Appendix A. Scaling an OpenShift Cluster

**Abstract**

|  |  |
| --- | --- |
| **Goal** | Control the size of an OpenShift cluster. |
| **Objectives** | * Describe resources used for scaling an OpenShift cluster, and identify methods to manually scale compute nodes. * Define parameters to automatically control the size of a cluster. |
| **Sections** | * Manually Scaling an OpenShift Cluster * Automatically Scaling an OpenShift Cluster |

Manually Scaling an OpenShift Cluster

Objectives

After completing this section, you should be able to describe resources used for scaling an OpenShift cluster, and identify methods to manually scale compute nodes.

Introducing the Machine API

OpenShift Container Platform can scale the cluster by adding or removing compute nodes through the Machine API. These scaling capabilities allow the cluster to provide enough computing power for your applications. The scaling process can be either manual or automatic, depending on your needs.

The following diagram shows the Machine API (running as an operator) and the API resources that it manages. The operator provides a variety of controllers that interact with the cluster resources, such as the MachineSet controller, which describes a group of compute nodes.

**IMPORTANT**

The machine API operator must be operational in order to perform either manual or automatic scaling. The machine API operator requires cloud provider integration and is available to clusters installed with the full-stack automation method.

|  |
| --- |
|  |

Figure A.1: The Machine API operator

**NOTE**

The API resources for automatic scaling (ClusterAutoscaler and MachineAutoscaler) are not managed by the Machine API operator. The Cluster Autoscaler operator manages these two resources.

The Machine API provides the following custom resources:

**Machines**

Machines are the fundamental compute unit in a cluster. Each machine resource correlates to a physical or virtual node. A machine is an instance from a machine set that inherits information defined in the machine set.

**MachineSets**

A MachineSet describes a group of machines, but does not include control plane nodes. Machine sets are to machines what replica sets are to pods. A machine set defines settings that apply to all machines in the machine set. These settings include labels, such as for regions and zones, and an instance type, such as the AWS m4.xlarge instance type. You can scale the number of machines in a machine set either by modifying the replicas setting in the machine set resource or by running the oc scale command.

Like replica sets, machine sets use labels to determine their members. The machine API uses this information to determine if the machine set has the correct number of machine replicas.

Use machine sets to customize your cluster topology. For example, you might define a unique machine set per region. An OpenShift cluster deployed to AWS creates one machine set per availability zone.

apiVersion: machine.openshift.io/v1beta1

kind: MachineSet

metadata:

name: ocp-xxyy-worker-us-west-2b

*...output omitted...*

spec:

replicas: 1

selector:

matchLabels:

machine.openshift.io/cluster-api-cluster: ocp-rnh2q

machine.openshift.io/cluster-api-machineset: ocp-xxyy-worker-us-west-2b

template:

metadata:

labels:

machine.openshift.io/cluster-api-cluster: ocp-xxyy

machine.openshift.io/cluster-api-machine-role: worker

machine.openshift.io/cluster-api-machine-type: worker

machine.openshift.io/cluster-api-machineset: ocp-xxyy-worker-us-west-2b

spec:

providerSpec:

*...output omitted...*

placement:

availabilityZone: us-west-2b

region: us-west-2

credentialsSecret:

name: aws-cloud-credentials

instanceType: m4.xlarge

*...output omitted...*

**NOTE**

Aside from scaling, changes to a machine set do not affect existing machines. New machines inherit changes made to a machine set, such as new or updated labels.

**MachineAutoscalers**

MachineAutoscalers define the machine sets that have the ability to scale automatically. Machine autoscalers are not required for manual scaling.

**ClusterAutoscaler**

A ClusterAutoscaler resource enables the automatic scaling of machines in a cloud deployment. Automatic scaling ensures that if a compute node becomes unresponsive, then pods are quickly evacuated to another compute node for improved availability. Automatic scaling is useful when the cluster is under stress, or for workloads whose computing requirements change.

The cluster autoscaler resource allows you to specify limits for a variety of components, such as nodes, cores, and memory. A cluster autoscaler is not required for manual scaling.

**NOTE**

The following section discusses both the MachineAutoscalers and ClusterAutoscaler resources.

**MachineHealthChecks**

A MachineHealthCheck verifies the health of a machine (such as a compute node) and takes action if the resource is unhealthy.

Manually Scaling Compute Nodes

Scaling a cluster can be done in two ways: manually and automatically. The availability of automatic scaling depends on the provider and is supported by the full-stack automation method. Reasons to scale a cluster include minimizing resource congestion and better management of resources, such as dedicating a set of compute nodes to a particular workload.

Like deployments and deployment configurations, you change the number of instances by adjusting the replicas specified by the machine set resource. Modify the replicas line in the machine set resource or use the oc scale command.

**[user@demo ~]$ oc scale --replicas 2 \**

**>**  **machineset *MACHINE-SET* -n openshift-machine-api**

If the number of machines in the machine set does not match the replica count, then OpenShift either adds or removes machines until the desired replica count is reached.

Scaling down the machine set instructs the scheduler to evacuate pods running on the machine targeted for removal. Reducing the machine replica count only works if the remaining compute nodes have enough capacity to run the pods. If there are not enough resources, then the controller refuses to reduce the machine replica count.