Date.....
Page.....

K8S

Automatic Deployment
Management of Containerized app
Scaling

Kubernets
↓
Containers

If we have 3 number of node and multiple Server then how we can control and monitor that all.
So we use K8S.

Deploy K8S we get cluster.
↳ Two important ~~Parts~~ Parts
1) Master
2) Worker nodes

master
↓
-) cluster is Group of nodes.

Master node (API Server, Scheduler, ETCD)
↓

Worker node 1, Worker node 2
↓

kubelet	kubelet
kub-proxy	Container-runtime

Date.....
Page.....

Component

1) API Server
2) ETCD
3) Container Runtime
4) Scheduler
5) kube Proxy
6) Kubelet
7) kube controller

Start kubernini

↳ minikube start

minikube dashboard

Date
Page

Creat pode in k8s
↳ Using nginx in docker

-> kubectl create deployment (name) --image = (reponame in docker):latest

If Service Stop

Check kubectl get deployments
kubectl get pods

↳ Command not work

pods is Create automatic

- - - - - -

Docker

Create a small node app.
↳ In Directory write Dockerfile. (Demo Build)

Creat a Docker img we Docker hub repository name.

-> docker build -t (Repo name of Docker):01 .
:01 is version name.

Repo name -> bhurguvbhalodiya04/myapp:01 .
Check Docker images.

After Run this command Push in docker
docker push (Repo name):(version)

Deployment of Our kubernet for WebApp

Repo name: bhuryuvbhulodiyaw/myapp

Date.....

Page.....

Deployment on Docker

kubectl create deployment my-webapp
--image (Repo name):01

kubectl get pods

kubectl get deployment

kubectl expose deployment my-webapp
--type=LoadBalancer --port=3000

kubectl get deployment

kubectl get Services

minikube
~~kubectl~~ services my-webapp

Create live env.

for container

Dashboard Pods -> kubectl, container

Date.....

Page.....

~~Create new img~~

fore learning create new img.

1) Deployment in img Point

③ kubectl set image deployment my-webapp ~~myapp=~~ myapp=bhuryuvbhulodiya04/myapp:02

☆ Simple Change in code Test on ~~local~~ local then type 1 command create new pods using that command.

① docker build -t bhuryuvbhulodiya04/myapp:03

② docker push bhuryuvbhulodiya04/myapp:03

Incorrect use

kubectl set image deployment my-web [old]

ok is not found.

& kubectl rollout status deployment my-webapp
kubectl rollout undo deployment my-webapp
\ Remove last.

Ctrl

Healing

Date
Page

Self Healing in k8s

If Pod have error then k8s automatic start the service

SCALING of App k8s

for scaling

kubectl scale deployment node-app
--replicas=2 (num of a pods)

Yml Config

Date.....
Page.....

-> kubernetes.io/docs/reference/One-Page-kubernetes-Ref

Run yml file deployment.yml
kubectl apply -f deployment-node-app.yml

~~api ve~~
apiversion: apps/v1
kind: Deployment

metadata:
  name: Give Unique name of instance

spec: -> Pods details
  replicas: numbe which u want to create
  selector:
    matchLabels
      app: node-app / Give your name
  template:
    metadata:
      labels:
        app: Give same name of matchLabels
    spec:
      containers:
      -name: node-app
       image: Docker img.

Date
Page

* Deployment

apiVersion: apps/v1
kind: Service

metadata:
  name: Unique name of services

spec:
  ports:
  - name: http
    port: 8081 / Give something else
    targetPort: 3000 / which app running
  selector:
    app: / Give name which in MatchLabels
  type: LoadBalancer

MongoDB

Date.....
Page.....

Download mongodb in Docker

docker run -p 27017:27017 -d --name mongodb mongo

docker pull bhuryuvbhulodiyaou/~~mgi~~node:01

Run Mongodb

docker run -d -p 27017:27017 --network my-net --name mongo mongo

To Run our Node App

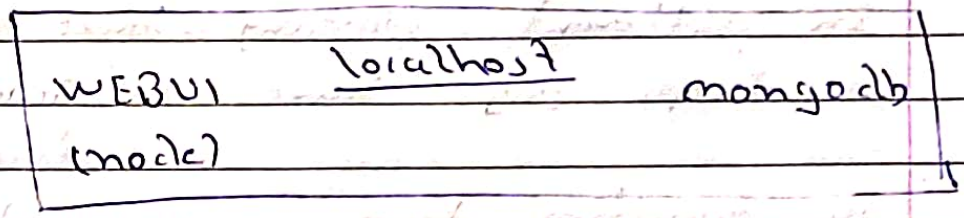
docker run --network my-net -p 3000:3000 --name myapp bhuryuvbhulodiyaou/node:01

Date.....
Page.....

Two way to Run Application with multiple containers

1) Run multiple container in same pod
2) Run each container in seperate pod

1) Run multiple container in same pod



Deployment.yml

Same yaml file

name: mongodb
app: nodedb-app
app: nodedb-app

Spec:
containers
-name: nodedb-app
image: Dockerimy.
-name: nodemongodb
image: mongo:latest