Toggle nav Documentation (/)

Page history (https://github.com/openshift/openshift-docs/commits/enterprise-4.1/installing/installing_bare_metal/installing-bare-metal.adoc) / Suggest an edit (https://github.com/openshift/openshift-docs/issues/new?title=[enterprise-4.1] Edit suggested in file installing/installing_bare_metal/installing-bare-metal.adoc)

Installing a cluster on bare metal

In OpenShift Container Platform version 4.1, you can install a cluster on bare metal infrastructure that you provision.

❗ While you might be able to follow this procedure to deploy a cluster on virtualized or cloud environments, you must be aware of additional considerations for non-bare metal platforms. Review the information in the guidelines for deploying OpenShift Container Platform on non-tested platforms (https://access.redhat.com/articles/4207611) before you attempt to install an OpenShift Container Platform cluster in such an environment.

Prerequisites

- Provision persistent storage (../../storage/understanding-persistent-storage.html#understanding-persistent-storage) for your cluster. To deploy a private image registry, your storage must provide ReadWriteMany access modes.
- Review details about the OpenShift Container Platform installation and update (../../architecture/architecture-installation.html#architecture-installation) processes.
- If you use a firewall, you must configure it to access Red Hat Insights (../../installing/install_config/configuring-firewall.html#configuring-firewall).

Internet and Telemetry access for OpenShift Container Platform

In OpenShift Container Platform 4.1, Telemetry is the component that provides metrics about cluster health and the success of updates. To perform subscription management, including legally entitling your purchase from Red Hat, you must use the Telemetry service and access the OpenShift Infrastructure Providers (https://cloud.redhat.com/openshift/install) page.

Because there is no disconnected subscription management, you cannot both opt out of sending data back to Red Hat and entitle your purchase. Support for disconnected subscription management might be added in future releases of OpenShift Container Platform

❗ Your machines must have direct internet access to install the cluster.

You must have internet access to:

- Access the OpenShift Infrastructure Providers (https://cloud.redhat.com/openshift/install) page to download the installation program
- Access Quay.io (http://quay.io) to obtain the packages that are required to install your cluster

- Obtain the packages that are required to perform cluster updates
- Access Red Hat's software as a service page (http://cloud.redhat.com) to perform subscription management

Machine requirements for a cluster with user-provisioned infrastructure

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

Required machines

The smallest OpenShift Container Platform clusters require the following hosts:

- One bootstrap machine
- Three control plane, or master, machines
- At least two compute, or worker, machines

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

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

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

Network connectivity requirements

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

Minimum resource requirements

Each cluster machine must meet the following minimum requirements:

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

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.

Creating the user-provisioned infrastructure

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

Prerequistes

- Review the OpenShift Container Platform 4.x Tested Integrations (https://access.redhat.com/articles/4128421) page before you create the supporting infrastructure for your cluster.

Procedure

1. Configure DHCP.

2. Provision the required load balancers.

3. Configure the ports for your machines.

4. Configure DNS.
5. Ensure network connectivity.

Networking requirements for user-provisioned infrastructure

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

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

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

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

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

Table 1. All machines to all machines

`2379-2380`	**etcd server, peer, and metrics ports**
`6443`	Kubernetes API
`9000-9999`	Host level services, including the node exporter on ports **`9100-9101`** and the Cluster Version Operator on port **`9099`**.
`10249-10259`	The default ports that Kubernetes reserves
`10256`	openshift-sdn
`30000-32767`	Kubernetes NodePort

NETWORK TOPOLOGY REQUIREMENTS

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

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

Load balancers

Before you install OpenShift Container Platform, you must provision two layer-4 load balancers.

Port	Machines	Internal	External	Description
6443	Bootstrap and control plane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane.	x	x	Kubernetes API server
22623	Bootstrap and control plane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane.	x		Machine Config server
443	The machines that run the Ingress router pods, compute, or worker, by default.	x	x	HTTPS traffic
80	The machines that run the Ingress router pods, compute, or worker by default.	x	x	HTTP traffic

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.

User-provisioned DNS requirements

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

Table 2. Required DNS records

Component	Record	Description
Kubernetes API	**`api.<cluster_name>.<base_domain>`**	This DNS record must point to the load balancer for the control plane machines. from all the nodes within the cluster.
	`api-int.<cluster_name>.<base_domain>`	This DNS record must point to the load balancer for the control plane machines. ❗ The API server must be able to resolve the worker nodes by the hos names, proxied API calls can fail, and you cannot retrieve logs from I
Routes	**`*.apps.<cluster_name>.<base_domain>`**	A wildcard DNS record that points to the load balancer that targets the machines This record must be resolvable by both clients external to the cluster and from all
etcd	**`etcd-<index>.<cluster_name>.<base_domain>`**	OpenShift Container Platform requires DNS records for each etcd instance to po are differentiated by **`<index>`** values, which start with **`0`** and end with **`n-1`**, where **n** resolve to an unicast IPV4 address for the control plane machine, and the records

	`_etcd-server-ssl._tcp.<cluster_name>.<base_domain>`	For each control plane machine, OpenShift Container Platform also requires a SF port **`2380`**. A cluster that uses three control plane machines requires the following
		`# _service._proto.name.                           TTL    class`
		`_etcd-server-ssl._tcp.<cluster_name>.<base_domain>  86400 IN`
		`_etcd-server-ssl._tcp.<cluster_name>.<base_domain>  86400 IN`
		`_etcd-server-ssl._tcp.<cluster_name>.<base_domain>  86400 IN`

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

Generating an SSH private key and adding it to the agent

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

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.

> You must use a local key, not one that you configured with platform-specific approaches such as AWS key pairs (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html).

Procedure

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

   ```
   $ ssh-keygen -t rsa -b 4096 -N '' \
       -f <path>/<file_name>  1
   ```

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

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

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

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

Agent pid 31874
```

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

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

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

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

Next steps

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

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 300 MB of local disk space to download the installation program.

Procedure

1. Access the OpenShift Infrastructure Providers (https://cloud.redhat.com/openshift/install) page. If you have a Red Hat account, log in with your credentials. If you do not, create an account.
2. Download the installation program for your operating system and place the file in the directory where you will store the installation configuration files.

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

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

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

4. From the OpenShift Infrastructure Providers (https://cloud.redhat.com/openshift/install) page, download your installation pull secret. 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.

Installing the OpenShift Command-line Interface

You can download and install the OpenShift Command-line Interface (CLI), commonly known as **oc**.

> If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.1. You must download and install the new version of **oc**.

Procedure

1. From the OpenShift Infrastructure Providers (https://cloud.redhat.com/openshift/install) page, click **Download Command-line Tools**.

2. From the site that is displayed, download the compressed file for your operating system.

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

3. Extract the compressed file and place it in a directory that is on your PATH.

Manually creating the installation configuration file

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

Prerequisites

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

Procedure

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

   ```
   $ mkdir <installation_directory>
   ```

   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>`**.

   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.

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

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

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

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

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

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

3 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 simultanous multithreading in some cluster machines, you must disable it in all cluster machines.

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

4 You must set the value of the `replicas` parameter to `0`. This parameter controls the number of workers that the cluster creates and manages for you, which are functions that the cluster does not perform when you use user-provisioned infrastructure. You must manually deploy worker machines for the cluster to use before you finish installing OpenShift Container Platform.

5 The number of control plane machines that you add to the cluster. Because the cluster uses this values as the number of etcd endpoints in the cluster, the value must match the number of control plane machines that you deploy.

6 The cluster name that you specified in your DNS records.

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

8 The subnet prefix length to assign to each individual node. For example, if `hostPrefix` is set to `23`, then each node is assigned a `/23` subnet out of the given `cidr`, which allows for 510 (2^(32 - 23) - 2) pod IPs addresses. If you are required to provide access to nodes from an external network, configure load balancers and routers to manage the traffic.

9 The IP address pool to use for service IP addresses. You can enter only one IP address pool. If you need to access the services from an external network, configure load balancers and routers to manage the traffic.

10 You must set the platform to `none`. You cannot provide additional platform configuration variables for bare metal infrastructure.

11 The pull secret that you obtained from the OpenShift Infrastructure Providers (https://cloud.redhat.com/openshift/install) page. 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.

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

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

Creating the Ignition config files

Because you must manually start the cluster machines, you must generate the Ignition config files that the cluster needs to make its machines.

> The Ignition config files that the installation program generates contain certificates that expire after 24 hours. You must complete your cluster installation and keep the cluster running for 24 hours in a non-degraded state to ensure that the first certificate rotation has finished.

Prerequisites

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

Procedure

1. Obtain the Ignition config files:

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

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

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

The following files are generated in the directory:

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

Creating Red Hat Enterprise Linux CoreOS (RHCOS) machines

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

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

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

Prerequisites

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

Procedure

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

2. Obtain the RHCOS images that are required for your preferred method of installing operating system instances from the Product Downloads (https://access.redhat.com/downloads/content/290) page on the Red Hat customer portal or the RHCOS image mirror (https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/4.1/) page.

   > The RHCOS images might not change with every release of OpenShift Container Platform. You must download images with the highest version that is less than or equal to the OpenShift Container Platform version that you install. Use the image versions that match your OpenShift Container Platform version if they are available.

   You must download the ISO file and either the BIOS or UEFI file. Those file names resemble the following examples:

   - ISO: **`rhcos-<version>-<architecture>-installer.iso`**
   - Compressed metal BIOS: **`rhcos-<version>-<architecture>-metal-bios.raw.gz`**
   - Compressed metal UEFI: **`rhcos-<version>-<architecture>-metal-uefi.raw.gz`**

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

4. Use the ISO to start the RHCOS installation. Use one of the following installation options:

   - Burn the ISO image to a disk and boot it directly.
   - Use ISO redirection via a LOM interface.

5. After the instance boots, press the **`TAB`** or **`E`** key to edit the kernel command line.

6. Add the parameters to the kernel command line:

   ```
   coreos.inst=yes
   coreos.inst.install_dev=sda  1
   coreos.inst.image_url=<bare_metal_image_URL>  2
   coreos.inst.ignition_url=http://example.com/config.ign  3
   ```

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

2 Specify the URL of the UEFI or BIOS image that you uploaded to your server.

3 Specify the URL of the Ignition config file for this machine type.

7. Press Enter to complete the installation. After RHCOS installs, the system reboots. After the system reboots, it applies the Ignition config file that you specified.

8. Continue to create the machines for your cluster.

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

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

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

Prerequisites

- Obtain the Ignition config files for your cluster.
- Configure suitable PXE or iPXE infrastructure.
- Have access to an HTTP server that you can access from your computer.

Procedure

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

2. Obtain the RHCOS ISO image, compressed metal BIOS, **`kernel`** and **`initramfs`** files from the Product Downloads (https://access.redhat.com/downloads/content/290) page on the Red Hat customer portal or the RHCOS image mirror (https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/4.1/) page.

> The RHCOS images might not change with every release of OpenShift Container Platform. You must download images with the highest version that is less than or equal to the OpenShift Container Platform version that you install. Use the image versions that match your OpenShift Container Platform version if they are available.

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

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

3. Upload the compressed metal BIOS file and the **`kernel`** and **`initramfs`** files to your HTTP server.

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

5. Configure PXE or iPXE installation for the RHCOS images.

   Modify one of the following example menu entries for your environment and verify that the image and Ignition files are properly accessible:

   - For PXE:

```
DEFAULT pxeboot
TIMEOUT 20
PROMPT 0
LABEL pxeboot
    KERNEL http://<HTTP_server>/rhcos-<version>-<architecture>-installer-kernel  1
    APPEND ip=dhcp rd.neednet=1 initrd=http://<HTTP_server>/rhcos-<version>-<architec
 3
```

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

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

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

- For iPXE:

```
kernel  http://<HTTP_server>/rhcos-<version>-<architecture>-installer-kernel ip=dhcp
initrd http://<HTTP_server>/rhcos-<version>-<architecture>-installer-initramfs.img  3
boot
```

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

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

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

6. If you use UEFI, edit the included **grub.conf** file that is included in the ISO that you downloaded to include the following installation options:

```
menuentry 'Install Red Hat Enterprise Linux CoreOS' --class fedora --class gnu-linux --clas
        linux /images/vmlinuz nomodeset rd.neednet=1 coreos.inst=yes coreos.inst.install_de
        initrd http://<HTTP_server>/rhcos-<version>-<architecture>-installer-initramfs.img
}
```

1 For the **`coreos.inst.image_url`** parameter value, specify the location of the compressed metal UEFI file that you uploaded to your HTTP server. For the **`coreos.inst.ignition_url`**, specify the location of the bootstrap Ingition config file that you uploaded to your HTTP server.

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

7. Continue to create the machines for your cluster.

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

Creating the cluster

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

Prerequisites

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

Procedure

1. Monitor the bootstrap process:

```
$ ./openshift-install --dir=<installation_directory> wait-for bootstrap-complete \  1
    --log-level info  2

INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com:6443...
INFO API v1.13.4+b626c2fe1 up
INFO Waiting up to 30m0s for the bootstrap-complete event...
```

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

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

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

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

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

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

$ oc whoami
system:admin
```

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

Approving the CSRs for your machines

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

Prerequisites

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

Procedure

1. Confirm that the cluster recognizes the machines:

```
$ oc get nodes

NAME      STATUS    ROLES   AGE  VERSION
master-0  Ready     master  63m  v1.13.4+b626c2fe1
master-1  Ready     master  63m  v1.13.4+b626c2fe1
master-2  Ready     master  64m  v1.13.4+b626c2fe1
worker-0  NotReady  worker  76s  v1.13.4+b626c2fe1
worker-1  NotReady  worker  70s  v1.13.4+b626c2fe1
```

The output lists all of the machines that you created.

2. Review the pending certificate signing requests (CSRs) and ensure that the you see a client and server request with **`Pending`** or **`Approved`** status for each machine that you added to the cluster:

```
$ oc get csr

NAME        AGE     REQUESTOR
csr-8b2br   15m     system:serviceaccount:openshift-machine-config-operator:node-bootstrapp
csr-8vnps   15m     system:serviceaccount:openshift-machine-config-operator:node-bootstrapp
csr-bfd72   5m26s   system:node:ip-10-0-50-126.us-east-2.compute.internal
csr-c57lv   5m26s   system:node:ip-10-0-95-157.us-east-2.compute.internal
...
```

1 A client request CSR.

2 A server request CSR.

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

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

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

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

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

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

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

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

Initial Operator configuration

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

Prerequisites

- Your control plane has initialized.

Procedure

1. Watch the cluster components come online:

```
$ watch -n5 oc get clusteroperators

NAME                                 VERSION   AVAILABLE   PROGRESSING   DEGRADED   SINCE
authentication                       4.1.0     True        False         False      69s
cloud-credential                     4.1.0     True        False         False      12m
cluster-autoscaler                   4.1.0     True        False         False      11m
console                              4.1.0     True        False         False      46s
dns                                  4.1.0     True        False         False      11m
image-registry                       4.1.0     False       True          False      5m26s
ingress                              4.1.0     True        False         False      5m36s
kube-apiserver                       4.1.0     True        False         False      8m53s
kube-controller-manager              4.1.0     True        False         False      7m24s
kube-scheduler                       4.1.0     True        False         False      12m
machine-api                          4.1.0     True        False         False      12m
machine-config                       4.1.0     True        False         False      7m36s
marketplace                          4.1.0     True        False         False      7m54m
monitoring                           4.1.0     True        False         False      7h54s
network                              4.1.0     True        False         False      5m9s
node-tuning                          4.1.0     True        False         False      11m
openshift-apiserver                  4.1.0     True        False         False      11m
openshift-controller-manager         4.1.0     True        False         False      5m943s
openshift-samples                    4.1.0     True        False         False      3m55s
operator-lifecycle-manager           4.1.0     True        False         False      11m
operator-lifecycle-manager-catalog   4.1.0     True        False         False      11m
service-ca                           4.1.0     True        False         False      11m
service-catalog-apiserver            4.1.0     True        False         False      5m26s
service-catalog-controller-manager   4.1.0     True        False         False      5m25s
storage                              4.1.0     True        False         False      5m30s
```

2. Configure the Operators that are not available.

Image registry storage configuration

If the **`image-registry`** Operator is not available, you must configure storage for it. Instructions for both configuring a PersistentVolume, which is required for production clusters, and for configuring an empty directory as the storage location, which is available for only non-production clusters, are shown.

CONFIGURING REGISTRY STORAGE FOR BARE METAL

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

Prerequisites

- Cluster administrator permissions.
- A cluster on bare metal.
- A provisioned persistent volume (PV) with **`ReadWriteMany`** access mode, such as **`NFS`**.
- Must have "100Gi" capacity.

Procedure

1. To configure your registry to use storage, change the **`spec.storage.pvc`** in the **`configs.imageregistry/cluster`** resource.
2. Verify you do not have a registry pod:

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

If the storage type is **`emptyDIR`**, the replica number can not be greater than **`1`**. If the storage type is **`NFS`**, and you want to scale up the registry Pod by setting **`replica>1`** you must enable the **`no_wdelay`** mount option. For example:

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

1. Check the registry configuration:

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

storage:
  pvc:
    claim:
```

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

2. Check the **`clusteroperator`** status:

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

CONFIGURING STORAGE FOR THE IMAGE REGISTRY IN NON-PRODUCTION CLUSTERS

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

Procedure

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

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

⚠ Configure this option for only non-production clusters.

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

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

Wait a few minutes and run the command again.

Completing installation on user-provisioned infrastructure

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

Prerequisites

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

Procedure

1. Confirm that all the cluster components are online:

```
$ watch -n5 oc get clusteroperators

NAME                                 VERSION   AVAILABLE   PROGRESSING   DEGRADED   SINCE
authentication                       4.1.0     True        False         False      10m
cloud-credential                     4.1.0     True        False         False      22m
cluster-autoscaler                   4.1.0     True        False         False      21m
console                              4.1.0     True        False         False      10m
dns                                  4.1.0     True        False         False      21m
image-registry                       4.1.0     True        False         False      16m
ingress                              4.1.0     True        False         False      16m
kube-apiserver                       4.1.0     True        False         False      19m
kube-controller-manager              4.1.0     True        False         False      18m
kube-scheduler                       4.1.0     True        False         False      22m
machine-api                          4.1.0     True        False         False      22m
machine-config                       4.1.0     True        False         False      18m
marketplace                          4.1.0     True        False         False      18m
monitoring                           4.1.0     True        False         False      18m
network                              4.1.0     True        False         False      16m
node-tuning                          4.1.0     True        False         False      21m
openshift-apiserver                  4.1.0     True        False         False      21m
openshift-controller-manager         4.1.0     True        False         False      17m
openshift-samples                    4.1.0     True        False         False      14m
operator-lifecycle-manager           4.1.0     True        False         False      21m
operator-lifecycle-manager-catalog   4.1.0     True        False         False      21m
service-ca                           4.1.0     True        False         False      21m
service-catalog-apiserver            4.1.0     True        False         False      16m
service-catalog-controller-manager   4.1.0     True        False         False      16m
storage                              4.1.0     True        False         False      16m
```

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

2. Monitor for cluster completion:

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

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

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

3. Confirm that the Kubernetes API server is communicating with the Pods.

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

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

NAMESPACE                         NAME                                            REA
openshift-apiserver-operator      openshift-apiserver-operator-85cb746d55-zqhs8   1/1
openshift-apiserver               apiserver-67b9g                                 1/1
openshift-apiserver               apiserver-ljcmx                                 1/1
openshift-apiserver               apiserver-z25h4                                 1/1
openshift-authentication-operator authentication-operator-69d5d8bf84-vh2n8        1/1
...
```

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

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

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

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

Next steps

- Customize your cluster (../../installing/install_config/customizations.html#customizations).
- If necessary, you can opt out of telemetry (../../telemetry/opting-out-of-telemetry.html#opting-out-of-telemetry).