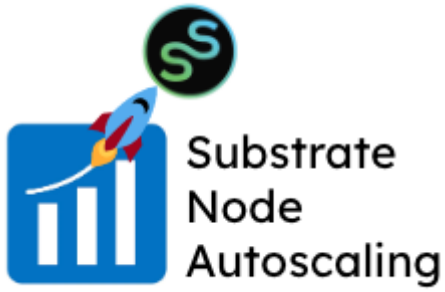
Scaling nodes intro



Name	Kubernetes Node Autoscaling
Description	Autoscaling dynamically scales the cloud infrastructure as to improve reliability, remove complexity, and reduce cost
Project Lead	Eric Greer
Priority	1 - High
Status	PRODUCTION
Next Milestone	26 Oct 2022

Due Date	31 Jan 2023
JIRA EPIC	IAAS-141 - Karpenter+Compactor Kubernetes Node Autos
Latest Update	*Deployed to all production clusters Oct 26th!*

RFC

https://docs.google.com/document/d/1GVjxR4KtmVI2_wDWzJ4Yere2zcTnYprHKsDaur8fvKs/edit?usp=sharing

Technical documentation

Karpenter+Compactor Node Autoscaling Operations Guide

Adding Non-Autoscaling Nodegroups to EKS Clusters

Custom Kubernetes Node Types for Karpenter Autoscaling

Assigning Workloads to Tainted Nodes

Kubernetes Node Autoscaling Operations Guide

Adding Additional Disk to Nodes and Pods

Project Summary

Node autoscaling enables automatic node provisioning in response to pods in the 'Pending' state. Additionally, when there is excess capacity int he cluster (above 30%), nodes will be automatically removed in order to bring the total wasted capacity within 30%. This means that nodes will come and go automatically in response to pods being created. Pods in the state of 'Pending' that require nodes normally take about 3 minutes to get scheduled. This includes the time it takes for a request for capacity to be sent, a node to be provisioned and bootstrapped, and the pod to get scheduled. Additionally, a new AWS API is used to provision nodes and many node types are available (all with NICs faster than 10Gbit

dedicated). This means that if a node type is unavailable for some reason, the next node type will be tried automatically until one is found that is available.

Node autoscaling enables you to stop thinking about and changing node counts in favor of changing pod counts. If you want to pre-scale your service for an event, you would just request more pods, or pods with bigger resource requests. This would in turn result in pods that need scheduled and thereafter nodes would be automatically provisioned for you. No longer will you need to determine how many nodes you should need. With node autoscaling, you just request pods and nodes come along automatically.

Benefits

- In periods of high load, additional node types will be provisioned, which helps work around node quotas and unavailable node types.
- No more guessing at and then monitoring the ideal number of nodes for your cluster. Just pick the right number of application pods and you'll get nodes automatically.
- When pod autoscaling is configured, clusters will scale down when not in use, which saves a lot of money.
- You can now request nodes of various architecture types and with various accelerators like GPU. See Custom Kubernetes Node Types for Karpenter Autoscaling for how to set this up.

How do I use it?

All you have to do is create pods! When any pod is in the `Pending` state but can not be assigned to any existing node, Karpenter will automatically create a new node for you from a list of compatible nodes. When your pods later terminate, your cluster will automatically shrink (slowly over several hours) to ensure that you never have more than 30% free capacity left unused. There is no maximum number of nodes set within Karpenter and nodes will continue to build until your account hits its

instance quota. The goal here is to think less about nodes and more about the pods you want to run.

Ending manual node scaling

You can continue to provision node counts manually, but this means you will continue to be responsible for monitoring these nodes' capacity to ensure that they don't fill up. If you would like to switch away from provisioning nodes, you will need to do the following as a baseline:

- Move all automation that works with upping node counts to instead modify pod counts.
- Implement an HPA (Horizontal Node Autoscaler) or KEDA to automatically create more pods in response to load.
- If you use an HPA or other autoscaling system, ensure that you set the minimum pod count high enough to be able to handle sudden surges in traffic. Pod autoscaling takes several minutes to kick in.

Please reach out to Eric Greer if you would like to talk about the exact behavior at play here or how exactly to do this.
