What's Next?

This lesson summarises what we have learned so far in this section and what we are going to learn next.

Summary

Everything we learned led us to Deployments. Pods must not be created directly, but through ReplicaSets which, similarly, must not be created directly, but through Deployments. They are the objects that allow us not only to create the ReplicaSets and Pods, but that can also be updated without producing any downtime (when applications are designed accordingly).

We combined Deployments with Services so that Pods can communicate with each other, or can be accessed from outside a cluster. All in all, we have everything we need to release our services to production. That is not to say that we understand all the crucial aspects of Kubernetes. We're not even close to that point. But, we do have almost everything we need for running some types of applications in production. What we're missing is networking.
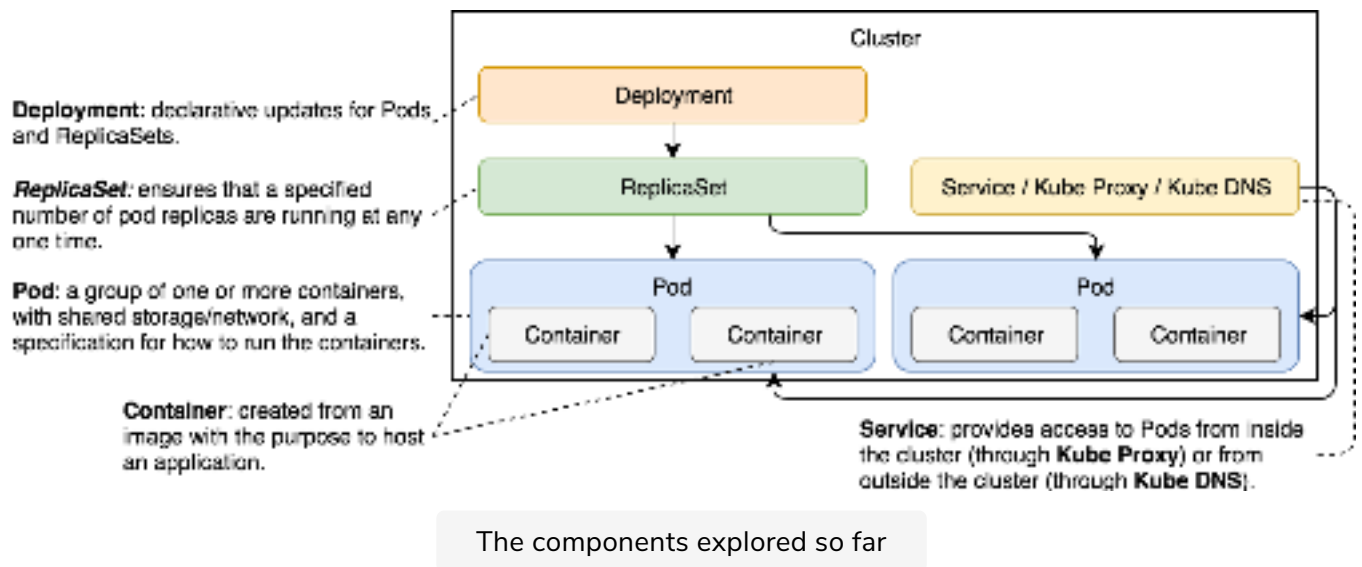
Destroying Everything

Before we enter the next stage of our knowledge seeking mission, we'll destroy the cluster we're running and give our machines a break.

```
minikube delete
```

> ℹ If you'd like to know more about Deployments, please explore
> [Deployment v1 apps](#) API documentation.

Deployment: declarative updates for Pods and ReplicaSets.

ReplicaSet: ensures that a specified number of pod replicas are running at any one time.

Pod: a group of one or more containers, with shared storage/network, and a specification for how to run the containers.

Container: created from an image with the purpose to host an application.

Service: provides access to Pods from inside the cluster (through **Kube Proxy**) or from outside the cluster (through **Kube DNS**).

The components explored so far

The above illustration shows and defines the Kubernetes components we have explored so far.

In the next chapter, we will learn using ingress to forward traffic.

Before we move onto the next chapter, in the next lesson, we'll explore the differences between Kubernetes Deployments and Docker Swarm stacks.