match the CIDR that the preferred NIC resides in.


#### 1.4.5.1.3. Optional configuration parameters

Optional installation configuration parameters are described in the following table:

Table 1.7. Optional parameters

Parameter	Description	Values
<b>additionalTrustBundle</b>	A PEM-encoded X.509 certificate bundle that is added to the nodes' trusted certificate store. This trust bundle may also be used when a proxy has been configured.	String
<b>compute</b>	The configuration for the machines that comprise the compute nodes.	Array of machine-pool objects. For details, see the following "Machine-pool" table.
<b>compute.architecture</b>	Determines the instruction set architecture of the machines in the pool. Currently, heterogeneous clusters are not supported, so all pools must specify the same architecture. Valid values are <b>amd64</b> (the default).	String

Parameter	Description	Values
<b>compute.hyperthreading</b>	<p>Whether to enable or disable simultaneous multithreading, or <b>hyperthreading</b>, on compute machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores.</p> <div style="display: flex; align-items: center;">  <div> <p><b>IMPORTANT</b></p> <p>If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.</p> </div> </div>	<b>Enabled</b> or <b>Disabled</b>
<b>compute.name</b>	Required if you use <b>compute</b> . The name of the machine pool.	<b>worker</b>
<b>compute.platform</b>	Required if you use <b>compute</b> . Use this parameter to specify the cloud provider to host the worker machines. This parameter value must match the <b>controlPlane.platform</b> parameter value.	<b>aws, azure, gcp, openstack, ovirt, vsphere</b> , or <b>{}</b>
<b>compute.replicas</b>	The number of compute machines, which are also known as worker machines, to provision.	A positive integer greater than or equal to <b>2</b> . The default value is <b>3</b> .
<b>controlPlane</b>	The configuration for the machines that comprise the control plane.	Array of <b>MachinePool</b> objects. For details, see the following "Machine-pool" table.
<b>controlPlane.architecture</b>	Determines the instruction set architecture of the machines in the pool. Currently, heterogeneous clusters are not supported, so all pools must specify the same architecture. Valid values are <b>amd64</b> (the default).	String

Parameter	Description	Values
<b>controlPlane.hyperthreading</b>	<p>Whether to enable or disable simultaneous multithreading, or <b>hyperthreading</b>, on control plane machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores.</p> <div style="display: flex; align-items: center;">  <div> <p><b>IMPORTANT</b></p> <p>If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.</p> </div> </div>	<b>Enabled</b> or <b>Disabled</b>
<b>controlPlane.name</b>	Required if you use <b>controlPlane</b> . The name of the machine pool.	<b>master</b>
<b>controlPlane.platform</b>	Required if you use <b>controlPlane</b> . Use this parameter to specify the cloud provider that hosts the control plane machines. This parameter value must match the <b>compute.platform</b> parameter value.	<b>aws, azure, gcp, openstack, ovirt, vsphere, or {}</b>
<b>controlPlane.replicas</b>	The number of control plane machines to provision.	The only supported value is <b>3</b> , which is the default value.
<b>fips</b>	<p>Enable or disable FIPS mode. The default is <b>false</b> (disabled). If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.</p> <div style="display: flex; align-items: center;">  <div> <p><b>NOTE</b></p> <p>If you are using Azure File storage, you cannot enable FIPS mode.</p> </div> </div>	<b>false</b> or <b>true</b>

Parameter	Description	Values
<b>imageContentSources</b>	Sources and repositories for the release-image content.	Array of objects. Includes a <b>source</b> and, optionally, <b>mirrors</b> , as described in the following rows of this table.
<b>imageContentSources.source</b>	Required if you use <b>imageContentSources</b> . Specify the repository that users refer to, for example, in image pull specifications.	String
<b>imageContentSources.mirrors</b>	Specify one or more repositories that may also contain the same images.	Array of strings
<b>publish</b>	How to publish or expose the user-facing endpoints of your cluster, such as the Kubernetes API, OpenShift routes.	<b>Internal</b> or <b>External</b> . To deploy a private cluster, which cannot be accessed from the internet, set <b>publish</b> to <b>Internal</b> . The default value is <b>External</b> .
<b>sshKey</b>	The SSH key to authenticate access to your cluster machines.   <p><b>NOTE</b></p> <p>For production OpenShift Container Platform clusters on which you want to perform installation debugging or disaster recovery, specify an SSH key that your <b>ssh-agent</b> process uses.</p>	For example, <b>sshKey: ssh-ed25519 AAAA...</b>

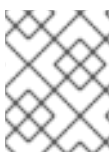
#### 1.4.5.1.4. Additional Azure configuration parameters

Additional Azure configuration parameters are described in the following table:

**Table 1.8. Additional Azure parameters**

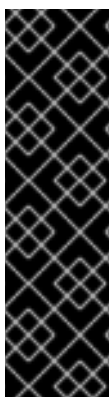
Parameter	Description	Values
<b>controlPlane.platform.azure.osDisk.diskSizeGB</b>	The Azure disk size for the VM.	Integer that represents the size of the disk in GB. The minimum supported disk size is <b>1024</b> .

Parameter	Description	Values
<b>platform.azure.baseDomainResourceGroupName</b>	The name of the resource group that contains the DNS zone for your base domain.	String, for example <b>production_cluster</b> .
<b>platform.azure.region</b>	The name of the Azure region that hosts your cluster.	Any valid region name, such as <b>centralus</b> .
<b>platform.azure.zone</b>	List of availability zones to place machines in. For high availability, specify at least two zones.	List of zones, for example <b>["1", "2", "3"]</b> .
<b>platform.azure.networkResourceGroupName</b>	The name of the resource group that contains the existing VNet that you want to deploy your cluster to. This name cannot be the same as the <b>platform.azure.baseDomainResourceGroupName</b> .	String.
<b>platform.azure.virtualNetwork</b>	The name of the existing VNet that you want to deploy your cluster to.	String.
<b>platform.azure.controlPlaneSubnet</b>	The name of the existing subnet in your VNet that you want to deploy your control plane machines to.	Valid CIDR, for example <b>10.0.0.0/16</b> .
<b>platform.azure.computeSubnet</b>	The name of the existing subnet in your VNet that you want to deploy your compute machines to.	Valid CIDR, for example <b>10.0.0.0/16</b> .



## NOTE

You cannot customize [Azure Availability Zones](#) or [Use tags to organize your Azure resources](#) with an Azure cluster.



## IMPORTANT

The Open Virtual Networking (OVN) Kubernetes network plug-in is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of the OVN Technology Preview, see <https://access.redhat.com/articles/4380121>.

### 1.4.5.2. Network configuration parameters

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



#### NOTE

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

Table 1.9. Required network parameters

Parameter	Description	Value
<b>networking.net workType</b>	The default Container Network Interface (CNI) network provider plug-in to deploy. The <b>OpenShiftSDN</b> plug-in is the only plug-in supported in OpenShift Container Platform 4.5. The <b>OVNKubernetes</b> plug-in is available as a Technology Preview in OpenShift Container Platform 4.5.	Either <b>OpenShiftSDN</b> or <b>OVNKubernetes</b> . The default value is <b>OpenShiftSDN</b> .
<b>networking.clusterNetwork[].cidr</b>	A block of IP addresses from which pod IP addresses are allocated. The <b>OpenShiftSDN</b> network plug-in supports multiple cluster networks. The address blocks for multiple cluster networks must not overlap. Select address pools large enough to fit your anticipated workload.	An IP address allocation in CIDR format. The default value is <b>10.128.0.0/14</b> .
<b>networking.clusterNetwork[].hostPrefix</b>	The subnet prefix length to assign to each individual node. For example, if <b>hostPrefix</b> is set to <b>23</b> , then each node is assigned a <b>/23</b> subnet out of the given <b>cidr</b> , allowing for 510 ( $2^{(32 - 23)} - 2$ ) pod IP addresses.	A subnet prefix. The default value is <b>23</b> .
<b>networking.serviceNetwork[]</b>	A block of IP addresses for services. <b>OpenShiftSDN</b> allows only one <b>serviceNetwork</b> block. The address block must not overlap with any other network block.	An IP address allocation in CIDR format. The default value is <b>172.30.0.0/16</b> .
<b>networking.machineNetwork[].cidr</b>	A block of IP addresses assigned to nodes created by the OpenShift Container Platform installation program while installing the cluster. The address block must not overlap with any other network block. Multiple CIDR ranges may be specified.	An IP address allocation in CIDR format. The default value is <b>10.0.0.0/16</b> .

#### 1.4.5.3. Sample customized install-config.yaml file for Azure

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 modify it.



```

apiVersion: v1
baseDomain: example.com 1
controlPlane: 2
  hyperthreading: Enabled 3 4
  name: master
  platform:
    azure:
      osDisk:
        diskSizeGB: 1024 5
        type: Standard_D8s_v3
      replicas: 3
compute: 6
- hyperthreading: Enabled 7
  name: worker
  platform:
    azure:
      type: Standard_D2s_v3
      osDisk:
        diskSizeGB: 512 8
      zones: 9
      - "1"
      - "2"
      - "3"
    replicas: 5
metadata:
  name: test-cluster 10
networking: 11
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
platform:
  azure:
    region: centralus 12
    baseDomainResourceGroupName: resource_group 13
pullSecret: '{"auths": ...}' 14
ifndef::openshift-origin
fips: false 15
sshKey: ssh-ed25519 AAAA... 16
endif::openshift-origin
ifndef::openshift-origin
sshKey: ssh-ed25519 AAAA... 17
endif::openshift-origin

```

1 10 12 14 Required. The installation program prompts you for this value.

2 6 11 If you do not provide these parameters and values, the installation program provides the default value.

3 7 The **controlPlane** section is a single mapping, but the compute section is a sequence of mappings.

To meet the requirements of the different data structures, the first line of the **compute** section must begin with a hyphen, -, and the first line of the **controlPlane** section must not. Although both sections currently define a single machine pool, it is possible that future versions of OpenShift Container Platform will support defining multiple compute pools during installation. Only one control plane pool is used.

- 4 Whether to enable or disable simultaneous multithreading, or **hypertreading**. 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 virtual machine types, such as **Standard\_D8s\_v3**, for your machines if you disable simultaneous multithreading.

- 5 8 You can specify the size of the disk to use in GB. Minimum recommendation for master nodes is 1024 GB.
- 9 Specify a list of zones to deploy your machines to. For high availability, specify at least two zones.
- 13 Specify the name of the resource group that contains the DNS zone for your base domain.
- 15 17 Whether to enable or disable FIPS mode. By default, FIPS mode is not enabled. If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.
- 16 You can optionally 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.

## 1.4.6. Modifying advanced network configuration parameters

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



### IMPORTANT

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

## Prerequisites

- Create the **install-config.yaml** file and complete any modifications to it.

## Procedure

1. Use the following command to create manifests:

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

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

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

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

- 1** For **<installation\_directory>**, specify the directory name that contains the **manifests/** directory for your cluster.

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

```
$ ls <installation_directory>/manifests/cluster-network-*
```

## Example output

```
cluster-network-01-crd.yml
cluster-network-02-config.yml
cluster-network-03-config.yml
```

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

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

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

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

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

### 1.4.7. Cluster Network Operator configuration

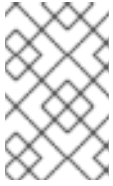
The configuration for the cluster network is specified as part of the Cluster Network Operator (CNO) configuration and stored in a CR object that is named **cluster**. The CR specifies the parameters for the **Network** API in the **operator.openshift.io** API group.

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

#### Cluster Network Operator CR

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  clusterNetwork: ❶
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  serviceNetwork: ❷
  - 172.30.0.0/16
  defaultNetwork: ❸
  ...
  kubeProxyConfig: ❹
  iptablesSyncPeriod: 30s ❺
  proxyArguments:
    iptables-min-sync-period: ❻
    - 0s
```

- ❶ ❷ Specified in the **install-config.yml** file.
- ❸ Configures the default Container Network Interface (CNI) network provider for the cluster network.
- ❹ The parameters for this object specify the **kube-proxy** configuration. If you do not specify the parameter values, the Cluster Network Operator applies the displayed default parameter values. If you are using the OVN-Kubernetes default CNI network provider, the kube-proxy configuration has no effect.
- ❺ The refresh period for **iptables** rules. The default value is **30s**. Valid suffixes include **s**, **m**, and **h** and are described in the [Go time package](#) documentation.

**NOTE**

Because of performance improvements introduced in OpenShift Container Platform 4.3 and greater, adjusting the **iptablesSyncPeriod** parameter is no longer necessary.

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

#### 1.4.7.1. Configuration parameters for the OpenShift SDN default CNI network provider

The following YAML object describes the configuration parameters for the OpenShift SDN default Container Network Interface (CNI) network provider.

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

- 1 Specified in the **install-config.yaml** file.
- 2 Specify only if you want to override part of the OpenShift SDN configuration.
- 3 Configures the network isolation mode for OpenShift SDN. The allowed values are **Multitenant**, **Subnet**, or **NetworkPolicy**. The default value is **NetworkPolicy**.
- 4 The maximum transmission unit (MTU) for the VXLAN 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 expected 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 **50** 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 **1450**.

- 5 The port to use for all VXLAN packets. The default value is **4789**. If you are running in a virtualized environment with existing nodes that are part of another VXLAN network, then you might be required to change this. For example, when running an OpenShift SDN overlay on top of VMware NSX-T, you must select an alternate port for VXLAN, since both SDNs use the same default VXLAN port number.

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

#### 1.4.7.2. Configuration parameters for the OVN-Kubernetes default CNI network provider

The following YAML object describes the configuration parameters for the OVN-Kubernetes default CNI network provider.

```
defaultNetwork:
  type: OVNKubernetes 1
  ovnKubernetesConfig: 2
    mtu: 1400 3
    genevePort: 6081 4
```

- 1** Specified in the **install-config.yaml** file.
- 2** Specify only if you want to override part of the OVN-Kubernetes configuration.
- 3** 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 expected 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**.

- 4** The UDP port for the Geneve overlay network.

### 1.4.7.3. Cluster Network Operator example configuration

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

#### Cluster Network Operator example CR

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  serviceNetwork:
    - 172.30.0.0/16
  defaultNetwork:
    type: OpenShiftSDN
    openshiftSDNConfig:
      mode: NetworkPolicy
      mtu: 1450
      vxlanPort: 4789
  kubeProxyConfig:
    iptablesSyncPeriod: 30s
  proxyArguments:
    iptables-min-sync-period:
      - 0s
```

## 1.4.8. 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

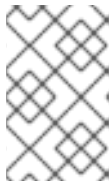
- Configure an account with the cloud platform that hosts your cluster.
- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.

### Procedure

1. Run the installation program:

```
$ ./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**.



### NOTE

If the cloud provider account that you configured on your host does not have sufficient permissions to deploy the cluster, the installation process stops, and the missing permissions are displayed.

When the cluster deployment completes, directions for accessing your cluster, including a link to its web console and credentials for the **kubeadmin** user, display in your terminal.



### 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.



### IMPORTANT

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

## 1.4.9. Installing the CLI by downloading the binary

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



### IMPORTANT

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.5. Download and install the new version of **oc**.

### 1.4.9.1. Installing the CLI on Linux

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

#### Procedure

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

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

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

```
$ echo $PATH
```

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

```
$ oc <command>
```

### 1.4.9.2. Installing the CLI on Windows

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

#### Procedure

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

-



```
C:\> path
```

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

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

### 1.4.9.3. Installing the CLI on macOS

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

#### Procedure

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

```
$ echo $PATH
```

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

```
$ oc <command>
```

### 1.4.10. Logging in to the cluster

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

#### Prerequisites

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

#### Procedure

1. Export the **kubeadmin** credentials:

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

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

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

```
$ oc whoami
```

#### Example output

```
system:admin
```

### 1.4.11. Next steps

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

## 1.5. INSTALLING A CLUSTER ON AZURE INTO AN EXISTING VNET

In OpenShift Container Platform version 4.5, you can install a cluster into an existing Azure Virtual Network (VNet) on Microsoft Azure. 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.

### 1.5.1. Prerequisites

- Review details about the [OpenShift Container Platform installation and update](#) processes.
- [Configure an Azure account](#) to host the cluster and determine the tested and validated region to deploy the cluster to.
- If you use a firewall, you must [configure it to allow the sites](#) that your cluster requires access to.
- If you do not allow the system to manage identity and access management (IAM), then a cluster administrator can [manually create and maintain IAM credentials](#). Manual mode can also be used in environments where the cloud IAM APIs are not reachable.

### 1.5.2. About reusing a VNet for your OpenShift Container Platform cluster

In OpenShift Container Platform 4.5, you can deploy a cluster into an existing Azure Virtual Network (VNet) in Microsoft Azure. If you do, you must also use existing subnets within the VNet and routing rules.

By deploying OpenShift Container Platform into an existing Azure VNet, you might be able to avoid service limit constraints in new accounts or more easily abide by the operational constraints that your company's guidelines set. This is a good option to use if you cannot obtain the infrastructure creation permissions that are required to create the VNet.



#### IMPORTANT

The use of an existing VNet requires the use of the updated Azure Private DNS (preview) feature. See [Announcing Preview Refresh for Azure DNS Private Zones](#) for more information about the limitations of this feature.

#### 1.5.2.1. Requirements for using your VNet

When you deploy a cluster by using an existing VNet, you must perform additional network configuration before you install the cluster. In installer-provisioned infrastructure clusters, the installer usually creates the following components, but it does not create them when you install into an existing VNet:

- Subnets
- Route tables
- VNets
- Network Security Groups

If you use a custom VNet, you must correctly configure it and its subnets for the installation program and the cluster to use. The installation program cannot subdivide network ranges for the cluster to use, set route tables for the subnets, or set VNet options like DHCP, so you must do so before you install the cluster.

The cluster must be able to access the resource group that contains the existing VNet and subnets. While all of the resources that the cluster creates are placed in a separate resource group that it creates, some network resources are used from a separate group. Some cluster Operators must be able to access resources in both resource groups. For example, the Machine API controller attaches NICS for the virtual machines that it creates to subnets from the networking resource group.

Your VNet must meet the following characteristics:

- The VNet's CIDR block must contain the **Networking.MachineCIDR** range, which is the IP address pool for cluster machines.
- The VNet and its subnets must belong to the same resource group, and the subnets must be configured to use Azure-assigned DHCP IP addresses instead of static IP addresses.

You must provide two subnets within your VNet, one for the control plane machines and one for the compute machines. Because Azure distributes machines in different availability zones within the region that you specify, your cluster will have high availability by default.

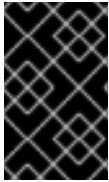
To ensure that the subnets that you provide are suitable, the installation program confirms the following data:

- All the subnets that you specify exist.
- You provide two private subnets, one for the control plane machines and one for the compute machines.
- The subnet CIDRs belong to the machine CIDR that you specified. Machines are not provisioned in availability zones that you do not provide private subnets for. If required, the installation program creates public load balancers that manage the control plane and worker nodes, and Azure allocates a public IP address to them.

If you destroy a cluster that uses an existing VNet, the VNet is not deleted.

#### 1.5.2.1.1. Network security group requirements

The network security groups for the subnets that host the compute and control plane machines require specific access to ensure that the cluster communication is correct. You must create rules to allow access to the required cluster communication ports.

**IMPORTANT**

The network security group rules must be in place before you install the cluster. If you attempt to install a cluster without the required access, the installation program cannot reach the Azure APIs, and installation fails.

Table 1.10. Required ports

Port	Description	Control plane	Compute
80	Allows HTTP traffic		x
443	Allows HTTPS traffic		x
6443	Allows communication to the control plane machines	x	
22623	Allows communication to the machine config server	x	

**NOTE**

Since cluster components do not modify the user-provided network security groups, which the Kubernetes controllers update, a pseudo-network security group is created for the Kubernetes controller to modify without impacting the rest of the environment.

**1.5.2.2. Division of permissions**

Starting with OpenShift Container Platform 4.3, you do not need all of the permissions that are required for an installation program-provisioned infrastructure cluster to deploy a cluster. This change mimics the division of permissions that you might have at your company: some individuals can create different resources in your clouds than others. For example, you might be able to create application-specific items, like instances, storage, and load balancers, but not networking-related components such as VNets, subnet, or ingress rules.

The Azure credentials that you use when you create your cluster do not need the networking permissions that are required to make VNets and core networking components within the VNet, such as subnets, routing tables, internet gateways, NAT, and VPN. You still need permission to make the application resources that the machines within the cluster require, such as load balancers, security groups, storage accounts, and nodes.

**1.5.2.3. Isolation between clusters**

Because the cluster is unable to modify network security groups in an existing subnet, there is no way to isolate clusters from each other on the VNet.

**1.5.3. Internet and Telemetry access for OpenShift Container Platform**

In OpenShift Container Platform 4.5, you require access to the Internet to install your cluster. The Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, also requires Internet access. If your cluster is connected to the Internet, Telemetry runs automatically, and your cluster is registered to the [Red Hat OpenShift Cluster Manager \(OCM\)](#).

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

You must have Internet access to:

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



### IMPORTANT

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

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

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



### NOTE

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

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



### NOTE

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

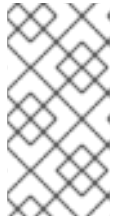
### Procedure

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

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

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

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



## NOTE

If you plan to install an OpenShift Container Platform cluster that uses FIPS Validated / Modules in Process cryptographic libraries on the **x86\_64** architecture, do not create a key that uses the **ed25519** algorithm. Instead, create a key that uses the **rsa** or **ecdsa** algorithm.

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

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

### Example output

```
Agent pid 31874
```

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.

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

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

### Example output

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

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

## Next steps

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

## 1.5.5. Obtaining the installation program

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

### Prerequisites

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

### Procedure

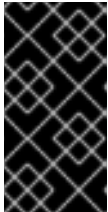
1. Access the [Infrastructure Provider](#) page on the Red Hat OpenShift Cluster Manager site. If you have a Red Hat account, log in with your credentials. If you do not, create an account.

2. Navigate to the page for your installation type, download the installation program for your operating system, and place the file in the directory where you will store the installation configuration files.



### IMPORTANT

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



### IMPORTANT

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

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

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

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

## 1.5.6. Creating the installation configuration file

You can customize the OpenShift Container Platform cluster you install on Microsoft Azure.

### Prerequisites

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

### Procedure

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

- a. Run the following command:

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

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



### IMPORTANT

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

- 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 **azure** as the platform to target.
- iii. If you do not have a Microsoft Azure profile stored on your computer, specify the following Azure parameter values for your subscription and service principal:
  - **azure subscription id** The subscription ID to use for the cluster. Specify the **id** value in your account output.
  - **azure tenant id** The tenant ID. Specify the **tenantId** value in your account output.
  - **azure service principal client id** The value of the **appId** parameter for the service principal.
  - **azure service principal client secret** The value of the **password** parameter for the service principal.
- iv. Select the region to deploy the cluster to.
- v. Select the base domain to deploy the cluster to. The base domain corresponds to the Azure DNS Zone that you created for your cluster.
- vi. Enter a descriptive name for your cluster.



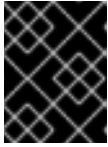
### IMPORTANT

All Azure resources that are available through public endpoints are subject to resource name restrictions, and you cannot create resources that use certain terms. For a list of terms that Azure restricts, see [Resolve reserved resource name errors](#) in the Azure documentation.

- vii. Paste the pull secret that you obtained from the [Pull Secret](#) page on the Red Hat OpenShift Cluster Manager site.
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.

#### 1.5.6.1. Installation configuration parameters

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



### NOTE

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



### IMPORTANT

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

##### 1.5.6.1.1. Required configuration parameters

Required installation configuration parameters are described in the following table:

Table 1.11. Required parameters

Parameter	Description	Values
<b>apiVersion</b>	The API version for the <b>install-config.yaml</b> content. The current version is <b>v1</b> . The installer may also support older API versions.	String
<b>baseDomain</b>	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 <b>baseDomain</b> and <b>metadata.name</b> parameter values that uses the <b>&lt;metadata.name&gt;</b> . <b>&lt;baseDomain&gt;</b> format.	A fully-qualified domain or subdomain name, such as <b>example.com</b> .

Parameter	Description	Values
<b>metadata</b>	Kubernetes resource <b>ObjectMeta</b> , from which only the <b>name</b> parameter is consumed.	Object
<b>metadata.name</b>	The name of the cluster. DNS records for the cluster are all subdomains of <b>{{.metadata.name}}</b> . <b>{{.baseDomain}}</b> .	String of lowercase letters, hyphens (-), and periods (.), such as <b>dev</b> .
<b>platform</b>	The configuration for the specific platform upon which to perform the installation: <b>aws, baremetal, azure, openstack, ovirt, vsphere</b> . For additional information about <b>platform.&lt;platform&gt;</b> parameters, consult the following table for your specific platform.	Object
<b>pullSecret</b>	Get a pull secret from <a href="https://cloud.redhat.com/openshift/install/pull-secret">https://cloud.redhat.com/openshift/install/pull-secret</a> to authenticate downloading container images for OpenShift Container Platform components from services such as Quay.io.	<pre>{   "auths":{     "cloud.openshift.com":{       "auth":"b3Blb=",       "email":"you@example.com"     },     "quay.io":{       "auth":"b3Blb=",       "email":"you@example.com"     }   } }</pre>


### 1.5.6.1.2. Network configuration parameters


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

Only IPv4 addresses are supported.

**Table 1.12. Network parameters**

Parameter	Description	Values
-----------	-------------	--------

Parameter	Description	Values
<b>networking</b>	The configuration for the cluster network.	Object  <b>NOTE</b> You cannot modify parameters specified by the <b>networking</b> object after installation.
<b>networking.networkType</b>	The cluster network provider Container Network Interface (CNI) plug-in to install.	Either <b>OpenShiftSDN</b> or <b>OVNKubernetes</b> . The default value is <b>OpenShiftSDN</b> .
<b>networking.clusterNetwork</b>	The IP address blocks for pods.  The default value is <b>10.128.0.0/14</b> with a host prefix of <b>/23</b> .  If you specify multiple IP address blocks, the blocks must not overlap.	An array of objects. For example:  <pre>networking:   clusterNetwork:     - cidr: 10.128.0.0/14       hostPrefix: 23</pre>
<b>networking.clusterNetwork.cidr</b>	Required if you use <b>networking.clusterNetwork</b> . An IP address block.  An IPv4 network.	An IP address block in Classless Inter-Domain Routing (CIDR) notation. The prefix length for an IPv4 block is between <b>0</b> and <b>32</b> .
<b>networking.clusterNetwork.hostPrefix</b>	The subnet prefix length to assign to each individual node. For example, if <b>hostPrefix</b> is set to <b>23</b> then each node is assigned a <b>/23</b> subnet out of the given <b>cidr</b> . A <b>hostPrefix</b> value of <b>23</b> provides 510 ( $2^{(32 - 23)} - 2$ ) pod IP addresses.	A subnet prefix.  The default value is <b>23</b> .
<b>networking.serviceNetwork</b>	The IP address block for services. The default value is <b>172.30.0.0/16</b> .  The OpenShift SDN and OVN-Kubernetes network providers support only a single IP address block for the service network.	An array with an IP address block in CIDR format. For example:  <pre>networking:   serviceNetwork:     - 172.30.0.0/16</pre>


Parameter	Description	Values
<b>networking.machineNetwork</b>	The IP address blocks for machines.  If you specify multiple IP address blocks, the blocks must not overlap.	An array of objects. For example:  <pre>networking:   machineNetwork:   - cidr: 10.0.0.0/16</pre>
<b>networking.machineNetwork.cidr</b>	Required if you use <b>networking.machineNetwork</b> . An IP address block. The default value is <b>10.0.0.0/16</b> for all platforms other than libvirt. For libvirt, the default value is <b>192.168.126.0/24</b> .	An IP network block in CIDR notation.  For example, <b>10.0.0.0/16</b> .   <p><b>NOTE</b></p> <p>Set the <b>networking.machineNetwork</b> to match the CIDR that the preferred NIC resides in.</p>


### 1.5.6.1.3. Optional configuration parameters



Optional installation configuration parameters are described in the following table:

Table 1.13. Optional parameters

Parameter	Description	Values
<b>additionalTrustBundle</b>	A PEM-encoded X.509 certificate bundle that is added to the nodes' trusted certificate store. This trust bundle may also be used when a proxy has been configured.	String
<b>compute</b>	The configuration for the machines that comprise the compute nodes.	Array of machine-pool objects. For details, see the following "Machine-pool" table.
<b>compute.architecture</b>	Determines the instruction set architecture of the machines in the pool. Currently, heterogeneous clusters are not supported, so all pools must specify the same architecture. Valid values are <b>amd64</b> (the default).	String

Parameter	Description	Values
<b>compute.hyperthreading</b>	<p>Whether to enable or disable simultaneous multithreading, or <b>hyperthreading</b>, on compute machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores.</p> <div style="display: flex; align-items: center;">  <div> <p><b>IMPORTANT</b></p> <p>If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.</p> </div> </div>	<b>Enabled</b> or <b>Disabled</b>
<b>compute.name</b>	Required if you use <b>compute</b> . The name of the machine pool.	<b>worker</b>
<b>compute.platform</b>	Required if you use <b>compute</b> . Use this parameter to specify the cloud provider to host the worker machines. This parameter value must match the <b>controlPlane.platform</b> parameter value.	<b>aws, azure, gcp, openstack, ovirt, vsphere</b> , or <b>{}</b>
<b>compute.replicas</b>	The number of compute machines, which are also known as worker machines, to provision.	A positive integer greater than or equal to <b>2</b> . The default value is <b>3</b> .
<b>controlPlane</b>	The configuration for the machines that comprise the control plane.	Array of <b>MachinePool</b> objects. For details, see the following "Machine-pool" table.
<b>controlPlane.architecture</b>	Determines the instruction set architecture of the machines in the pool. Currently, heterogeneous clusters are not supported, so all pools must specify the same architecture. Valid values are <b>amd64</b> (the default).	String

Parameter	Description	Values
<b>controlPlane.hypert hreading</b>	<p>Whether to enable or disable simultaneous multithreading, or <b>hyperthreading</b>, on control plane machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores.</p> <div style="display: flex; align-items: center;">  <div> <p><b>IMPORTANT</b></p> <p>If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.</p> </div> </div>	<b>Enabled</b> or <b>Disabled</b>
<b>controlPlane.name</b>	Required if you use <b>controlPlane</b> . The name of the machine pool.	<b>master</b>
<b>controlPlane.platfor m</b>	Required if you use <b>controlPlane</b> . Use this parameter to specify the cloud provider that hosts the control plane machines. This parameter value must match the <b>compute.platform</b> parameter value.	<b>aws, azure, gcp, openstack, ovirt, vsphere, or {}</b>
<b>controlPlane.replica s</b>	The number of control plane machines to provision.	The only supported value is <b>3</b> , which is the default value.

Parameter	Description	Values
<b>fips</b>	<p>Enable or disable FIPS mode. The default is <b>false</b> (disabled). If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.</p>  <p><b>NOTE</b></p> <p>If you are using Azure File storage, you cannot enable FIPS mode.</p>	<b>false</b> or <b>true</b>
<b>imageContentSources</b>	Sources and repositories for the release-image content.	Array of objects. Includes a <b>source</b> and, optionally, <b>mirrors</b> , as described in the following rows of this table.
<b>imageContentSources.source</b>	Required if you use <b>imageContentSources</b> . Specify the repository that users refer to, for example, in image pull specifications.	String
<b>imageContentSources.mirrors</b>	Specify one or more repositories that may also contain the same images.	Array of strings
<b>publish</b>	How to publish or expose the user-facing endpoints of your cluster, such as the Kubernetes API, OpenShift routes.	<b>Internal</b> or <b>External</b> . To deploy a private cluster, which cannot be accessed from the internet, set <b>publish</b> to <b>Internal</b> . The default value is <b>External</b> .
<b>sshKey</b>	<p>The SSH key to authenticate access to your cluster machines.</p>  <p><b>NOTE</b></p> <p>For production OpenShift Container Platform clusters on which you want to perform installation debugging or disaster recovery, specify an SSH key that your <b>ssh-agent</b> process uses.</p>	For example, <b>sshKey: ssh-ed25519 AAAA...</b>

#### 1.5.6.1.4. Additional Azure configuration parameters

Additional Azure configuration parameters are described in the following table:

Table 1.14. Additional Azure parameters

Parameter	Description	Values
<b>controlPlane.platform.azure.osDisk.diskSizeGB</b>	The Azure disk size for the VM.	Integer that represents the size of the disk in GB. The minimum supported disk size is <b>1024</b> .
<b>platform.azure.baseDomainResourceGroupName</b>	The name of the resource group that contains the DNS zone for your base domain.	String, for example <b>production_cluster</b> .
<b>platform.azure.region</b>	The name of the Azure region that hosts your cluster.	Any valid region name, such as <b>centralus</b> .
<b>platform.azure.zone</b>	List of availability zones to place machines in. For high availability, specify at least two zones.	List of zones, for example <b>["1", "2", "3"]</b> .
<b>platform.azure.networkResourceGroupName</b>	The name of the resource group that contains the existing VNet that you want to deploy your cluster to. This name cannot be the same as the <b>platform.azure.baseDomainResourceGroupName</b> .	String.
<b>platform.azure.virtualNetwork</b>	The name of the existing VNet that you want to deploy your cluster to.	String.
<b>platform.azure.controlPlaneSubnet</b>	The name of the existing subnet in your VNet that you want to deploy your control plane machines to.	Valid CIDR, for example <b>10.0.0.0/16</b> .
<b>platform.azure.computeSubnet</b>	The name of the existing subnet in your VNet that you want to deploy your compute machines to.	Valid CIDR, for example <b>10.0.0.0/16</b> .



#### NOTE

You cannot customize [Azure Availability Zones](#) or [Use tags to organize your Azure resources](#) with an Azure cluster.

#### 1.5.6.2. Sample customized `install-config.yaml` file for Azure

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 modify it.

```

apiVersion: v1
baseDomain: example.com 1
controlPlane: 2
  hyperthreading: Enabled 3 4
  name: master
  platform:
    azure:
      osDisk:
        diskSizeGB: 1024 5
        type: Standard_D8s_v3
      replicas: 3
compute: 6
- hyperthreading: Enabled 7
  name: worker
  platform:
    azure:
      type: Standard_D2s_v3
      osDisk:
        diskSizeGB: 512 8
      zones: 9
      - "1"
      - "2"
      - "3"
    replicas: 5
metadata:
  name: test-cluster 10
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
platform:
  azure:
    region: centralus 11
    baseDomainResourceGroupName: resource_group 12
    networkResourceGroupName: vnet_resource_group 13
    virtualNetwork: vnet 14
    controlPlaneSubnet: control_plane_subnet 15
    computeSubnet: compute_subnet 16
  pullSecret: '{"auths": ...}' 17
  fips: false 18
  sshKey: ssh-ed25519 AAAA... 19

```

1 10 11 17 Required. The installation program prompts you for this value.

- 2 6 If you do not provide these parameters and values, the installation program provides the default value.
- 3 7 The **controlPlane** section is a single mapping, but the compute section is a sequence of mappings. To meet the requirements of the different data structures, the first line of the **compute** section must begin with a hyphen, -, and the first line of the **controlPlane** section must not. Although both sections currently define a single machine pool, it is possible that future versions of OpenShift Container Platform will support defining multiple compute pools during installation. Only one control plane pool is used.
- 4 Whether to enable or disable simultaneous multithreading, or **hypertreading**. 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 virtual machine types, such as **Standard\_D8s\_v3**, for your machines if you disable simultaneous multithreading.

- 5 8 You can specify the size of the disk to use in GB. Minimum recommendation for master nodes is 1024 GB.
- 9 Specify a list of zones to deploy your machines to. For high availability, specify at least two zones.
- 12 Specify the name of the resource group that contains the DNS zone for your base domain.
- 13 If you use an existing VNet, specify the name of the resource group that contains it.
- 14 If you use an existing VNet, specify its name.
- 15 If you use an existing VNet, specify the name of the subnet to host the control plane machines.
- 16 If you use an existing VNet, specify the name of the subnet to host the compute machines.
- 18 Whether to enable or disable FIPS mode. By default, FIPS mode is not enabled. If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.
- 19 You can optionally 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.

#### 1.5.6.3. Configuring the cluster-wide proxy during installation

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

## Prerequisites

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



## NOTE

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

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

## Procedure

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

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

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

**NOTE**

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

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

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

**NOTE**

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

### 1.5.7. 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

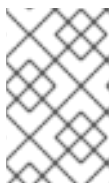
- Configure an account with the cloud platform that hosts your cluster.
- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.

#### Procedure

1. Run the installation program:

```
$ ./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**.

**NOTE**

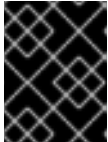
If the cloud provider account that you configured on your host does not have sufficient permissions to deploy the cluster, the installation process stops, and the missing permissions are displayed.

When the cluster deployment completes, directions for accessing your cluster, including a link to its web console and credentials for the **kubeadmin** user, display in your terminal.



### 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.

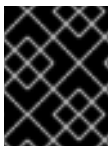


### IMPORTANT

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

## 1.5.8. Installing the CLI by downloading the binary

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



### IMPORTANT

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.5. Download and install the new version of **oc**.

### 1.5.8.1. Installing the CLI on Linux

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

#### Procedure

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

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

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

```
$ echo $PATH
```

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

```
$ oc <command>
```

### 1.5.8.2. Installing the CLI on Windows

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

### Procedure

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

```
C:\> path
```

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

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

### 1.5.8.3. Installing the CLI on macOS

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

### Procedure

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

```
$ echo $PATH
```

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

```
$ oc <command>
```

### 1.5.9. Logging in to the cluster

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

## Prerequisites

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

## Procedure

1. Export the **kubeadmin** credentials:

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

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

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

```
$ oc whoami
```

### Example output

```
system:admin
```

## 1.5.10. Next steps

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

## 1.6. INSTALLING A PRIVATE CLUSTER ON AZURE

In OpenShift Container Platform version 4.5, you can install a private cluster into an existing Azure Virtual Network (VNet) on Microsoft Azure. 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.

### 1.6.1. Prerequisites

- Review details about the [OpenShift Container Platform installation and update](#) processes.
- [Configure an Azure account](#) to host the cluster and determine the tested and validated region to deploy the cluster to.
- If you use a firewall, you must [configure it to allow the sites](#) that your cluster requires access to.
- If you do not allow the system to manage identity and access management (IAM), then a cluster administrator can [manually create and maintain IAM credentials](#). Manual mode can also be used in environments where the cloud IAM APIs are not reachable.

### 1.6.2. Private clusters

If your environment does not require an external Internet connection, 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.

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.

Additionally, you must deploy a private cluster from a machine that has access the API services for the cloud you provision to, the hosts on the network that you provision, and to the internet to obtain installation media. You can use any machine that meets these access requirements and follows your company's guidelines. For example, this machine can be a bastion host on your cloud network or a machine that has access to the network through a VPN.

### 1.6.2.1. Private clusters in Azure

To create a private cluster on Microsoft Azure, you must provide an existing private VNet 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.

Depending how your network connects to the private VNET, you might need to use a DNS forwarder in order to resolve the cluster's private DNS records. The cluster's machines use **168.63.129.16** internally for DNS resolution. For more information, see [What is Azure Private DNS?](#) and [What is IP address 168.63.129.16?](#) in the Azure documentation.

The cluster still requires access to Internet to access the Azure APIs.

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

- A **BaseDomainResourceGroup**, since the cluster does not create public records
- Public IP addresses
- Public DNS records
- Public endpoints

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.

#### 1.6.2.1.1. Limitations

Private clusters on Azure are subject to only the limitations that are associated with the use of an existing VNet.

### 1.6.3. About reusing a VNet for your OpenShift Container Platform cluster

In OpenShift Container Platform 4.5, you can deploy a cluster into an existing Azure Virtual Network (VNet) in Microsoft Azure. If you do, you must also use existing subnets within the VNet and routing rules.



By deploying OpenShift Container Platform into an existing Azure VNet, you might be able to avoid service limit constraints in new accounts or more easily abide by the operational constraints that your company's guidelines set. This is a good option to use if you cannot obtain the infrastructure creation permissions that are required to create the VNet.



### IMPORTANT

The use of an existing VNet requires the use of the updated Azure Private DNS (preview) feature. See [Announcing Preview Refresh for Azure DNS Private Zones](#) for more information about the limitations of this feature.

#### 1.6.3.1. Requirements for using your VNet

When you deploy a cluster by using an existing VNet, you must perform additional network configuration before you install the cluster. In installer-provisioned infrastructure clusters, the installer usually creates the following components, but it does not create them when you install into an existing VNet:

- Subnets
- Route tables
- VNets
- Network Security Groups

If you use a custom VNet, you must correctly configure it and its subnets for the installation program and the cluster to use. The installation program cannot subdivide network ranges for the cluster to use, set route tables for the subnets, or set VNet options like DHCP, so you must do so before you install the cluster.

The cluster must be able to access the resource group that contains the existing VNet and subnets. While all of the resources that the cluster creates are placed in a separate resource group that it creates, some network resources are used from a separate group. Some cluster Operators must be able to access resources in both resource groups. For example, the Machine API controller attaches NICs for the virtual machines that it creates to subnets from the networking resource group.

Your VNet must meet the following characteristics:

- The VNet's CIDR block must contain the **Networking.MachineCIDR** range, which is the IP address pool for cluster machines.
- The VNet and its subnets must belong to the same resource group, and the subnets must be configured to use Azure-assigned DHCP IP addresses instead of static IP addresses.

You must provide two subnets within your VNet, one for the control plane machines and one for the compute machines. Because Azure distributes machines in different availability zones within the region that you specify, your cluster will have high availability by default.

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

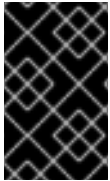
- All the subnets that you specify exist.
- You provide two private subnets, one for the control plane machines and one for the compute machines.

- The subnet CIDRs belong to the machine CIDR that you specified. Machines are not provisioned in availability zones that you do not provide private subnets for. If required, the installation program creates public load balancers that manage the control plane and worker nodes, and Azure allocates a public IP address to them.

If you destroy a cluster that uses an existing VNet, the VNet is not deleted.

### 1.6.3.1.1. Network security group requirements

The network security groups for the subnets that host the compute and control plane machines require specific access to ensure that the cluster communication is correct. You must create rules to allow access to the required cluster communication ports.



#### IMPORTANT

The network security group rules must be in place before you install the cluster. If you attempt to install a cluster without the required access, the installation program cannot reach the Azure APIs, and installation fails.

Table 1.15. Required ports

Port	Description	Control plane	Compute
80	Allows HTTP traffic		x
443	Allows HTTPS traffic		x
6443	Allows communication to the control plane machines	x	
22623	Allows communication to the machine config server	x	



#### NOTE

Since cluster components do not modify the user-provided network security groups, which the Kubernetes controllers update, a pseudo-network security group is created for the Kubernetes controller to modify without impacting the rest of the environment.

### 1.6.3.2. Division of permissions

Starting with OpenShift Container Platform 4.3, you do not need all of the permissions that are required for an installation program-provisioned infrastructure cluster to deploy a cluster. This change mimics the division of permissions that you might have at your company: some individuals can create different resources in your clouds than others. For example, you might be able to create application-specific items, like instances, storage, and load balancers, but not networking-related components such as VNets, subnet, or ingress rules.

The Azure credentials that you use when you create your cluster do not need the networking permissions that are required to make VNets and core networking components within the VNet, such as subnets, routing tables, internet gateways, NAT, and VPN. You still need permission to make the application resources that the machines within the cluster require, such as load balancers, security groups, storage accounts, and nodes.

### 1.6.3.3. Isolation between clusters

Because the cluster is unable to modify network security groups in an existing subnet, there is no way to isolate clusters from each other on the VNet.

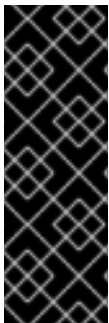
### 1.6.4. Internet and Telemetry access for OpenShift Container Platform

In OpenShift Container Platform 4.5, you require access to the Internet to install your cluster. The Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, also requires Internet access. If your cluster is connected to the Internet, Telemetry runs automatically, and your cluster is registered to the [Red Hat OpenShift Cluster Manager \(OCM\)](#).

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

You must have Internet access to:

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



#### IMPORTANT

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

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

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



#### NOTE

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

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



#### NOTE

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

## Procedure

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

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

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

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



### NOTE

If you plan to install an OpenShift Container Platform cluster that uses FIPS Validated / Modules in Process cryptographic libraries on the **x86\_64** architecture, do not create a key that uses the **ed25519** algorithm. Instead, create a key that uses the **rsa** or **ecdsa** algorithm.

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

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

### Example output

```
Agent pid 31874
```

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.

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

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

### Example output

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

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

## Next steps

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

## 1.6.6. Obtaining the installation program

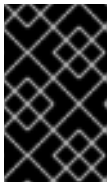
Before you install OpenShift Container Platform, download the installation file on a local computer.

## Prerequisites

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

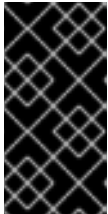
## Procedure

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



### IMPORTANT

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



### IMPORTANT

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

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

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

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

## 1.6.7. Manually creating the installation configuration file

For installations of a private OpenShift Container Platform cluster that are only accessible from an internal network and are not visible to the Internet, you must manually generate your installation configuration file.

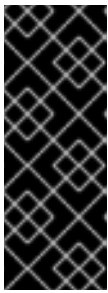
## Prerequisites

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

## Procedure

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

```
$ mkdir <installation_directory>
```

**IMPORTANT**

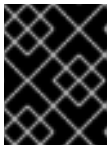
You must create a directory. Some installation assets, like bootstrap X.509 certificates have short expiration intervals, so you must not reuse an installation directory. If you want to reuse individual files from another cluster installation, you can copy them into your directory. However, the file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

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

**NOTE**

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

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

**IMPORTANT**

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

### 1.6.7.1. Installation configuration parameters

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

**NOTE**

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

**IMPORTANT**

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

#### 1.6.7.1.1. Required configuration parameters

Required installation configuration parameters are described in the following table:

Table 1.16. Required parameters

Parameter	Description	Values
-----------	-------------	--------

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


Parameter	Description	Values
<b>pullSecret</b>	Get a pull secret from <a href="https://cloud.redhat.com/openshift/install/pull-secret">https://cloud.redhat.com/openshift/install/pull-secret</a> to authenticate downloading container images for OpenShift Container Platform components from services such as Quay.io.	<pre>{   "auths":{     "cloud.openshift.com":{       "auth":"b3Blb=",       "email":"you@example.com"     },     "quay.io":{       "auth":"b3Blb=",       "email":"you@example.com"     }   } }</pre>

### 1.6.7.1.2. Network configuration parameters


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

Only IPv4 addresses are supported.

**Table 1.17. Network parameters**

Parameter	Description	Values
<b>networking</b>	The configuration for the cluster network.	Object   <b>NOTE</b> You cannot modify parameters specified by the <b>networking</b> object after installation.
<b>networking.networkType</b>	The cluster network provider Container Network Interface (CNI) plug-in to install.	Either <b>OpenShiftSDN</b> or <b>OVNKubernetes</b> . The default value is <b>OpenShiftSDN</b> .
<b>networking.clusterNetwork</b>	The IP address blocks for pods.  The default value is <b>10.128.0.0/14</b> with a host prefix of <b>/23</b> .  If you specify multiple IP address blocks, the blocks must not overlap.	An array of objects. For example:  <pre>networking:   clusterNetwork:     - cidr: 10.128.0.0/14       hostPrefix: 23</pre>





Parameter	Description	Values
<b>networking.clusterNetwork.cidr</b>	Required if you use <b>networking.clusterNetwork</b> . An IP address block.  An IPv4 network.	An IP address block in Classless Inter-Domain Routing (CIDR) notation. The prefix length for an IPv4 block is between <b>0</b> and <b>32</b> .
<b>networking.clusterNetwork.hostPrefix</b>	The subnet prefix length to assign to each individual node. For example, if <b>hostPrefix</b> is set to <b>23</b> then each node is assigned a <b>/23</b> subnet out of the given <b>cidr</b> . A <b>hostPrefix</b> value of <b>23</b> provides 510 ( $2^{(32 - 23)} - 2$ ) pod IP addresses.	A subnet prefix.  The default value is <b>23</b> .
<b>networking.serviceNetwork</b>	The IP address block for services. The default value is <b>172.30.0.0/16</b> .  The OpenShift SDN and OVN-Kubernetes network providers support only a single IP address block for the service network.	An array with an IP address block in CIDR format. For example:  <pre>networking:   serviceNetwork:     - 172.30.0.0/16</pre>
<b>networking.machineNetwork</b>	The IP address blocks for machines.  If you specify multiple IP address blocks, the blocks must not overlap.	An array of objects. For example:  <pre>networking:   machineNetwork:     - cidr: 10.0.0.0/16</pre>
<b>networking.machineNetwork.cidr</b>	Required if you use <b>networking.machineNetwork</b> . An IP address block. The default value is <b>10.0.0.0/16</b> for all platforms other than libvirt. For libvirt, the default value is <b>192.168.126.0/24</b> .	An IP network block in CIDR notation.  For example, <b>10.0.0.0/16</b> .   <b>NOTE</b>  Set the <b>networking.machineNetwork</b> to match the CIDR that the preferred NIC resides in.


### 1.6.7.1.3. Optional configuration parameters


Optional installation configuration parameters are described in the following table:

Table 1.18. Optional parameters

Parameter	Description	Values
<b>additionalTrustBundle</b>	A PEM-encoded X.509 certificate bundle that is added to the nodes' trusted certificate store. This trust bundle may also be used when a proxy has been configured.	String
<b>compute</b>	The configuration for the machines that comprise the compute nodes.	Array of machine-pool objects. For details, see the following "Machine-pool" table.
<b>compute.architecture</b>	Determines the instruction set architecture of the machines in the pool. Currently, heterogeneous clusters are not supported, so all pools must specify the same architecture. Valid values are <b>amd64</b> (the default).	String
<b>compute.hyperthreading</b>	Whether to enable or disable simultaneous multithreading, or <b>hyperthreading</b> , on compute machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores.   <b>IMPORTANT</b>  If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.	<b>Enabled</b> or <b>Disabled</b>
<b>compute.name</b>	Required if you use <b>compute</b> . The name of the machine pool.	<b>worker</b>
<b>compute.platform</b>	Required if you use <b>compute</b> . Use this parameter to specify the cloud provider to host the worker machines. This parameter value must match the <b>controlPlane.platform</b> parameter value.	<b>aws, azure, gcp, openstack, ovirt, vsphere</b> , or <b>{}</b>
<b>compute.replicas</b>	The number of compute machines, which are also known as worker machines, to provision.	A positive integer greater than or equal to <b>2</b> . The default value is <b>3</b> .

Parameter	Description	Values
<b>controlPlane</b>	The configuration for the machines that comprise the control plane.	Array of <b>MachinePool</b> objects. For details, see the following "Machine-pool" table.
<b>controlPlane.architecture</b>	Determines the instruction set architecture of the machines in the pool. Currently, heterogeneous clusters are not supported, so all pools must specify the same architecture. Valid values are <b>amd64</b> (the default).	String
<b>controlPlane.hyperthreading</b>	<p>Whether to enable or disable simultaneous multithreading, or <b>hyperthreading</b>, on control plane machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores.</p> <div style="display: flex; align-items: center;">  <div> <p><b>IMPORTANT</b></p> <p>If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.</p> </div> </div>	<b>Enabled</b> or <b>Disabled</b>
<b>controlPlane.name</b>	Required if you use <b>controlPlane</b> . The name of the machine pool.	<b>master</b>
<b>controlPlane.platform</b>	Required if you use <b>controlPlane</b> . Use this parameter to specify the cloud provider that hosts the control plane machines. This parameter value must match the <b>compute.platform</b> parameter value.	<b>aws, azure, gcp, openstack, ovirt, vsphere</b> , or <b>{}</b>
<b>controlPlane.replicas</b>	The number of control plane machines to provision.	The only supported value is <b>3</b> , which is the default value.

Parameter	Description	Values
<b>fips</b>	<p>Enable or disable FIPS mode. The default is <b>false</b> (disabled). If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.</p> <div style="display: flex; align-items: flex-start;">  <div> <p><b>NOTE</b></p> <p>If you are using Azure File storage, you cannot enable FIPS mode.</p> </div> </div>	<b>false</b> or <b>true</b>
<b>imageContentSources</b>	Sources and repositories for the release-image content.	Array of objects. Includes a <b>source</b> and, optionally, <b>mirrors</b> , as described in the following rows of this table.
<b>imageContentSources.source</b>	Required if you use <b>imageContentSources</b> . Specify the repository that users refer to, for example, in image pull specifications.	String
<b>imageContentSources.mirrors</b>	Specify one or more repositories that may also contain the same images.	Array of strings
<b>publish</b>	How to publish or expose the user-facing endpoints of your cluster, such as the Kubernetes API, OpenShift routes.	<b>Internal</b> or <b>External</b> . To deploy a private cluster, which cannot be accessed from the internet, set <b>publish</b> to <b>Internal</b> . The default value is <b>External</b> .

Parameter	Description	Values
<b>sshKey</b>	<p>The SSH key to authenticate access to your cluster machines.</p>  <p><b>NOTE</b></p> <p>For production OpenShift Container Platform clusters on which you want to perform installation debugging or disaster recovery, specify an SSH key that your <b>ssh-agent</b> process uses.</p>	For example, <b>sshKey: ssh-ed25519 AAAA...</b>

#### 1.6.7.1.4. Additional Azure configuration parameters

Additional Azure configuration parameters are described in the following table:

Table 1.19. Additional Azure parameters

Parameter	Description	Values
<b>controlPlane.platform.azure.osDisk.diskSizeGB</b>	The Azure disk size for the VM.	Integer that represents the size of the disk in GB. The minimum supported disk size is <b>1024</b> .
<b>platform.azure.baseDomainResourceGroupName</b>	The name of the resource group that contains the DNS zone for your base domain.	String, for example <b>production_cluster</b> .
<b>platform.azure.region</b>	The name of the Azure region that hosts your cluster.	Any valid region name, such as <b>centralus</b> .
<b>platform.azure.zone</b>	List of availability zones to place machines in. For high availability, specify at least two zones.	List of zones, for example <b>["1", "2", "3"]</b> .
<b>platform.azure.networkResourceGroupName</b>	The name of the resource group that contains the existing VNet that you want to deploy your cluster to. This name cannot be the same as the <b>platform.azure.baseDomainResourceGroupName</b> .	String.
<b>platform.azure.virtualNetwork</b>	The name of the existing VNet that you want to deploy your cluster to.	String.

Parameter	Description	Values
<b>platform.azure.controlPlaneSubnet</b>	The name of the existing subnet in your VNet that you want to deploy your control plane machines to.	Valid CIDR, for example <b>10.0.0.0/16</b> .
<b>platform.azure.computeSubnet</b>	The name of the existing subnet in your VNet that you want to deploy your compute machines to.	Valid CIDR, for example <b>10.0.0.0/16</b> .

**NOTE**

You cannot customize [Azure Availability Zones](#) or [Use tags to organize your Azure resources](#) with an Azure cluster.

### 1.6.7.2. Sample customized `install-config.yaml` file for Azure

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 modify it.

```

apiVersion: v1
baseDomain: example.com 1
controlPlane: 2
  hyperthreading: Enabled 3 4
  name: master
  platform:
    azure:
      osDisk:
        diskSizeGB: 1024 5
        type: Standard_D8s_v3
      replicas: 3
compute: 6
- hyperthreading: Enabled 7
  name: worker
  platform:
    azure:
      type: Standard_D2s_v3
      osDisk:
        diskSizeGB: 512 8
      zones: 9
      - "1"
      - "2"
      - "3"
  replicas: 5
metadata:

```

```

name: test-cluster 10
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  machineNetwork:
    - cidr: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
    - 172.30.0.0/16
platform:
  azure:
    region: centralus 11
    baseDomainResourceGroupName: resource_group 12
    networkResourceGroupName: vnet_resource_group 13
    virtualNetwork: vnet 14
    controlPlaneSubnet: control_plane_subnet 15
    computeSubnet: compute_subnet 16
  pullSecret: '{"auths": ...}' 17
  fips: false 18
  sshKey: ssh-ed25519 AAAA... 19
  publish: Internal 20

```

**1 10 11 17** Required. The installation program prompts you for this value.

**2 6** If you do not provide these parameters and values, the installation program provides the default value.

**3 7** The **controlPlane** section is a single mapping, but the compute section is a sequence of mappings. To meet the requirements of the different data structures, the first line of the **compute** section must begin with a hyphen, -, and the first line of the **controlPlane** section must not. Although both sections currently define a single machine pool, it is possible that future versions of OpenShift Container Platform will support defining multiple compute pools during installation. Only one control plane pool is used.

**4** Whether to enable or disable simultaneous multithreading, or **hyperthreading**. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores. You can disable it by setting the parameter value to **Disabled**. If you disable simultaneous multithreading in some cluster machines, you must disable it in all cluster machines.



### IMPORTANT

If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance. Use larger virtual machine types, such as **Standard\_D8s\_v3**, for your machines if you disable simultaneous multithreading.

**5 8** You can specify the size of the disk to use in GB. Minimum recommendation for master nodes is 1024 GB.

**9** Specify a list of zones to deploy your machines to. For high availability, specify at least two zones.

**12** Specify the name of the resource group that contains the DNS zone for your base domain.

- 13 If you use an existing VNet, specify the name of the resource group that contains it.
- 14 If you use an existing VNet, specify its name.
- 15 If you use an existing VNet, specify the name of the subnet to host the control plane machines.
- 16 If you use an existing VNet, specify the name of the subnet to host the compute machines.
- 18 Whether to enable or disable FIPS mode. By default, FIPS mode is not enabled. If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.
- 19 You can optionally 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.

- 20 How to publish the user-facing endpoints of your cluster. Set **publish** to **Internal** to deploy a private cluster, which cannot be accessed from the Internet. The default value is **External**.

### 1.6.7.3. Configuring the cluster-wide proxy during installation

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

#### Prerequisites

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



#### NOTE

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

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

#### Procedure

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

```
apiVersion: v1
```

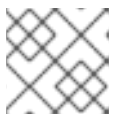


```

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

```

- 1 A proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**. If you use an MITM transparent proxy network that does not require additional proxy configuration but requires additional CAs, you must not specify an **httpProxy** value.
- 2 A proxy URL to use for creating HTTPS connections outside the cluster. If this field is not specified, then **httpProxy** is used for both HTTP and HTTPS connections. If you use an MITM transparent proxy network that does not require additional proxy configuration but requires additional CAs, you must not specify an **httpsProxy** value.
- 3 A comma-separated list of destination domain names, domains, IP addresses, or other network CIDRs to exclude proxying. Preface a domain with **.** to match subdomains only. For example, **.y.com** matches **x.y.com**, but not **y.com**. Use **\*** to bypass proxy for all destinations.
- 4 If provided, the installation program generates a config map that is named **user-ca-bundle** in the **openshift-config** namespace that contains one or more additional CA certificates that are required for proxying HTTPS connections. The Cluster Network Operator then creates a **trusted-ca-bundle** config map that merges these contents with the Red Hat Enterprise Linux CoreOS (RHCOS) trust bundle, and this config map is referenced in the **Proxy** object's **trustedCA** field. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle. If you use an MITM transparent proxy network that does not require additional proxy configuration but requires additional CAs, you must provide the MITM CA certificate.

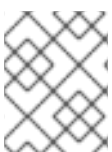


#### NOTE

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

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

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

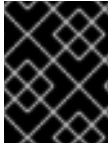


#### NOTE

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

### 1.6.8. 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

- Configure an account with the cloud platform that hosts your cluster.
- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.

### Procedure

1. Run the installation program:

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

1 For **<installation\_directory>**, specify the

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



### NOTE

If the cloud provider account that you configured on your host does not have sufficient permissions to deploy the cluster, the installation process stops, and the missing permissions are displayed.

When the cluster deployment completes, directions for accessing your cluster, including a link to its web console and credentials for the **kubeadmin** user, display in your terminal.



### 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.

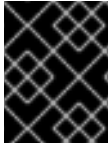


### IMPORTANT

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

## 1.6.9. Installing the CLI by downloading the binary

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



## IMPORTANT

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.5. Download and install the new version of **oc**.

### 1.6.9.1. Installing the CLI on Linux

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

#### Procedure

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

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

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

```
$ echo $PATH
```

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

```
$ oc <command>
```

### 1.6.9.2. Installing the CLI on Windows

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

#### Procedure

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

```
C:\> path
```

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

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

### 1.6.9.3. Installing the CLI on macOS

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

#### Procedure

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

```
$ echo $PATH
```

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

```
$ oc <command>
```

### 1.6.10. Logging in to the cluster

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

#### Prerequisites

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

#### Procedure

1. Export the **kubeadmin** credentials:

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

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

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

```
$ oc whoami
```

### Example output

```
system:admin
```

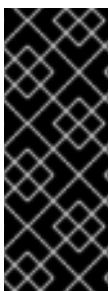
#### 1.6.11. Next steps

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

## 1.7. INSTALLING A CLUSTER ON AZURE USING ARM TEMPLATES

In OpenShift Container Platform version 4.5, you can install a cluster on Microsoft Azure by using infrastructure that you provide.

Several [Azure Resource Manager](#) (ARM) templates are provided to assist in completing these steps or to help model your own.



### IMPORTANT

The steps for performing a user-provisioned infrastructure installation are provided as an example only. Installing a cluster with infrastructure you provide requires knowledge of the cloud provider and the installation process of OpenShift Container Platform. Several ARM templates are provided to assist in completing these steps or to help model your own. You are also free to create the required resources through other methods; the templates are just an example.

#### 1.7.1. Prerequisites

- Review details about the [OpenShift Container Platform installation and update](#) processes.
- [Configure an Azure account](#) to host the cluster.
- Download the Azure CLI and install it on your computer. See [Install the Azure CLI](#) in the Azure documentation. The documentation below was last tested using version **2.2.0** of the Azure CLI. Azure CLI commands might perform differently based on the version you use.
- If you use a firewall and plan to use telemetry, you must [configure the firewall to allow the sites](#) that your cluster requires access to.
- If you do not allow the system to manage identity and access management (IAM), then a cluster administrator can [manually create and maintain IAM credentials](#) . Manual mode can also be used in environments where the cloud IAM APIs are not reachable.



### NOTE

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

#### 1.7.2. Internet and Telemetry access for OpenShift Container Platform

In OpenShift Container Platform 4.5, you require access to the Internet to install your cluster. The

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

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

You must have Internet access to:

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

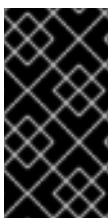


### IMPORTANT

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

## 1.7.3. Configuring your Azure project

Before you can install OpenShift Container Platform, you must configure an Azure project to host it.



### IMPORTANT

All Azure resources that are available through public endpoints are subject to resource name restrictions, and you cannot create resources that use certain terms. For a list of terms that Azure restricts, see [Resolve reserved resource name errors](#) in the Azure documentation.

### 1.7.3.1. Azure account limits

The OpenShift Container Platform cluster uses a number of Microsoft Azure components, and the default [Azure subscription and service limits, quotas, and constraints](#) affect your ability to install OpenShift Container Platform clusters.



### IMPORTANT

Default limits vary by offer category types, such as Free Trial and Pay-As-You-Go, and by series, such as Dv2, F, and G. For example, the default for Enterprise Agreement subscriptions is 350 cores.

Check the limits for your subscription type and if necessary, increase quota limits for your account before you install a default cluster on Azure.

The following table summarizes the Azure components whose limits can impact your ability to install and run OpenShift Container Platform clusters.

Component	Number of components required by default	Default Azure limit	Description
vCPU	40	20 per region	<p>A default cluster requires 40 vCPUs, so you must increase the account limit.</p> <p>By default, each cluster creates the following instances:</p> <ul style="list-style-type: none"> <li>● One bootstrap machine, which is removed after installation</li> <li>● Three control plane machines</li> <li>● Three compute machines</li> </ul> <p>Because the bootstrap machine uses <b>Standard_D4s_v3</b> machines, which use 4 vCPUs, the control plane machines use <b>Standard_D8s_v3</b> virtual machines, which use 8 vCPUs, and the worker machines use <b>Standard_D4s_v3</b> virtual machines, which use 4 vCPUs, a default cluster requires 40 vCPUs. The bootstrap node VM, which uses 4 vCPUs, is used only during installation.</p> <p>To deploy more worker nodes, enable autoscaling, deploy large workloads, or use a different instance type, you must further increase the vCPU limit for your account to ensure that your cluster can deploy the machines that you require.</p> <p>By default, the installation program distributes control plane and compute machines across <a href="#">all availability zones</a> within a <a href="#">region</a>. To ensure high availability for your cluster, select a region with at least three availability zones. If your region contains fewer than three availability zones, the installation program places more than one control plane machine in the available zones.</p>
VNet	1	1000 per region	Each default cluster requires one Virtual Network (VNet), which contains two subnets.
Network interfaces	6	65,536 per region	Each default cluster requires six network interfaces. If you create more machines or your deployed workloads create load balancers, your cluster uses more network interfaces.

Component	Number of components required by default	Default Azure limit	Description						
Network security groups	2	5000	<p>Each default cluster Each cluster creates network security groups for each subnet in the VNet. The default cluster creates network security groups for the control plane and for the compute node subnets:</p> <table border="1"> <tr> <td><b>control plane</b></td> <td>Allows the control plane machines to be reached on port 6443 from anywhere</td> </tr> <tr> <td><b>node</b></td> <td>Allows worker nodes to be reached from the Internet on ports 80 and 443</td> </tr> </table>	<b>control plane</b>	Allows the control plane machines to be reached on port 6443 from anywhere	<b>node</b>	Allows worker nodes to be reached from the Internet on ports 80 and 443		
<b>control plane</b>	Allows the control plane machines to be reached on port 6443 from anywhere								
<b>node</b>	Allows worker nodes to be reached from the Internet on ports 80 and 443								
Network load balancers	3	1000 per region	<p>Each cluster creates the following <a href="#">load balancers</a>:</p> <table border="1"> <tr> <td><b>default</b></td> <td>Public IP address that load balances requests to ports 80 and 443 across worker machines</td> </tr> <tr> <td><b>internal</b></td> <td>Private IP address that load balances requests to ports 6443 and 22623 across control plane machines</td> </tr> <tr> <td><b>external</b></td> <td>Public IP address that load balances requests to port 6443 across control plane machines</td> </tr> </table> <p>If your applications create more Kubernetes <b>LoadBalancer</b> service objects, your cluster uses more load balancers.</p>	<b>default</b>	Public IP address that load balances requests to ports 80 and 443 across worker machines	<b>internal</b>	Private IP address that load balances requests to ports 6443 and 22623 across control plane machines	<b>external</b>	Public IP address that load balances requests to port 6443 across control plane machines
<b>default</b>	Public IP address that load balances requests to ports 80 and 443 across worker machines								
<b>internal</b>	Private IP address that load balances requests to ports 6443 and 22623 across control plane machines								
<b>external</b>	Public IP address that load balances requests to port 6443 across control plane machines								
Public IP addresses	3		Each of the two public load balancers uses a public IP address. The bootstrap machine also uses a public IP address so that you can SSH into the machine to troubleshoot issues during installation. The IP address for the bootstrap node is used only during installation.						
Private IP addresses	7		The internal load balancer, each of the three control plane machines, and each of the three worker machines each use a private IP address.						

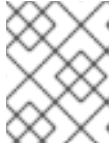
### 1.7.3.2. Configuring a public DNS zone in Azure



To install OpenShift Container Platform, the Microsoft Azure account you use must have a dedicated public hosted DNS zone in your account. This zone must be authoritative for the domain. This service provides cluster DNS resolution and name lookup for external connections to the cluster.

### Procedure

1. Identify your domain, or subdomain, and registrar. You can transfer an existing domain and registrar or obtain a new one through Azure or another source.



#### NOTE

For more information about purchasing domains through Azure, see [Buy a custom domain name for Azure App Service](#) in the Azure documentation.

2. If you are using an existing domain and registrar, migrate its DNS to Azure. See [Migrate an active DNS name to Azure App Service](#) in the Azure documentation.
3. Configure DNS for your domain. Follow the steps in the [Tutorial: Host your domain in Azure DNS](#) in the Azure documentation to create a public hosted zone for your domain or subdomain, extract the new authoritative name servers, and update the registrar records for the name servers that your domain uses.  
Use an appropriate root domain, such as **openshiftcorp.com**, or subdomain, such as **clusters.openshiftcorp.com**.
4. If you use a subdomain, follow your company's procedures to add its delegation records to the parent domain.

You can view Azure's DNS solution by visiting this [example for creating DNS zones](#).

### 1.7.3.3. Increasing Azure account limits

To increase an account limit, file a support request on the Azure portal.



#### NOTE

You can increase only one type of quota per support request.

### Procedure

1. From the Azure portal, click **Help + support** in the lower left corner.
2. Click **New support request** and then select the required values:
  - a. From the **Issue type** list, select **Service and subscription limits (quotas)**
  - b. From the **Subscription** list, select the subscription to modify.
  - c. From the **Quota type** list, select the quota to increase. For example, select **Compute-VM (cores-vCPUs) subscription limit increases** to increase the number of vCPUs, which is required to install a cluster.
  - d. Click **Next: Solutions**.
3. On the **Problem Details** page, provide the required information for your quota increase:

- a. Click **Provide details** and provide the required details in the **Quota details** window.
  - b. In the SUPPORT METHOD and CONTACT INFO sections, provide the issue severity and your contact details.
4. Click **Next: Review + create** and then click **Create**.

#### 1.7.3.4. Certificate signing requests management

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

#### 1.7.3.5. Required Azure roles

Your Microsoft Azure account must have the following roles for the subscription that you use:

- **User Access Administrator**

To set roles on the Azure portal, see the [Manage access to Azure resources using RBAC and the Azure portal](#) in the Azure documentation.

#### 1.7.3.6. Creating a service principal

Because OpenShift Container Platform and its installation program must create Microsoft Azure resources through Azure Resource Manager, you must create a service principal to represent it.

#### Prerequisites

- Install or update the [Azure CLI](#).
- Install the **jq** package.
- Your Azure account has the required roles for the subscription that you use.

#### Procedure

1. Log in to the Azure CLI:

```
$ az login
```

Log in to Azure in the web console by using your credentials.

2. If your Azure account uses subscriptions, ensure that you are using the right subscription.
  - a. View the list of available accounts and record the **tenantId** value for the subscription you want to use for your cluster:

```
$ az account list --refresh
```

#### Example output

```
[
  {
    "cloudName": "AzureCloud",
    "id": "9bab1460-96d5-40b3-a78e-17b15e978a80",
    "isDefault": true,
    "name": "Subscription Name",
    "state": "Enabled",
    "tenantId": "6057c7e9-b3ae-489d-a54e-de3f6bf6a8ee",
    "user": {
      "name": "you@example.com",
      "type": "user"
    }
  }
]
```

- b. View your active account details and confirm that the **tenantId** value matches the subscription you want to use:

```
$ az account show
```

### Example output

```
{
  "environmentName": "AzureCloud",
  "id": "9bab1460-96d5-40b3-a78e-17b15e978a80",
  "isDefault": true,
  "name": "Subscription Name",
  "state": "Enabled",
  "tenantId": "6057c7e9-b3ae-489d-a54e-de3f6bf6a8ee", 1
  "user": {
    "name": "you@example.com",
    "type": "user"
  }
}
```

- 1** Ensure that the value of the **tenantId** parameter is the UUID of the correct subscription.

- c. If you are not using the right subscription, change the active subscription:

```
$ az account set -s <id> 1
```

- 1** Substitute the value of the **id** for the subscription that you want to use for **<id>**.

- d. If you changed the active subscription, display your account information again:

```
$ az account show
```

### Example output

```
{
```

```

"environmentName": "AzureCloud",
"id": "33212d16-bdf6-45cb-b038-f6565b61edda",
"isDefault": true,
"name": "Subscription Name",
"state": "Enabled",
"tenantId": "8049c7e9-c3de-762d-a54e-dc3f6be6a7ee",
"user": {
  "name": "you@example.com",
  "type": "user"
}
}

```

- Record the values of the **tenantId** and **id** parameters from the previous output. You need these values during OpenShift Container Platform installation.
- Create the service principal for your account:

```
$ az ad sp create-for-rbac --role Contributor --name <service_principal> 1
```

- Replace **<service\_principal>** with the name to assign to the service principal.

### Example output

```

Changing "<service_principal>" to a valid URI of "http://<service_principal>", which is the
required format used for service principal names
Retrying role assignment creation: 1/36
Retrying role assignment creation: 2/36
Retrying role assignment creation: 3/36
Retrying role assignment creation: 4/36
{
  "appId": "8bd0d04d-0ac2-43a8-928d-705c598c6956",
  "displayName": "<service_principal>",
  "name": "http://<service_principal>",
  "password": "ac461d78-bf4b-4387-ad16-7e32e328aec6",
  "tenant": "6048c7e9-b2ad-488d-a54e-dc3f6be6a7ee"
}

```

- Record the values of the **appId** and **password** parameters from the previous output. You need these values during OpenShift Container Platform installation.
- Grant additional permissions to the service principal. The service principal requires the legacy **Azure Active Directory Graph** → **Application.ReadWrite.OwnedBy** permission and the **User Access Administrator** role for the cluster to assign credentials for its components.
  - To assign the **User Access Administrator** role, run the following command:

```

$ az role assignment create --role "User Access Administrator" \
  --assignee-object-id $(az ad sp list --filter "appId eq '<appId>'" \
  | jq '[0].objectId' -r) 1

```

- Replace **<appId>** with the **appId** parameter value for your service principal.

- To assign the **Azure Active Directory Graph** permission, run the following command:

```
$ az ad app permission add --id <appld> \ 1
--api 00000002-0000-0000-c000-000000000000 \
--api-permissions 824c81eb-e3f8-4ee6-8f6d-de7f50d565b7=Role
```

- 1 Replace **<appld>** with the **appld** parameter value for your service principal.

### Example output

```
Invoking "az ad app permission grant --id 46d33abc-b8a3-46d8-8c84-f0fd58177435 --api
00000002-0000-0000-c000-000000000000" is needed to make the change effective
```

For more information about the specific permissions that you grant with this command, see the [GUID Table for Windows Azure Active Directory Permissions](#) .

- c. Approve the permissions request. If your account does not have the Azure Active Directory tenant administrator role, follow the guidelines for your organization to request that the tenant administrator approve your permissions request.

```
$ az ad app permission grant --id <appld> \ 1
--api 00000002-0000-0000-c000-000000000000
```

- 1 Replace **<appld>** with the **appld** parameter value for your service principal.

### 1.7.3.7. Supported Azure regions

The installation program dynamically generates the list of available Microsoft Azure regions based on your subscription. The following Azure regions were tested and validated in OpenShift Container Platform version 4.5.4:

- **australiacentral** (Australia Central)
- **australiaeast** (Australia East)
- **australiasoutheast** (Australia South East)
- **brazilsouth** (Brazil South)
- **canadacentral** (Canada Central)
- **canadaeast** (Canada East)
- **centralindia** (Central India)
- **centralus** (Central US)
- **eastasia** (East Asia)
- **eastus** (East US)
- **eastus2** (East US 2)
- **francecentral** (France Central)

- **germanywestcentral** (Germany West Central)
- **japaneast** (Japan East)
- **japanwest** (Japan West)
- **koreacentral** (Korea Central)
- **koreasouth** (Korea South)
- **northcentralus** (North Central US)
- **northeurope** (North Europe)
- **norwayeast** (Norway East)
- **southafricanorth** (South Africa North)
- **southcentralus** (South Central US)
- **southeastasia** (Southeast Asia)
- **southindia** (South India)
- **switzerlandnorth** (Switzerland North)
- **uaenorth** (UAE North)
- **uksouth** (UK South)
- **ukwest** (UK West)
- **westcentralus** (West Central US)
- **westeurope** (West Europe)
- **westindia** (West India)
- **westus** (West US)
- **westus2** (West US 2)

#### 1.7.4. Obtaining the installation program

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

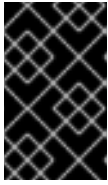
##### Prerequisites

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

##### Procedure

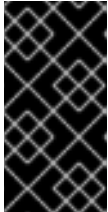
1. Access the [Infrastructure Provider](#) page on the Red Hat OpenShift Cluster Manager site. If you have a Red Hat account, log in with your credentials. If you do not, create an account.

2. Navigate to the page for your installation type, download the installation program for your operating system, and place the file in the directory where you will store the installation configuration files.



### IMPORTANT

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



### IMPORTANT

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

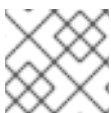
3. Extract the installation program. For example, on a computer that uses a Linux operating system, run the following command:

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

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

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

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



### NOTE

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

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



### NOTE

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

### Procedure

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

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

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

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



#### NOTE

If you plan to install an OpenShift Container Platform cluster that uses FIPS Validated / Modules in Process cryptographic libraries on the **x86\_64** architecture, do not create a key that uses the **ed25519** algorithm. Instead, create a key that uses the **rsa** or **ecdsa** algorithm.

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

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

#### Example output

```
Agent pid 31874
```

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.

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

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

#### Example output

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

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

### Next steps

- When you install OpenShift Container Platform, provide the SSH public key to the installation program. If you install a cluster on infrastructure that you provision, you must provide this key to your cluster's machines.

## 1.7.6. Creating the installation files for Azure

To install OpenShift Container Platform on Microsoft Azure using user-provisioned infrastructure, you must generate the files that the installation program needs to deploy your cluster and modify them so that the cluster creates only the machines that it will use. You generate and customize the **install-config.yaml** file, Kubernetes manifests, and Ignition config files.

### 1.7.6.1. Creating the installation configuration file

You can customize the OpenShift Container Platform cluster you install on Microsoft Azure.



## Prerequisites

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

## Procedure

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

- a. Run the following command:

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

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



### IMPORTANT

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

- 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 **azure** as the platform to target.
- iii. If you do not have a Microsoft Azure profile stored on your computer, specify the following Azure parameter values for your subscription and service principal:
  - **azure subscription id** The subscription ID to use for the cluster. Specify the **id** value in your account output.
  - **azure tenant id** The tenant ID. Specify the **tenantId** value in your account output.
  - **azure service principal client id** The value of the **appId** parameter for the service principal.
  - **azure service principal client secret** The value of the **password** parameter for the service principal.
- iv. Select the region to deploy the cluster to.

- v. Select the base domain to deploy the cluster to. The base domain corresponds to the Azure DNS Zone that you created for your cluster.
- vi. Enter a descriptive name for your cluster.



### IMPORTANT

All Azure resources that are available through public endpoints are subject to resource name restrictions, and you cannot create resources that use certain terms. For a list of terms that Azure restricts, see [Resolve reserved resource name errors](#) in the Azure documentation.

- vii. Paste the pull secret that you obtained from the [Pull Secret](#) page on the Red Hat OpenShift Cluster Manager site.
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.

#### 1.7.6.2. Configuring the cluster-wide proxy during installation

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

#### Prerequisites

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



### NOTE

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

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

#### Procedure

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

```

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

```

- 1 A proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**. If you use an MITM transparent proxy network that does not require additional proxy configuration but requires additional CAs, you must not specify an **httpProxy** value.
- 2 A proxy URL to use for creating HTTPS connections outside the cluster. If this field is not specified, then **httpProxy** is used for both HTTP and HTTPS connections. If you use an MITM transparent proxy network that does not require additional proxy configuration but requires additional CAs, you must not specify an **httpsProxy** value.
- 3 A comma-separated list of destination domain names, domains, IP addresses, or other network CIDRs to exclude proxying. Preface a domain with **.** to match subdomains only. For example, **.y.com** matches **x.y.com**, but not **y.com**. Use **\*** to bypass proxy for all destinations.
- 4 If provided, the installation program generates a config map that is named **user-ca-bundle** in the **openshift-config** namespace that contains one or more additional CA certificates that are required for proxying HTTPS connections. The Cluster Network Operator then creates a **trusted-ca-bundle** config map that merges these contents with the Red Hat Enterprise Linux CoreOS (RHCOS) trust bundle, and this config map is referenced in the **Proxy** object's **trustedCA** field. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle. If you use an MITM transparent proxy network that does not require additional proxy configuration but requires additional CAs, you must provide the MITM CA certificate.



#### NOTE

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

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

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



#### NOTE

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

### 1.7.6.3. Exporting common variables for ARM templates

You must export a common set of variables that are used with the provided Azure Resource Manager (ARM) templates used to assist in completing a user-provided infrastructure install on Microsoft Azure.



## NOTE

Specific ARM templates can also require additional exported variables, which are detailed in their related procedures.

## Prerequisites

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

## Procedure

1. Export common variables found in the **install-config.yaml** to be used by the provided ARM templates:

```
$ export CLUSTER_NAME=<cluster_name> 1
$ export AZURE_REGION=<azure_region> 2
$ export SSH_KEY=<ssh_key> 3
$ export BASE_DOMAIN=<base_domain> 4
$ export BASE_DOMAIN_RESOURCE_GROUP=<base_domain_resource_group> 5
```

- 1 The value of the **.metadata.name** attribute from the **install-config.yaml** file.
- 2 The region to deploy the cluster into, for example **centralus**. This is the value of the **.platform.azure.region** attribute from the **install-config.yaml** file.
- 3 The SSH RSA public key file as a string. You must enclose the SSH key in quotes since it contains spaces. This is the value of the **.sshKey** attribute from the **install-config.yaml** file.
- 4 The base domain to deploy the cluster to. The base domain corresponds to the public DNS zone that you created for your cluster. This is the value of the **.baseDomain** attribute from the **install-config.yaml** file.
- 5 The resource group where the public DNS zone exists. This is the value of the **.platform.azure.baseDomainResourceGroupName** attribute from the **install-config.yaml** file.

For example:

```
$ export CLUSTER_NAME=test-cluster
$ export AZURE_REGION=centralus
$ export SSH_KEY="ssh-rsa xxx/xxx/xxx= user@email.com"
$ export BASE_DOMAIN=example.com
$ export BASE_DOMAIN_RESOURCE_GROUP=ocp-cluster
```

2. Export the kubeadmin credentials:

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

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

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

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



#### IMPORTANT

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

#### Prerequisites

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

#### Procedure

1. Generate the Kubernetes manifests for the cluster:

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

#### Example output

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

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

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

2. Remove the Kubernetes manifest files that define the control plane machines:

```
$ rm -f <installation_directory>/openshift/99_openshift-cluster-api_master-machines-*.yaml
```

By removing these files, you prevent the cluster from automatically generating control plane machines.

3. Remove the Kubernetes manifest files that define the worker machines:

```
$ rm -f <installation_directory>/openshift/99_openshift-cluster-api_worker-machineset-*.yaml
```

Because you create and manage the worker machines yourself, you do not need to initialize these machines.

4. Modify the `<installation_directory>/manifests/cluster-scheduler-02-config.yml` Kubernetes manifest file to prevent pods from being scheduled on the control plane machines:
  - a. Open the `<installation_directory>/manifests/cluster-scheduler-02-config.yml` file.
  - b. Locate the `mastersSchedulable` parameter and set its value to **False**.
  - c. Save and exit the file.
5. Optional: If you do not want [the Ingress Operator](#) to create DNS records on your behalf, remove the `privateZone` and `publicZone` sections from the `<installation_directory>/manifests/cluster-dns-02-config.yml` DNS configuration file:

```
apiVersion: config.openshift.io/v1
kind: DNS
metadata:
  creationTimestamp: null
  name: cluster
spec:
  baseDomain: example.openshift.com
  privateZone: ❶
    id: mycluster-100419-private-zone
  publicZone: ❷
    id: example.openshift.com
status: {}
```

- ❶ ❷ Remove this section completely.

If you do so, you must add ingress DNS records manually in a later step.

6. When configuring Azure on user-provisioned infrastructure, you must export some common variables defined in the manifest files to use later in the Azure Resource Manager (ARM) templates:
  - a. Export the infrastructure ID by using the following command:

```
$ export INFRA_ID=<infra_id> ❶
```

- ❶ The OpenShift Container Platform cluster has been assigned an identifier (`INFRA_ID`) in the form of `<cluster_name>-<random_string>`. This will be used as the base name for most resources created using the provided ARM templates. This is the value of the `.status.infrastructureName` attribute from the `manifests/cluster-infrastructure-02-config.yml` file.

- b. Export the resource group by using the following command:

```
$ export RESOURCE_GROUP=<resource_group> ❶
```

- ❶ All resources created in this Azure deployment exists as part of a [resource group](#). The resource group name is also based on the `INFRA_ID`, in the form of `<cluster_name>-<random_string>-rg`. This is the value of the

`.status.platformStatus.azure.resourceGroupName` attribute from the `manifests/cluster-infrastructure-02-config.yml` file.

7. Obtain the Ignition config files:

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

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

The following files are generated in the directory:

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

### 1.7.7. Creating the Azure resource group and identity

You must create a Microsoft Azure [resource group](#) and an identity for that resource group. These are both used during the installation of your OpenShift Container Platform cluster on Azure.

#### Prerequisites

- Configure an Azure account.
- Generate the Ignition config files for your cluster.

#### Procedure

1. Create the resource group in a supported Azure region:

```
┌ $ az group create --name ${RESOURCE_GROUP} --location ${AZURE_REGION}
```

2. Create an Azure identity for the resource group:

```
┌ $ az identity create -g ${RESOURCE_GROUP} -n ${INFRA_ID}-identity
```

This is used to grant the required access to Operators in your cluster. For example, this allows the Ingress Operator to create a public IP and its load balancer. You must assign the Azure identity to a role.

3. Grant the Contributor role to the Azure identity:

a. Export the following variables required by the Azure role assignment:

```
┌ $ export PRINCIPAL_ID=`az identity show -g ${RESOURCE_GROUP} -n ${INFRA_ID}-
identity --query principalId --out tsv`
```

```
$ export RESOURCE_GROUP_ID=`az group show -g ${RESOURCE_GROUP} --query id --out tsv`
```

- b. Assign the Contributor role to the identity:

```
$ az role assignment create --assignee "${PRINCIPAL_ID}" --role 'Contributor' --scope "${RESOURCE_GROUP_ID}"
```

### 1.7.8. Uploading the RHCOS cluster image and bootstrap Ignition config file

The Azure client does not support deployments based on files existing locally; therefore, you must copy and store the RHCOS virtual hard disk (VHD) cluster image and bootstrap Ignition config file in a storage container so they are accessible during deployment.

#### Prerequisites

- Configure an Azure account.
- Generate the Ignition config files for your cluster.

#### Procedure

1. Create an Azure storage account to store the VHD cluster image:

```
$ az storage account create -g ${RESOURCE_GROUP} --location ${AZURE_REGION} --name ${CLUSTER_NAME}sa --kind Storage --sku Standard_LRS
```



#### WARNING

The Azure storage account name must be between 3 and 24 characters in length and use numbers and lower-case letters only. If your **CLUSTER\_NAME** variable does not follow these restrictions, you must manually define the Azure storage account name. For more information on Azure storage account name restrictions, see [Resolve errors for storage account names](#) in the Azure documentation.

2. Export the storage account key as an environment variable:

```
$ export ACCOUNT_KEY=`az storage account keys list -g ${RESOURCE_GROUP} --account-name ${CLUSTER_NAME}sa --query "[0].value" -o tsv`
```

3. Choose the RHCOS version to use and export the URL of its VHD to an environment variable:

```
$ export VHD_URL=`curl -s https://raw.githubusercontent.com/openshift/installer/release-4.5/data/data/rhcos.json | jq -r .azure.url`
```





## IMPORTANT

The RHCOS images might not change with every release of OpenShift Container Platform. You must specify an image with the highest version that is less than or equal to the OpenShift Container Platform version that you install. Use the image version that matches your OpenShift Container Platform version if it is available.

4. Copy the chosen VHD to a blob:

```
$ az storage container create --name vhd --account-name ${CLUSTER_NAME}sa --account-key ${ACCOUNT_KEY}
```

```
$ az storage blob copy start --account-name ${CLUSTER_NAME}sa --account-key ${ACCOUNT_KEY} --destination-blob "rhcos.vhd" --destination-container vhd --source-uri "${VHD_URL}"
```

To track the progress of the VHD copy task, run this script:

```
status="unknown"
while [ "$status" != "success" ]
do
  status=`az storage blob show --container-name vhd --name "rhcos.vhd" --account-name ${CLUSTER_NAME}sa --account-key ${ACCOUNT_KEY} -o tsv --query properties.copy.status`
  echo $status
done
```

5. Create a blob storage container and upload the generated **bootstrap.ign** file:

```
$ az storage container create --name files --account-name ${CLUSTER_NAME}sa --account-key ${ACCOUNT_KEY} --public-access blob
```

```
$ az storage blob upload --account-name ${CLUSTER_NAME}sa --account-key ${ACCOUNT_KEY} -c "files" -f "<installation_directory>/bootstrap.ign" -n "bootstrap.ign"
```

### 1.7.9. Example for creating DNS zones

DNS records are required for clusters that use user-provisioned infrastructure. You should choose the DNS strategy that fits your scenario.

For this example, [Azure's DNS solution](#) is used, so you will create a new public DNS zone for external (internet) visibility and a private DNS zone for internal cluster resolution.



## NOTE

The public DNS zone is not required to exist in the same resource group as the cluster deployment and might already exist in your organization for the desired base domain. If that is the case, you can skip creating the public DNS zone; be sure the installation config you generated earlier reflects that scenario.

### Prerequisites

- Configure an Azure account.

- Generate the Ignition config files for your cluster.

## Procedure

1. Create the new public DNS zone in the resource group exported in the **BASE\_DOMAIN\_RESOURCE\_GROUP** environment variable:

```
$ az network dns zone create -g ${BASE_DOMAIN_RESOURCE_GROUP} -n
${CLUSTER_NAME}.${BASE_DOMAIN}
```

You can skip this step if you are using a public DNS zone that already exists.

2. Create the private DNS zone in the same resource group as the rest of this deployment:

```
$ az network private-dns zone create -g ${RESOURCE_GROUP} -n
${CLUSTER_NAME}.${BASE_DOMAIN}
```

You can learn more about [configuring a public DNS zone in Azure](#) by visiting that section.

### 1.7.10. Creating a VNet in Azure

You must create a virtual network (VNet) in Microsoft Azure for your OpenShift Container Platform cluster to use. You can customize the VNet to meet your requirements. One way to create the VNet is to modify the provided Azure Resource Manager (ARM) template.



#### NOTE

If you do not use the provided ARM template to create your Azure infrastructure, you must review the provided information and manually create the infrastructure. If your cluster does not initialize correctly, you might have to contact Red Hat support with your installation logs.

## Prerequisites

- Configure an Azure account.
- Generate the Ignition config files for your cluster.

## Procedure

1. Copy the template from the **ARM template for the VNet** section of this topic and save it as **01\_vnet.json** in your cluster's installation directory. This template describes the VNet that your cluster requires.
2. Create the deployment by using the **az** CLI:

```
$ az deployment group create -g ${RESOURCE_GROUP} \
  --template-file "<installation_directory>/01_vnet.json" \
  --parameters baseName="${INFRA_ID}" 1
```

- 1** The base name to be used in resource names; this is usually the cluster's infrastructure ID.

3. Link the VNet template to the private DNS zone:

```
$ az network private-dns link vnet create -g ${RESOURCE_GROUP} -z
${CLUSTER_NAME}.${BASE_DOMAIN} -n ${INFRA_ID}-network-link -v "${INFRA_ID}-vnet"
-e false
```

### 1.7.10.1. ARM template for the VNet

You can use the following Azure Resource Manager (ARM) template to deploy the VNet that you need for your OpenShift Container Platform cluster:

#### Example 1.1. 01\_vnet.json ARM template

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    }
  },
  "variables" : {
    "location" : "[resourceGroup().location]",
    "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
    "addressPrefix" : "10.0.0.0/16",
    "masterSubnetName" : "[concat(parameters('baseName'), '-master-subnet')]",
    "masterSubnetPrefix" : "10.0.0.0/24",
    "nodeSubnetName" : "[concat(parameters('baseName'), '-worker-subnet')]",
    "nodeSubnetPrefix" : "10.0.1.0/24",
    "clusterNsgName" : "[concat(parameters('baseName'), '-nsg')]"
  },
  "resources" : [
    {
      "apiVersion" : "2018-12-01",
      "type" : "Microsoft.Network/virtualNetworks",
      "name" : "[variables('virtualNetworkName')]",
      "location" : "[variables('location')]",
      "dependsOn" : [
        "[concat('Microsoft.Network/networkSecurityGroups/', variables('clusterNsgName'))]"
      ],
      "properties" : {
        "addressSpace" : {
          "addressPrefixes" : [
            "[variables('addressPrefix')]"
          ]
        },
        "subnets" : [
          {
            "name" : "[variables('masterSubnetName')]",
            "properties" : {
              "addressPrefix" : "[variables('masterSubnetPrefix')]",
              "serviceEndpoints": [],

```

```

        "networkSecurityGroup" : {
            "id" : "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('clusterNsgName'))]"
        }
    },
    {
        "name" : "[variables('nodeSubnetName')]",
        "properties" : {
            "addressPrefix" : "[variables('nodeSubnetPrefix')]",
            "serviceEndpoints" : [],
            "networkSecurityGroup" : {
                "id" : "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('clusterNsgName'))]"
            }
        }
    }
]
},
{
    "type" : "Microsoft.Network/networkSecurityGroups",
    "name" : "[variables('clusterNsgName')]",
    "apiVersion" : "2018-10-01",
    "location" : "[variables('location')]",
    "properties" : {
        "securityRules" : [
            {
                "name" : "apiserver_in",
                "properties" : {
                    "protocol" : "Tcp",
                    "sourcePortRange" : "*",
                    "destinationPortRange" : "6443",
                    "sourceAddressPrefix" : "*",
                    "destinationAddressPrefix" : "*",
                    "access" : "Allow",
                    "priority" : 101,
                    "direction" : "Inbound"
                }
            }
        ]
    }
}
]
}
}

```

### 1.7.11. Deploying the RHCOS cluster image for the Azure infrastructure

You must use a valid Red Hat Enterprise Linux CoreOS (RHCOS) image for Microsoft Azure for your OpenShift Container Platform nodes.

#### Prerequisites

- Configure an Azure account.

- Generate the Ignition config files for your cluster.
- Store the RHCOS virtual hard disk (VHD) cluster image in an Azure storage container.
- Store the bootstrap Ignition config file in an Azure storage container.

## Procedure

1. Copy the template from the **ARM template for image storage** section of this topic and save it as **02\_storage.json** in your cluster's installation directory. This template describes the image storage that your cluster requires.
2. Export the RHCOS VHD blob URL as a variable:

```
$ export VHD_BLOB_URL=`az storage blob url --account-name ${CLUSTER_NAME}sa --
account-key ${ACCOUNT_KEY} -c vhd -n "rhcos.vhd" -o tsv`
```

3. Deploy the cluster image:

```
$ az deployment group create -g ${RESOURCE_GROUP} \
--template-file "<installation_directory>/02_storage.json" \
--parameters vhdBlobURL="${VHD_BLOB_URL}" \ 1
--parameters baseName="${INFRA_ID}" 2
```

**1** The blob URL of the RHCOS VHD to be used to create master and worker machines.

**2** The base name to be used in resource names; this is usually the cluster's infrastructure ID.

### 1.7.11.1. ARM template for image storage

You can use the following Azure Resource Manager (ARM) template to deploy the stored Red Hat Enterprise Linux CoreOS (RHCOS) image that you need for your OpenShift Container Platform cluster:

#### Example 1.2. 02\_storage.json ARM template

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-
01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "vhdBlobURL" : {
      "type" : "string",
      "metadata" : {
        "description" : "URL pointing to the blob where the VHD to be used to create master and
worker machines is located"
      }
    }
  }
}
```

```

},
"variables" : {
  "location" : "[resourceGroup().location]",
  "imageName" : "[concat(parameters('baseName'), '-image')]"
},
"resources" : [
  {
    "apiVersion" : "2018-06-01",
    "type": "Microsoft.Compute/images",
    "name": "[variables('imageName')]",
    "location" : "[variables('location')]",
    "properties": {
      "storageProfile": {
        "osDisk": {
          "osType": "Linux",
          "osState": "Generalized",
          "blobUri": "[parameters('vhdBlobURL')]",
          "storageAccountType": "Standard_LRS"
        }
      }
    }
  }
]
}

```

### 1.7.12. Networking requirements for user-provisioned infrastructure

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

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

**Table 1.20. All machines to all machines**

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

Protocol	Port	Description
	<b>9000-9999</b>	Host level services, including the node exporter on ports <b>9100-9101</b> .
TCP/UDP	<b>30000-32767</b>	Kubernetes node port

Table 1.21. All machines to control plane

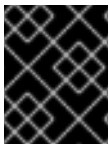
Protocol	Port	Description
TCP	<b>6443</b>	Kubernetes API

Table 1.22. Control plane machines to control plane machines

Protocol	Port	Description
TCP	<b>2379-2380</b>	etcd server and peer ports

### Network topology requirements

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



#### IMPORTANT

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

### Load balancers

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

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



#### NOTE

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

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

Table 1.23. API load balancer

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

**NOTE**

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

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

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

**Table 1.24. Application Ingress load balancer**

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



**TIP**

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

**NOTE**

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

**1.7.13. Creating networking and load balancing components in Azure**

You must configure networking and load balancing in Microsoft Azure for your OpenShift Container Platform cluster to use. One way to create these components is to modify the provided Azure Resource Manager (ARM) template.

**NOTE**

If you do not use the provided ARM template to create your Azure infrastructure, you must review the provided information and manually create the infrastructure. If your cluster does not initialize correctly, you might have to contact Red Hat support with your installation logs.

**Prerequisites**

- Configure an Azure account.
- Generate the Ignition config files for your cluster.
- Create and configure a VNet and associated subnets in Azure.

**Procedure**

1. Copy the template from the **ARM template for the network and load balancers** section of this topic and save it as **03\_infra.json** in your cluster's installation directory. This template describes the networking and load balancing objects that your cluster requires.
2. Create the deployment by using the **az** CLI:

```
$ az deployment group create -g ${RESOURCE_GROUP} \
  --template-file "<installation_directory>/03_infra.json" \
  --parameters privateDNSZoneName="${CLUSTER_NAME}.${BASE_DOMAIN}" \ 1
  --parameters baseName="${INFRA_ID}" 2
```

- 1** The name of the private DNS zone.
- 2** The base name to be used in resource names; this is usually the cluster's infrastructure ID.

3. Create an **api** DNS record in the public zone for the API public load balancer. The **\${BASE\_DOMAIN\_RESOURCE\_GROUP}** variable must point to the resource group where the public DNS zone exists.
  - a. Export the following variable:

```
$ export PUBLIC_IP=`az network public-ip list -g ${RESOURCE_GROUP} --query "[?name=='${INFRA_ID}-master-pip'] | [0].ipAddress" -o tsv`
```

- b. Create the DNS record in a new public zone:

```
$ az network dns record-set a add-record -g ${BASE_DOMAIN_RESOURCE_GROUP} -z ${CLUSTER_NAME}.${BASE_DOMAIN} -n api -a ${PUBLIC_IP} --ttl 60
```

- c. If you are adding the cluster to an existing public zone, you can create the DNS record in it instead:

```
$ az network dns record-set a add-record -g ${BASE_DOMAIN_RESOURCE_GROUP} -z ${BASE_DOMAIN} -n api.${CLUSTER_NAME} -a ${PUBLIC_IP} --ttl 60
```

### 1.7.13.1. ARM template for the network and load balancers

You can use the following Azure Resource Manager (ARM) template to deploy the networking objects and load balancers that you need for your OpenShift Container Platform cluster:

#### Example 1.3. 03\_infra.json ARM template

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "privateDNSZoneName" : {
      "type" : "string",
      "metadata" : {
        "description" : "Name of the private DNS zone"
      }
    }
  },
  "variables" : {
    "location" : "[resourceGroup().location]",
    "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
    "virtualNetworkID" : "[resourceId('Microsoft.Network/virtualNetworks', variables('virtualNetworkName'))]",
    "masterSubnetName" : "[concat(parameters('baseName'), '-master-subnet')]",
    "masterSubnetRef" : "[concat(variables('virtualNetworkID'), '/subnets/', variables('masterSubnetName'))]",
    "masterPublicIpAddressName" : "[concat(parameters('baseName'), '-master-pip')]",
    "masterPublicIpAddressID" : "[resourceId('Microsoft.Network/publicIPAddresses', variables('masterPublicIpAddressName'))]",
    "masterLoadBalancerName" : "[concat(parameters('baseName'), '-public-lb')]",
    "masterLoadBalancerID" : "[resourceId('Microsoft.Network/loadBalancers', variables('masterLoadBalancerName'))]",
```

```

    "internalLoadBalancerName" : "[concat(parameters('baseName'), '-internal-lb')]",
    "internalLoadBalancerID" : "[resourceId('Microsoft.Network/loadBalancers',
variables('internalLoadBalancerName'))]",
    "skuName": "Standard"
  },
  "resources" : [
    {
      "apiVersion" : "2018-12-01",
      "type" : "Microsoft.Network/publicIPAddresses",
      "name" : "[variables('masterPublicIpAddressName')]",
      "location" : "[variables('location')]",
      "sku": {
        "name": "[variables('skuName')]"
      },
      "properties" : {
        "publicIPAllocationMethod" : "Static",
        "dnsSettings" : {
          "domainNameLabel" : "[variables('masterPublicIpAddressName')]"
        }
      }
    },
    {
      "apiVersion" : "2018-12-01",
      "type" : "Microsoft.Network/loadBalancers",
      "name" : "[variables('masterLoadBalancerName')]",
      "location" : "[variables('location')]",
      "sku": {
        "name": "[variables('skuName')]"
      },
      "dependsOn" : [
        "[concat('Microsoft.Network/publicIPAddresses/', variables('masterPublicIpAddressName'))]"
      ],
      "properties" : {
        "frontendIPConfigurations" : [
          {
            "name" : "public-lb-ip",
            "properties" : {
              "publicIPAddress" : {
                "id" : "[variables('masterPublicIpAddressID')]"
              }
            }
          }
        ],
        "backendAddressPools" : [
          {
            "name" : "public-lb-backend"
          }
        ],
        "loadBalancingRules" : [
          {
            "name" : "api-internal",
            "properties" : {
              "frontendIPConfiguration" : {
                "id" : "[concat(variables('masterLoadBalancerID'), '/frontendIPConfigurations/public-lb-
ip')]"
              },

```

```

    "backendAddressPool" : {
      "id" : "[concat(variables('masterLoadBalancerID'), '/backendAddressPools/public-lb-backend']]"
    },
    "protocol" : "Tcp",
    "loadDistribution" : "Default",
    "idleTimeoutInMinutes" : 30,
    "frontendPort" : 6443,
    "backendPort" : 6443,
    "probe" : {
      "id" : "[concat(variables('masterLoadBalancerID'), '/probes/api-internal-probe']]"
    }
  }
],
"probes" : [
  {
    "name" : "api-internal-probe",
    "properties" : {
      "protocol" : "Https",
      "port" : 6443,
      "requestPath" : "/readyz",
      "intervalInSeconds" : 10,
      "numberOfProbes" : 3
    }
  }
]
},
{
  "apiVersion" : "2018-12-01",
  "type" : "Microsoft.Network/loadBalancers",
  "name" : "[variables('internalLoadBalancerName')]",
  "location" : "[variables('location')]",
  "sku" : {
    "name" : "[variables('skuName')]"
  },
  "properties" : {
    "frontendIPConfigurations" : [
      {
        "name" : "internal-lb-ip",
        "properties" : {
          "privateIPAllocationMethod" : "Dynamic",
          "subnet" : {
            "id" : "[variables('masterSubnetRef')]"
          },
          "privateIPAddressVersion" : "IPv4"
        }
      }
    ],
    "backendAddressPools" : [
      {
        "name" : "internal-lb-backend"
      }
    ],
    "loadBalancingRules" : [

```

```

    {
      "name" : "api-internal",
      "properties" : {
        "frontendIPConfiguration" : {
          "id" : "[concat(variables('internalLoadBalancerID'), '/frontendIPConfigurations/internal-lb-
ip')]"
        },
        "frontendPort" : 6443,
        "backendPort" : 6443,
        "enableFloatingIP" : false,
        "idleTimeoutInMinutes" : 30,
        "protocol" : "Tcp",
        "enableTcpReset" : false,
        "loadDistribution" : "Default",
        "backendAddressPool" : {
          "id" : "[concat(variables('internalLoadBalancerID'), '/backendAddressPools/internal-lb-
backend')]"
        },
        "probe" : {
          "id" : "[concat(variables('internalLoadBalancerID'), '/probes/api-internal-probe')]"
        }
      }
    },
    {
      "name" : "sint",
      "properties" : {
        "frontendIPConfiguration" : {
          "id" : "[concat(variables('internalLoadBalancerID'), '/frontendIPConfigurations/internal-lb-
ip')]"
        },
        "frontendPort" : 22623,
        "backendPort" : 22623,
        "enableFloatingIP" : false,
        "idleTimeoutInMinutes" : 30,
        "protocol" : "Tcp",
        "enableTcpReset" : false,
        "loadDistribution" : "Default",
        "backendAddressPool" : {
          "id" : "[concat(variables('internalLoadBalancerID'), '/backendAddressPools/internal-lb-
backend')]"
        },
        "probe" : {
          "id" : "[concat(variables('internalLoadBalancerID'), '/probes/sint-probe')]"
        }
      }
    }
  ],
  "probes" : [
    {
      "name" : "api-internal-probe",
      "properties" : {
        "protocol" : "Https",
        "port" : 6443,
        "requestPath" : "/readyz",
        "intervalInSeconds" : 10,
        "numberOfProbes" : 3
      }
    }
  ]
}

```

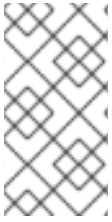
```

    }
  },
  {
    "name": "sint-probe",
    "properties": {
      "protocol": "Https",
      "port": 22623,
      "requestPath": "/healthz",
      "intervalInSeconds": 10,
      "numberOfProbes": 3
    }
  }
]
}
},
{
  "apiVersion": "2018-09-01",
  "type": "Microsoft.Network/privateDnsZones/A",
  "name": "[concat(parameters('privateDNSZoneName'), '/api')]",
  "location": "[variables('location')]",
  "dependsOn": [
    "[concat('Microsoft.Network/loadBalancers/', variables('internalLoadBalancerName'))]"
  ],
  "properties": {
    "ttl": 60,
    "aRecords": [
      {
        "ipv4Address": "[reference(variables('internalLoadBalancerName')).frontendIPConfigurations[0].properties.privateIP
Address]"
      }
    ]
  }
},
{
  "apiVersion": "2018-09-01",
  "type": "Microsoft.Network/privateDnsZones/A",
  "name": "[concat(parameters('privateDNSZoneName'), '/api-int')]",
  "location": "[variables('location')]",
  "dependsOn": [
    "[concat('Microsoft.Network/loadBalancers/', variables('internalLoadBalancerName'))]"
  ],
  "properties": {
    "ttl": 60,
    "aRecords": [
      {
        "ipv4Address": "[reference(variables('internalLoadBalancerName')).frontendIPConfigurations[0].properties.privateIP
Address]"
      }
    ]
  }
}
]
}

```

## 1.7.14. Creating the bootstrap machine in Azure

You must create the bootstrap machine in Microsoft Azure to use during OpenShift Container Platform cluster initialization. One way to create this machine is to modify the provided Azure Resource Manager (ARM) template.



### NOTE

If you do not use the provided ARM template to create your bootstrap machine, you must review the provided information and manually create the infrastructure. If your cluster does not initialize correctly, you might have to contact Red Hat support with your installation logs.

### Prerequisites

- Configure an Azure account.
- Generate the Ignition config files for your cluster.
- Create and configure a VNet and associated subnets in Azure.
- Create and configure networking and load balancers in Azure.
- Create control plane and compute roles.

### Procedure

1. Copy the template from the **ARM template for the bootstrap machine** section of this topic and save it as **04\_bootstrap.json** in your cluster's installation directory. This template describes the bootstrap machine that your cluster requires.
2. Export the following variables required by the bootstrap machine deployment:

```
$ export BOOTSTRAP_URL=`az storage blob url --account-name ${CLUSTER_NAME}sa --
account-key ${ACCOUNT_KEY} -c "files" -n "bootstrap.ign" -o tsv`
$ export BOOTSTRAP_IGNITION=`jq -rcnM --arg v "2.2.0" --arg url ${BOOTSTRAP_URL}
'[{ignition:{version:$v,config:{replace:{source:$url}}}]' | base64 -w0`
```

3. Create the deployment by using the **az** CLI:

```
$ az deployment group create -g ${RESOURCE_GROUP} \
--template-file "<installation_directory>/04_bootstrap.json" \
--parameters bootstrapIgnition="${BOOTSTRAP_IGNITION}" \ 1
--parameters sshKeyData="${SSH_KEY}" \ 2
--parameters baseName="${INFRA_ID}" \ 3
```

- 1** The bootstrap Ignition content for the bootstrap cluster.
- 2** The SSH RSA public key file as a string.
- 3** The base name to be used in resource names; this is usually the cluster's infrastructure ID.

### 1.7.14.1. ARM template for the bootstrap machine

You can use the following Azure Resource Manager (ARM) template to deploy the bootstrap machine that you need for your OpenShift Container Platform cluster:

#### Example 1.4. 04\_bootstrap.json ARM template

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "bootstrapIgnition" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Bootstrap ignition content for the bootstrap cluster"
      }
    },
    "sshKeyData" : {
      "type" : "securestring",
      "metadata" : {
        "description" : "SSH RSA public key file as a string."
      }
    },
    "bootstrapVMSize" : {
      "type" : "string",
      "defaultValue" : "Standard_D4s_v3",
      "allowedValues" : [
        "Standard_A2",
        "Standard_A3",
        "Standard_A4",
        "Standard_A5",
        "Standard_A6",
        "Standard_A7",
        "Standard_A8",
        "Standard_A9",
        "Standard_A10",
        "Standard_A11",
        "Standard_D2",
        "Standard_D3",
        "Standard_D4",
        "Standard_D11",
        "Standard_D12",
        "Standard_D13",
        "Standard_D14",
        "Standard_D2_v2",
        "Standard_D3_v2",
        "Standard_D4_v2",
        "Standard_D5_v2",
        "Standard_D8_v3",
```



```

"Standard_D11_v2",
"Standard_D12_v2",
"Standard_D13_v2",
"Standard_D14_v2",
"Standard_E2_v3",
"Standard_E4_v3",
"Standard_E8_v3",
"Standard_E16_v3",
"Standard_E32_v3",
"Standard_E64_v3",
"Standard_E2s_v3",
"Standard_E4s_v3",
"Standard_E8s_v3",
"Standard_E16s_v3",
"Standard_E32s_v3",
"Standard_E64s_v3",
"Standard_G1",
"Standard_G2",
"Standard_G3",
"Standard_G4",
"Standard_G5",
"Standard_DS2",
"Standard_DS3",
"Standard_DS4",
"Standard_DS11",
"Standard_DS12",
"Standard_DS13",
"Standard_DS14",
"Standard_DS2_v2",
"Standard_DS3_v2",
"Standard_DS4_v2",
"Standard_DS5_v2",
"Standard_DS11_v2",
"Standard_DS12_v2",
"Standard_DS13_v2",
"Standard_DS14_v2",
"Standard_GS1",
"Standard_GS2",
"Standard_GS3",
"Standard_GS4",
"Standard_GS5",
"Standard_D2s_v3",
"Standard_D4s_v3",
"Standard_D8s_v3"
],
"metadata" : {
  "description" : "The size of the Bootstrap Virtual Machine"
}
},
"variables" : {
  "location" : "[resourceGroup().location]",
  "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
  "virtualNetworkID" : "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]",
  "masterSubnetName" : "[concat(parameters('baseName'), '-master-subnet')]",

```

```

    "masterSubnetRef" : "[concat(variables('virtualNetworkID'), '/subnets/',
variables('masterSubnetName'))]",
    "masterLoadBalancerName" : "[concat(parameters('baseName'), '-public-lb')]",
    "internalLoadBalancerName" : "[concat(parameters('baseName'), '-internal-lb')]",
    "sshKeyPath" : "/home/core/.ssh/authorized_keys",
    "identityName" : "[concat(parameters('baseName'), '-identity')]",
    "vmName" : "[concat(parameters('baseName'), '-bootstrap')]",
    "nicName" : "[concat(variables('vmName'), '-nic')]",
    "imageName" : "[concat(parameters('baseName'), '-image')]",
    "clusterNsgName" : "[concat(parameters('baseName'), '-nsg')]",
    "sshPublicIpAddressName" : "[concat(variables('vmName'), '-ssh-pip')]"
  },
  "resources" : [
    {
      "apiVersion" : "2018-12-01",
      "type" : "Microsoft.Network/publicIPAddresses",
      "name" : "[variables('sshPublicIpAddressName')]",
      "location" : "[variables('location')]",
      "sku": {
        "name": "Standard"
      },
      "properties" : {
        "publicIPAllocationMethod" : "Static",
        "dnsSettings" : {
          "domainNameLabel" : "[variables('sshPublicIpAddressName')]"
        }
      }
    },
    {
      "apiVersion" : "2018-06-01",
      "type" : "Microsoft.Network/networkInterfaces",
      "name" : "[variables('nicName')]",
      "location" : "[variables('location')]",
      "dependsOn" : [
        "[resourceId('Microsoft.Network/publicIPAddresses', variables('sshPublicIpAddressName'))]"
      ],
      "properties" : {
        "ipConfigurations" : [
          {
            "name" : "pipConfig",
            "properties" : {
              "privateIPAllocationMethod" : "Dynamic",
              "publicIPAddress": {
                "id": "[resourceId('Microsoft.Network/publicIPAddresses',
variables('sshPublicIpAddressName'))]"
              },
              "subnet" : {
                "id" : "[variables('masterSubnetRef')]"
              },
              "loadBalancerBackendAddressPools" : [
                {
                  "id" : "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/loadBalancers/',
variables('masterLoadBalancerName'), '/backendAddressPools/public-lb-backend')]"
                },
                {

```

```

        "id" : "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/loadBalancers/',
variables('internalLoadBalancerName'), '/backendAddressPools/internal-lb-backend')]"
    }
  ]
}
]
},
{
  "apiVersion" : "2018-06-01",
  "type" : "Microsoft.Compute/virtualMachines",
  "name" : "[variables('vmName')]",
  "location" : "[variables('location')]",
  "identity" : {
    "type" : "userAssigned",
    "userAssignedIdentities" : {
      "[resourceID('Microsoft.ManagedIdentity/userAssignedIdentities/',
variables('identityName'))]" : {}
    }
  },
  "dependsOn" : [
    "[concat('Microsoft.Network/networkInterfaces/', variables('nicName'))]"
  ],
  "properties" : {
    "hardwareProfile" : {
      "vmSize" : "[parameters('bootstrapVMSize')]"
    },
    "osProfile" : {
      "computerName" : "[variables('vmName')]",
      "adminUsername" : "core",
      "customData" : "[parameters('bootstrapIgnition')]",
      "linuxConfiguration" : {
        "disablePasswordAuthentication" : true,
        "ssh" : {
          "publicKeys" : [
            {
              "path" : "[variables('sshKeyPath')]",
              "keyData" : "[parameters('sshKeyData')]"
            }
          ]
        }
      }
    },
    "storageProfile" : {
      "imageReference": {
        "id": "[resourceId('Microsoft.Compute/images', variables('imageName'))]"
      },
      "osDisk" : {
        "name": "[concat(variables('vmName'),'_OSDisk')]",
        "osType" : "Linux",
        "createOption" : "FromImage",
        "managedDisk": {
          "storageAccountType": "Premium_LRS"
        }
      },

```

```

        "diskSizeGB" : 100
      }
    },
    "networkProfile" : {
      "networkInterfaces" : [
        {
          "id" : "[resourceId('Microsoft.Network/networkInterfaces', variables('nicName'))]"
        }
      ]
    }
  },
  {
    "apiVersion" : "2018-06-01",
    "type" : "Microsoft.Network/networkSecurityGroups/securityRules",
    "name" : "[concat(variables('clusterNsgName'), '/bootstrap_ssh_in')]",
    "location" : "[variables('location')]",
    "dependsOn" : [
      "[resourceId('Microsoft.Compute/virtualMachines', variables('vmName'))]"
    ],
    "properties" : {
      "protocol" : "Tcp",
      "sourcePortRange" : "*",
      "destinationPortRange" : "22",
      "sourceAddressPrefix" : "*",
      "destinationAddressPrefix" : "*",
      "access" : "Allow",
      "priority" : 100,
      "direction" : "Inbound"
    }
  }
]
}

```

### 1.7.15. Creating the control plane machines in Azure

You must create the control plane machines in Microsoft Azure for your cluster to use. One way to create these machines is to modify the provided Azure Resource Manager (ARM) template.



#### NOTE

If you do not use the provided ARM template to create your control plane machines, you must review the provided information and manually create the infrastructure. If your cluster does not initialize correctly, you might have to contact Red Hat support with your installation logs.

#### Prerequisites

- Configure an Azure account.
- Generate the Ignition config files for your cluster.
- Create and configure a VNet and associated subnets in Azure.

- Create and configure networking and load balancers in Azure.
- Create control plane and compute roles.
- Create the bootstrap machine.

## Procedure

1. Copy the template from the **ARM template for control plane machines** section of this topic and save it as **05\_masters.json** in your cluster's installation directory. This template describes the control plane machines that your cluster requires.
2. Export the following variable needed by the control plane machine deployment:

```
$ export MASTER_IGNITION=`cat <installation_directory>/master.ign | base64`
```

3. Create the deployment by using the **az** CLI:

```
$ az deployment group create -g ${RESOURCE_GROUP} \
  --template-file "<installation_directory>/05_masters.json" \
  --parameters masterIgnition="${MASTER_IGNITION}" \ 1
  --parameters sshKeyData="${SSH_KEY}" \ 2
  --parameters privateDNSZoneName="${CLUSTER_NAME}.${BASE_DOMAIN}" 3
  --parameters baseName="${INFRA_ID}" 4
```

- 1** The Ignition content for the master nodes.
- 2** The SSH RSA public key file as a string.
- 3** The name of the private DNS zone to which the master nodes are attached.
- 4** The base name to be used in resource names; this is usually the cluster's infrastructure ID.

### 1.7.15.1. ARM template for control plane machines

You can use the following Azure Resource Manager (ARM) template to deploy the control plane machines that you need for your OpenShift Container Platform cluster:

#### Example 1.5. 05\_masters.json ARM template

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "masterIgnition" : {
```

```
"type" : "string",
"metadata" : {
  "description" : "Ignition content for the master nodes"
}
},
"numberOfMasters" : {
  "type" : "int",
  "defaultValue" : 3,
  "minValue" : 2,
  "maxValue" : 30,
  "metadata" : {
    "description" : "Number of OpenShift masters to deploy"
  }
},
"sshKeyData" : {
  "type" : "securestring",
  "metadata" : {
    "description" : "SSH RSA public key file as a string"
  }
},
"privateDNSZoneName" : {
  "type" : "string",
  "metadata" : {
    "description" : "Name of the private DNS zone the master nodes are going to be attached to"
  }
},
"masterVMSize" : {
  "type" : "string",
  "defaultValue" : "Standard_D8s_v3",
  "allowedValues" : [
    "Standard_A2",
    "Standard_A3",
    "Standard_A4",
    "Standard_A5",
    "Standard_A6",
    "Standard_A7",
    "Standard_A8",
    "Standard_A9",
    "Standard_A10",
    "Standard_A11",
    "Standard_D2",
    "Standard_D3",
    "Standard_D4",
    "Standard_D11",
    "Standard_D12",
    "Standard_D13",
    "Standard_D14",
    "Standard_D2_v2",
    "Standard_D3_v2",
    "Standard_D4_v2",
    "Standard_D5_v2",
    "Standard_D8_v3",
    "Standard_D11_v2",
    "Standard_D12_v2",
    "Standard_D13_v2",
    "Standard_D14_v2",
```

```

"Standard_E2_v3",
"Standard_E4_v3",
"Standard_E8_v3",
"Standard_E16_v3",
"Standard_E32_v3",
"Standard_E64_v3",
"Standard_E2s_v3",
"Standard_E4s_v3",
"Standard_E8s_v3",
"Standard_E16s_v3",
"Standard_E32s_v3",
"Standard_E64s_v3",
"Standard_G1",
"Standard_G2",
"Standard_G3",
"Standard_G4",
"Standard_G5",
"Standard_DS2",
"Standard_DS3",
"Standard_DS4",
"Standard_DS11",
"Standard_DS12",
"Standard_DS13",
"Standard_DS14",
"Standard_DS2_v2",
"Standard_DS3_v2",
"Standard_DS4_v2",
"Standard_DS5_v2",
"Standard_DS11_v2",
"Standard_DS12_v2",
"Standard_DS13_v2",
"Standard_DS14_v2",
"Standard_GS1",
"Standard_GS2",
"Standard_GS3",
"Standard_GS4",
"Standard_GS5",
"Standard_D2s_v3",
"Standard_D4s_v3",
"Standard_D8s_v3"
],
"metadata" : {
  "description" : "The size of the Master Virtual Machines"
}
},
"diskSizeGB" : {
  "type" : "int",
  "defaultValue" : 1024,
  "metadata" : {
    "description" : "Size of the Master VM OS disk, in GB"
  }
}
},
"variables" : {
  "location" : "[resourceGroup().location]",
  "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",

```

```

    "virtualNetworkID" : "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]",
    "masterSubnetName" : "[concat(parameters('baseName'), '-master-subnet')]",
    "masterSubnetRef" : "[concat(variables('virtualNetworkID'), '/subnets/',
variables('masterSubnetName'))]",
    "masterLoadBalancerName" : "[concat(parameters('baseName'), '-public-lb')]",
    "internalLoadBalancerName" : "[concat(parameters('baseName'), '-internal-lb')]",
    "sshKeyPath" : "/home/core/.ssh/authorized_keys",
    "identityName" : "[concat(parameters('baseName'), '-identity')]",
    "imageName" : "[concat(parameters('baseName'), '-image')]",
    "copy" : [
      {
        "name" : "vmNames",
        "count" : "[parameters('numberOfMasters')]",
        "input" : "[concat(parameters('baseName'), '-master-', copyIndex('vmNames'))]"
      }
    ]
  },
  "resources" : [
    {
      "apiVersion" : "2018-06-01",
      "type" : "Microsoft.Network/networkInterfaces",
      "copy" : {
        "name" : "nicCopy",
        "count" : "[length(variables('vmNames'))]"
      },
      "name" : "[concat(variables('vmNames')[copyIndex()], '-nic')]",
      "location" : "[variables('location')]",
      "properties" : {
        "ipConfigurations" : [
          {
            "name" : "pipConfig",
            "properties" : {
              "privateIPAllocationMethod" : "Dynamic",
              "subnet" : {
                "id" : "[variables('masterSubnetRef')]"
              },
              "loadBalancerBackendAddressPools" : [
                {
                  "id" : "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/loadBalancers/',
variables('masterLoadBalancerName'), '/backendAddressPools/public-lb-backend')]"
                },
                {
                  "id" : "[concat('/subscriptions/', subscription().subscriptionId, '/resourceGroups/',
resourceGroup().name, '/providers/Microsoft.Network/loadBalancers/',
variables('internalLoadBalancerName'), '/backendAddressPools/internal-lb-backend')]"
                }
              ]
            }
          }
        ]
      }
    }
  ]
},
{
  "apiVersion": "2018-09-01",

```



```

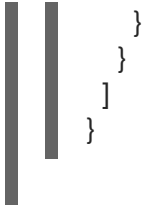
"type": "Microsoft.Network/privateDnsZones/SRV",
"name": "[concat(parameters('privateDNSZoneName'), '/_etcd-server-ssl._tcp')]",
"location" : "[variables('location')]",
"properties": {
  "ttl": 60,
  "copy": [{
    "name": "srvRecords",
    "count": "[length(variables('vmNames'))]",
    "input": {
      "priority": 0,
      "weight" : 10,
      "port" : 2380,
      "target" : "[concat('etcd-', copyIndex('srvRecords'), '.',
parameters('privateDNSZoneName'))]"
    }
  ]
}
},
{
  "apiVersion": "2018-09-01",
  "type": "Microsoft.Network/privateDnsZones/A",
  "copy" : {
    "name" : "dnsCopy",
    "count" : "[length(variables('vmNames'))]"
  },
  "name": "[concat(parameters('privateDNSZoneName'), '/etcd-', copyIndex())]",
  "location" : "[variables('location')]",
  "dependsOn" : [
    "[concat('Microsoft.Network/networkInterfaces/', concat(variables('vmNames')[copyIndex()], '-
nic'))]"
  ],
  "properties": {
    "ttl": 60,
    "aRecords": [
      {
        "ipv4Address": "[reference(concat(variables('vmNames')[copyIndex()], '-
nic')).ipConfigurations[0].properties.privateIPAddress]"
      }
    ]
  }
},
{
  "apiVersion" : "2018-06-01",
  "type" : "Microsoft.Compute/virtualMachines",
  "copy" : {
    "name" : "vmCopy",
    "count" : "[length(variables('vmNames'))]"
  },
  "name" : "[variables('vmNames')[copyIndex()]]",
  "location" : "[variables('location')]",
  "identity" : {
    "type" : "userAssigned",
    "userAssignedIdentities" : {
      "[resourceID('Microsoft.ManagedIdentity/userAssignedIdentities/',
variables('identityName'))]" : {}
    }
  }
}

```

```

    },
    "dependsOn" : [
      "[concat('Microsoft.Network/networkInterfaces/', concat(variables('vmNames')[copyIndex()], '-nic'))]",
      "[concat('Microsoft.Network/privateDnsZones/', parameters('privateDNSZoneName'), '/A/etcd-', copyIndex())]",
      "[concat('Microsoft.Network/privateDnsZones/', parameters('privateDNSZoneName'), '/SRV/_etcd-server-ssl._tcp')]"
    ],
    "properties" : {
      "hardwareProfile" : {
        "vmSize" : "[parameters('masterVMSize')]"
      },
      "osProfile" : {
        "computerName" : "[variables('vmNames')[copyIndex()]]",
        "adminUsername" : "core",
        "customData" : "[parameters('masterIgnition')]",
        "linuxConfiguration" : {
          "disablePasswordAuthentication" : true,
          "ssh" : {
            "publicKeys" : [
              {
                "path" : "[variables('sshKeyPath')]",
                "keyData" : "[parameters('sshKeyData')]"
              }
            ]
          }
        }
      },
      "storageProfile" : {
        "imageReference": {
          "id": "[resourceId('Microsoft.Compute/images', variables('imageName'))]"
        },
        "osDisk" : {
          "name": "[concat(variables('vmNames')[copyIndex()], '_OSDisk')]",
          "osType" : "Linux",
          "createOption" : "FromImage",
          "caching": "ReadOnly",
          "writeAcceleratorEnabled": false,
          "managedDisk": {
            "storageAccountType": "Premium_LRS"
          },
          "diskSizeGB" : "[parameters('diskSizeGB')]"
        }
      },
      "networkProfile" : {
        "networkInterfaces" : [
          {
            "id" : "[resourceId('Microsoft.Network/networkInterfaces', concat(variables('vmNames')[copyIndex()], '-nic'))]",
            "properties": {
              "primary": false
            }
          }
        ]
      }
    }
  }

```



## 1.7.16. Wait for bootstrap completion and remove bootstrap resources in Azure

After you create all of the required infrastructure in Microsoft Azure, wait for the bootstrap process to complete on the machines that you provisioned by using the Ignition config files that you generated with the installation program.

### Prerequisites

- Configure an Azure account.
- Generate the Ignition config files for your cluster.
- Create and configure a VNet and associated subnets in Azure.
- Create and configure networking and load balancers in Azure.
- Create control plane and compute roles.
- Create the bootstrap machine.
- Create the control plane machines.

### Procedure

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

```
$ ./openshift-install wait-for bootstrap-complete --dir=<installation_directory> \ 1
--log-level info 2
```

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

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

If the command exits without a **FATAL** warning, your production control plane has initialized.

2. Delete the bootstrap resources:

```
$ az network nsg rule delete -g ${RESOURCE_GROUP} --nsg-name ${INFRA_ID}-nsg --
name bootstrap_ssh_in
$ az vm stop -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap
$ az vm deallocate -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap
$ az vm delete -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap --yes
$ az disk delete -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap_OSDisk --no-
wait --yes
$ az network nic delete -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap-nic --no-
wait
$ az storage blob delete --account-key ${ACCOUNT_KEY} --account-name
```

```

${CLUSTER_NAME}sa --container-name files --name bootstrap.ign
$ az network public-ip delete -g ${RESOURCE_GROUP} --name ${INFRA_ID}-bootstrap-ssh-pip

```

### 1.7.17. Creating additional worker machines in Azure

You can create worker machines in Microsoft Azure for your cluster to use by launching individual instances discretely or by automated processes outside the cluster, such as auto scaling groups. You can also take advantage of the built-in cluster scaling mechanisms and the machine API in OpenShift Container Platform.

In this example, you manually launch one instance by using the Azure Resource Manager (ARM) template. Additional instances can be launched by including additional resources of type **06\_workers.json** in the file.



#### NOTE

If you do not use the provided ARM template to create your worker machines, you must review the provided information and manually create the infrastructure. If your cluster does not initialize correctly, you might have to contact Red Hat support with your installation logs.

#### Prerequisites

- Configure an Azure account.
- Generate the Ignition config files for your cluster.
- Create and configure a VNet and associated subnets in Azure.
- Create and configure networking and load balancers in Azure.
- Create control plane and compute roles.
- Create the bootstrap machine.
- Create the control plane machines.

#### Procedure

1. Copy the template from the **ARM template for worker machines** section of this topic and save it as **06\_workers.json** in your cluster's installation directory. This template describes the worker machines that your cluster requires.
2. Export the following variable needed by the worker machine deployment:

```
$ export WORKER_IGNITION=`cat <installation_directory>/worker.ign | base64`
```

3. Create the deployment by using the **az** CLI:

```

$ az deployment group create -g ${RESOURCE_GROUP} \
  --template-file "<installation_directory>/06_workers.json" \
  --parameters workerIgnition="${WORKER_IGNITION}" \ 1

```

```
--parameters sshKeyData="${SSH_KEY}" \ 2
--parameters baseName="${INFRA_ID}" 3
```

- 1 The Ignition content for the worker nodes.
- 2 The SSH RSA public key file as a string.
- 3 The base name to be used in resource names; this is usually the cluster's infrastructure ID.

### 1.7.17.1. ARM template for worker machines

You can use the following Azure Resource Manager (ARM) template to deploy the worker machines that you need for your OpenShift Container Platform cluster:

#### Example 1.6. 06\_workers.json ARM template

```
{
  "$schema" : "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion" : "1.0.0.0",
  "parameters" : {
    "baseName" : {
      "type" : "string",
      "minLength" : 1,
      "metadata" : {
        "description" : "Base name to be used in resource names (usually the cluster's Infra ID)"
      }
    },
    "workerIgnition" : {
      "type" : "string",
      "metadata" : {
        "description" : "Ignition content for the worker nodes"
      }
    },
    "numberOfNodes" : {
      "type" : "int",
      "defaultValue" : 3,
      "minValue" : 2,
      "maxValue" : 30,
      "metadata" : {
        "description" : "Number of OpenShift compute nodes to deploy"
      }
    },
    "sshKeyData" : {
      "type" : "securestring",
      "metadata" : {
        "description" : "SSH RSA public key file as a string"
      }
    },
    "nodeVMSize" : {
      "type" : "string",
      "defaultValue" : "Standard_D4s_v3",
      "allowedValues" : [
        "Standard_A2",
```

"Standard\_A3",  
"Standard\_A4",  
"Standard\_A5",  
"Standard\_A6",  
"Standard\_A7",  
"Standard\_A8",  
"Standard\_A9",  
"Standard\_A10",  
"Standard\_A11",  
"Standard\_D2",  
"Standard\_D3",  
"Standard\_D4",  
"Standard\_D11",  
"Standard\_D12",  
"Standard\_D13",  
"Standard\_D14",  
"Standard\_D2\_v2",  
"Standard\_D3\_v2",  
"Standard\_D4\_v2",  
"Standard\_D5\_v2",  
"Standard\_D8\_v3",  
"Standard\_D11\_v2",  
"Standard\_D12\_v2",  
"Standard\_D13\_v2",  
"Standard\_D14\_v2",  
"Standard\_E2\_v3",  
"Standard\_E4\_v3",  
"Standard\_E8\_v3",  
"Standard\_E16\_v3",  
"Standard\_E32\_v3",  
"Standard\_E64\_v3",  
"Standard\_E2s\_v3",  
"Standard\_E4s\_v3",  
"Standard\_E8s\_v3",  
"Standard\_E16s\_v3",  
"Standard\_E32s\_v3",  
"Standard\_E64s\_v3",  
"Standard\_G1",  
"Standard\_G2",  
"Standard\_G3",  
"Standard\_G4",  
"Standard\_G5",  
"Standard\_DS2",  
"Standard\_DS3",  
"Standard\_DS4",  
"Standard\_DS11",  
"Standard\_DS12",  
"Standard\_DS13",  
"Standard\_DS14",  
"Standard\_DS2\_v2",  
"Standard\_DS3\_v2",  
"Standard\_DS4\_v2",  
"Standard\_DS5\_v2",  
"Standard\_DS11\_v2",  
"Standard\_DS12\_v2",  
"Standard\_DS13\_v2",

```

    "Standard_DS14_v2",
    "Standard_GS1",
    "Standard_GS2",
    "Standard_GS3",
    "Standard_GS4",
    "Standard_GS5",
    "Standard_D2s_v3",
    "Standard_D4s_v3",
    "Standard_D8s_v3"
  ],
  "metadata" : {
    "description" : "The size of the each Node Virtual Machine"
  }
},
"variables" : {
  "location" : "[resourceGroup().location]",
  "virtualNetworkName" : "[concat(parameters('baseName'), '-vnet')]",
  "virtualNetworkID" : "[resourceId('Microsoft.Network/virtualNetworks',
variables('virtualNetworkName'))]",
  "nodeSubnetName" : "[concat(parameters('baseName'), '-worker-subnet')]",
  "nodeSubnetRef" : "[concat(variables('virtualNetworkID'), '/subnets/',
variables('nodeSubnetName'))]",
  "infraLoadBalancerName" : "[parameters('baseName')]",
  "sshKeyPath" : "/home/capi/.ssh/authorized_keys",
  "identityName" : "[concat(parameters('baseName'), '-identity')]",
  "imageName" : "[concat(parameters('baseName'), '-image')]",
  "copy" : [
    {
      "name" : "vmNames",
      "count" : "[parameters('numberOfNodes')]",
      "input" : "[concat(parameters('baseName'), '-worker-', variables('location'), '-',
copyIndex('vmNames', 1))]"
    }
  ]
},
"resources" : [
  {
    "apiVersion" : "2019-05-01",
    "name" : "[concat('node', copyIndex())]",
    "type" : "Microsoft.Resources/deployments",
    "copy" : {
      "name" : "nodeCopy",
      "count" : "[length(variables('vmNames'))]"
    },
    "properties" : {
      "mode" : "Incremental",
      "template" : {
        "$schema" : "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
        "contentVersion" : "1.0.0.0",
        "resources" : [
          {
            "apiVersion" : "2018-06-01",
            "type" : "Microsoft.Network/networkInterfaces",
            "name" : "[concat(variables('vmNames')[copyIndex()], '-nic')]",

```

```

"location" : "[variables('location')]",
"properties" : {
  "ipConfigurations" : [
    {
      "name" : "pipConfig",
      "properties" : {
        "privateIPAllocationMethod" : "Dynamic",
        "subnet" : {
          "id" : "[variables('nodeSubnetRef')]"
        }
      }
    }
  ]
},
{
  "apiVersion" : "2018-06-01",
  "type" : "Microsoft.Compute/virtualMachines",
  "name" : "[variables('vmNames')[copyIndex()]]",
  "location" : "[variables('location')]",
  "tags" : {
    "kubernetes.io-cluster-ffranzupi": "owned"
  },
  "identity" : {
    "type" : "userAssigned",
    "userAssignedIdentities" : {
      "[resourceID('Microsoft.ManagedIdentity/userAssignedIdentities/',
variables('identityName'))]" : {}
    }
  },
  "dependsOn" : [
    "[concat('Microsoft.Network/networkInterfaces/', concat(variables('vmNames')
[copyIndex()], '-nic'))]"
  ],
  "properties" : {
    "hardwareProfile" : {
      "vmSize" : "[parameters('nodeVMSize')]"
    },
    "osProfile" : {
      "computerName" : "[variables('vmNames')[copyIndex()]]",
      "adminUsername" : "capi",
      "customData" : "[parameters('workerIgnition')]",
      "linuxConfiguration" : {
        "disablePasswordAuthentication" : true,
        "ssh" : {
          "publicKeys" : [
            {
              "path" : "[variables('sshKeyPath')]",
              "keyData" : "[parameters('sshKeyData')]"
            }
          ]
        }
      }
    }
  },
  "storageProfile" : {
    "imageReference": {

```





3. In the **Command line interface** section, select **Linux** from the drop-down menu and click **Download command-line tools**.
4. Unpack the archive:

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

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

```
$ echo $PATH
```

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

```
$ oc <command>
```

### 1.7.18.2. Installing the CLI on Windows

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

#### Procedure

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

```
C:\> path
```

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

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

### 1.7.18.3. Installing the CLI on macOS

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

#### Procedure

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

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

```
$ echo $PATH
```

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

```
$ oc <command>
```

### 1.7.19. Logging in to the cluster

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

#### Prerequisites

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

#### Procedure

1. Export the **kubeadmin** credentials:

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

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

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

```
$ oc whoami
```

#### Example output

```
system:admin
```

### 1.7.20. Approving the certificate signing requests for your machines

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

#### Prerequisites

- You added machines to your cluster.

## Procedure

1. Confirm that the cluster recognizes the machines:

```
$ oc get nodes
```

### Example output

```
NAME      STATUS    ROLES    AGE   VERSION
master-0  Ready    master   63m   v1.18.3
master-1  Ready    master   63m   v1.18.3
master-2  Ready    master   64m   v1.18.3
worker-0  NotReady worker   76s   v1.18.3
worker-1  NotReady worker   70s   v1.18.3
```

The output lists all of the machines that you created.

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

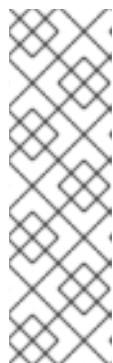
```
$ oc get csr
```

### Example output

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

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

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



### NOTE

Because the CSRs rotate automatically, approve your CSRs within an hour of adding the machines to the cluster. If you do not approve them within an hour, the certificates will rotate, and more than two certificates will be present for each node. You must approve all of these certificates. Once the client CSR is approved, the Kubelet creates a secondary CSR for the serving certificate, which requires manual approval. Then, subsequent serving certificate renewal requests are automatically approved by the **machine-approver** if the Kubelet requests a new certificate with identical parameters.

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

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

**1** **<csr\_name>** is the name of a CSR from the list of current CSRs.

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

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

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

```
$ oc get csr
```

### Example output

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

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

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

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

**1** **<csr\_name>** is the name of a CSR from the list of current CSRs.

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

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

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

```
$ oc get nodes
```

### Example output

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



### NOTE

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

## Additional information

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

### 1.7.21. Adding the Ingress DNS records

If you removed the DNS Zone configuration when creating Kubernetes manifests and generating Ignition configs, you must manually create DNS records that point at the Ingress load balancer. You can create either a wildcard **\*.apps.{baseDomain}**, or specific records. You can use A, CNAME, and other records per your requirements.

#### Prerequisites

- You deployed an OpenShift Container Platform cluster on Microsoft Azure by using infrastructure that you provisioned.
- Install the OpenShift CLI (**oc**).
- Install the **jq** package.
- Install or update the [Azure CLI](#).

#### Procedure

1. Confirm the Ingress router has created a load balancer and populated the **EXTERNAL-IP** field:

```
$ oc -n openshift-ingress get service router-default
```

#### Example output

```
NAME           TYPE           CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
router-default LoadBalancer   172.30.20.10 35.130.120.110
80:32288/TCP,443:31215/TCP 20
```

2. Export the Ingress router IP as a variable:

```
$ export PUBLIC_IP_ROUTER=`oc -n openshift-ingress get service router-default --no-headers | awk '{print $4}'`
```

3. Add a **\*.apps** record to the public DNS zone.

- a. If you are adding this cluster to a new public zone, run:

```
$ az network dns record-set a add-record -g ${BASE_DOMAIN_RESOURCE_GROUP} -z ${CLUSTER_NAME}.${BASE_DOMAIN} -n *.apps -a ${PUBLIC_IP_ROUTER} --ttl 300
```

- b. If you are adding this cluster to an already existing public zone, run:

```
$ az network dns record-set a add-record -g ${BASE_DOMAIN_RESOURCE_GROUP} -z ${BASE_DOMAIN} -n *.apps.${CLUSTER_NAME} -a ${PUBLIC_IP_ROUTER} --ttl 300
```

4. Add a **\*.apps** record to the private DNS zone:

- a. Create a **\*.apps** record by using the following command:

```
$ az network private-dns record-set a create -g ${RESOURCE_GROUP} -z
${CLUSTER_NAME}.${BASE_DOMAIN} -n *.apps --ttl 300
```

- b. Add the **\*.apps** record to the private DNS zone by using the following command:

```
$ az network private-dns record-set a add-record -g ${RESOURCE_GROUP} -z
${CLUSTER_NAME}.${BASE_DOMAIN} -n *.apps -a ${PUBLIC_IP_ROUTER}
```

If you prefer to add explicit domains instead of using a wildcard, you can create entries for each of the cluster's current routes:

```
$ oc get --all-namespaces -o jsonpath='{range .items[*]}{range .status.ingress[*]}{.host}{"\n"}{end}
{end}' routes
```

### Example output

```
oauth-openshift.apps.cluster.basedomain.com
console-openshift-console.apps.cluster.basedomain.com
downloads-openshift-console.apps.cluster.basedomain.com
alertmanager-main-openshift-monitoring.apps.cluster.basedomain.com
grafana-openshift-monitoring.apps.cluster.basedomain.com
prometheus-k8s-openshift-monitoring.apps.cluster.basedomain.com
```

## 1.7.22. Completing an Azure installation on user-provisioned infrastructure

After you start the OpenShift Container Platform installation on Microsoft Azure user-provisioned infrastructure, you can monitor the cluster events until the cluster is ready.

### Prerequisites

- Deploy the bootstrap machine for an OpenShift Container Platform cluster on user-provisioned Azure infrastructure.
- Install the **oc** CLI and log in.

### Procedure

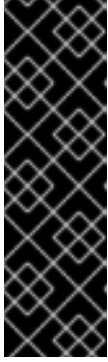
- Complete the cluster installation:

```
$ ./openshift-install --dir=<installation_directory> wait-for install-complete 1
```

### Example output

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

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



## 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.

## 1.8. UNINSTALLING A CLUSTER ON AZURE

You can remove a cluster that you deployed to Microsoft Azure.

### 1.8.1. Removing a cluster that uses installer-provisioned infrastructure

You can remove a cluster that uses installer-provisioned infrastructure from your cloud.



## NOTE

After uninstallation, check your cloud provider for any resources not removed properly, especially with User Provisioned Infrastructure (UPI) clusters. There might be resources that the installer did not create or that the installer is unable to access.

### Prerequisites

- Have a copy of the installation program that you used to deploy the cluster.
- Have the files that the installation program generated when you created your cluster.

### Procedure

1. From the computer that you used to install the cluster, run the following command:

```

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

```

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

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



## NOTE

You must specify the directory that contains the cluster definition files for your cluster. The installation program requires the **metadata.json** file in this directory to delete the cluster.

2. Optional: Delete the **<installation\_directory>** directory and the OpenShift Container Platform installation program.



