Getting Started with Pods

In this lesson, we will understand Pods and create a Kubernetes cluster.

WE'LL COVER THE FOLLOWING

Understanding Pods

Pods are equivalent to bricks we use to build houses. Both are uneventful and not much by themselves. Yet, they are fundamental building blocks without which we could not construct the solution we are set to build.

If you have used *Docker or Docker Swarm*, you're probably used to thinking that a *container is the smallest unit* and that more complex patterns are built on top of it. With Kubernetes, the *smallest unit is a Pod*.

> A **Pod** is a way to represent a running process in a cluster.

From the Kubernetes' perspective, there's nothing smaller than a Pod.

A Pod encapsulates one or more containers. It provides a unique network IP, attaches storage resources, and also decides how containers should run. Everything in a Pod is tightly coupled.

We should clarify that containers in a Pod are not necessarily made by Docker. Other container runtimes are supported as well. Still, at the time of this writing, Docker is the most commonly used container runtime, and all our examples will use it.

Since we cannot create Pods without a Kubernetes cluster, our first order of

business is to create one.

> ℹ All the commands from this chapter are available in the 03-pods.sh Gist.

Creating A Cluster

We'll create a local Kubernetes cluster using Minikube.

```
minikube start --vm-driver=virtualbox
kubectl get nodes
```

The **output** of the latter command is as follows.

```
NAME     STATUS ROLES  AGE VERSION
minikube Ready  master 47s v1.14.0
```

To simplify the process and save you from writing all the configuration files, we'll clone the GitHub repository. It contains everything we'll need for this chapter, as well as for most of the others in this course.

```
git clone https://github.com/vfarcic/k8s-specs.git
cd k8s-specs
```

We cloned the repository and entered into the directory that was created.

Now, in the next lesson, we will run our first Pod.