# Learning Objectives

By the end of this chapter, you should be able to:

- Discuss about high availability in Kubernetes.
- Discuss about collocated and non-collocated databases.
- Learn steps to achieve high availability in Kubernetes.

# HIGH AVAILABILITY

## Cluster High Availability

A newer feature of **kubeadm** is the integrated ability to join multiple master nodes with collocated etcd databases. This allows for higher redundancy and fault tolerance. As long as the database services the cluster will continue to run and catch up with kubelet information should the master node go down and be brought back online.

Three instances are required for etcd to be able to determine quorum if the data is accurate, or if the data is corrupt, the database could become unavailable. Once etcd is able to determine quorum, it will elect a leader and return to functioning as it had before failure.

One can either collocate the database with control planes or use an external etcd database cluster. The **kubeadm** command makes the collocated deployment easier to use.

To ensure that workers and other control planes continue to have access, it is a good idea to use a load balancer. The default configuration leverages SSL, so you may need to configure the load balancer as a TCP pass through unless you want the extra work of certificate configuration. As the certificates will be decoded only for particular node names, it is a good idea to use a FQDN instead of an IP address, although there are many possible ways to handle access.

## Collocated Databases

The easiest way to gain higher availability is to use the **kubeadm** command and join at least two more master servers to the cluster. The command is almost the same as a worker join except an additional **--control-plane** flag and a **certificate-key**. The key will probably need to be generated unless the other master nodes are added within two hours of the cluster initialization.

Should a node fail, you would lose both a control plane and a database. As the database is the one object that cannot be rebuilt, this may not be an important issue.

## Non-Collocated Databases

Using an external cluster of etcd allows for less interruption should a node fail. Creating a cluster in this manner requires a lot more equipment to properly spread out services and takes more work to configure.

The external etcd cluster needs to be configured first. The **kubeadm** command has options to configure this cluster, or other options are available. Once the etcd cluster is running, the certificates need to be manually copied to the intended first control plane node.

The **kubeadm-config.yaml** file needs to be populated with the etcd set to external, endpoints, and the certificate locations. Once the first control plane is fully initialized, the redundant control planes need to be added one at a time, each fully initialized before the next is added.