COMPONENTS OF THE CONTROL PLANE

The Control Plane is what controls and makes the whole cluster function. To refresh

your memory, the components that make up the Control Plane are

 The etcd distributed persistent storage

 The API server

 The Scheduler

 The Controller Manager

These components store and manage the state of the cluster, but they aren’t what runs

the application containers.

COMPONENTS RUNNING ON THE WORKER NODES

The task of running your containers is up to the components running on each

worker node:

 The Kubelet

 The Kubernetes Service Proxy (kube-proxy)

 The Container Runtime (Docker, rkt, or others)

ADD-ON COMPONENTS

Beside the Control Plane components and the components running on the nodes, a

few add-on components are required for the cluster to provide everything discussed

so far. This includes

 The Kubernetes DNS server

 The Dashboard

 An Ingress controller

 Heapster, which we’ll talk about in chapter 14

 The Container Network Interface network plugin

HOW THESE COMPONENTS COMMUNICATE

Kubernetes system components communicate only with the API server. They don’t

talk to each other directly. The API server is the only component that communicates

with etcd. None of the other components communicate with etcd directly, but instead

modify the cluster state by talking to the API server.

Although the components on the worker nodes all need to run on the same node,

the components of the Control Plane can easily be split across multiple servers.

While multiple instances of etcd and API server can be active at the

same time and do perform their jobs in parallel, only a single instance of the Scheduler

and the Controller Manager may be active at a given time

HOW COMPONENTS ARE RUN :

The Control Plane components, as well as kube-proxy, can either be deployed on the

system directly or they can run as pods (as shown in listing 11.1). You may be surprised

to hear this, but it will all make sense later when we talk about the Kubelet.

The Kubelet is the only component that always runs as a regular system component,

and it’s the Kubelet that then runs all the other components as pods. To run the

Control Plane components as pods, the Kubelet is also deployed on the master.

***How Kubernetes uses etcd***

All the objects you’ve created throughout this book—Pods, ReplicationControllers,

Services, Secrets, and so on—need to be stored somewhere in a persistent manner so

their manifests survive API server restarts and failures. For this, Kubernetes uses etcd,ods

which is a fast, distributed, and consistent key-value store. Because it’s distributed,

you can run more than one etcd instance to provide both high availability and better

performance.

It’s worth emphasizing that etcd is the *only* place Kubernetes stores cluster state and metadata.

Kubernetes stores all its data in etcd under /registry.

Usually, for large clusters, an etcd cluster of five or seven nodes is sufficient. It can handle a two- or a three-node failure, respectively, which suffices in almost all situations.

***What the API server does***

The Kubernetes API server is the central component used by all other components and by clients, such as kubectl.

It provides a CRUD (Create, Read, Update, Delete) interface for querying and modifying the cluster state over a RESTful API. It stores that state in etcd.