A CLOUD GURU

K8s Management Overview

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

K8s Management Overview

- Introduction to High Availability in K8s
- Introducing K8s Management Tools
- Safely Draining a K8s Node
- Upgrading K8s with kubeadm
- Backing Up and Restoring etcd Cluster Data

A CLOUD GURU

Introduction to High Availability in K8s

Will Boyd
DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

Introduction to High Availability in K8s

High Availability in K8s

High Availability Control Plane

Stacked etcd

External etcd

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

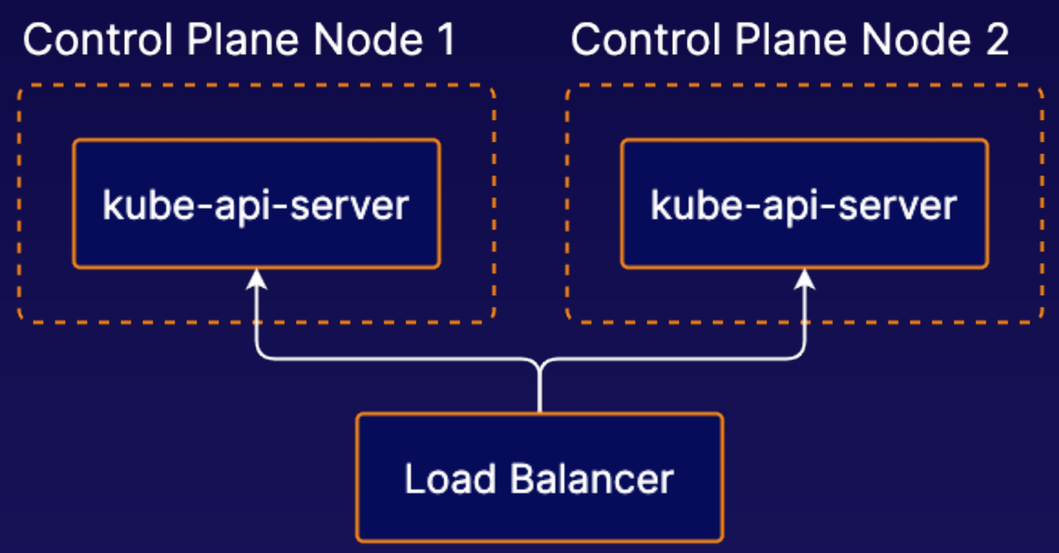
High Availability in K8s

K8s facilitates high-availability applications, but you can also design the cluster itself to be highly available.

To do this, you need multiple control plane nodes.

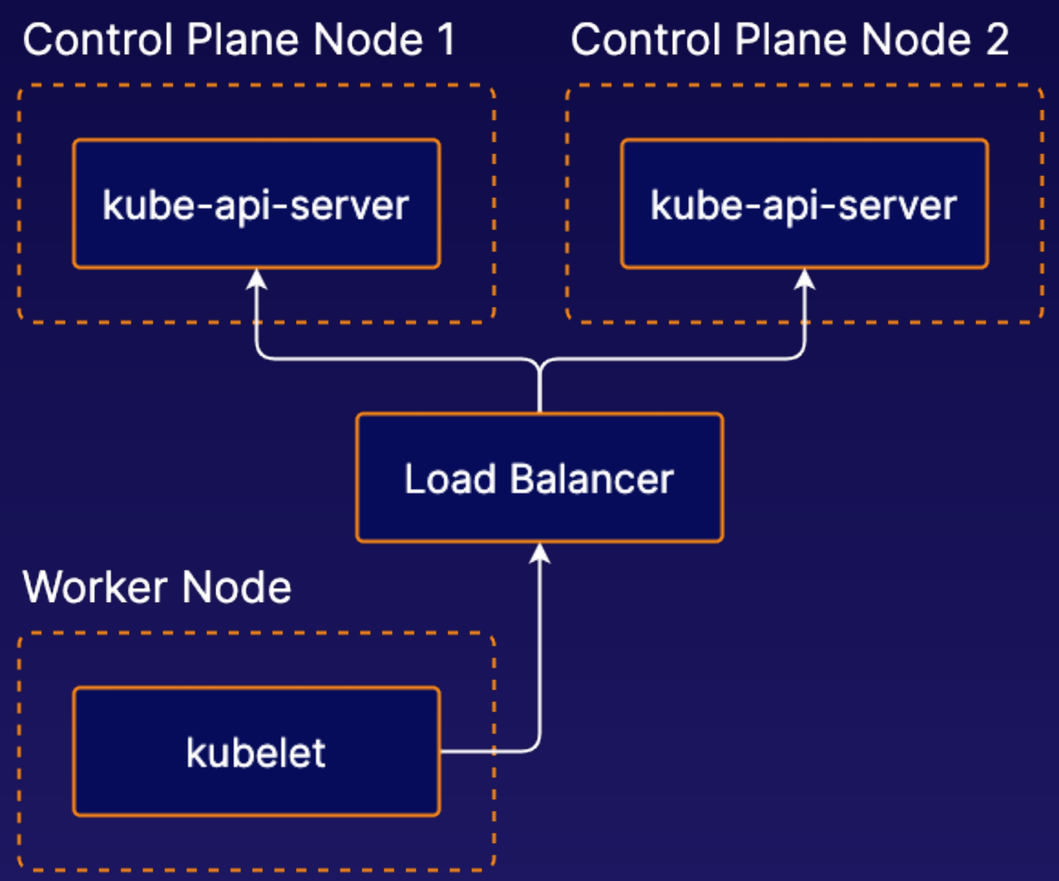
High Availability Control Plane

High Availability Control Plane



When using multiple control planes for high availability, you will likely need to communicate with the Kubernetes API through a **load balancer**.

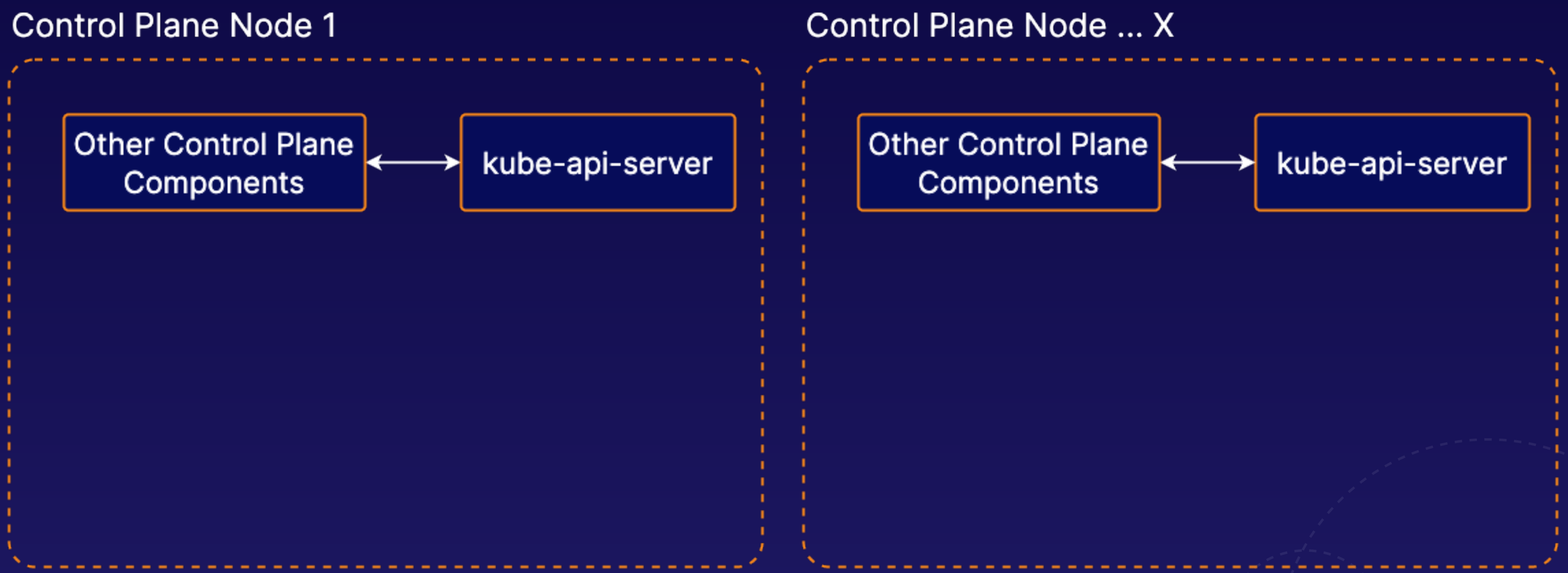
High Availability Control Plane



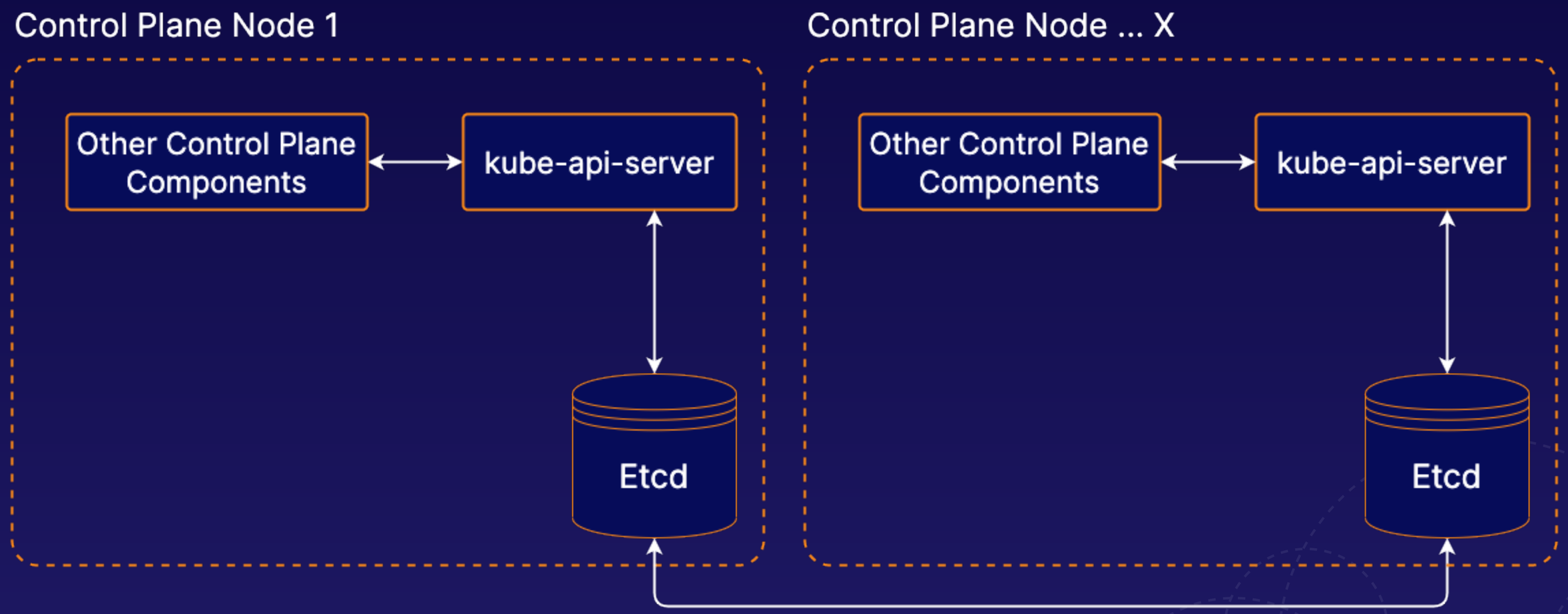
When using multiple control planes for high availability, you will likely need to communicate with the Kubernetes API through a **load balancer**.

This includes clients such as **kubelet** instances running on worker nodes.

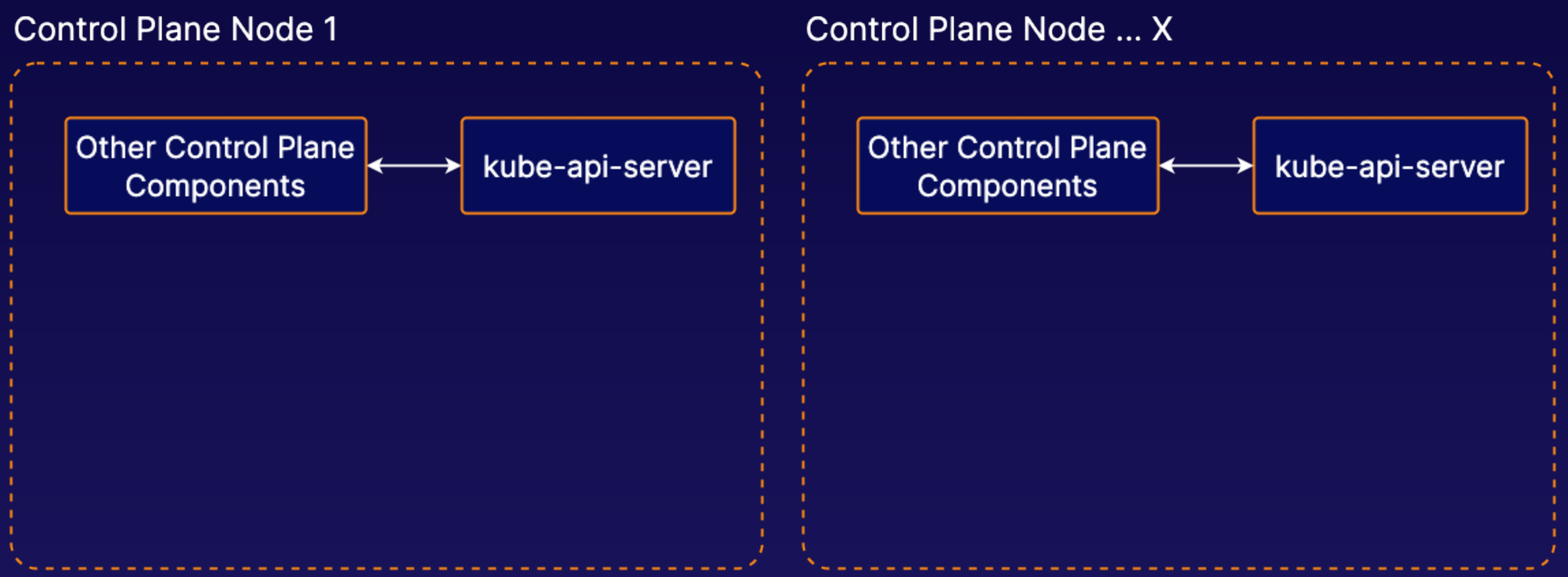
Stacked etcd



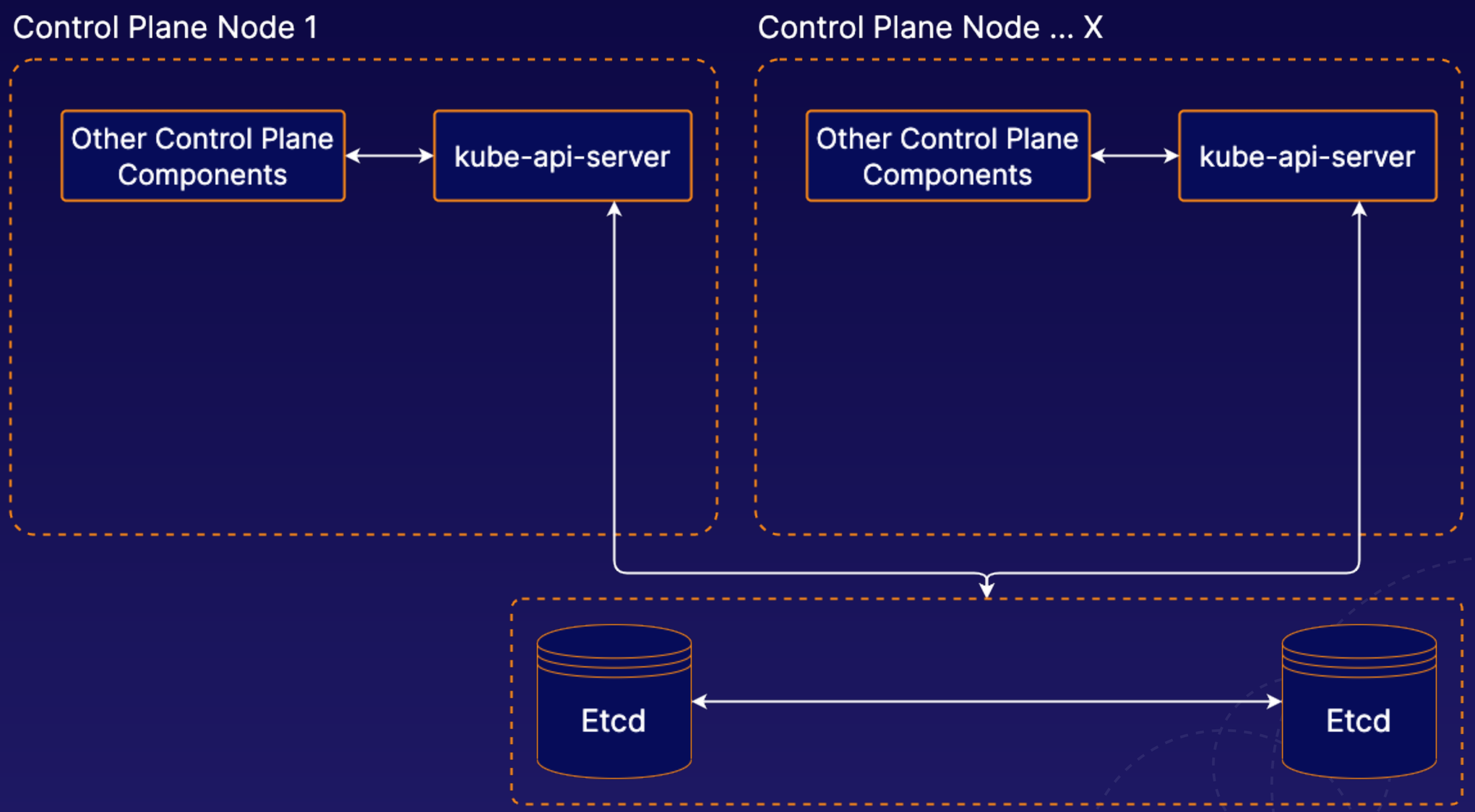
Stacked etcd



External Etcd



External Etcd



LESSON BREAKDOWN

Introduction to High Availability in K8s

High Availability in K8s

High Availability Control Plane

Stacked etcd

External etcd

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

Introduction to K8s Management Tools

Will Boyd
DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

Introduction to K8s Management Tools

K8s Management Tools

kubectl

kubeadm

Minikube

Helm

Kompose

Kustomize

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

K8s Management Tools

There is a variety of **management tools** available for Kubernetes. These tools interface with Kubernetes to provide additional functionality. When using Kubernetes, it is a good idea to be aware of some of these tools.

A CLOUD GURU

kubectl

kubectl is the official command line interface for Kubernetes. It is the main method you will use to work with Kubernetes in this course.

A CLOUD GURU

kubeadm

We have already used **kubeadm**, which is a tool for quickly and easily creating Kubernetes clusters.

A CLOUD GURU

Minikube

Minikube allows you to automatically set up a local single-node Kubernetes cluster. It is great for getting Kubernetes up and running quickly for development purposes.

Helm

Helm provides templating and package management for Kubernetes objects. You can use it to manage your own templates (known as charts). You can also download and use shared templates.

A CLOUD GURU

Kompose

Kompose helps you translate Docker compose files into Kubernetes objects. If you are using Docker compose for some part of your workflow, you can move your application to Kubernetes easily with Kompose.

A CLOUD GURU

Kustomize

Kustomize is a configuration management tool for managing Kubernetes object configurations. It allows you to share and re-use templated configurations for Kubernetes applications.

LESSON BREAKDOWN

Introduction to K8s Management Tools

K8s Management Tools

kubectl

kubeadm

Minikube

Helm

Kompose

Kustomize

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

Safely Draining a K8s Node

Will Boyd
DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

Safely Draining a K8s Node

- What Is Draining?
- Draining a Node
- Ignoring DaemonSets
- Uncordoning a Node
- Hands-On Demonstration

Will Boyd

DEVOPS TRAINING ARCHITECT

A CLOUD GURU

What Is Draining?

When performing maintenance, you may sometimes need to remove a Kubernetes node from service.

To do this, you can **drain** the node. Containers running on the node will be gracefully terminated (and potentially rescheduled on another node).

Draining a Node

To drain a node, use the **`kubectl drain`** command.

```
$ kubectl drain <node name>
```

A CLOUD GURU

Ignoring DaemonSets

When draining a node, you may need to ignore DaemonSets (pods that are tied to each node). If you have any DaemonSet pods running on the node, you will likely need to use the **`--ignore-daemonsets`** flag.

```
$ kubectl drain <node name> --ignore-daemonsets
```

A CLOUD GURU

Uncordoning a Node

If the node remains part of the cluster, you can allow pods to run on the node again when maintenance is complete using the **`kubectl uncordon`** command.

```
$ kubectl uncordon <node name>
```

A CLOUD GURU

Hands-On Demonstration

LESSON BREAKDOWN

Safely Draining a K8s Node

- What Is Draining?
- Draining a Node
- Ignoring DaemonSets
- Uncordoning a Node
- Hands-On Demonstration

Will Boyd

DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

Upgrading K8s With Kubeadm

Will Boyd
DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

Upgrading K8s with kubeadm

Upgrading with kubeadm

Control Plane Upgrade Steps

Worker Node Upgrade Steps

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

Upgrading with kubeadm

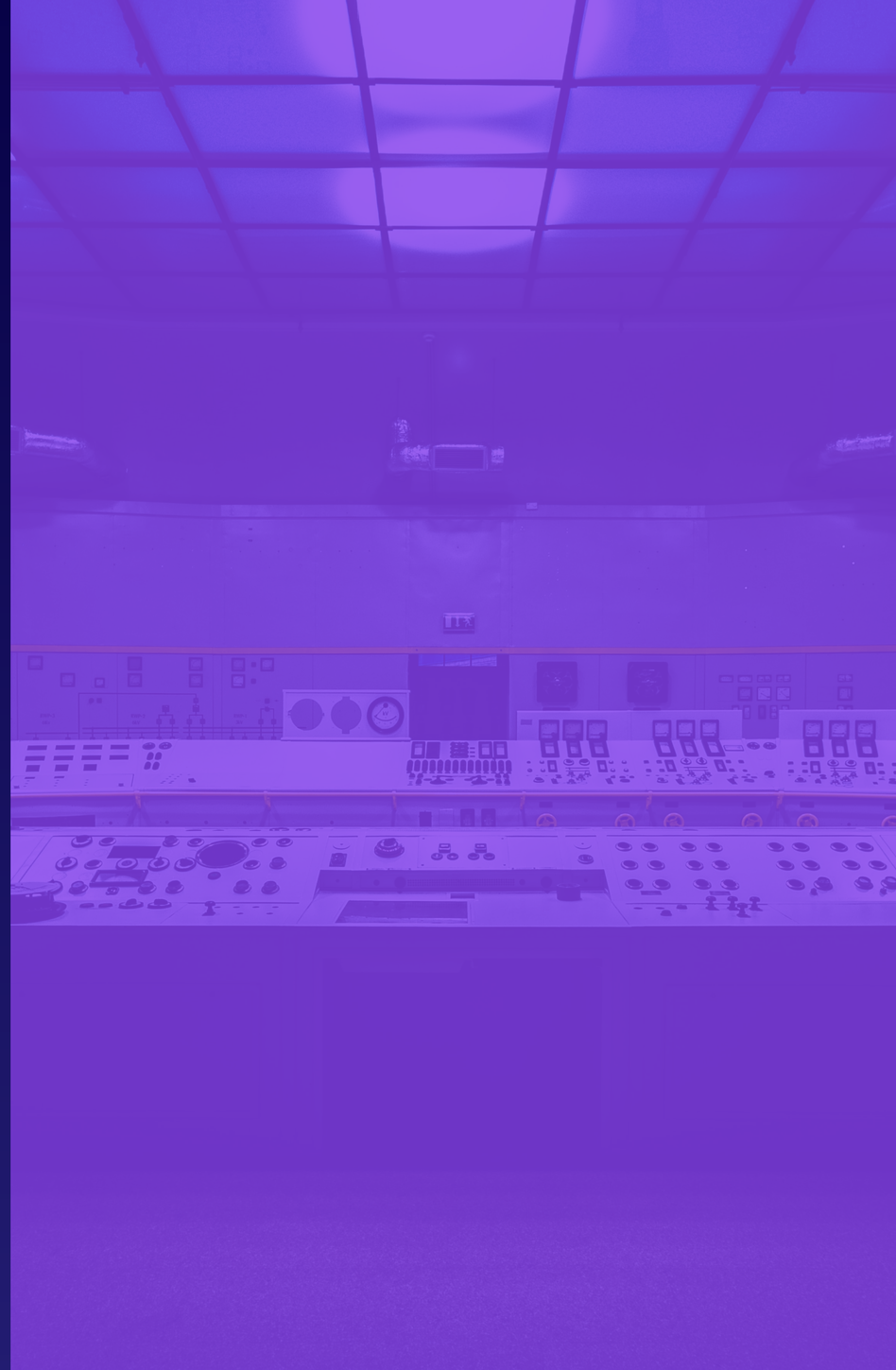
When using Kubernetes, you will likely want to periodically upgrade Kubernetes to keep your cluster up to date.

Luckily, kubeadm makes this process easier.

A CLOUD GURU

Control Plane Upgrade Steps

- Upgrade kubeadm on the control plane node.
- Drain the control plane node.
- Plan the upgrade (`kubeadm upgrade plan`).
- Apply the upgrade (`kubeadm upgrade apply`).
- Uncordon the control plane node.
- Upgrade kubelet and kubectl on the control plane node.



A CLOUD GURU

Worker Node Upgrade Steps

- Upgrade kubeadm.
- Drain the node.
- Upgrade the kubelet configuration (`kubeadm upgrade node`).
- Upgrade kubelet and kubectl.
- Uncordon the node.

A CLOUD GURU

Hands-On Demonstration

LESSON BREAKDOWN

Upgrading K8s with kubeadm

Upgrading with kubeadm

Control Plane Upgrade Steps

Worker Node Upgrade Steps

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

Backing Up and Restoring Etcd Cluster Data

Will Boyd
DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

Backing Up and Restoring etcd Cluster Data

Why Back Up etcd?

Backing Up etcd

Restoring etcd

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

Why Back Up etcd?

etcd is the backend data storage solution for your Kubernetes cluster. As such, all your Kubernetes objects, applications, and configurations are stored in etcd.

Therefore, you will likely want to be able to back up your cluster's data by backing up etcd.

A CLOUD GURU

Backing Up etcd

You can back up etcd data using the etcd command line tool, **etcdctl**.

Use the **`etcdctl snapshot save`** command to back up the data.

```
$ ETCDCTL_API=3 etcdctl --endpoints $ENDPOINT
snapshot save <file name>
```

A CLOUD GURU

Restoring etcd

You can restore etcd data from a backup using the **`etcdctl snapshot restore`** command.

You will need to supply some additional parameters, as the restore operation creates a new logical cluster.

```
$ ETCDCTL_API=3 etcdctl snapshot restore <file
name>
```

LESSON BREAKDOWN

Backing Up and Restoring etcd Cluster Data

Why Back Up etcd?

Backing Up etcd

Restoring etcd

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

Cluster Management Summary

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

Cluster Management Summary

- Introduction to High Availability in K8s
- Introducing K8s Management Tools
- Safely Draining a K8s Node
- Upgrading K8s with kubeadm
- Backing Up and Restoring etcd Cluster Data