docker run -d -v /var/jenkins\_home:/var/jenkins\_home:z -p 8080:8080 -p 50000:50000 --name myjenkins jenkins/jenkins:latest

kubernets learn the basics now for this demo, may be we can do advanced stuff after this

Each pod in k8s gets a virtual ip address, and when pod gets killed and the pod is restarted,

the ip address changes

Components of K8 cluster->Pod, Service, Ingress, volume, deployment,config map, secret, stateful set

"Service has a lot more to do than just being static-ip(it is one of the functions of the service)

,it can be a load balancerit is communication endpoint to the external world"

What is a service:it is a permant ip address for the pod, service ipaddress is constant,

though the pod is restarted

"Service is also load balancer"

What is Ingress : The external requests to the applications are handled and routed by

ingress for ex: if a user wants to access the app in a browser, the request is served to the

application server through ingress

Ingress forwards that request to the service(maybe\*\* the apiserver forwards the request to

ingress then it forwards it to service then to the application service)

Config map - contains data about external configuration of the application, environment variable

are used to use the variables(used in config map) inside the application

Secret:

Volume persistance:

->->Deployment is an abstraction on top of pod

->->\_\_\_ Stateful Set(used for applications like Databases)-If there are two clone pods of a db,

and that dp is persisting its volume, to sync the data between the two clones stateful set is used

Three things needs to be installed in the nodes for k8s to work:

->Container Runtime

->kubectl

->Kubeproxy(manages the routing of requests)

4 process should run in Master node:

1)Api Server ->Gateway into the cluster(any request passes through apiserver, it also manages

authenticartion of the request)

2)Scheduler->Schedules the pods(Checks where to create new pod depending on the node resource usage)

3)Controller Manager->Checkes the status of pods(if a pod dies, it tells the scheduler to create

a new pod)

4)etcd -> Brain of the cluster(\*\*Every transaction is stored in the form of key value pair\*\*, \*data

of the cluster is stored in the etcd, like if a new pod is created its data is stored in etcd)

\*Application data is not stored in etcd\*

###two different pods on a node communicate using services

\*\*\*\*\*\*\*Configure yaml file explained:

"'targetPort' is the port that the service communicates with pod" this targetPort is in \*service yaml\*,

that means any request that comes to the service is routed to the pod through target port

any app service listens on the key called "port"(port:80) key in service yaml, (if nginx has to communicate

with db service nginx service will be listening on port->80 )

\*\*\*target port in service yaml should match, "containerPort" key in pod specification

containerPort is specified in pod scpecification of deployment yaml

readyness probs and lightress probs

What are different services

use ingress next time

What is cluster ip

create a feature branch then do the triggering

hard code the port in yaml file, so the port given in yamlshould be used for running the app

add image tag in the google container registry, avoid using latest tag

include static ip in gke config file - service yaml

Pod is a collection of 1 or more docker containers. It is an atomic unit of scaling in Kubernetes

Readiness probe is to check whether the application is accepting the traffic to serve requests

or not

Liveness probe is to check the whether the application is working properly or not

