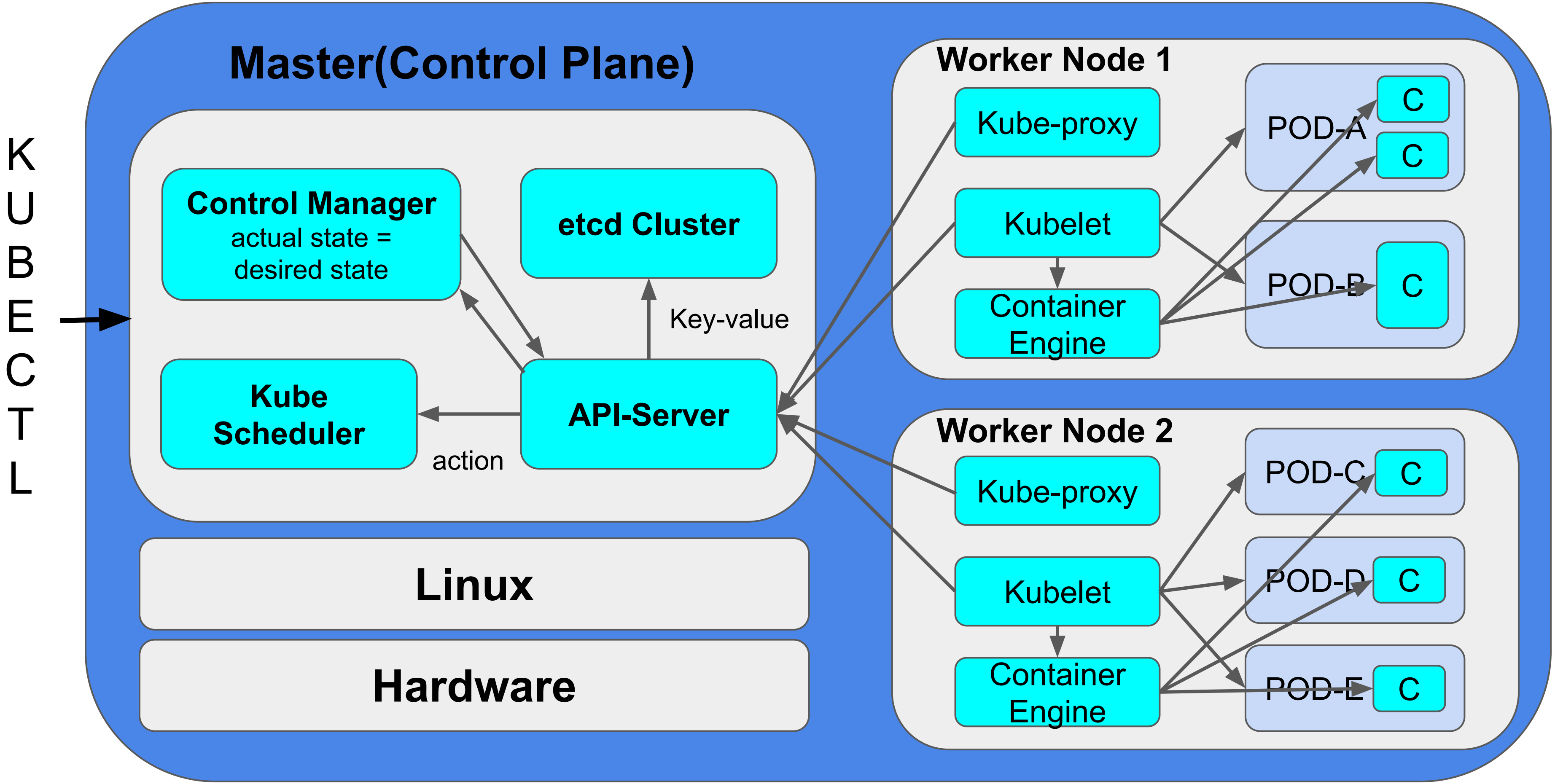
Assalam-u-alaikum

BILAL KHAN

This is my 22nd video of DevOps

Kubernetes Architecture

Cluster



KUBECTL

Master(Control Plane)

Control Manager
actual state =
desired state

etcd Cluster

Key-value

Kube Scheduler

action

API-Server

Linux

Hardware

Worker Node 1

Kube-proxy

Kubelet

Container Engine

POD-A

C

C

POD-B

C

Worker Node 2

Kube-proxy

Kubelet

Container Engine

POD-C

C

POD-D

C

POD-E

C

Roles of master node

- It manages the health of the cluster, like creating, deleting pods etc.
- Kubernetes cluster contains containers running in bare metal, vm instances, cloud instances, or mix.
- Kubernetes designates one or more of these as master and all others as workers.
- Cluster is a group of nodes. A cluster contains at least one master node, and one worker node.
- The master is now going to run set of k8s processes. These processes will ensure smooth functioning of cluster. These processes are called control plane.
- You can create multiple masters for high availability.
- Control plane manages the worker nodes.
- Pod runs on node. Node is controlled by master.

Components of Control plane(Master)

API-Server:

API server communicates directly with the user. You can write YAML/JSON or write commands in kubectl to communicate with it. It is used to send and receive the requests inside a master node and to worker nodes also. It only builds the connection and does not provide the solution.

Control manager

It will make a balance by contacting the etcd cluster that what actually happened(desired state) and what should happen(actual state). Let’s say a person wants 4 containers(desired state) and is only given 3 containers(actual state). The containers are mismatched. So the control manager won’t accept it.

etcd Cluster

It is a kind of database that contains all the information(how many containers are present there, what are their IPs, etc) of the cluster.

Control manager knows the status with the help of etcd cluster. From an etcd cluster, the actual and the desired states data will be given to the control manager.

That's how the control manager will know that what are the states of the data? etcd cluster is not directly a part of kubernetes. API server can only access the etcd cluster. etcd cluster stores metadata(data of data) and the status of cluster. The entire state of etcd cluster is available on every node.

Kube Scheduler

Scheduler will perform an actual task that the controller has requested to it. It does not care, what are these tasks? Kube scheduler handles pod creation and management and then control manager will check that whether the actual state is equal to the desired state. The scheduler watches for the newly created pods that have no node assigned to it and is responsible for finding the best node for that pod to run on.

Declarative or imperative way to interact with kubernetes

For interaction, the commands can either be written in kubectl(CLI tool) called **imperative way** or in YAML or JSON as manifest file called **declarative way** & It’ll be executed with *kubectl apply -f <filename>*. The request will go to the API server & from there it will go to the control manager and the control manager will send it to the scheduler.

Components of worker nodes

Kubelet

Kubelet is an agent running on every worker node that talks to the API server. It controls the pod and sends the pod creation request to the master node.

Container Engine(Docker, Containerd, Rocket)

It works with kubelet. It pulls the images, create the containers, run the images on containers. It starts and stops the containers. And ask you which port number to open and communicate with it.

Kube-proxy

Kube-proxy assigns ip address to the pod. Containers don’t have an ip address. Kube proxy runs on each node and this makes sure that each pod will get its own unique ip address.

Two pods are not connected with each other directly. Instead the pod will give a message to the kube-proxy and kube-proxy will deliver that message to another pod.

Pod

Pod is the smallest unit of the kubernetes. Inside pod, you can add containers. The reason for creating the pod is because there are multiple types of containers and they have different settings. Pod is a common space in which any type of container can be added. In kubernetes, the control unit is pod, not containers.

Pod may contain multiple containers but usually it contains one container. Because multiple containers are tightly coupled. So if one container is failed, other will also fail. As a result, a new pod will be created with a new ip address.

The Steps

- Create microservice.
- Containerize it.
- Put containers in pods.
- Deploy these pods to API server.
- Control manager will take the information from the etcd cluster to check the states.
- If the actual state and the desired state is equal then it will be transferred to the kube scheduler and the kube scheduler will assign Pods to the Nodes then it will again contact the etcd cluster to update the data there.

Multi-Containers-POD

Multiple containers inside a pod share their memory. Although the pod is provided an ip address but multiple containers inside the pod are connected using localhost. Either the required number of containers will run or not. If the containers didn't run then a new pod will be created again.

POD has the limitations that you cannot perform auto scaling and auto healing, versioning, rollback, etc by default. You have to install the plugins manually.

Higher level kubernetes objects/plugins

- **Replication set** → Auto scaling and auto healing.
- **Deployment** →Versioning and rollback.
- **Service** → Static IP and networking.
- **Volume** → Storage outside the node.

What we have learned?

Roles of control plan/master

Components of master node

Components of worker node

Higher level kubernetes objects and tools

That’s It

I hope you will like this video.

Make sure to subscribe to my channel.

Ask questions in the comment section.