OpenShift Dedicated 4

Nodes

Managing nodes in OpenShift Dedicated clusters
OpenShift Dedicated 4 Nodes

Managing nodes in OpenShift Dedicated clusters
Abstract

This document provides instructions for managing machine pools and node autoscaling on OpenShift Dedicated clusters.
# Table of Contents

## CHAPTER 1. ABOUT MACHINE POOLS

1.1. MACHINES .................................................. 3
1.2. MACHINE SETS .......................................... 3
1.3. MACHINE POOLS ........................................ 3
1.4. MACHINE POOLS IN MULTIPLE ZONE CLUSTERS .... 3
1.5. ADDITIONAL RESOURCES ......................... 3

## CHAPTER 2. MANAGING COMPUTE NODES

2.1. CREATING A MACHINE POOL ................. 4
2.2. SCALING COMPUTE NODES MANUALLY ....... 6
2.3. ADDING NODE LABELS TO A MACHINE POOL ... 7
2.4. ADDING Taints TO A MACHINE POOL .......... 7
2.5. ADDITIONAL RESOURCES ................... 8

## CHAPTER 3. ABOUT AUTOSCALING NODES ON A CLUSTER

3.1. ENABLING AUTOSCALING NODES ON A CLUSTER 9
   Enabling autoscaling nodes in an existing cluster using Red Hat OpenShift Cluster Manager 9
3.2. DISABLING AUTOSCALING NODES ON A CLUSTER 9
   Disabling autoscaling nodes in an existing cluster using Red Hat OpenShift Cluster Manager 10
3.3. ABOUT THE CLUSTER AUTOSCALER .......... 10
3.4. ADDITIONAL RESOURCES ................... 12
CHAPTER 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.

1.1. MACHINES

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

1.2. MACHINE SETS

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

1.3. MACHINE POOLS

Machine pools are a higher level construct to machine sets.

A machine pool creates 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.

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 machine sets - one machine set for each zone in the cluster. Each of those 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.

1.5. ADDITIONAL RESOURCES

- About autoscaling
CHAPTER 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.

2.1. CREATING A MACHINE POOL

A default 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 to you 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 Worker 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 **Worker 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 **Worker 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.

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: 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.

9. 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.

2.2. 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.

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
CHAPTER 2. MANAGING COMPUTE NODES

2.3. 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.

**NOTE**

Adding node labels to an existing machine pool adds the labels to only new nodes that are created in that pool. The labels are not applied to the existing nodes. If you need nodes with the labels, you can scale down the nodes in the machine pool to zero, add the node labels, and then scale back up to the desired amount. Alternatively, you can create a new machine pool and define the node labels at creation time. You cannot scale the default machine pool to zero.

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.

2.4. 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**
Adding taints to an existing machine pool adds the taints to only new nodes that are created in that pool. The taints are not applied to the existing nodes. If you need nodes with the taint, you can scale down the nodes in the machine pool to zero, add the taints, and then scale back up to the desired amount. Alternatively, you can create a new machine pool and define the taints at creation time. You cannot scale the default machine pool to zero.

**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 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.

**2.5. ADDITIONAL RESOURCES**
- About machine pools
- About autoscaling
- Enabling autoscaling
- Disabling autoscaling
- OpenShift Dedicated service definition
Important

Autoscaling is available only on clusters that were purchased through the 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.

3.1. 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.

3.2. 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. 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 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.

3.4. ADDITIONAL RESOURCES

- About machinepools