What's Next?

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

WE'LL COVER THE FOLLOWING

Summary

Authorization and authentication are critical security components. Without a proper set of permissions, we are risking exposure with potentially devastating results. Moreover, with appropriate Rules, Roles, and RoleBindings, we can make a cluster not only more secure but also increase collaboration between different members of our organization. The only trick is to find a right balance between tight security and freedom. It takes time until that equilibrium is established.

RBAC combined with Namespaces provides an excellent separation. Without Namespaces, we'd need to create multiple clusters. Without RBAC, those clusters would be exposed or locked down to only a handful of users. The two combined provide an excellent way to increase collaboration without sacrificing security.

We did not explore Service Accounts. They are the third kind of Subjects, besides Users and Groups. We'll leave that for some other time and place since they are used primarily for Pods that need to access the Kubernetes API. This chapter focused on humans and the ways we can enable them to reach a cluster in a safe and controlled manner.

We are still missing one important restriction. By combining Namespaces and RBAC, we can restrict what users can do. However, that will not prevent them from deploying applications that could potentially bring down the whole cluster. We need to add Resource Quotas to the mix. That will be the subject of the next chapter.

Destroying Everything

For now, we'll destroy the cluster and take a rest. We covered a lot of ground in this chapter. We deserve a break.

```
minikube delete
```

> **i** If you'd like to know more about Roles, please explore the Role v1 rbac and ClusterRole v1 rbac API documentation. Similarly, you might want to visit the RoleBinding v1 rbac and ClusterRoleBinding v1 rbac API documentation as well.

In the next chapter, we will explore how to manage resources such as memory, CPU, etc. when running Kubernetes Clusters.

In the next lesson, we will compare Kubernetes Role Based Acess Control(RBAC) with the Docker Swarm equivalent.