Automatically Scaling an OpenShift Cluster

Objectives

After completing this section, you should be able to define parameters to automatically control the size of a cluster.

Automatically Scaling a Cluster

The Machine API provides several resources for managing the workloads of your cluster. You can scale your cluster resources in two ways: manually and automatically.

Manual scaling requires updating the number of replicas in a machine set. Automatic scaling of a cluster involves using two custom resources: MachineAutoscaler and ClusterAutoscaler.

**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.

A machine autoscaler resource automatically scales the number of replicas in a machine set, depending on the load. This API resource interacts with machine sets and instructs them to add or remove compute nodes. The machine autoscaler resource supports the definition of lower and upper boundaries. The cluster autoscaler resource enforces limits for the whole cluster, such as the total number of nodes. For example, MaxNodesTotal sets the maximum number of nodes in the whole cluster, and MaxMemoryTotal sets the maximum memory in the whole cluster. If desired, then configure the cluster autoscaler to scale down the number of nodes as load decreases.

Each OpenShift cluster can only have one cluster autoscaler resource. The cluster autoscaler resource operates at a higher level and defines the maximum number of nodes and other resources, such as cores, memory, and Graphical Processing Units (GPUs). These limits prevent machine autoscaler resources from scaling out in an uncontrolled manner.

**NOTE**

You must define a ClusterAutoscaler resource for your cluster, otherwise, automatic scaling does not work.

The following excerpt describes a cluster autoscaler resource:

apiVersion: "autoscaling.openshift.io/v1"

kind: "ClusterAutoscaler"

metadata:

name: "default"

spec:

podPriorityThreshold: -10

resourceLimits:

**maxNodesTotal: 6**

scaleDown:

**enabled: true**

delayAfterAdd: 3m

unneededTime: 3m

Use a machine autoscaler resource to scale the number of machines defined by a machine set resource. Scaling only works if adding a new machine does not exceed any of the values defined by the cluster autoscaler resource. For example, adding one m4.xlarge machine from AWS adds one node, 4 CPU cores, and 16 GB of memory. On its own, a machine autoscaler does not increase or decrease the number of machines unless cluster autoscaling is allowed.

The following diagram shows how the machine autoscaler resource interacts with machine sets to scale the number of compute nodes.

* If the machine autoscaler resource must scale, then it checks to see if a cluster autoscaler resource exists. If it does not, then no scaling occurs.

If a cluster autoscaler resource does exist, then the machine autoscaler resource evaluates whether adding the new machine violates any of the limits defined in the cluster autoscaler resource.

* Provided the request does not exceeds the limits, a new machine is created.
* When the new machine is ready, OpenShift schedules pods to it as a new node.

|  |
| --- |
|  |

Figure A.2: Dynamic scaling with full-stack automation

There is a one-to-one mapping between a machine autoscaler and a machine set resource. This mapping allows each machine autoscaler resource to manage its own machine set. If you want a machine set to have the ability to scale, then create a machine autoscaler resource for it. Alternatively, do not create machine autoscaler resources for machine sets that should not scale.

In a default full-stack automation deployment, each machine set resource maps to an availability zone, but this design is arbitrary. For example, you can use a machine set resource to separate compute nodes based on their disk speeds or functional roles.

Implementing Automatic Scaling

For successful automatic scaling, the following are required:

* A cluster deployed in full-stack automation, because automatic scaling must interact with a cloud service when adding or removing compute nodes.
* A cluster autoscaler resource, if the infrastructure supports it. Additionally, the cluster autoscaler resource might limit the maximum number of nodes, and define minimum and maximum values for cores, memory, and GPUs. The enabled: true entry in the scaleDown section of the cluster autoscaler resource authorizes this resource to automatically scale down the number of machines when they are not used.
* At least one machine autoscaler resource. Each machine autoscaler resource defines the minimum and maximum number of replicas for a specific machine set.
* apiVersion: "autoscaling.openshift.io/v1beta1"
* kind: "MachineAutoscaler"
* metadata:
* name: "***MACHINE-AUTOSCALER-NAME***"
* namespace: "openshift-machine-api"
* spec:
* **minReplicas: 1**
* **maxReplicas: 4**
* scaleTargetRef:
* apiVersion: machine.openshift.io/v1beta1 kind: MachineSet

name: ***MACHINE-SET-NAME***

After creating these resources, if the cluster cannot manage a load, then the automatic addition of compute nodes is triggered.

The cluster autoscaler resource scales out the number of compute nodes when pods fail to schedule on any of the current nodes due to insufficient resources, or when another node is necessary to meet deployment needs.