

# Getting Started with Managing Resources

In this lesson, we will find out why resource management is important.

## WE'LL COVER THE FOLLOWING



- Why Resource Management is Important?
- Creating A Cluster
  - Enabling Metrics Server

## Why Resource Management is Important? #

Without an indication how much CPU and memory a container needs, Kubernetes has no other option than to treat all containers equally. That often produces a very uneven distribution of resource usage. Asking Kubernetes to schedule containers without resource specifications is like entering a taxi driven by a blind person.

We have come a long way, from humble beginnings, towards understanding many of the essential Kubernetes object types and principles. One of the most important things we're missing is resource management. Kubernetes was blindly scheduling the applications we deployed so far. We never gave it any indication how much resources we expect those applications to use, nor established any limits.

Without these indications, Kubernetes was carrying out its tasks in a very myopic fashion. Kubernetes could see a lot, but not enough. We'll change that soon. We'll give Kubernetes a pair of glasses that will provide it a much better vision.

Once we learn how to define resources, we'll go further and make sure that certain limitations are set, that some defaults are determined, and that there are quotas that will prevent applications from overloading the cluster.

After this chapter, we'll have enough knowledge to think about running Kubernetes in a production environment.

❗ All the commands from this chapter are available in the [13-resource.sh](#) Gist.

## Creating A Cluster #

We'll go through almost the same routine as we did in the previous chapters. We'll enter the directory where we cloned the *vfarcic/k8s-specs* repository, pull the latest code, start a Minikube cluster, and so on and so forth.

## Enabling Metrics Server #

The only new thing we'll do this time is to enable one more addon. We'll add Metrics Server to the cluster. It's too soon to explain what it does and why we'll need it. That will come later. For now, just remember that there will soon be something in your cluster called Metrics Server. If you do not already know what it is, consider this a teaser meant to build suspense.

```
cd k8s-specs
git pull
minikube start --vm-driver=virtualbox
kubectl config current-context
minikube addons enable ingress
minikube addons enable metrics-server
```



Now that the latest code is pulled, the cluster is running, and the add-ons are enabled.

---

In the next lesson, we'll proceed and explore how to define the allocation of memory and CPU resources to a container.