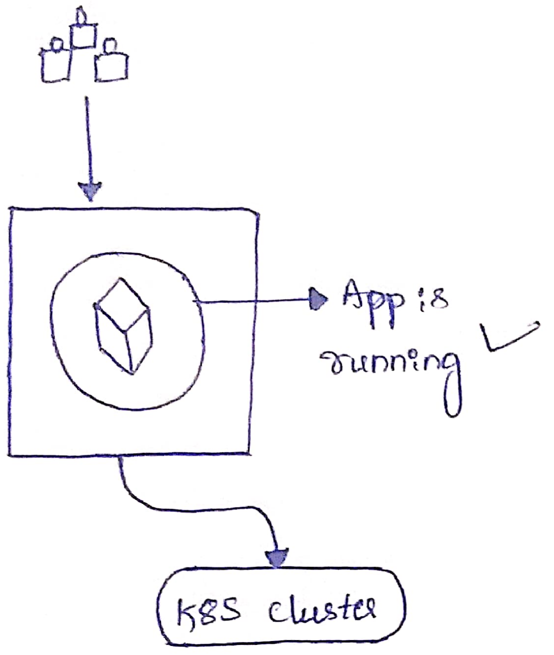
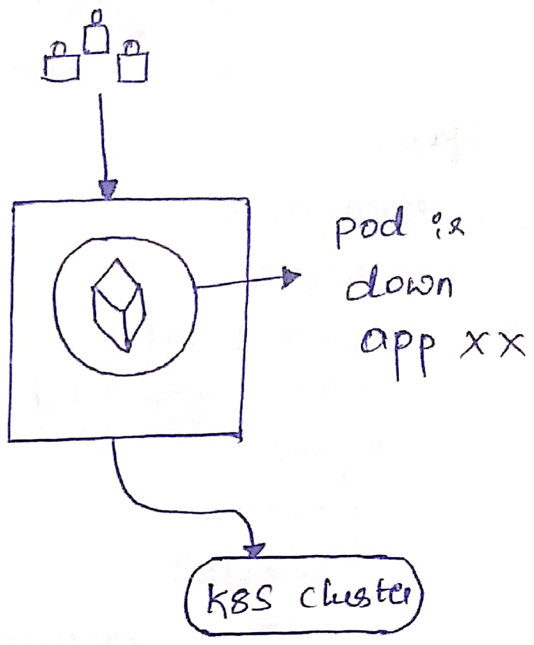
Day - 8

A pod is running on a single node in K8S cluster. What if the pod is dead? our application is down checkout below:-





- In this case, Someone has to identify that pod is down and try to create a new pod But we already know that kubernetes supports self-healing. ~~Here~~ kubernetes do self healing/scaling is done through Replicaset.

- Replica Set:-
  - It is to maintain stable set of pods at any given time.
  - It will count the replicas using Label of pods.
  - It can be used for scaling up or down according to requirement.

→ Checkout below Sample YAML Code for Replicaset.

```
apiVersion : apps/v1
kind : Replicaset
metadata :
    name : frontend
    Labels :
        app : guestbook
        tier : frontend
spec :
  replicas : 3
  Selector :
     matchLabels :
     tier : frontend
  template :
     metadata :
       Labels :
          tier : frontend
   Spec :
      Containers :
         - name : php-redis
          image : gcr.io/google-Samples/gb-frontend:v3
```

→ <u>Basic Kubectl Commands</u> :

```
# kubectl create -f replicaset.yaml
```
to create any k8s object using manifest.file.

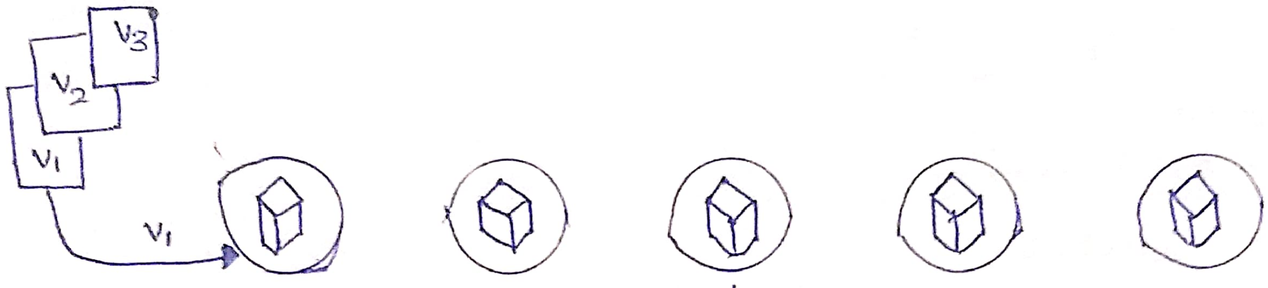
```
# kubectl get rs
```
to get replicaset details here.

```
# kubectl get pod
```
to get pods list in default namespace.

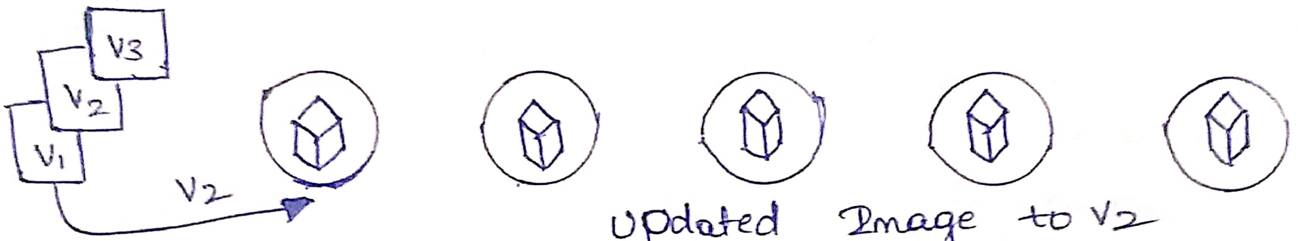
```
# kubectl scale --replicas=1 rs/frontend
```
to scale down the replicas to 1

```
# kubectl delete rs/frontend.
```
to delete frontend Replicaset

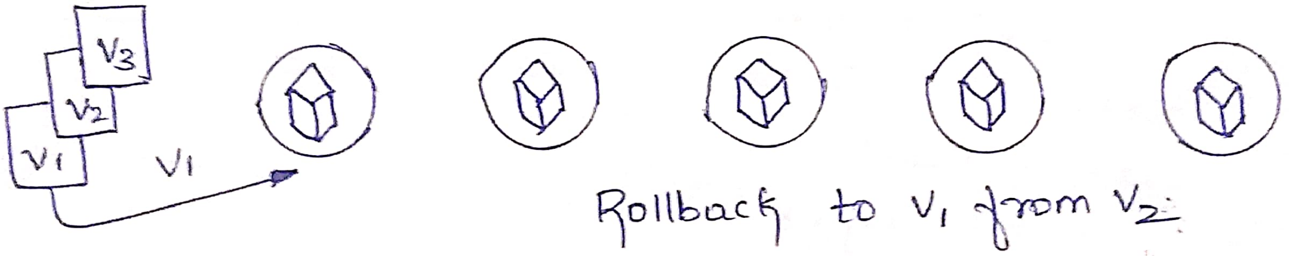
- Deployment:
  - This object in Kubernetes provides upgrade / Rollback / changes gracefully in cluster.
  - Deployment manages Replicaset and Replicaset Controls the numbers of pods to be running.



current version



updated Image to V2



Rollback to V1 from V2:

- This can be done easily through deployment object in Kubernetes.

→ Lets Do Small Handson on RS and Deployment:-

1) I am using minikube cluster you can setup any Kops / AKS / EKS K8S cluster of your choice.

```
# minikube start
```

2) Create a YAML file with below data :-

```
apiVersion : apps/v1
kind: Deployment
metadata:
  name: nginx- deployment
  labels:
    app: nginx
spec:
  replicas: 3
  Selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      Labels:
        app: nginx
    Spec:
      containers:
      - name: nginx
        Image: nginx:1.14.2
        ports:
        - Containers: 80
```

3) Now, use below Commands to create deployment objects and Check the followup Commands as well.

```
# kubectl create -f <deploy_name>.yml
# kubectl get deploy
# kubectl get rs
```

4> What if we have given wrong image tag/ Image name then we don't need to worry kubernetes automatically do checks and only if new pod is up and running then it will delete older pods. check below I have given wrong version.

```
# kubectl set image deployment.v1.apps/nginx-depolyment
  nginx=nginx:1.16.2 deployment.apps/nginx-deployment
  image updated.

# kubectl get deploy

# kubectl get rs
```

5> Updating the deployment image with correct version using below command.
(check 1-pod is having error)

```
# kubectl get pods.

# kubectl set image deployment.v1.app/nginx-deployment
  nginx=nginx:1.16.1 deployment.apps/nginx-deployment
  image updated.

# kubectl get deploy.

# kubectl get rs
```

6) Rolling back to previous version can be done Seamlessly.

```
# kubectl rollout undo deployment/nginx-deployment
deployment.apps/nginx-deployment rolled back.
```

```
# kubectl get rs.
```

7) Delete the deployment and boom all Gone!

Replicaset, pods, Deployments all Gone ✓

```
# kubectl delete depolyment nginx-deployment
deployment.apps "nginx-deployment" deleted
```

8) For minikube Cleanup all System resources.

```
# minikube delete --all
```