Kubeadm v/s kops

Kubeadm is a toolkit for bootstrapping a best-practice k8s cluster on existing infrastructure.

Kops stands for Kubernetes operations and lets you create, destroy and upgrade K8s cluster &

Is supported on AWS , GCP and Vmware.

And this is the easiest way to get a production-grade k8s cluster up and running.

Kubeadm cannot provision your infrastructure, it works not only as installer but also building block.

It sets up minimal viable cluster.

Kops on the other hand is responsible for entire lifecycle of the cluster, from infra provisioning to upgrading to deleting, and it knows about everything: nodes, master, lb, cloud providers, monitoring, n/w, logging etc.

**CNI - the Container Network Interface, to configure network interfaces in Linux containers**

**Antrea, Calico, Cilium, Flannel, Coil**

Kubernetes architecture

Role of of KubeAPI server

Q. Suppose I am create a pod. How k8s component will work to deploy this pod?

A. To run a pod first we need a deployment. This deployment instructs k8s how to create and update instances of our app. And once we have create a deployment, k8s control plane schedules the app instance included in the deployment to run on individual node of the cluster.

What is control manager?

Control manager consists of two parts:

node-controller: takes care of node, responsible for on boarding new nodes to the cluster,

handling situations where nodes become unavailable or gets destroyed.

replication-controller: ensures actual capacity = desired capacity

(or desired no. of containers are running at all times)

Scheduler v/s kubelet?

If anything happen to master node, how app will work?

What is special pods running on master nodes, compare to worker nodes?

Deployments , daemon-set & stateful-set?

If we have 1 pv. Can we create 2 pvc of 500Mb ?

How does pv and pvc works?

Different storage type in k8s?

What kind of storage class you have used in your env?

Vpc-peering in different A-Z?

If we have a vpc in zone-A and another in zone-B. Can we do vpc-peering?

Is auto-scaling possible in Amazon-aurora?

Jenikns architecture?

To access a k8s cluster using Jenkins, what kind of access you need? And how you will create that access, to run kubectl commands in Jenkins?

What is roles in Ansible?

What is modules?

What is custom inventory fields?

Create dynamic inventory field in ansible?

How you will access the app from outside in k8s?

What is ingress controller?

What type of operator you have used in ingress?

Which component will take care of ingress part in cluster?

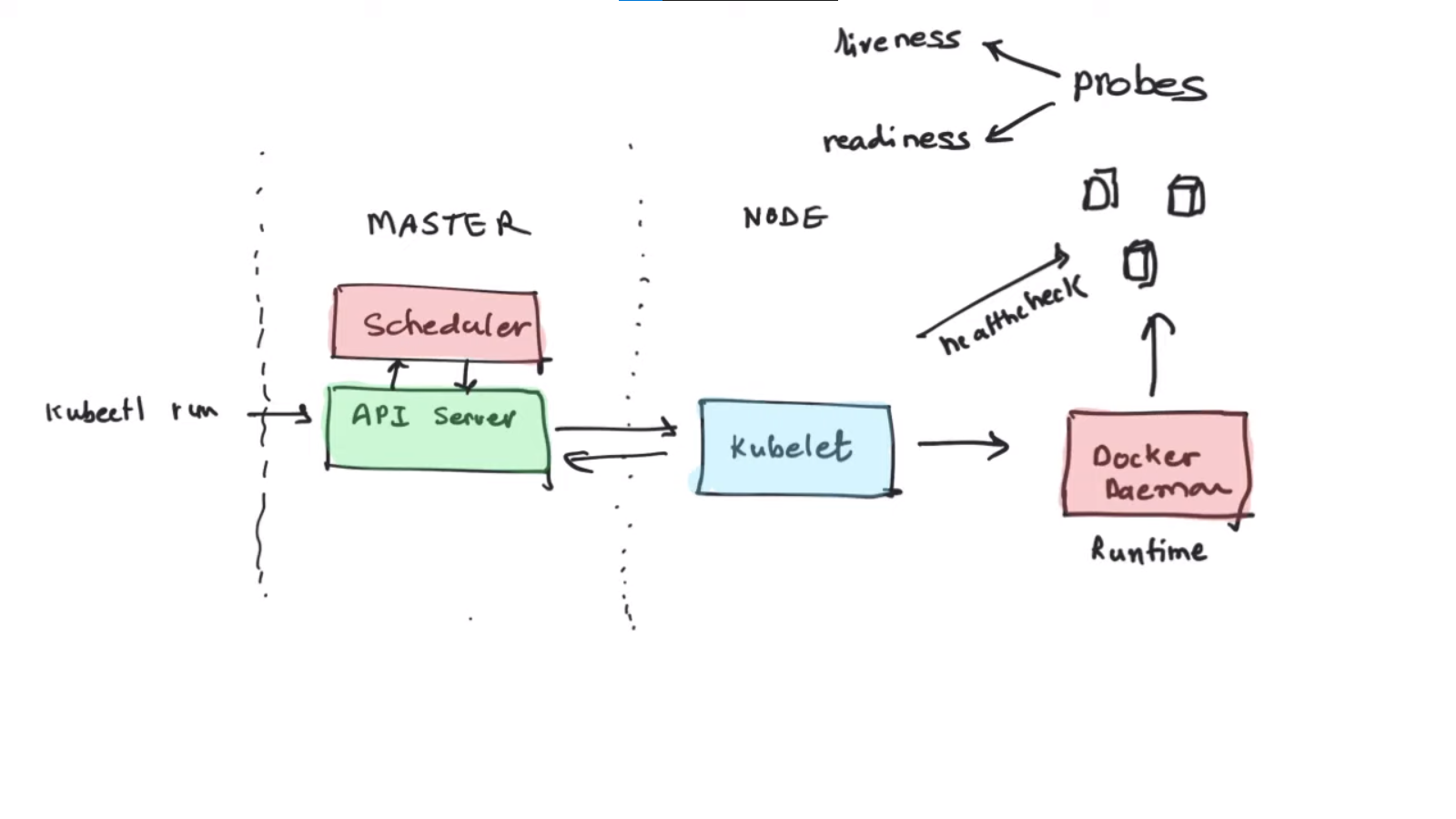
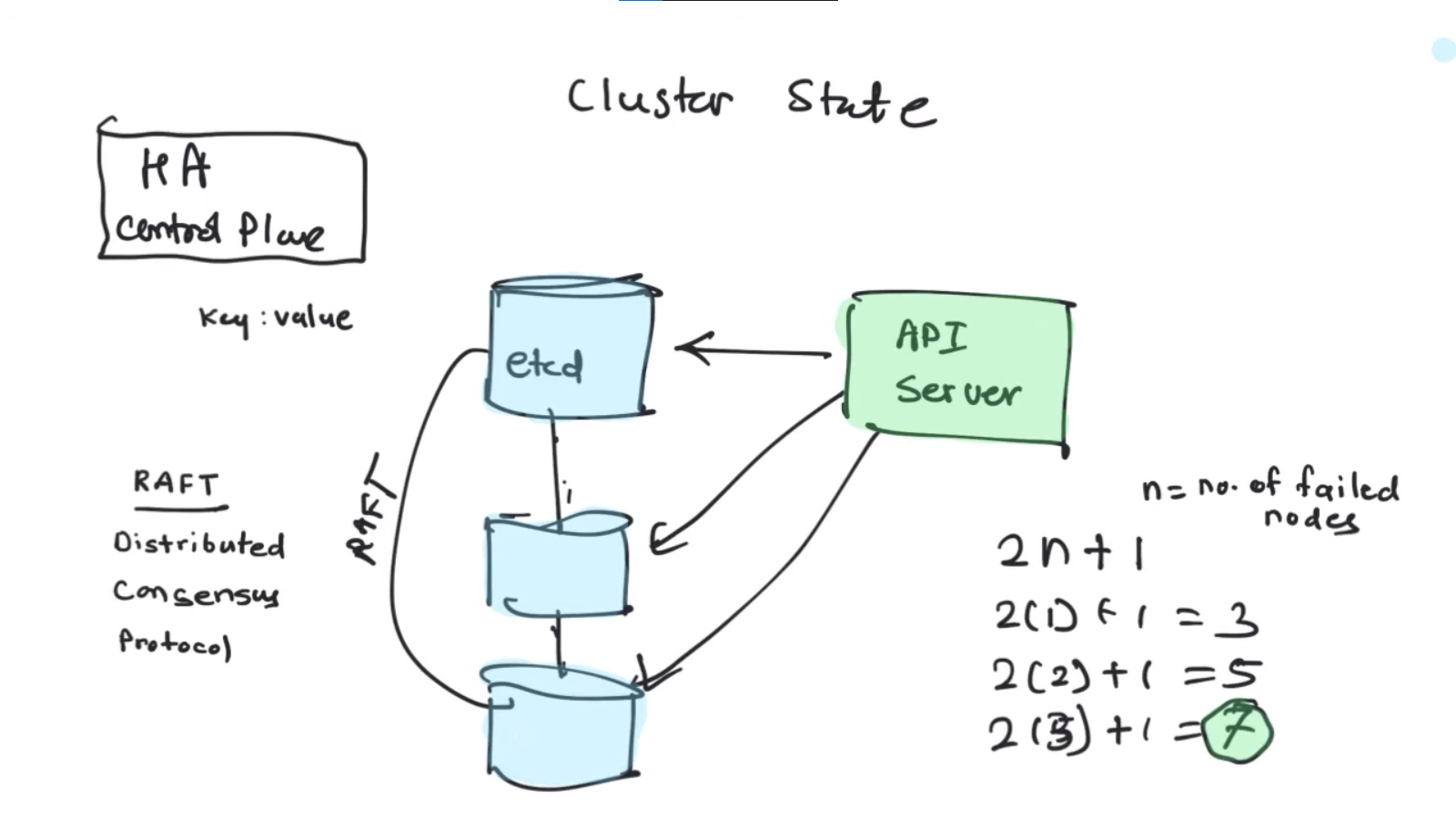
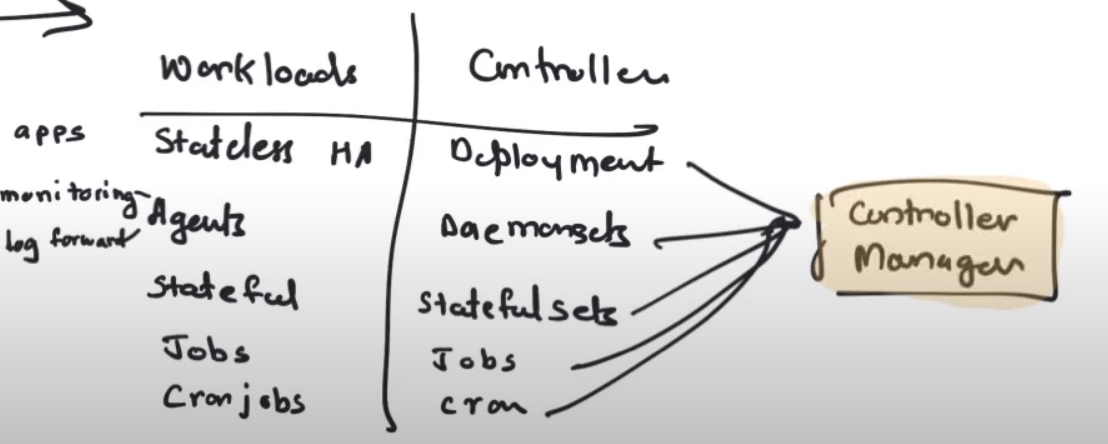
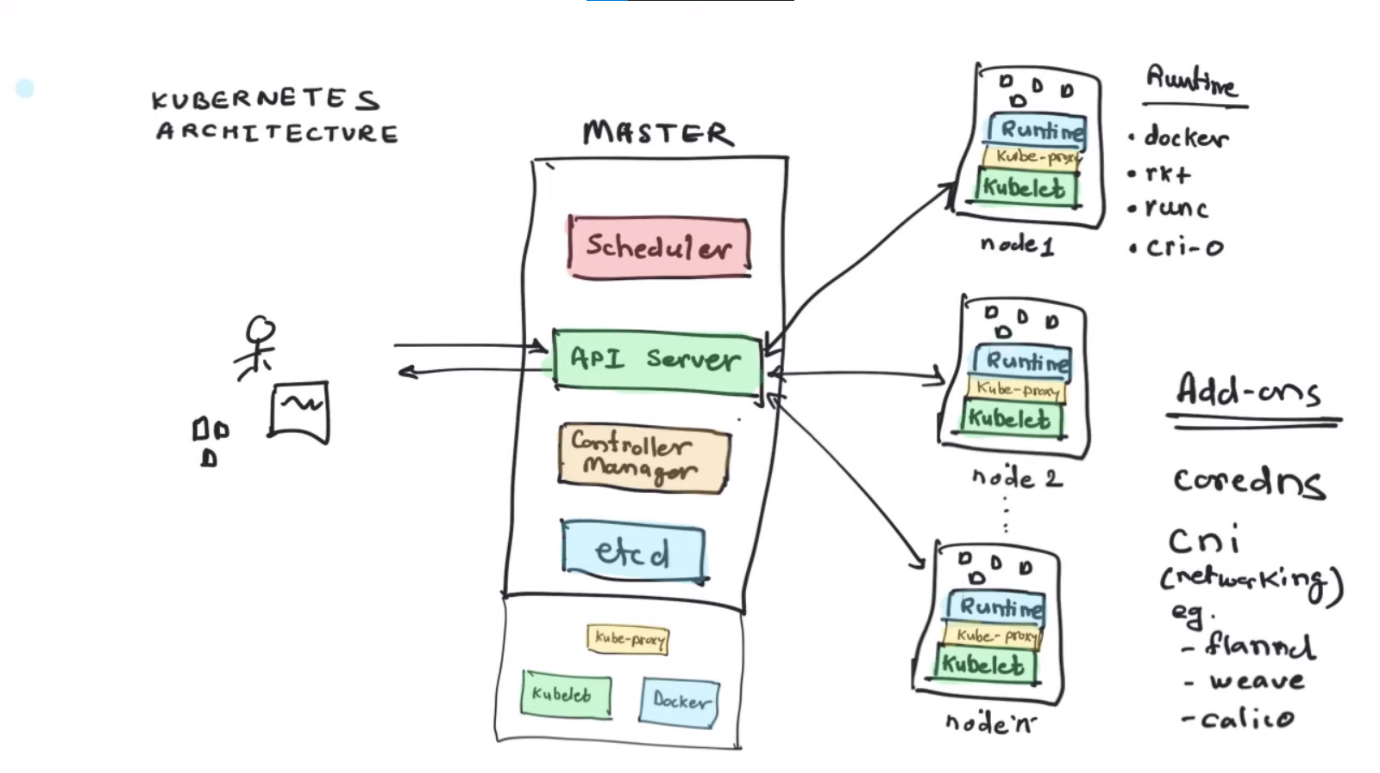
Sto

How you monitor cluster using Prometheus?

Difference b/w prome and Grafana?

Get metrics of pods?

How to fetch data from pods?



------------------------------- Master-Node----------------------------

Api-Server: in order to interact with k8s whether it be k8s gui dashboard, or kubectl cli we have api-server

whenever we give instruction to create a new pod the request first goes to api-server.

scheduler: decides which container runs on which node,

or, where to run which container...

or launch particular container on particular node like:

you may have contraints like: cpu intensive operations on cpu optimized machines and so on...

control manager: responsible for providing app deployment, its versioning.

container scaling, replication

etcd: state of cluster is stored here.

it is key:value pair based db.

config. changes

any req. to api-server, first it's entry is made here.

probes: used to do health checks

liveness: kubelet uses this to know when to restart a container in case of deadlock.

readiness: kubelet uses this to know when a container is ready to start accept traffic,

a pod is considered ready when all of its containers are ready.

cre: we need to have installed cre on every node, whether it be master or worker.

bcz all the services in the cluster will be running on containers.

we can use container-runtime-engine like:

docker, podman , rkt, runc, cri-o and many more...

cni: network plugins are also used now-a-days for better n/w stability of our cluster.

eg: flannel, waves, calico and many more....

core dns: for provide dns facility to our pods...

-------------------------------worker-Node----------------------------

kubelet: this is agent of api-server on worker node, api-server periodically sync data from kubelet

to know the helath status of worker node.

this is also responsible for launching new pods on the node after receiving info from kube-scheduler.

kube-proxy: this is used to provide communication among pods running inside nodes(same or different)