OpenShift Dedicated 4

Cluster administration

Configuring OpenShift Dedicated clusters

Last Updated: 2024-04-25

OpenShift Dedicated 4 Cluster administration

Configuring OpenShift Dedicated clusters

Legal Notice

Abstract

This document provides information about configuring OpenShift Dedicated clusters.

Table of Contents

CHAPTER 1. CONFIGURING PRIVATE CONNECTIONS

1.1. CONFIGURING PRIVATE CONNECTIONS FOR AWS

1.1.1. Understanding AWS cloud infrastructure access

NOTE

AWS cloud infrastructure access does not apply to the Customer Cloud Subscription (CCS) infrastructure type that is chosen when you create a cluster because CCS clusters are deployed onto your account.

Amazon Web Services (AWS) infrastructure access permits Customer Portal Organization Administrators and cluster owners to enable AWS Identity and Access Management (IAM) users to have federated access to the AWS Management Console for their OpenShift Dedicated cluster. AWS access can be granted for customer AWS users, and private cluster access can be implemented to suit the needs of your OpenShift Dedicated environment.

1. Get started with configuring AWS infrastructure access for your OpenShift Dedicated cluster. By creating an AWS user and account and providing that user with access to the OpenShift Dedicated AWS account.

2. After you have access to the OpenShift Dedicated AWS account, use one or more of the following methods to establish a private connection to your cluster:

   - Configuring AWS VPC peering: Enable VPC peering to route network traffic between two private IP addresses.

   - Configuring AWS VPN: Establish a Virtual Private Network to securely connect your private network to your Amazon Virtual Private Cloud.

   - Configuring AWS Direct Connect: Configure AWS Direct Connect to establish a dedicated network connection between your private network and an AWS Direct Connect location.

After configuring your cloud infrastructure access, learn more about Configuring a private cluster.

1.1.2. Configuring AWS infrastructure access

Amazon Web Services (AWS) infrastructure access allows Customer Portal Organization Administrators and cluster owners to enable AWS Identity and Access Management (IAM) users to have federated access to the AWS Management Console for their OpenShift Dedicated cluster. Administrators can select between **Network Management** or **Read-only** access options.

Prerequisites

- An AWS account with IAM permissions.

Procedure

1. Log in to your AWS account. If necessary, you can create a new AWS account by following the AWS documentation.

2. Create an IAM user with **STS:AllowAssumeRole** permissions within the AWS account.

a. Open the IAM dashboard of the AWS Management Console.

b. In the **Policies** section, click **Create Policy**.

c. Select the **JSON** tab and replace the existing text with the following:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "*"
    }
  ]
}
```

d. Click **Next:Tags**.

e. Optional: Add tags. Click **Next:Review**

f. Provide an appropriate name and description, then click **Create Policy**.

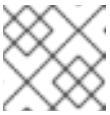
g. In the **Users** section, click **Add user**.

h. Provide an appropriate user name.

i. Select **AWS Management Console access** as the AWS access type.

j. Adjust the password requirements as necessary for your organization, then click **Next:Permissions**.

k. Click the **Attach existing policies directly** option. Search for and check the policy created in previous steps.



NOTE

It is not recommended to set a permissions boundary.

l. Click **Next: Tags**, then click **Next: Review**. Confirm the configuration is correct.

m. Click **Create user**, a success page appears.

n. Gather the IAM user's Amazon Resource Name (ARN). The ARN will have the following format: **arn:aws:iam::000111222333:user/username**. Click **Close**.

3. Open OpenShift Cluster Manager in your browser and select the cluster you want to allow AWS infrastructure access.

4. Select the **Access control** tab, and scroll to the **AWS Infrastructure Access** section.

5. Paste the **AWS IAM ARN** and select **Network Management** or **Read-only** permissions, then click **Grant role**.

6. Copy the **AWS OSD console URL** to your clipboard.

7. Sign in to your AWS account with your Account ID or alias, IAM user name, and password.
8. In a new browser tab, paste the AWS OSD Console URL that will be used to route to the AWS Switch Role page.
9. Your account number and role will be filled in already. Choose a display name if necessary, then click **Switch Role**.

Verification

- You now see **VPC** under **Recently visited services**.

1.1.3. Configuring AWS VPC peering

A Virtual Private Cloud (VPC) peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. You can configure an Amazon Web Services (AWS) VPC containing an OpenShift Dedicated cluster to peer with another AWS VPC network.

WARNING

Before you attempt to uninstall a cluster, you must remove any VPC peering connections from the cluster's VPC. Failure to do so might result in a cluster not completing the uninstall process.

AWS supports inter-region VPC peering between all commercial regions excluding China.

Prerequisites

- Gather the following information about the Customer VPC that is required to initiate the peering request:
  - Customer AWS account number
  - Customer VPC ID
  - Customer VPC Region
  - Customer VPC CIDR
- Check the CIDR block used by the OpenShift Dedicated Cluster VPC. If it overlaps or matches the CIDR block for the Customer VPC, then peering between these two VPCs is not possible; see the Amazon VPC Unsupported VPC peering configurations documentation for details. If the CIDR blocks do not overlap, you can proceed with the procedure.

Procedure

1. Initiate the VPC peering request.
2. Accept the VPC peering request.

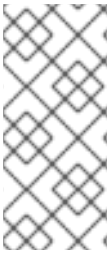
3. Update your Route tables for the VPC peering connection .

Additional resources

- For more information and troubleshooting help, see the AWS VPC guide.

1.1.4. Configuring an AWS VPN

You can configure an Amazon Web Services (AWS) OpenShift Dedicated cluster to use a customer's on-site hardware Virtual Private Network (VPN) device. By default, instances that you launch into an AWS Virtual Private Cloud (VPC) cannot communicate with your own (remote) network. You can enable access to your remote network from your VPC by creating an AWS Site-to-Site VPN connection, and configuring routing to pass traffic through the connection.



NOTE

AWS VPN does not currently provide a managed option to apply NAT to VPN traffic. See the AWS Knowledge Center for more details.

Routing all traffic, for example **0.0.0.0/0**, through a private connection is not supported. This requires deleting the internet gateway, which disables SRE management traffic.

Prerequisites

- Hardware VPN gateway device model and software version, for example Cisco ASA running version 8.3. See the AWS documentation to confirm whether your gateway device is supported by AWS.
- Public, static IP address for the VPN gateway device.
- BGP or static routing: if BGP, the ASN is required. If static routing, you must configure at least one static route.
- Optional: IP and port/protocol of a reachable service to test the VPN connection.

Procedure

1. Create a customer gateway to configure the VPN connection.
2. If you do not already have a Virtual Private Gateway attached to the intended VPC, create and attach a Virtual Private Gateway.
3. Configure routing and enable VPN route propagation .
4. Update your security group .
5. Establish the Site-to-Site VPN connection .

NOTE

Note the VPC subnet information, which you must add to your configuration as the remote network.

Additional resources

- For more information and troubleshooting help, see the AWS VPN guide.

1.1.5. Configuring AWS Direct Connect

Amazon Web Services (AWS) Direct Connect requires a hosted Virtual Interface (VIF) connected to a Direct Connect Gateway (DXGateway), which is in turn associated to a Virtual Gateway (VGW) or a Transit Gateway in order to access a remote Virtual Private Cloud (VPC) in the same or another account.

If you do not have an existing DXGateway, the typical process involves creating the hosted VIF, with the DXGateway and VGW being created in your AWS account.

If you have an existing DXGateway connected to one or more existing VGWs, the process involves your AWS account sending an Association Proposal to the DXGateway owner. The DXGateway owner must ensure that the proposed CIDR will not conflict with any other VGWs they have associated.

Prerequisites

- Confirm the CIDR range of the OpenShift Dedicated VPC will not conflict with any other VGWs you have associated.
- Gather the following information:
  - The Direct Connect Gateway ID.
  - The AWS Account ID associated with the virtual interface.
  - The BGP ASN assigned for the DXGateway. Optional: the Amazon default ASN may also be used.

Procedure

1. Create a VIF or view your existing VIFs to determine the type of direct connection you need to create.
2. Create your gateway.
   a. If the Direct Connect VIF type is **Private**, create a virtual private gateway.
   b. If the Direct Connect VIF is **Public**, create a Direct Connect gateway.
3. If you have an existing gateway you want to use, create an association proposal and send the proposal to the DXGateway owner for approval.

WARNING

When connecting to an existing DXGateway, you are responsible for the costs.

Additional resources

- For more information and troubleshooting help, see the AWS Direct Connect guide.

1.2. CONFIGURING A PRIVATE CLUSTER

An OpenShift Dedicated cluster can be made private so that internal applications can be hosted inside a corporate network. In addition, private clusters can be configured to have only internal API endpoints for increased security.

OpenShift Dedicated administrators can choose between public and private cluster configuration from within **OpenShift Cluster Manager**. Privacy settings can be configured during cluster creation or after a cluster is established.

1.2.1. Enabling a private cluster during cluster creation

You can enable private cluster settings when creating a new cluster.

Prerequisites

- The following private connections must be configured to allow private access:
  - VPC Peering
  - Cloud VPN
  - DirectConnect (AWS only)
  - TransitGateway (AWS only)
  - Cloud Interconnect (GCP only)

Procedure

1. Log in to OpenShift Cluster Manager.
2. Click **Create cluster → OpenShift Dedicated → Create cluster**.
3. Configure your cluster details.
4. When selecting your preferred network configuration, select **Advanced**.
5. Select **Private**.

WARNING

When set to **Private**, you cannot access your cluster unless you have configured the private connections in your cloud provider as outlined in the prerequisites.

6. Click **Create cluster**. The cluster creation process begins and takes about 30-40 minutes to complete.

Verification

- The **Installing cluster** heading, under the **Overview** tab, indicates that the cluster is installing and you can view the installation logs from this heading. The **Status** indicator under the **Details** heading indicates when your cluster is **Ready** for use.

1.2.2. Enabling an existing cluster to be private

After a cluster has been created, you can later enable the cluster to be private.

Prerequisites

- The following private connections must be configured to allow private access:
  - VPC Peering
  - Cloud VPN
  - DirectConnect (AWS only)
  - TransitGateway (AWS only)
  - Cloud Interconnect (GCP only)

Procedure

1. Log in to OpenShift Cluster Manager.
2. Select the public cluster you would like to make private.
3. On the **Networking** tab, select **Make API private** under **Control Plane API endpoint**

WARNING

When set to **Private**, you cannot access your cluster unless you have configured the private connections in your cloud provider as outlined in the prerequisites.

4. Click **Change settings**.

NOTE

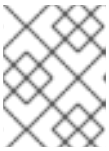
Transitioning your cluster between private and public can take several minutes to complete.

1.2.3. Enabling an existing private cluster to be public

After a private cluster has been created, you can later enable the cluster to be public.

Procedure

1. Log in to OpenShift Cluster Manager.
2. Select the private cluster you would like to make public.
3. On the **Networking** tab, deselect **Make API private** under **Control Plane API endpoint**
4. Click **Change settings**.



NOTE

Transitioning your cluster between private and public can take several minutes to complete.

CHAPTER 2. CLUSTER AUTOSCALING

Applying autoscaling to OpenShift Dedicated clusters involves configuring a cluster autoscaler and then configuring a machine autoscaler for at least one machine pool in your cluster.

IMPORTANT

You can configure the cluster autoscaler only in clusters where the machine API is operational.

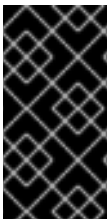
Only one cluster autoscaler can be created per cluster.

2.1. ABOUT THE CLUSTER AUTOSCALER

The cluster autoscaler adjusts the size of an OpenShift Dedicated cluster to meet its current deployment needs. It uses declarative, Kubernetes-style arguments to provide infrastructure management that does not rely on objects of a specific cloud provider. The cluster autoscaler has a cluster scope, and is not associated with a particular namespace.

The cluster autoscaler increases the size of the cluster when there are pods that fail to schedule on any of the current worker nodes due to insufficient resources or when another node is necessary to meet deployment needs. The cluster autoscaler does not increase the cluster resources beyond the limits that you specify.

The cluster autoscaler computes the total memory and CPU on all nodes the cluster, even though it does not manage the control plane nodes. These values are not single-machine oriented. They are an aggregation of all the resources in the entire cluster. For example, if you set the maximum memory resource limit, the cluster autoscaler includes all the nodes in the cluster when calculating the current memory usage. That calculation is then used to determine if the cluster autoscaler has the capacity to add more worker resources.



IMPORTANT

Ensure that the **maxNodesTotal** value in the **ClusterAutoscaler** resource definition that you create is large enough to account for the total possible number of machines in your cluster. This value must encompass the number of control plane machines and the possible number of compute machines that you might scale to.

Every 10 seconds, the cluster autoscaler checks which nodes are unnecessary in the cluster and removes them. The cluster autoscaler considers a node for removal if the following conditions apply:

- The node utilization is less than the *node utilization level* threshold for the cluster. The node utilization level is the sum of the requested resources divided by the allocated resources for the node. If you do not specify a value in the **ClusterAutoscaler** custom resource, the cluster autoscaler uses a default value of **0.5**, which corresponds to 50% utilization.
- The cluster autoscaler can move all pods running on the node to the other nodes. The Kubernetes scheduler is responsible for scheduling pods on the nodes.
- The cluster autoscaler does not have scale down disabled annotation.

If the following types of pods are present on a node, the cluster autoscaler will not remove the node:

- Pods with restrictive pod disruption budgets (PDBs).

- Kube-system pods that do not run on the node by default.
- Kube-system pods that do not have a PDB or have a PDB that is too restrictive.
- Pods that are not backed by a controller object such as a deployment, replica set, or stateful set.
- Pods with local storage.
- Pods that cannot be moved elsewhere because of a lack of resources, incompatible node selectors or affinity, matching anti-affinity, and so on.
- Unless they also have a **"cluster-autoscaler.kubernetes.io/safe-to-evict": "true"** annotation, pods that have a **"cluster-autoscaler.kubernetes.io/safe-to-evict": "false"** annotation.

For example, you set the maximum CPU limit to 64 cores and configure the cluster autoscaler to only create machines that have 8 cores each. If your cluster starts with 30 cores, the cluster autoscaler can add up to 4 more nodes with 32 cores, for a total of 62.

If you configure the cluster autoscaler, additional usage restrictions apply:

- Do not modify the nodes that are in autoscaled node groups directly. All nodes within the same node group have the same capacity and labels and run the same system pods.
- Specify requests for your pods.
- If you have to prevent pods from being deleted too quickly, configure appropriate PDBs.
- Confirm that your cloud provider quota is large enough to support the maximum node pools that you configure.
- Do not run additional node group autoscalers, especially the ones offered by your cloud provider.

The horizontal pod autoscaler (HPA) and the cluster autoscaler modify cluster resources in different ways. The HPA changes the deployment's or replica set's number of replicas based on the current CPU load. If the load increases, the HPA creates new replicas, regardless of the amount of resources available to the cluster. If there are not enough resources, the cluster autoscaler adds resources so that the HPA-created pods can run. If the load decreases, the HPA stops some replicas. If this action causes some nodes to be underutilized or completely empty, the cluster autoscaler deletes the unnecessary nodes.

The cluster autoscaler takes pod priorities into account. The Pod Priority and Preemption feature enables scheduling pods based on priorities if the cluster does not have enough resources, but the cluster autoscaler ensures that the cluster has resources to run all pods. To honor the intention of both features, the cluster autoscaler includes a priority cutoff function. You can use this cutoff to schedule "best-effort" pods, which do not cause the cluster autoscaler to increase resources but instead run only when spare resources are available.

Pods with priority lower than the cutoff value do not cause the cluster to scale up or prevent the cluster from scaling down. No new nodes are added to run the pods, and nodes running these pods might be deleted to free resources.

Cluster autoscaling is supported for the platforms that have machine API available on it.

2.2. ENABLE AUTOSCALING DURING CLUSTER CREATION WITH OPENSHIFT CLUSTER MANAGER

You can use OpenShift Cluster Manager to autoscale during cluster creation.

Procedure

1. During cluster creation, check the **Enable autoscaling** box. The **Edit cluster autoscaling settings** button becomes selectable.
    a. You can also choose the minimum or maximum amount of nodes to autoscale.
2. Click **Edit cluster autoscaling settings**
3. Edit any settings you want and then click **Close**.

2.3. ENABLE AUTOSCALING AFTER CLUSTER CREATION WITH OPENSHIFT CLUSTER MANAGER

You can use OpenShift Cluster Manager to autoscale after cluster creation.

Procedure

1. In OpenShift Cluster Manager, click the name of the cluster you want to autoscale. The Overview page for the cluster has a **Autoscaling** item that indicates if it is enabled or disabled.
2. Click the **Machine Pools** tab.
3. Click the **Edit cluster autoscaling** button. The **Edit cluster autoscaling** settings window is shown.
4. Click the **Autoscale cluster** toggle at the top of the window. All the settings are now editable.
5. Edit any settings you want and then click **Save**.
6. Click the **x** at the top right of the screen to close the settings window.

To revert all autoscaling settings to the defaults if they have been changed, click the **Revert all to defaults** button.

2.4. CLUSTER AUTOSCALING SETTINGS USING OPENSHIFT CLUSTER MANAGER

The tables explain all the configurable UI settings when using cluster autoscaling with OpenShift Cluster Manager.

2.4.1. General settings

Table 2.1. Configurable general settings for cluster autoscaling when using the OpenShift Cluster Manager

Setting	Description	Type or Range	Default

Setting	Description	Type or Range	Default
log-verbosity	Sets the autoscaler log level. The default value is 1. Level 4 is recommended for debugging. Level 6 enables almost everything.	**integer**	1
skip-nodes-with-local-storage	If **true**, the cluster autoscaler never deletes nodes with pods with local storage, e.g. EmptyDir or HostPath.	**boolean**	true
max-pod-grace-period	Gives pods graceful termination time in seconds before scaling down.	**integer**	600
max-node-provision-time	Maximum time the cluster autoscaler waits for nodes to be provisioned.	**string**	15m
pod-priority-threshold	Allows users to schedule "best-effort" pods, which are not expected to trigger cluster autoscaler actions. These pods only run when spare resources are available.	**integer**	-10
ignore-daemonsets-utilization	Determines whether the cluster autoscaler ignores daemon set pods when calculating resource utilization for scaling down.	**boolean**	false
balance-similar-node-groups	If **true**, this setting automatically identifies node groups with the same instance type and the same set of labels and tries to keep the respective sizes of those node groups balanced.	**boolean**	false

Setting	Description	Type or Range	Default
balancing-ignored-labels	This option specifies labels that the cluster autoscaler should ignore when considering node group similarity. For example, if you have nodes with a "topology.ebs.csi.aws.com/zone" label, you can add the name of this label to prevent the cluster autoscaler from splitting nodes into different node groups based on its value. This option cannot contain spaces.	**array (string)**	Format should be a comma-separated list of labels.

2.4.2. Resource limits

Table 2.2. Configurable resource limit settings for cluster autoscaling when using the OpenShift Cluster Manager

Setting	Description	Type or Range	Default
cores-total-min	Minimum number of cores in cluster. The cluster autoscaler does not scale the cluster less than this number.	**object**	0
cores-total-max	Maximum number of cores in cluster. The cluster autoscaler does not scale the cluster greater than this number.	**object**	180 * 64 (11520)
memory-total-min	Minimum number of gigabytes of memory in cluster. The cluster autoscaler does not scale the cluster less than this number.	**object**	0

Setting	Description	Type or Range	Default
memory-total-max	Maximum number of gigabytes of memory in cluster. The cluster autoscaler does not scale the cluster greater than this number.	**object**	180 * 64 * 20 (230400)
max-nodes-total	Maximum number of nodes in all node groups. Includes all nodes, not just automatically scaled nodes. The cluster autoscaler does not grow the cluster greater than this number.	**integer**	180
GPUs	Minimum and maximum number of different GPUs in cluster. The cluster autoscaler does not scale the cluster less than or greater than these numbers.	**array**	Format should be a comma-separated list of "<gpu_type>:<min>:<max>".

2.4.3. Scale down configuration

Table 2.3. Configurable scale down settings for cluster autoscaling when using the OpenShift Cluster Manager

Setting	Description	Type or Range	Default
scale-down-enabled	Should the cluster autoscaler scale down the cluster.	**boolean**	true
scale-down-utilization-threshold	Node utilization level, defined as the sum of the requested resources divided by capacity, below which a node can be considered for scale down.	**float**	0.5
scale-down-unneeded-time	How long a node should be unneeded before it is eligible for scale down.	**string**	10m

Setting	Description	Type or Range	Default
scale-down-delay-after-add	How long after scale up that scale-down evaluation resumes.	**string**	10m
scale-down-delay-after-delete	How long after node deletion that scale-down evaluation resumes.	**string**	0s
scale-down-delay-after-failure	How long after scale down failure that scale-down evaluation resumes.	**string**	3m

CHAPTER 3. NODES

3.1. ABOUT MACHINE POOLS

OpenShift Dedicated uses machine pools as an elastic, dynamic provisioning method on top of your cloud infrastructure.

The primary resources are machines, machine sets, and machine pools.

IMPORTANT

As of OpenShift Dedicated 4.11, the default per-pod PID limit is **4096**. If you want to enable this PID limit, you must upgrade your OpenShift Dedicated clusters to this version or later. OpenShift Dedicated clusters running on earlier versions use a default PID limit of **1024**.

You cannot configure the per-pod PID limit on any OpenShift Dedicated cluster.

3.1.1. Machines

A machine is a fundamental unit that describes the host for a worker node.

3.1.2. Machine sets

MachineSet resources are groups of compute machines. If you need more machines or must scale them down, change the number of replicas in the machine pool to which the compute machine sets belong.

3.1.3. Machine pools

Machine pools are a higher level construct to compute machine sets.

A machine pool creates compute machine sets that are all clones of the same configuration across availability zones. Machine pools perform all of the host node provisioning management actions on a worker node. If you need more machines or must scale them down, change the number of replicas in the machine pool to meet your compute needs. You can manually configure scaling or set autoscaling.

By default, a cluster is created with one machine pool. You can add additional machine pools to an existing cluster, modify the default machine pool, and delete machine pools.

Multiple machine pools can exist on a single cluster, and they can each have different types or different size nodes.

3.1.4. Machine pools in multiple zone clusters

When you create a machine pool in a multiple availability zone (Multi-AZ) cluster, that one machine pool has 3 zones. The machine pool, in turn, creates a total of 3 compute machine sets - one for each zone in the cluster. Each of those compute machine sets manages one or more machines in its respective availability zone.

If you create a new Multi-AZ cluster, the machine pools are replicated to those zones automatically. If you add a machine pool to an existing Multi-AZ, the new pool is automatically created in those zones. Similarly, deleting a machine pool will delete it from all zones. Due to this multiplicative effect, using machine pools in Multi-AZ cluster can consume more of your project's quota for a specific region when creating machine pools.

3.1.5. Additional resources

- About autoscaling

3.2. MANAGING COMPUTE NODES

This document describes how to manage compute (also known as worker) nodes with OpenShift Dedicated.

The majority of changes for compute nodes are configured on machine pools. A machine pool is a group of compute nodes in a cluster that have the same configuration, providing ease of management.

You can edit machine pool configuration options such as scaling, adding node labels, and adding taints.

3.2.1. Creating a machine pool

A machine pool is created when you install an OpenShift Dedicated cluster. After installation, you can create additional machine pools for your cluster by using OpenShift Cluster Manager.

IMPORTANT

The compute (also known as worker) node instance types, autoscaling options, and node counts that are available depend on your OpenShift Dedicated subscriptions, resource quotas and deployment scenario. For more information, contact your sales representative or Red Hat support.

Prerequisites

- You created an OpenShift Dedicated cluster.

Procedure

1. Navigate to OpenShift Cluster Manager and select your cluster.
2. Under the **Machine pools** tab, click **Add machine pool**.
3. Add a **Machine pool name**.
4. Select a **Compute node instance type** from the drop-down menu. The instance type defines the vCPU and memory allocation for each compute node in the machine pool.

NOTE

You cannot change the instance type for a machine pool after the pool is created.

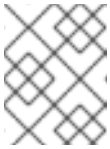
5. Optional: Configure autoscaling for the machine pool:
   a. Select **Enable autoscaling** to automatically scale the number of machines in your machine pool to meet the deployment needs.

NOTE

The **Enable autoscaling** option is only available for OpenShift Dedicated if you have the **capability.cluster.autoscale_clusters** subscription. For more information, contact your sales representative or Red Hat support.

b. Set the minimum and maximum node count limits for autoscaling. The cluster autoscaler does not reduce or increase the machine pool node count beyond the limits that you specify.

- If you deployed your cluster using a single availability zone, set the **Minimum and maximum node count**. This defines the minimum and maximum compute node limits in the availability zone.
- If you deployed your cluster using multiple availability zones, set the **Minimum nodes per zone** and **Maximum nodes per zone**. This defines the minimum and maximum compute node limits per zone.



NOTE

Alternatively, you can set your autoscaling preferences for the machine pool after the machine pool is created.

6. If you did not enable autoscaling, select a compute node count:

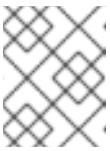
- If you deployed your cluster using a single availability zone, select a **Compute node count** from the drop-down menu. This defines the number of compute nodes to provision to the machine pool for the zone.
- If you deployed your cluster using multiple availability zones, select a **Compute node count (per zone)** from the drop-down menu. This defines the number of compute nodes to provision to the machine pool per zone.

7. Optional: Add node labels and taints for your machine pool:

a. Expand the **Edit node labels and taints** menu.

b. Under **Node labels**, add **Key** and **Value** entries for your node labels.

c. Under **Taints**, add **Key** and **Value** entries for your taints.



NOTE

Creating a machine pool with taints is only possible if the cluster already has at least one machine pool without a taint.

d. For each taint, select an **Effect** from the drop-down menu. Available options include **NoSchedule**, **PreferNoSchedule**, and **NoExecute**.

NOTE

Alternatively, you can add the node labels and taints after you create the machine pool.

8. Optional: Select additional custom security groups to use for nodes in this machine pool. You must have already created the security groups and associated them with the VPC that you selected for this cluster. You cannot add or edit security groups after you create the machine pool. For more information, see the requirements for security groups in the "Additional resources" section.

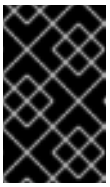
IMPORTANT

You can use up to ten additional security groups for machine pools on ROSA with HCP clusters.

9. Optional: If you deployed OpenShift Dedicated on AWS using the Customer Cloud Subscription (CCS) model, use Amazon EC2 Spot Instances if you want to configure your machine pool to deploy machines as non-guaranteed AWS Spot Instances:

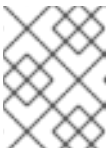
   a. Select **Use Amazon EC2 Spot Instances**.

   b. Leave **Use On-Demand instance price** selected to use the on-demand instance price. Alternatively, select **Set maximum price** to define a maximum hourly price for a Spot Instance.
   For more information about Amazon EC2 Spot Instances, see the AWS documentation.



IMPORTANT

Your Amazon EC2 Spot Instances might be interrupted at any time. Use Amazon EC2 Spot Instances only for workloads that can tolerate interruptions.



NOTE

If you select **Use Amazon EC2 Spot Instances** for a machine pool, you cannot disable the option after the machine pool is created.

10. Click **Add machine pool** to create the machine pool.

Verification

- Verify that the machine pool is visible on the **Machine pools** page and the configuration is as expected.

3.2.2. Deleting a machine pool

You can delete a machine pool in the event that your workload requirements have changed and your current machine pools no longer meet your needs.

You can delete machine pools using the OpenShift Cluster Manager.

Prerequisites

- You have created an OpenShift Dedicated cluster.
- The cluster is in the ready state.

- You have an existing machine pool without any taints and with at least two replicas for a Single-AZ cluster or three replicas for a Multi-AZ cluster.

Procedure

1. From OpenShift Cluster Manager, navigate to the **Clusters** page and select the cluster that contains the machine pool that you want to delete.
2. On the selected cluster, select the **Machine pools** tab.
3. Under the **Machine pools** tab, click the options menu  for the machine pool that you want to delete.
4. Click **Delete**.

The selected machine pool is deleted.

3.2.3. Scaling compute nodes manually

If you have not enabled autoscaling for your machine pool, you can manually scale the number of compute (also known as worker) nodes in the pool to meet your deployment needs.

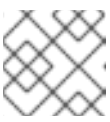
You must scale each machine pool separately.

Prerequisites

- You created an OpenShift Dedicated cluster.
- You have an existing machine pool.

Procedure

1. Navigate to OpenShift Cluster Manager and select your cluster.
2. Under the **Machine pools** tab, click the options menu  for the machine pool that you want to scale.
3. Select **Scale**.
4. Specify the node count:
   - If you deployed your cluster using a single availability zone, specify the **Node count** in the drop-down menu.
   - If you deployed your cluster using multiple availability zones, specify the **Node count per zone** in the drop-down menu.



NOTE

Your subscription determines the number of nodes that you can select.

5. Click **Apply** to scale the machine pool.

Verification

- Under the **Machine pools** tab, verify that the **Node count** for your machine pool is as expected.

3.2.4. Node labels

A label is a key-value pair applied to a **Node** object. You can use labels to organize sets of objects and control the scheduling of pods.

You can add labels during cluster creation or after. Labels can be modified or updated at any time.

Additional resources

- For more information about labels, see Kubernetes Labels and Selectors overview .
- For more information about custom additional security group requirements, see Additional custom security groups.

3.2.4.1. Adding node labels to a machine pool

Add or edit labels for compute (also known as worker) nodes at any time to manage the nodes in a manner that is relevant to you. For example, you can assign types of workloads to specific nodes.

Labels are assigned as key-value pairs. Each key must be unique to the object it is assigned to.

Prerequisites

- You created an OpenShift Dedicated cluster.
- You have an existing machine pool.

Procedure

1. Navigate to OpenShift Cluster Manager and select your cluster.
2. Under the **Machine pools** tab, click the options menu  for the machine pool that you want to add a label to.
3. Select **Edit labels**.
4. If you have existing labels in the machine pool that you want to remove, select **x** next to the label to delete it.
5. Add a label using the format **<key>=<value>** and press enter. For example, add **app=db** and then press enter. If the format is correct, the key value pair is then highlighted.
6. Repeat the previous step if you want to add additional labels.
7. Click **Save** to apply the labels to the machine pool.

Verification

1. Under the **Machine pools** tab, select **>** next to your machine pool to expand the view.

2. Verify that your labels are listed under **Labels** in the expanded view.

3.2.5. Adding taints to a machine pool

You can add taints for compute (also known as worker) nodes in a machine pool to control which pods are scheduled to them. When you apply a taint to a machine pool, the scheduler cannot place a pod on the nodes in the pool unless the pod specification includes a toleration for the taint.

NOTE

A cluster must have at least one machine pool that does not contain any taints.

Prerequisites

- You created an OpenShift Dedicated cluster.
- You have an existing machine pool that does not contain any taints and contains at least two instances.

Procedure

1. Navigate to OpenShift Cluster Manager and select your cluster.
2. Under the **Machine pools** tab, click the options menu  for the machine pool that you want to add a taint to.
3. Select **Edit taints**.
4. Add **Key** and **Value** entries for your taint.
5. Select an **Effect** for your taint from the drop-down menu. Available options include **NoSchedule**, **PreferNoSchedule**, and **NoExecute**.
6. Select **Add taint** if you want to add more taints to the machine pool.
7. Click **Save** to apply the taints to the machine pool.

Verification

1. Under the **Machine pools** tab, select > next to your machine pool to expand the view.
2. Verify that your taints are listed under **Taints** in the expanded view.

3.2.6. Additional resources

- About machine pools
- Enabling autoscaling
- Disabling autoscaling
- OpenShift Dedicated service definition

3.3. ABOUT AUTOSCALING NODES ON A CLUSTER

IMPORTANT

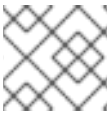
Autoscaling is available only on clusters that were purchased through Google Cloud Marketplace and Red Hat Marketplace.

The autoscaler option can be configured to automatically scale the number of machines in a cluster.

The cluster autoscaler increases the size of the cluster when there are pods that failed to schedule on any of the current nodes due to insufficient resources or when another node is necessary to meet deployment needs. The cluster autoscaler does not increase the cluster resources beyond the limits that you specify.

Additionally, the cluster autoscaler decreases the size of the cluster when some nodes are consistently not needed for a significant period, such as when it has low resource use and all of its important pods can fit on other nodes.

When you enable autoscaling, you must also set a minimum and maximum number of worker nodes.



NOTE

Only cluster owners and organization admins can scale or delete a cluster.

3.3.1. Enabling autoscaling nodes on a cluster

You can enable autoscaling on worker nodes to increase or decrease the number of nodes available by editing the machine pool definition for an existing cluster.

Enabling autoscaling nodes in an existing cluster using Red Hat OpenShift Cluster Manager
Enable autoscaling for worker nodes in the machine pool definition from OpenShift Cluster Manager console.

Procedure

1. From OpenShift Cluster Manager, navigate to the **Clusters** page and select the cluster that you want to enable autoscaling for.
2. On the selected cluster, select the **Machine pools** tab.
3. Click the Options menu  at the end of the machine pool that you want to enable autoscaling for and select **Scale**.
4. On the **Edit node count** dialog, select the **Enable autoscaling** checkbox.
5. Select **Apply** to save these changes and enable autoscaling for the cluster.

3.3.2. Disabling autoscaling nodes on a cluster

You can disable autoscaling on worker nodes to increase or decrease the number of nodes available by editing the machine pool definition for an existing cluster.

You can disable autoscaling on a cluster using OpenShift Cluster Manager console.

Disabling autoscaling nodes in an existing cluster using Red Hat OpenShift Cluster Manager
Disable autoscaling for worker nodes in the machine pool definition from OpenShift Cluster Manager console.

Procedure

1. From OpenShift Cluster Manager, navigate to the **Clusters** page and select the cluster with autoscaling that must be disabled.
2. On the selected cluster, select the **Machine pools** tab.
3. Click the Options menu  at the end of the machine pool with autoscaling and select **Scale**.
4. On the "Edit node count" dialog, deselect the **Enable autoscaling** checkbox.
5. Select **Apply** to save these changes and disable autoscaling from the cluster.

Applying autoscaling to an OpenShift Dedicated cluster involves deploying a cluster autoscaler and then deploying machine autoscalers for each machine type in your cluster.

IMPORTANT

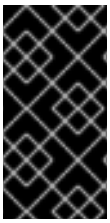
You can configure the cluster autoscaler only in clusters where the Machine API is operational.

3.3.3. About the cluster autoscaler

The cluster autoscaler adjusts the size of an OpenShift Dedicated cluster to meet its current deployment needs. It uses declarative, Kubernetes-style arguments to provide infrastructure management that does not rely on objects of a specific cloud provider. The cluster autoscaler has a cluster scope, and is not associated with a particular namespace.

The cluster autoscaler increases the size of the cluster when there are pods that fail to schedule on any of the current worker nodes due to insufficient resources or when another node is necessary to meet deployment needs. The cluster autoscaler does not increase the cluster resources beyond the limits that you specify.

The cluster autoscaler computes the total memory and CPU on all nodes the cluster, even though it does not manage the control plane nodes. These values are not single-machine oriented. They are an aggregation of all the resources in the entire cluster. For example, if you set the maximum memory resource limit, the cluster autoscaler includes all the nodes in the cluster when calculating the current memory usage. That calculation is then used to determine if the cluster autoscaler has the capacity to add more worker resources.



IMPORTANT

Ensure that the **maxNodesTotal** value in the **ClusterAutoscaler** resource definition that you create is large enough to account for the total possible number of machines in your cluster. This value must encompass the number of control plane machines and the possible number of compute machines that you might scale to.

Every 10 seconds, the cluster autoscaler checks which nodes are unnecessary in the cluster and removes them. The cluster autoscaler considers a node for removal if the following conditions apply:

- The node utilization is less than the *node utilization level* threshold for the cluster. The node utilization level is the sum of the requested resources divided by the allocated resources for the node. If you do not specify a value in the **ClusterAutoscaler** custom resource, the cluster autoscaler uses a default value of **0.5**, which corresponds to 50% utilization.
- The cluster autoscaler can move all pods running on the node to the other nodes. The Kubernetes scheduler is responsible for scheduling pods on the nodes.
- The cluster autoscaler does not have scale down disabled annotation.

If the following types of pods are present on a node, the cluster autoscaler will not remove the node:

- Pods with restrictive pod disruption budgets (PDBs).
- Kube-system pods that do not run on the node by default.
- Kube-system pods that do not have a PDB or have a PDB that is too restrictive.
- Pods that are not backed by a controller object such as a deployment, replica set, or stateful set.
- Pods with local storage.
- Pods that cannot be moved elsewhere because of a lack of resources, incompatible node selectors or affinity, matching anti-affinity, and so on.
- Unless they also have a **"cluster-autoscaler.kubernetes.io/safe-to-evict": "true"** annotation, pods that have a **"cluster-autoscaler.kubernetes.io/safe-to-evict": "false"** annotation.

For example, you set the maximum CPU limit to 64 cores and configure the cluster autoscaler to only create machines that have 8 cores each. If your cluster starts with 30 cores, the cluster autoscaler can add up to 4 more nodes with 32 cores, for a total of 62.

If you configure the cluster autoscaler, additional usage restrictions apply:

- Do not modify the nodes that are in autoscaled node groups directly. All nodes within the same node group have the same capacity and labels and run the same system pods.
- Specify requests for your pods.
- If you have to prevent pods from being deleted too quickly, configure appropriate PDBs.
- Confirm that your cloud provider quota is large enough to support the maximum node pools that you configure.
- Do not run additional node group autoscalers, especially the ones offered by your cloud provider.

The horizontal pod autoscaler (HPA) and the cluster autoscaler modify cluster resources in different ways. The HPA changes the deployment's or replica set's number of replicas based on the current CPU load. If the load increases, the HPA creates new replicas, regardless of the amount of resources available to the cluster. If there are not enough resources, the cluster autoscaler adds resources so that the HPA-created pods can run. If the load decreases, the HPA stops some replicas. If this action causes some nodes to be underutilized or completely empty, the cluster autoscaler deletes the unnecessary nodes.

The cluster autoscaler takes pod priorities into account. The Pod Priority and Preemption feature enables scheduling pods based on priorities if the cluster does not have enough resources, but the cluster autoscaler ensures that the cluster has resources to run all pods. To honor the intention of both

features, the cluster autoscaler includes a priority cutoff function. You can use this cutoff to schedule "best-effort" pods, which do not cause the cluster autoscaler to increase resources but instead run only when spare resources are available.

Pods with priority lower than the cutoff value do not cause the cluster to scale up or prevent the cluster from scaling down. No new nodes are added to run the pods, and nodes running these pods might be deleted to free resources.

Cluster autoscaling is supported for the platforms that have machine API available on it.

3.3.4. Additional resources

- About machinepools

CHAPTER 4. LOGGING

4.1. ACCESSING THE SERVICE LOGS FOR OPENSHIFT DEDICATED CLUSTERS

You can view the service logs for your OpenShift Dedicated clusters by using Red Hat OpenShift Cluster Manager. The service logs detail cluster events such as load balancer quota updates and scheduled maintenance upgrades. The logs also show cluster resource changes such as the addition or deletion of users, groups, and identity providers.

Additionally, you can add notification contacts for an OpenShift Dedicated cluster. Subscribed users receive emails about cluster events that require customer action, known cluster incidents, upgrade maintenance, and other topics.

4.1.1. Viewing the service logs by using OpenShift Cluster Manager

You can view the service logs for an OpenShift Dedicated cluster by using Red Hat OpenShift Cluster Manager.

Prerequisites

- You have installed an OpenShift Dedicated cluster.

Procedure

1. Navigate to OpenShift Cluster Manager and select your cluster.
2. In the **Overview** page for your cluster, view the service logs in the **Cluster history** section.
3. Optional: Filter the cluster service logs by **Description** or **Severity** from the drop-down menu. You can filter further by entering a specific item in the search bar.
4. Optional: Click **Download history** to download the service logs for your cluster in JSON or CSV format.

4.1.2. Adding cluster notification contacts

You can add notification contacts for your OpenShift Dedicated cluster. When an event occurs that triggers a cluster notification email, subscribed users are notified.

Procedure

1. Navigate to OpenShift Cluster Manager and select your cluster.
2. On the **Support** tab, under the **Notification contacts** heading, click **Add notification contact**.
3. Enter the Red Hat username or email of the contact you want to add.

NOTE

The username or email address must relate to a user account in the Red Hat organization where the cluster is deployed.

4. Click **Add contact**.

Verification

- You see a confirmation message when you have successfully added the contact. The user appears under the **Notification contacts** heading on the **Support** tab.