Introduction

This lesson provides an Introduction to this course and lists all the four phases of Kubernetes.

WE'LL COVER THE FOLLOWING

Introduction to Kubernetes

Kubernetes is probably the biggest project we know. It is vast, and yet many think that after a few weeks or months of reading and practice they know all there is to know about it. It's much bigger than that, and it is growing faster than most of us can follow. How far did you get in Kubernetes adoption?

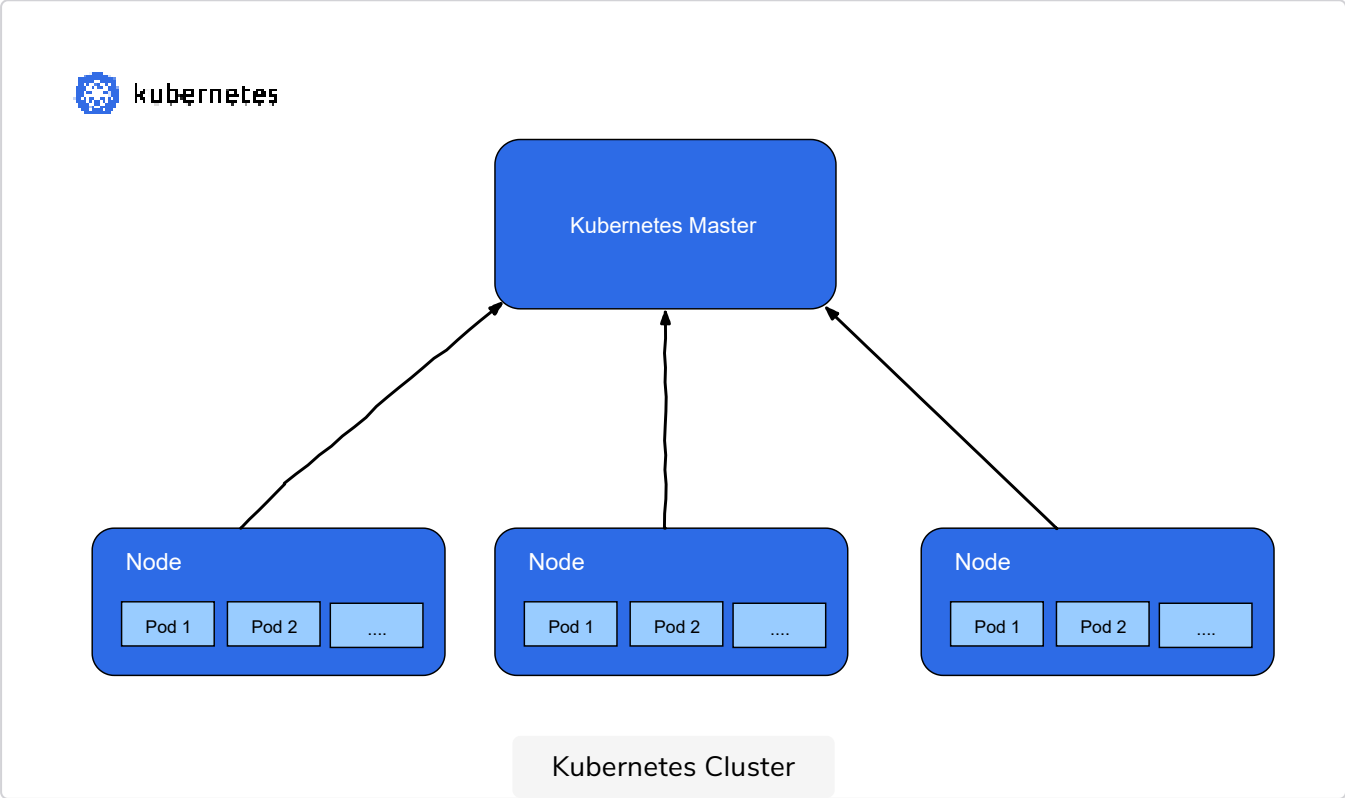
From my experience, there are **four** main phases in Kubernetes adoption.

First Phase

In the first phase, we create a cluster and learn intricacies of Kube API and different types of resources (e.g., Pods, Ingress, Deployments, StatefulSets, and so on). Once we are comfortable with the way Kubernetes works, we start

deploying and managing our applications. By the end of this phase, we can

shout “**look at me, I have things running in my production Kubernetes cluster, and nothing blew up!**”



Second Phase

The second phase is often automation. Once we become comfortable with how Kubernetes works and we are running production loads, we can move to automation. We often adopt some form of continuous delivery (CD) or continuous deployment (CDP). We create Pods with the tools we need, we build our software and container images, we run tests, and we deploy to production. When we’re finished, most of our processes are automated, and we do not perform manual deployments to Kubernetes anymore. We can say that **things are working and I’m not even touching my keyboard**.

Third Phase

The third phase is in many cases related to monitoring, alerting, logging, and scaling. The fact that we can run (almost) anything in Kubernetes and that it will do its best to make it fault-tolerant and highly available, does not mean that our applications and clusters are bulletproof. We need to monitor the cluster, and we need alerts that will notify us of potential issues. When we do discover that there is a problem, we need to be able to query metrics and logs

of the whole system. We can fix an issue only once we know what the root cause is. In highly dynamic distributed systems like Kubernetes, that is not as easy as it looks.

Further on, we need to learn how to scale (and de-scale) everything. The number of Pods of an application should change over time to accommodate fluctuations in traffic and demand. Nodes should scale as well to fulfill the needs of our applications.

Kubernetes already has the tools that provide metrics and visibility into logs. It allows us to create auto-scaling rules. Yet, we might discover that Kubernetes alone is not enough and that we might need to extend our system with additional processes and tools. This phase is the subject of this course. By the time you finish reading it, you'll be able to say that **your clusters and applications are truly dynamic and resilient and that they require minimal manual involvement. We'll try to make our system self-adaptive.**

Fourth Phase

I mentioned the fourth phase. That, dear reader, is everything else. The last phase is mostly about keeping up with all the other goodies Kubernetes provides. It's about following its roadmap and adapting our processes to get the benefits of each new release.

In the next lesson, we will see the intended audience and course requirements of the course.