StormForge

SOLUTION BRIEF

Karpenter + StormForge for the Win

Intelligent Cluster and Pod Autoscaling for Peak Kubernetes Efficiency

Kubernetes autoscaling is essential for maintaining efficiency and maximizing cloud ROI, but it's also complicated. There are three primary ways to scale your Kubernetes environment, listed from most granular to least:

- **Vertical Pod Autoscaling** increases or decreases the resources (CPU and memory) within existing pods based on changing usage.
- **Horizontal Pod Autoscaling** adds or removes pods in response to changing usage.
- **Cluster Autoscaling** increases or decreases the number of nodes in a cluster based on node utilization metrics and the existence of pending pods.

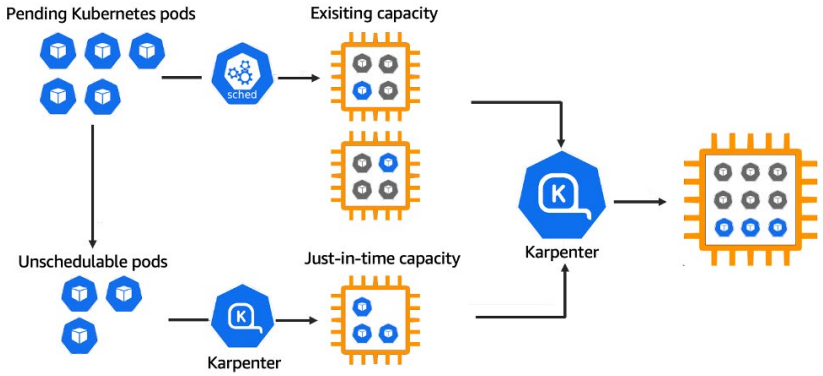
Kubernetes includes default tools for each of the three types of autoscaling: Vertical Pod Autoscaler (VPA), Horizontal Pod Autoscaler (HPA), and Cluster Autoscaler (CA). But each of these tools can be complex. They lack flexibility and scale based on simplistic metrics using minimal intelligence – and they don't always work together nicely. For example, by default the VPA and HPA both scale based on the same metric (CPU utilization), which invariably leads to thrashing. Also, if resource inefficiencies aren't addressed at the pod level, they can easily multiply when scaling horizontally or at the cluster level.

Karpenter for Flexible Cluster Autoscaling

AWS **introduced Karpenter**, an open-source, flexible, high-performance Kubernetes cluster autoscaler, in November of 2021. Karpenter is designed to work with any Kubernetes cluster running in any environment, including all major cloud providers and on-premises environments.

Karpenter automatically provisions new nodes in response to unschedulable pods. Karpenter evaluates the aggregate resource requirements of the pending pods and chooses the optimal instance type to run them.

Over time, those instances can become under-utilized as some workloads scale down or are removed from the cluster. Karpenter can also automatically look for opportunities to reschedule these workloads onto a set of more cost-efficient EC2 instances, whether they are already in the cluster or need to be launched.



Karpenter simplifies Kubernetes infrastructure with the right nodes at the right time. (Image credit: AWS)

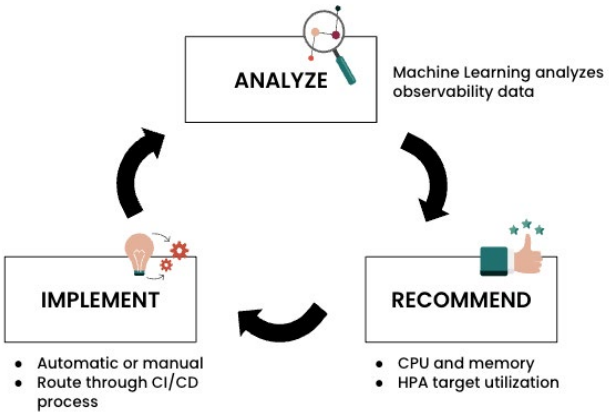
But What About the Pods?

Karpenter drastically simplifies the configuration of cluster autoscaling and can improve application availability and cluster efficiency to reduce cloud costs. However, Karpenter is not designed to address the more granular need for pod right-sizing nor does it provide the ability to add and remove pods in response to bursty workloads. Absent these other forms of autoscaling, Karpenter continues to perform its role of provisioning resources based on existing suboptimal resource requests. This results in wasted resources and excessive cloud costs. Meanwhile, performance risks and availability issues remain unaddressed. That's where StormForge comes in.

StormForge for Automatic, ML-Powered Pod-Level Efficiency

StormForge Optimize Live right-sizes pods automatically, using machine learning to analyze actual usage data from your observability solution and then recommend the most efficient CPU and memory settings. Recommendations can be automatically implemented to keep pods effortlessly right-sized as usage fluctuates.

In addition, StormForge also works with the HPA, recommending the best target utilization to ensure the HPA adds and removes pods as efficiently as possible. Because StormForge is managing both the vertical scaling and the HPA configuration, vertical and horizontal autoscaling can now work together - eliminating thrashing and maximizing pod-level efficiency.



StormForge Optimize Live recommends CPU and memory to continuously right-size pods, and also recommends the HPA target utilization to ensure pod autoscaling efficiency.

Karpenter and StormForge Together

Karpenter and StormForge are both great solutions on their own. But if you use Karpenter without StormForge, you're wasting resources and leaving money on the table by not keeping pods right-sized or scaling them in an efficient way. Conversely, if you use StormForge without Karpenter, your pods may be operating efficiently but you're still missing the efficient "bin packing", i.e. the most efficient placement of those pods on the right nodes, which is needed to realize material cloud cost savings.

By deploying Karpenter and StormForge together, it becomes easy to maximize efficiency – automatically and intelligently. The results?

- Reduce Kubernetes-related cloud costs and resource usage by 30% or more.
- Ensure application performance and reliability for higher productivity and a positive customer experience
- Free up operational bandwidth from Kubernetes management to focus on innovation and other strategic activities

 | +1 (857) 233-9831 | info@stormforge.io | stormforge.io