SESSION 5

Containers ll not support auto scaling

Scale up : whenever the traffic increase towards app we have to increase no of servers
There are 2 kind of scaling : vertical scaling and horoizontal

Verticals scaling means we have to increase the system resources
Horizontals auto scaling : means we have to add additional server to existing infrastructure
So the container ll not support it

Scale down :
When traffic goes slow towards the application .. we have to remove the additional servers

Containers ll not supports load balancing
If we lose the container we are going to get application downtime
Containers l not supports self healing system
To over come the problem we are moving towards the

What is Kubernetes : kuburnetes is a tool or it is going to manage all the containers ..
It performs the automated deployment of the application

Kubernetes feature :
1. Orchitrtion
2. Autoscaling
3. Load balancing
4. Self healing

It is a platform independent
We can run it on a any cloud
To perform the automation Kubernetes ..we are going to write manifested files

1. Create a jump server or
2.We have to install kupecel , ekcplkubernetes service , aws

Understand what is cluster :
1.Kubernetes ll manage the application in cluster
Cluster is a group of nodes it contains master nodes and worker nodes
Master node is the hero of the cluster which is going to take care cluster head
Master node is responsible for total health of the cluster
Worker node : it is nothing but like a server where we can
In a cluster at least we should have one master node and one worker node

There are 2 types of cluster :
On premises cluster : manage cluster by ourself if something goes down in application v are the responsible for that
Cloud manage cluster : this clusters manage by the cloud providers
We are responsible for our app
In was if we want to create a cluster we have a service called eks
In a master node there are 4 components we have

1. Api server hero: whenever we want tp perform auto scaling it is playing a crucial role
2. Etcd : it is a distributed database where we are going to store info about the cluster and application inside the etc
3. Controller : it is responsible for monitoring health of the application : this is always to check the desired state == actual state \
4. Scheduler : scheduler is a component it is going to schedule a pod in a node it is responsible for scheduling a pod in a node Kubernetes ll run the app in a pod ..it contains containers …pod is the smallest deployment unit in a kubenetes

Worker node conponent:

1. Kubelet : it is responsible for creating pods it is going to act as an agent . Something goes wrong in a worker node it ll communicate in a master node
2. Container run time : is nothing but like docker .. it is responsible for pulling the docker image creating the containers starting the containers it is responsible for managing the containers lifecycle
3. kube=proxy : it is a networking component in wokrer node . It is responsible for creating deployment exposing app to internet

Api version : it defines schema representation of the object
Kind defies what kind of object that you are creating
Meta data defines name of the object
Spec defines behaviour of the object