Page No.
Date

Kubernetes Management Overview

we learn follow:-

1) Kubernetes High availability
2) K8s management Tools.
3) Setup K8s HA Cluster
4) K8s Maintenance Window mangement
5) Upgrading Kubernetes cluster.
6) Backing up & Restoring Etcd cluster.

Kubernetes High availability :-

1) Kubernetes is designed to facilitate the high availability in application.

2) K8s as a tool is providing the Execution Environment for application and as the Environment K8s is providing the high availibility to the application.

But application is available if my cluster is available. what happens if your cluster will itself go down? In that Case, the application availability will also be impacted.

So application availability is another aspect & cluster of ability is another aspect.

here we are taking about the high availability of Kubernetes cluster & this can be achieved by the infra level.

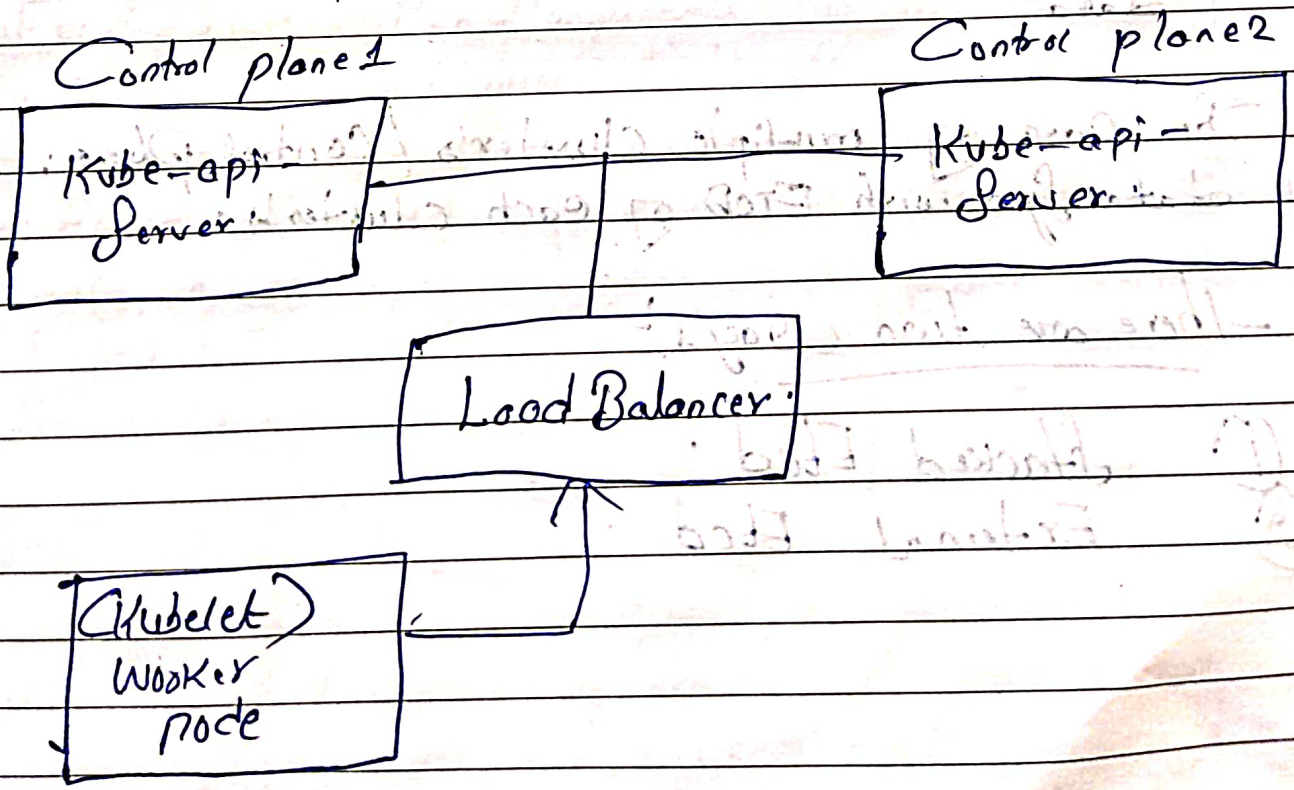
So infra high ability is very necessary for the production Environments.
whenever I'm talking about infra, I mean the infrastructure on which the application is executing
in kubernetes we can achieve the high availability by supporting the multiple Control plane.

It means in production Kubernetes Environment, we must have the multi master & multi worker node.

(3) Cluster also needs to be highly available to support Application HA

(4) To facilitate cluster HA, we need multiple Control planes.

(5) User needs Load Balancer to communicate with multiple Control planes.



Page No.
Date

→ Server.

(a) On different nodes we deploy the Control plane (master node).

(b) To Communicate b/w these nodes & we need something in b/w so, "Load Balancer" will act as Communicater b/w these node.

(c) User send request will go to Load balancer. Load balancer will propagate to the Control Panels.

(d) Worker node also send request or communication via Load balancer.

Let talk about ETCD Management

what is ETCD, which store Key-Value data. By default each cluster has ETCD in it which store data.

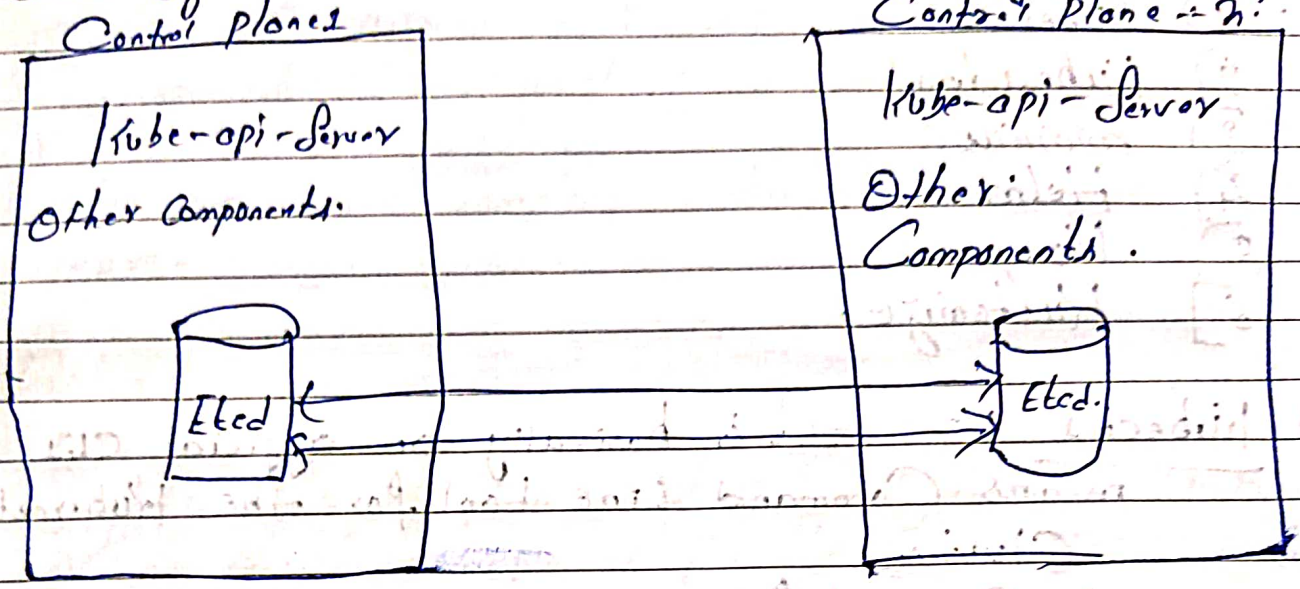
In case of multiple cluster's / control plane; how data sync with ETCD of each cluster's.

There are two way's :-

(1) Stacked Etcd.
(2) External Etcd

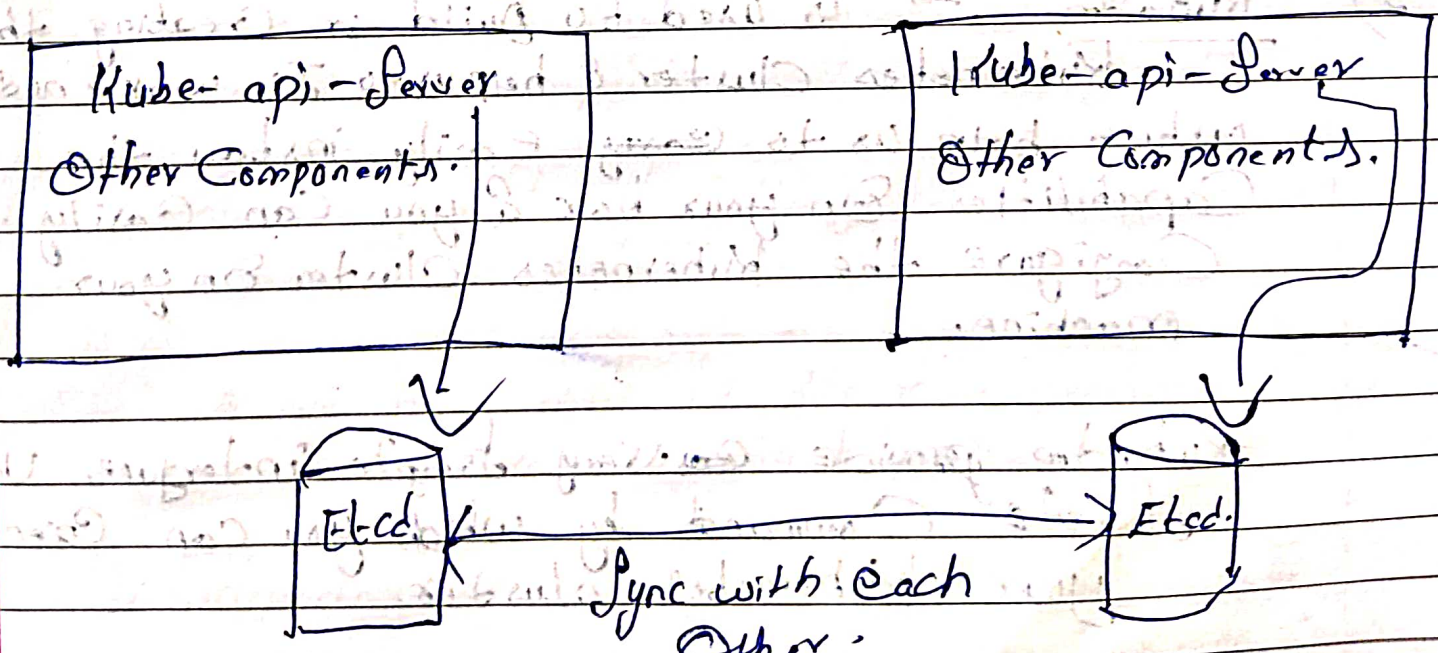
Stacked Etcd.

We use inbuilt Etcd of each cluster/Control planes and Sync data blw the each Etcd of control planes



External Etcd :-

We remove the Etcd from Control planes & add External Etcd, note: Other Components are intact.. Only Etcd is remove & Configured Externaly.



Page No.
Date

K8s Management Tools :-

These tool help us to Manage Kubernets.

1) Kubectl
2) Kubeadm
3) minikube
4) Helm
5) Kompose
6) Kustomize.

1) **Kubectl :-** It is basically an official CLI, which means Command Line tool for the Kubernetes Cluster.

CLI is Command line interface, which you can use to interact or Communicate with the Kubernetes Cluster.

API Rest api is another interface by which you can Communicate with the Kubernetes Cluster.

2) **Kubadm :-** is used to build or Creating the Kubernetes Cluster & help us to Join worker node. which help us to ~~Easy~~ easily instal the Capabilities On your box & you can easily Configure the Kubernetes Cluster On your machine.

Kubadm provide ~~easy~~ very simple interface, very simple Commands by which you can Execute your Kubernetes cluster.

Page No.
Date

(3) Minicube :- minicube is a developer's tool. / practicing. advantage; it will set up the master & worker node in a single machine.

So with the help of mini cube, you can setup the development environment very quickly.

Can set up the Complete Kubernetes cluster on a single node.

(4) Helm :-

(a) Helm is a very powerful tool & help us to create template & package management for the communities.

(b) Can create the multiple object in K8S & create a template of these object so that you can use that template multiple times to spin up a new object on different different K8S.

(c) Helm have the ability to convert the K8S object in to a reusable template which is a very good thing.

(d) Helm also provide the ability to configure the compler ~~for~~ templates

it means K8S cluster have the interdependency, it means where K8S objects have interdependency.

interdependency means whenever the K8S object like the deployment, pods & services they have the own dependency.

Page No.
Date

you can create the complex charts, you can create the complex templates of these objects with the help of the helm.

(5) Kompose :-

(1) helps to translate Docker Compose files into a K8s object.

(2) Ability to ship container from Docker Compose to K8s

(6) Kustomize :- used to manage the configuration of your K8s cluster. That is somewhere similar to the helm. But it has a few things which are missing in the helm.

(7) Kubelet :- Run of an agent node.

☆ HA Kubernetes installation.

Please refer youtube video for installing on Ubuntu 22.04.

(1) We need all these three for HA

(1) Kubectl
(2) Kubeadm
(3) Kubelet

Page No.
Date

Create-regenerate
To get Token again On master-node

```
kubeadm token create --print-join-command.
```

`kubeadm token list` -> will show all token we have generated.

K8S CLUSTER MAINTENANCE WINDOW

i] we need to remove some nodes from my cluster & after the recovery we again need to add the node in the cluster.

Remove Node from Cluster

1) what is Node Draining.
2) How to Drain a Node
3) Ignore DaemonSets
4) uncordoning a Node
5) Hand-On-Demonstration.

Node Draining :-

1) Sometimes, we need to remove a Node from cluster in service.

We need to remove a node from a Kubernetes Cluster & there could be the multiple reasons to removing a node from a Kubernetes Cluster.

Page No.
Date

Sometimes we need the Maintenance window, might be for applying patch, Resources like CPU, increase, additional disk -- etc. we fix the issues & add the cluster back.
This process is called "Node Draining."

(2) Application should'nt be impacted by this process.

(3) Draining node: Containers running on that node will be gracefully terminated & re-schedule to another available node.

Standard process to Draining K8s node:-

(1) We can use Kubectl to drain node

Kubectl drain <node_name>

Deamon Set:- there are the pods that are tightly coupled to that node.
it means sometimes we have a requirement that we need to execute a particular process on all of my worker nodes. that could be monitoring, that could be the health check, that would be the performance check.
So this is called deamon set

(2) if we are running deamon set drain node could be difficult. To remove the node we have to pass parameter --ignore-daemonsets.

Kubectl drain <node-name> --ignore-daemonsets.

Page No.
Date

(3) Uncordoning K8S Node :-

If node remain part of Container. User can allow pods to run on that node.

~~If that node is~~

Once you drain the node you put the fix on that node & if that node is still a part of the cluster & you want to allow the pods on that particular node. & also you want to allow that master can schedule the additional resources or addition port in that particular node.

Then you need to uncord that particular node & you can do that with the command kubes.

Kubectl uncordon <node_name>

Hands-On :-

(1) Kubectl get nodes. → list the nodes.

(2) Kubectl get pods -o wide. → list all pods.

(3) To create ~~the~~ pods, we creat Yaml file.

```
mkdir drain_pod
cd drain_pod
vi drain.yaml
↓
add the Contents.
```

This file will create a pod.

Kubectl apply -f <file.yaml>.

↳ will Create a pod.

(4) Deploy the Service / Container on Pod.

Creat a Yaml file for deployment.

Kubectl apply -f <deploy.yaml>.

(5) Kubectl get pods -o wide → get all pods Details.

(6) To Drain the pod.

Kubectl drain <node_name>.

Kubectl drain k8s1p37worker-node --ignore-daemonsets --force.

we note the pod on worker-node is removed but other pod which was part of deployment on worker-node moved to other node. but pod we create via pod.yaml file is drain.

(7) Kubectl get nodes → Status of nodes

(8) Re-join the pod to cluster :- (Uncordon)

Kubectl Uncordon k8s-ip-37-worker-node

Kubernetes upgrade K8s Cluster :-

Video no:- 194, Course :- devops-training

1. Upgrading with Kubeadm
2. master node upgrade steps
3. worker node upgrade steps
4. Hands-on Demonstration.

Why upgrade necessary :-

1. K8s cluster update is periodically task to keep the cluster sync with latest K8s release.
2. Kubeadm makes the cluster updat very easy

Master node upgrade :- process

1. Drain Control plane node.
2. plan the upgrade
3. Apply the upgrade
4. upgrade Kubectl & Kubelet on Control plane node.
5. Uncordon (Join) the node.

Page No.
Date

Worker node upgrade process :-

1. Drain worker node.
2. Upgrade Kubeadm
3. upgrade Kubeadm
4. upgrade Kubelet Config
5. upgrade Kubectl & Kubelet.
6. uncordon (Join) the node.

what do you mean Cluster :-

A Kubernetes Cluster is a grouping of nodes that run Containerized apps in a efficient, automated, distributed, and Scalable manner.

Hands On :-

Video :- 194 : Course :- Udemy Course :- devops-training.

Kubernetes working with Kubectl :-

1) Kubectl is Command line tool for Kubernetes Cluster.
2) Kubectl uses the K8s API Internally to Carryout the Commands

Kubectl get :- get the object : present in K8s cluster.

Kubectl get <object_type> <object_name> -o <output> --sort-by <JSonpath> --Selector [Selector]

Page No.
Date

Kubectl Support Output in: JSon / XML.

-o = Set Output format YML/JSON
--Sort-by = Sort Ouput using JSON path Expression
-- Selector :- filter results by label.

Kubectl describe :- Command.

you can get Detailed information about any Kubernetes object.

```
Kubectl describe <object_type> <object_name>
```

Kubectl Create Command.

1) You can Create any K8s object using Kubectl create.
2) file descriptor must be YAML.

```
Kubectl Create -f <file.name.YAML>
```

Note: file should be in YAML.

Kubectl Apply Command

1) Kubectl apply is similar to Kubectl Create.
2) if user use Kubectl apply on already existing object. It will modify the existing object, if possible.

```
Kubetcl apply -f <filename.YAML>
```

Page No.
Date

Kubectl delete Command

Kubectl delete: deletes the object from K8s cluster.

```
Kubectl delete <objet_type> <objet_name>
```

Kubectl Exec Command

1) Kubectl exec, used to run commands inside container.
2) K8s resource must be running state.

```
Kubectl exec <pod_names> -c <Container_name>
```

Hand On. :- Video on 196,

(1) Create a pod:

Create a Yaml to create pod :→ object type.

```
Kubectl Create -f pod.Yaml.
```

(2) `Kubectl get pod.` → object type.

(3) `Kubectl api-resources` → list all supported by resources. (objects)

(4) `Kubectl get pods draing-node -o wide`

Show all details along with cluster Ip address & more.

(5) `Kubectl get pods draing-node -o JSON`

Output in JSON format.

-o = Yaml = output in Yaml format.

(6) `Kubectl describe pods draing-node`

Complete information of pod.

(7) `Kubectl exec [redacted] draining-node -c nginx -- cat /etc/nginx/nginx.conf`

(Space)

To run Command on Pod Container.

(8) `Kubectl delete pod <pod-name>`