Cluster Autoscaling

Automatically manage the nodes in your cluster to adapt to demand.

Kubernetes requires nodes in your cluster to run pods. This means providing capacity for the workload Pods and for Kubernetes itself.

You can adjust the amount of resources available in your cluster automatically: *node autoscaling*. You can either change the number of nodes, or change the capacity that nodes provide. The first approach is referred to as *horizontal scaling*, while the second is referred to as *vertical scaling*.

Kubernetes can even provide multidimensional automatic scaling for nodes.

Manual node management

You can manually manage node-level capacity, where you configure a fixed amount of nodes; you can use this approach even if the provisioning (the process to set up, manage, and decommission) for these nodes is automated.

This page is about taking the next step, and automating management of the amount of node capacity (CPU, memory, and other node resources) available in your cluster.

Automatic horizontal scaling

Cluster Autoscaler

You can use the Cluster Autoscaler to manage the scale of your nodes automatically. The cluster autoscaler can integrate with a cloud provider, or with Kubernetes' cluster API, to achieve the actual node management that's needed.

The cluster autoscaler adds nodes when there are unschedulable Pods, and removes nodes when those nodes are empty.

Cloud provider integrations

The README for the cluster autoscaler lists some of the cloud provider integrations that are available.

Cost-aware multidimensional scaling

Karpenter

Karpenter supports direct node management, via plugins that integrate with specific cloud providers, and can manage nodes for you whilst optimizing for overall cost.

> Karpenter automatically launches just the right compute resources to handle your cluster's applications. It is designed to let you take full advantage of the cloud with fast and simple compute provisioning for Kubernetes clusters.

The Karpenter tool is designed to integrate with a cloud provider that provides API-driven server management, and where the price information for available servers is also available via a web API.

For example, if you start some more Pods in your cluster, the Karpenter tool might buy a new node that is larger than one of the nodes you are already using, and then shut down an existing node once the new node is in service.

Cloud provider integrations

> **Note:** Items on this page refer to vendors external to Kubernetes. The Kubernetes project authors aren't responsible for those third-party products or projects. To add a vendor, product or project to this list, read the content guide before submitting a change. More information.

There are integrations available between Karpenter's core and the following cloud providers:

- Amazon Web Services
- Azure

Related components

Descheduler

The descheduler can help you consolidate Pods onto a smaller number of nodes, to help with automatic scale down when the cluster has space capacity.

Sizing a workload based on cluster size

Cluster proportional autoscaler

For workloads that need to be scaled based on the size of the cluster (for example `cluster-dns` or other system components), you can use the *Cluster Proportional Autoscaler*.

The Cluster Proportional Autoscaler watches the number of schedulable nodes and cores, and scales the number of replicas of the target workload accordingly.

Cluster proportional vertical autoscaler

If the number of replicas should stay the same, you can scale your workloads vertically according to the cluster size using the *Cluster Proportional Vertical Autoscaler*. This project is in **beta** and can be found on GitHub.

While the Cluster Proportional Autoscaler scales the number of replicas of a workload, the Cluster Proportional Vertical Autoscaler adjusts the resource requests for a workload (for example a Deployment or DaemonSet) based on the number of nodes and/or cores in the cluster.

What's next

- Read about workload-level autoscaling

Items on this page refer to third party products or projects that provide functionality required by Kubernetes. The Kubernetes project authors aren't responsible for those third-party products or projects. See the CNCF website guidelines for more details.

You should read the content guide before proposing a change that adds an extra third-party link.

Feedback

Was this page helpful?

Yes No

Last modified February 18, 2024 at 2:59 PM PST: Add concept page about cluster autoscaling (b39e01b971)