Introduction

This lesson points to the issue of no nodes with available resources and introduces Cluster Autoscaler to solve that issue.

WE'LL COVER THE FOLLOWING

No nodes with available resources

Usage of `HorizontalPodAutoscaler (HPA)` is one of the most critical aspects of making a resilient, fault-tolerant, and highly-available system. However, it is of no use if there are no nodes with available resources. When Kubernetes cannot schedule new Pods because there's not enough available memory or CPU, new Pods will be unschedulable and in the *pending* status. If we do not increase the capacity of our cluster, *pending* Pods might stay in that state indefinitely. To make things more complicated, Kubernetes might start removing other Pods to make room for those that are in the pending state. That, as you might have guessed, might lead to worse problems than the issue of our applications not having enough replicas to serve the demand.

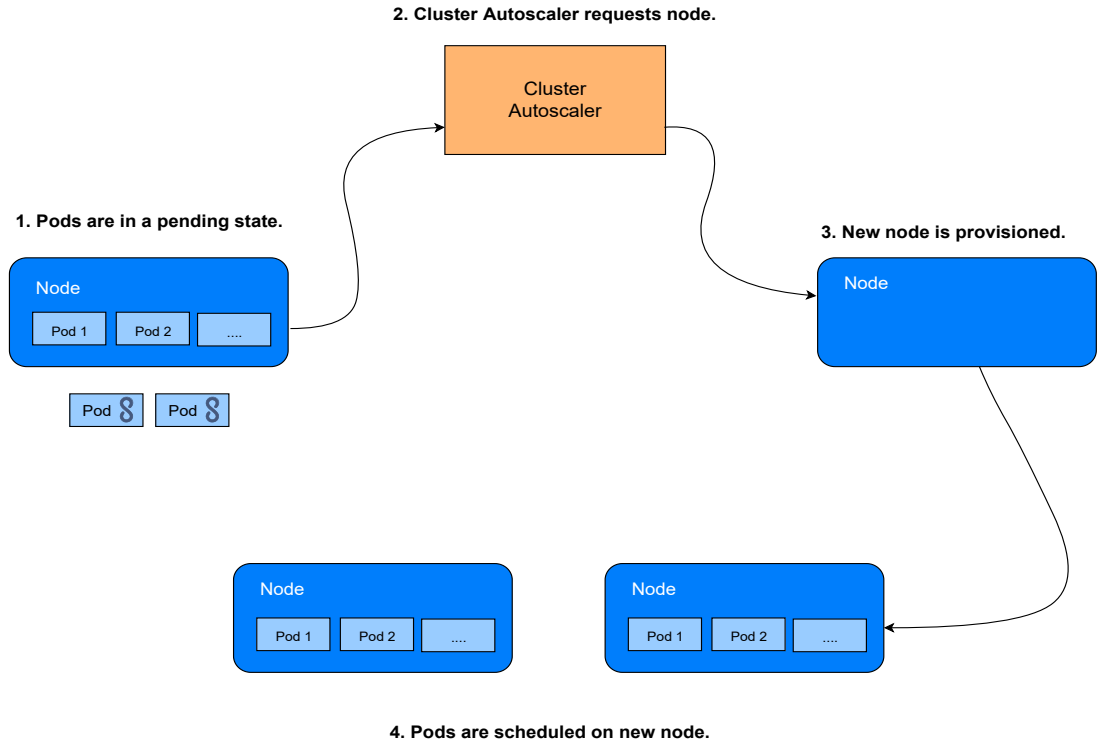
Kubernetes solves the problem of scaling nodes through Cluster Autoscaler.

Purpose of `Cluster Autoscaler`

`Cluster Autoscaler` has a single purpose: to adjust the size of the cluster by adding or removing worker nodes. It adds new nodes when Pods cannot be scheduled due to insufficient resources. Similarly, it eliminates nodes when they are underutilized for a period of time and when Pods running on one such node can be rescheduled somewhere else.

The logic behind `Cluster Autoscaler` is simple to grasp. We are yet to see whether it is simple to use as well.

whether it is simple to use as well.



How Cluster Autoscaler works?

In the next lesson, we will create a cluster (unless you already have one) and prepare it for autoscaling.