166-199 • WK 24

15 JUNE THURSDAY

Kubernates

① ⊙ It is open-source container engine or container management tool
⊙ It automate deploy, scaling, & managing container appln

② Kubernates → K8S

③ google → CNCF handing Kubrate
[Cloud Native Computing Foundation]

IMP

⊙ It is hard to handle multiple containers at a time so that why It comes into pitcure & It automate process also & manage container appln.

Deploy, scaling, scheduling, Load Balancing, monitoring, Roll Back, Batch Execution

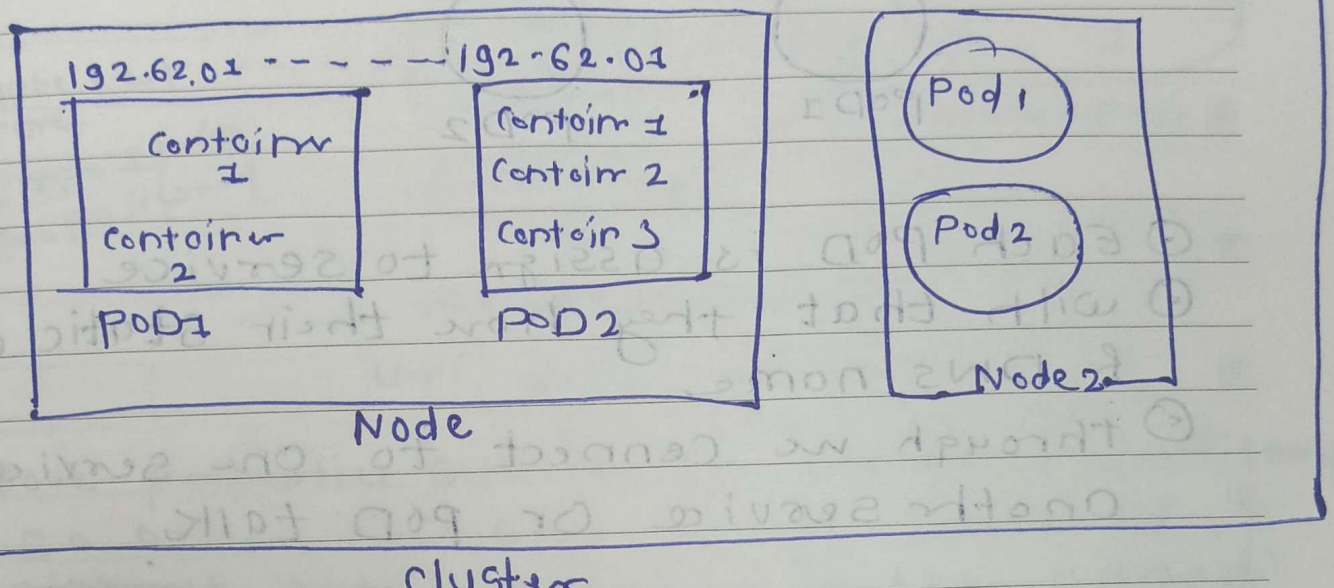
Components

- POD
- Node
- cluster
- Replication set / controller
- Service
- Deployment
- Secrets
- Config Map
- ETCD

2023

MAY 2023

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				



POD → ① ह्यामध्ये 1 containr किंवा multiple containr असू शकतात.
② Every POD has unique IP address.
③ multipe POD connect with Each other with that Unique address

Node → ① Node contain one or more POD

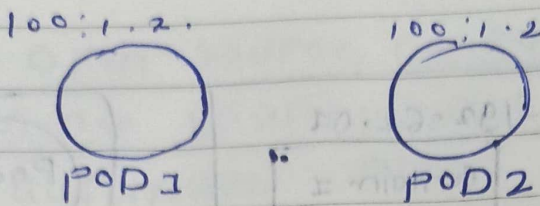
Cluster - ① Cluster contain one or more Node inside cluster.

Replicaset → ① Each POD have unique address
② Due to some issue POD goes crash or down.
③ at that time we have Backup POD
(self healing)
∴ For POD1 → I have 3 Backup POD with diffrent address
④ so it will replace that Existing POD & assign new IP of that specific POD

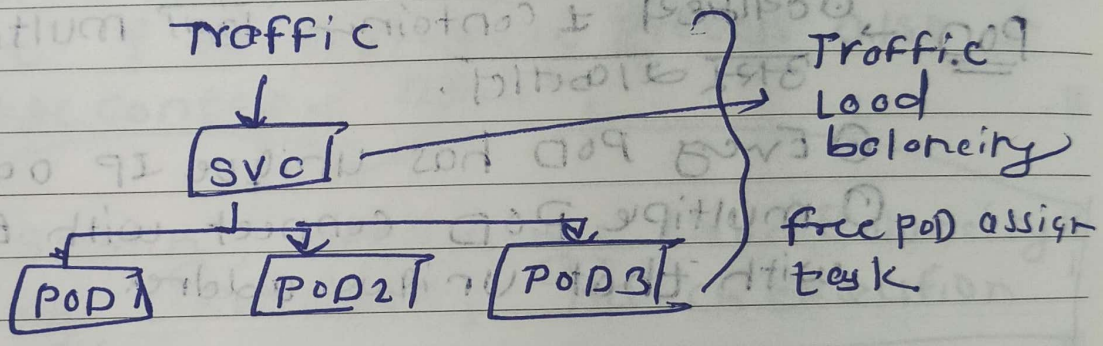
17 JUNE SATURDAY

SVC — DNS — SVC

Service



- Each POD is assign to service
- with that they have their static address & DNS name
- through we connect to one service to another service or POD talk.



Service Type - ① cluster IP
② Node Port
③ Load Balancer

Deployment

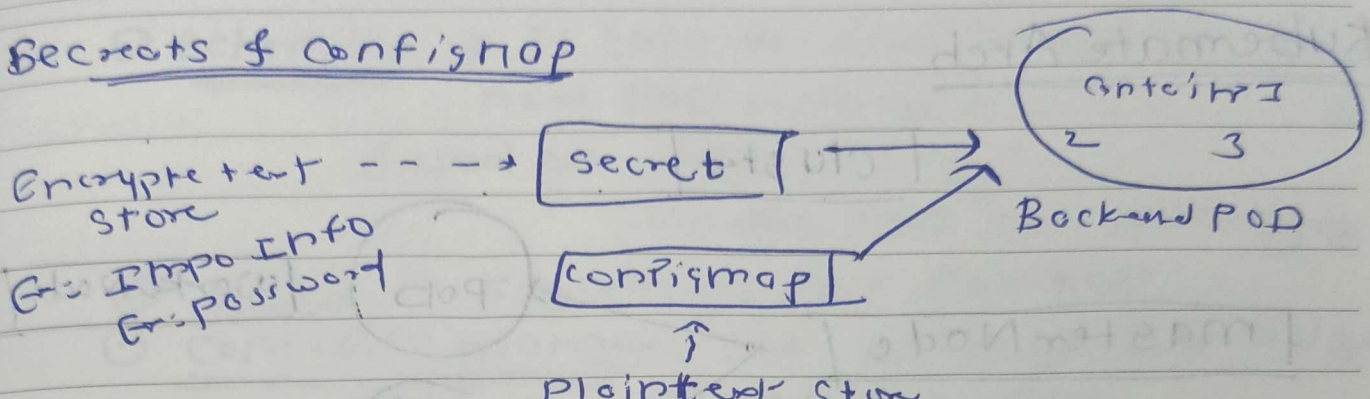
→ ① It is used to managed PODS
② We can scale appln by increasing No. of POD or update running Appn using Deployment obj

Kubectl create deployment first-deploy --image=<Docker_Img_Name> --port=8080 --replicas=4

Present outside POD

↑

Secrets & Configmap



- Store properties outside in secret & configmap so other POD also easily access it.
- ~~[illegible]~~ security extra added.

* ETCD

- Kubernate ETCD uses to store config of Kubernete cluster
- It is key-value database store
- It store all secret & configmap data inside etcD database
- Max Limit is 1mb to store secrets.

JULY

Kubernative → minimum 1 cluster → at least one master & worker Node

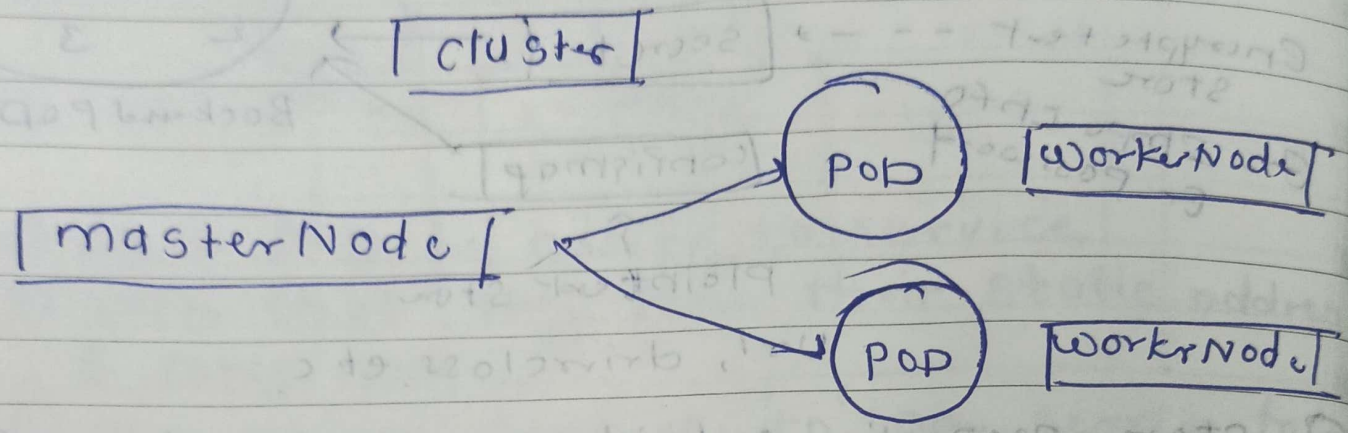
170-195 • WK 25

19 JUNE MONDAY

MAY

M	T	W	T	F	S	S
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Kubernate Arch



① master Node will monitor ~~worker~~ Node running smoothly or Not

② master Node will check POD working or Not

③ If any POD goes down new POD will assign with New IP

④ If any Nodes are down then he will tag all Pods to New Node.

★ masterNode → ① API server
② scheduler
③ Controller manager
④ ETCD
} components

① API server → ① cluster gatway.
② K8S UI
③ Command Line Tool (Kubectl)

2023 ⇨ health of POD = Kubectl get nodes
Kubectl get pods

⊙ schedular → It take decision on which Node new POD will add.
Based on CPU memory,
current POD use, memory

⊙ ETCD → Key-value pair database

⊙ controllr manager → manage controllr.
Node → i] Node controllr
heath monit → ii] Replication Controllr
servic → iii] Endpoint controllr

* Worker Node (some work)
① Kubelet → Communicate with master Node using API Server.
② Kube-Proxy → It runs on Each node
∴ It is responsible for maintaining Network config & rules.
iii] Container runtime → help to run containr inside pods.

* Minikube → ① run k8s cluster in Local machin
② It runs single-node kubernate cluster on Local machine

* Kubectl → command Line tool Communicate with Kubernate
chaclately install for windo.

Both Install.

172-193 • WK 25

21 JUNE WEDNESDAY

MAY 2023

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Start Kubernate Dashboard,
Commd → minikube dashboard

★ Command → minikube start
minikube stop
minikube status
minikube ~~#~~.

kubectl logs <pod-name>
kubectl get pods
kubectl get service
kubectl get deployment.

Deploy app → kubectl apply -f deploy.yaml

Increase decrease No of pod Scale Appln → kubectl scale deployment <name> --replicas=3

kubectl delete pods / service / deployment
kubectl delete all --all

kubectl edit deployment

★ Yaml. Config

① apiVersion:
② kind : ③ metadata:
④ specification: (configuration)
⑤ status (once we create resource)

Ex: Service Registry

deployment.yaml

```
apiVersion: v1
kind: Deployment
metadata:
  name: service-registry
spec:
  replicas: 3
  selector:
    matchLabels:
      app: service-registry
  template:
    metadata:
      labels:
        app: service-registry
    spec:
      containers:
        - name: service-registry
          image: lokeshkhodke/<img name>:tag
          imagePullPolicy: Always
          ports:
            - containerPort: 8761
```

(connection)

Service.yaml

```
apiVersion: v1
kind: service
metadata:
  name: service-registry-svc
spec:
  selector:
    app: service-registry
  ports:
    - port: 80
      targetPort: 8761
```

80 → http.

8761 → Service Port

8761 → service Port

Service Create For Expose so other can use & view it

174-191 • WK 25

23 JUNE FRIDAY

MAY 2023

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

✱ Kubernate namespace

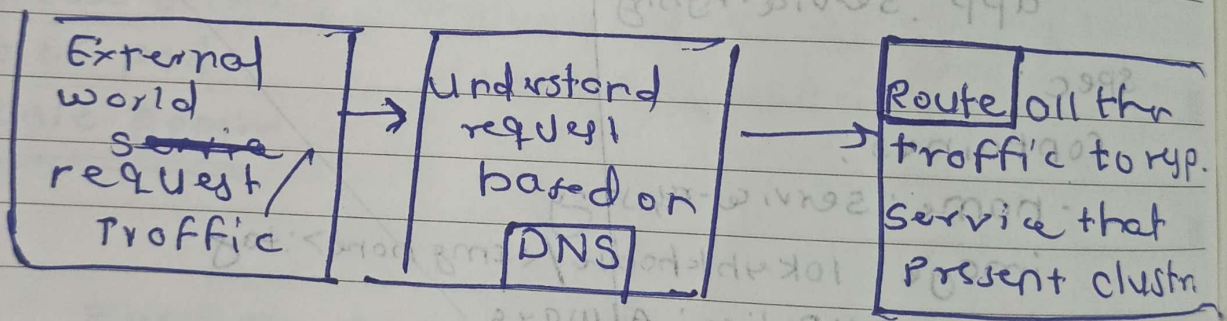
It is use to give name to component of kubernate.

Ex. Kubectl create namespace myName

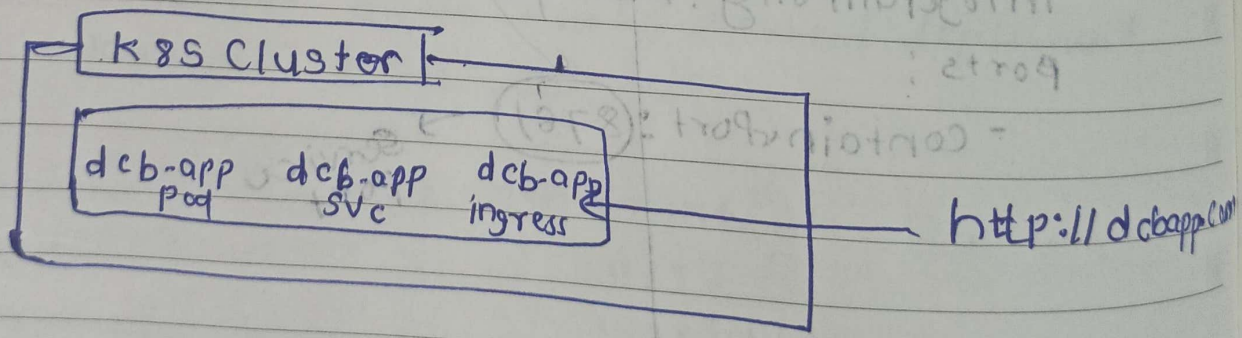
Kubectl apply -f depoly.yaml -n deployServiceName

Kubectl get all -n deployServiceName

✱ Kubernate Ingress



Ex:



Routing handle By Ingress Controller Pod

2023

A ingress.yaml

```
apiversion: networking.k8s.io/v1
Kind: Ingress
metadata:
  name: food-catring-ingress
spec:
  rules:
    - host: dcbapp.com  ⟶ Ingress host name
      http:
        Paths:
          -path: /service1  ⟶ ही path आली की
           PathType: Prefix
           backend:
             Service:
               name: service1   ⟶ या service redirect करायची.
               port:
                 number: 80  ⟶ default App port
```

(headless service)
ClusterIP: none

✱ Kubernate Statefulsets (mostly use with dB) (store state meandata)

mySQL

Based on previous data

Pod 0 master | Pod 1 work | Pod 2 work | Pod 3 main

(delete 3 then 2 soon)

① It scale based on previous pod
② & It delete last pods then 2nd last like that

25 JUNE SUNDAY

(Backup Plan data)

✱ Kubernates Volumes

⊙ Outside cluster Store data into Volume

1. Storage should not depend on POD Lifecycle
   Colwoy Available not based on old pod snapped
2. Storage should be all nodes available for
3. Storage should survive cluster crashes

[PersistentVolume.yaml] & [Claim] also create

(For access POD & claim

```
apiVersion: v1
kind: PersistentVolume
metadata:
    name: Volume-Backup
    labels:
      type: local
  Spec:
    storageClassName: manual
    capacity:
      storage: 10Gi
    accessModes:
      - ReadWriteOnce
    hostPath:
      path: "/mnt/data"
```

★ Kubernate Health Probes

① Liveness Probes → It check your appl^n is Live or not

② readiness probe → your appl^n is ready to serve request or not

★ Deploy all files at a single time

kubectl apply -f foldrname → It all deploy. Struct, secret, configmap file Project

★ delete all

kubectl delete -f foldrname →