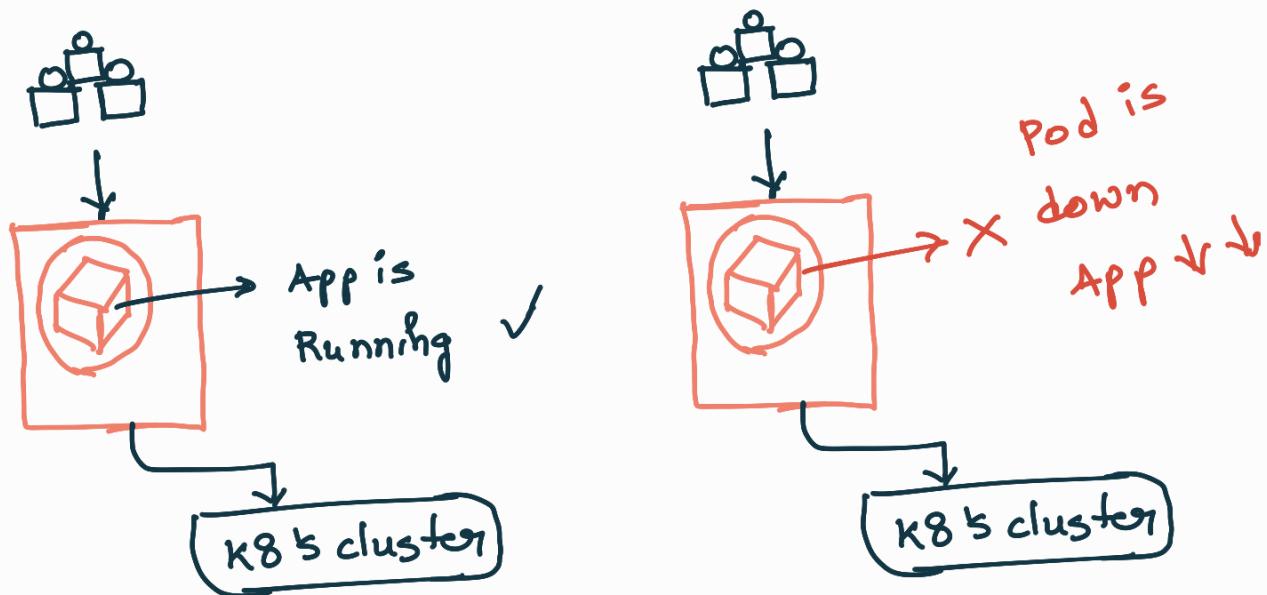


KubernetesIn30Days challenge :-

#Day8 :-

A Pod is running on a single Node in K8 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 K8 supports self-healing. Here K8 do self healing / scaling is done through Replicaset.

Replicaset :- 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:
  # modify replicas according to your case
  replicas: 3
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
        - name: php-redis
          image: gcr.io/google_samples/gb-frontend:v3
```

Basic Kubectl commands :-

kubectl create -f replicaset.yaml

to create any k8 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

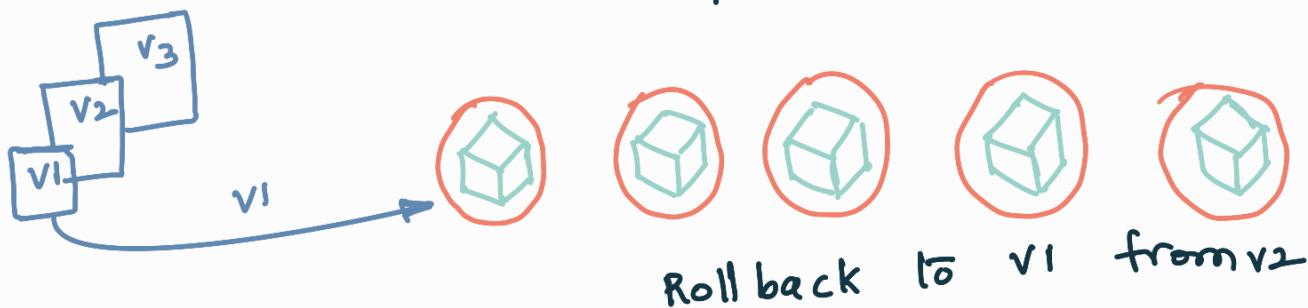
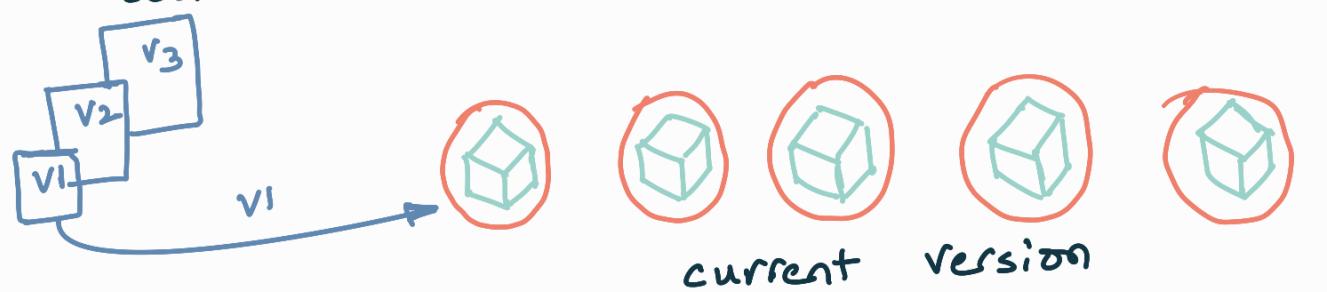
to scale down the replicas to 1.

kubectl delete rs/frontend

to delete frontend Replicaset.

Deployment:-

- This object in k8 provides upgrade/ Rollback/ changes gracefully in cluster.
- Deployment manages Replicaset and Replicaset controls the No. of Pods to be running.



→ This can be done easily through deployment object in kubernetes.

Let's Do small handson on RS & Deployment:-

- ① I am using minikube cluster. you can setup any kops/ EKS/ AKS K8 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:
            - containerPort: 80
```

deploy.yaml [unix] (18:03 16/12/2023)

③ Now, use below command to create deployment object and check the followup commands as well.

```
betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl create -f deploy.yaml
deployment.apps/nginx-deployment created
```

```
betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   0/3       3           0           8s
```

```
betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
nginx-deployment-86dcfdf4c6   3         3         2       16s
```

④ what if we have given wrong image tag / img name then we don't need to worry k8 automatically do checks and only if new pod is up & running then it'll delete older pods. check below I've given wrong version

```
betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.16.2
deployment.apps/nginx-deployment image updated

betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3      1            3          4m12s

betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl get rs
NAME                           DESIRED   CURRENT   READY   AGE
nginx-deployment-86c7fc5d7d     1          1          0       15s
nginx-deployment-86dcfdf4c6     3          3          3       4m20s
```

⑤ Updating the deployment image with correct version using below cmd: (check 1-pod is having Error)

```
betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
nginx-deployment-86c7fc5d7d-75vcc  0/1     ErrImagePull  0          72s
nginx-deployment-86dcfdf4c6-2fm7t  1/1     Running   0          5m17s
nginx-deployment-86dcfdf4c6-nnj7b  1/1     Running   0          5m17s
nginx-deployment-86dcfdf4c6-q18tp  1/1     Running   0          5m17s

betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.16.1
deployment.apps/nginx-deployment image updated

betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3      1            3          5m57s

betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl get rs
NAME                           DESIRED   CURRENT   READY   AGE
nginx-deployment-848dd6cfb5     1          1          0       9s
nginx-deployment-86c7fc5d7d     0          0          0       117s
nginx-deployment-86dcfdf4c6     3          3          3       6m2s
```

⑥ Rolling back to previous version can be done seamlessly.

```
betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl rollout undo deployment/nginx-deployment
deployment.apps/nginx-deployment rolled back
```

```
betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
nginx-deployment-848dd6cfb5   2         2         2       2m17s
nginx-deployment-86dcfdf4c6   2         2         1       8m10s
```

```
betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
nginx-deployment-848dd6cfb5   0         0         0       2m23s
nginx-deployment-86dcfdf4c6   3         3         3       8m16s
```

⑦ delete the deployment and Boom All Gone!

Replicaset, Pods, Deployments all gone ✓

```
betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl get deploy
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3      3           3          9m17s
```

```
betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl delete deployment nginx-deployment
deployment.apps "nginx-deployment" deleted
```

```
betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl get deploy
No resources found in default namespace.
```

```
betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl get rs
No resources found in default namespace.
```

```
betha@Rajasekhar_PC MINGW64 ~/OneDrive/Desktop/kubernetes_practice
$ kubectl get pods
No resources found in default namespace.
```

⑧ for minikube cleanup all system resources

minikube delete --all