

Running Multiple Instances of Your App

Objectives

- Scale an app using kubectl.

Scaling an application

In the previous modules we created a [Deployment](#), and then exposed it publicly via a [Service](#). The Deployment created only one Pod for running our application. When traffic increases, we will need to scale the application to keep up with user demand.

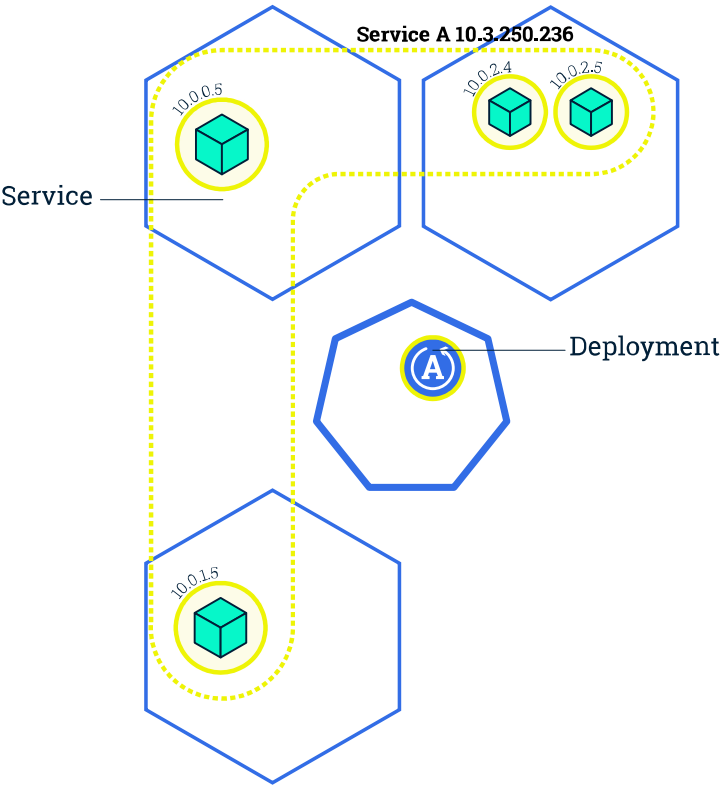
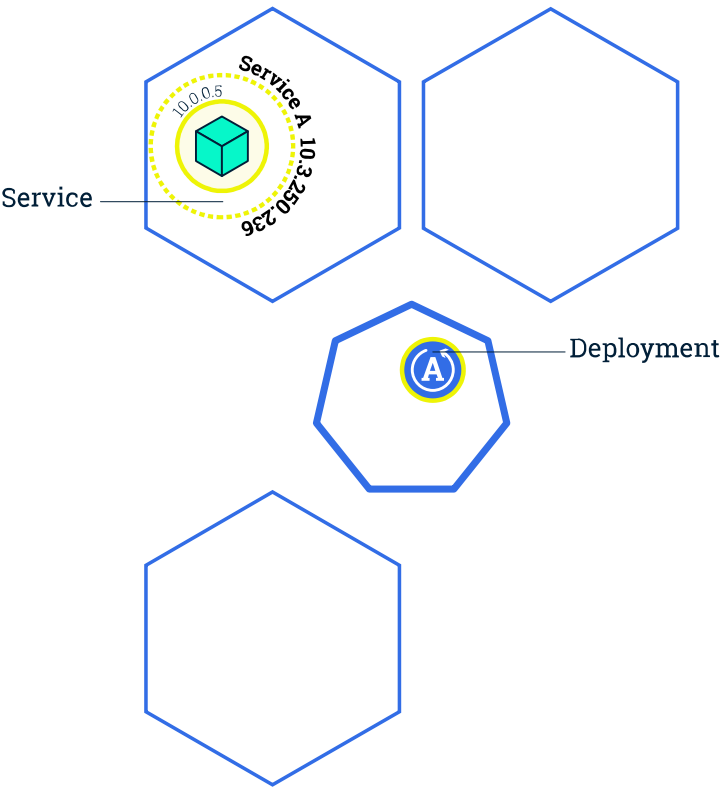
Scaling is accomplished by changing the number of replicas in a Deployment

Summary:

- Scaling a Deployment

You can create from the start a Deployment with multiple instances using the --replicas parameter for the kubectl create deployment command

Scaling overview



Scaling out a Deployment will ensure new Pods are created and scheduled to Nodes with available resources. Scaling will increase the number of Pods to the new desired state. Kubernetes also supports [autoscaling](#) of Pods, but it is outside of the scope of this tutorial. Scaling to zero is also possible, and it will terminate all Pods of the specified Deployment.

Running multiple instances of an application will require a way to distribute the traffic to all of them. Services have an integrated load-balancer that will distribute network traffic to all Pods of an exposed Deployment. Services will monitor continuously the running Pods using endpoints, to ensure the traffic is sent only to available Pods.

Scaling is accomplished by changing the number of replicas in a Deployment.

Once you have multiple instances of an Application running, you would be able to do Rolling updates without downtime. We'll cover that in the next module. Now, let's go to the online terminal and scale our application.