

Deployment

- ❖ A Deployment in Kubernetes manages ReplicaSets to ensure the desired state of applications.
- ❖ It provides mechanisms for updating applications, such as rolling updates and rollbacks, ensuring minimal downtime.
- ❖ Deployments are used for declaratively managing application states and scaling pod replicas efficiently.

Imperative Methods

- ❖ Imperative commands in Kubernetes are used to directly manage cluster resources through the kubectl command-line tool.

Declarative Methods

- ❖ Declarative management in Kubernetes involves defining the desired state of resources in configuration files (YAML or JSON).

Key Components of Deployments:

- ❖ **Pod Template:** Specifies the configuration of the pods, including container images, labels, and other settings.
- ❖ **ReplicaSet:** Ensures that a specified number of pod replicas are running at all times.
- ❖ **Strategy:** Defines how updates to the Deployment should be performed, such as rolling updates or recreations.
- ❖ **Labels and Selectors:** Used to identify and manage the pods that belong to the Deployment.
- ❖ **Revision History:** Keeps track of previous states of the Deployment for rollback purposes.

Step: 1

Create a Basic Deployment

vi deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: httpd-deploy
spec:
  replicas: 2
  selector:
    matchLabels:
      app: httpd
  template:
    metadata:
      labels:
        app: httpd
    spec:
      containers:
      - name: httpd
        image: httpd:2.4
```

Step: 2

To create the pods

```
controlplane $ kubectl create -f depoly.yaml
deployment.apps/httpd-deploy created
```

Step: 3

To verify the pods

```
controlplane $ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
httpd-deploy-7b64654c86-9fjl9      1/1     Running   0           65s
httpd-deploy-7b64654c86-p9rdb      1/1     Running   0           65s
```

Step:4

To verify kubectl get deployments.apps

```
controlplane $ kubectl get deployments.apps
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
httpd-deploy  2/2     2             2           85s
```

Step:5

To verify the logs

```
controlplane $ kubectl logs deployments/httpd-deploy -f
Found 2 pods, using pod/httpd-deploy-7b64654c86-p9rdb
```

Step:6

To verify the version

```
controlplane $ kubectl describe deployment/httpd-deploy |grep Image
Image:          httpd:2.4
```

Step:7

To verify the history

```
controlplane $ kubectl rollout history deployment httpd-deploy
deployment.apps/httpd-deploy
REVISION  CHANGE-CAUSE
1          <none>
```

Step: 8

To verify the revision

```
controlplane $ kubectl rollout history deployment httpd-deploy --revision 1
deployment.apps/httpd-deploy with revision #1
Pod Template:
  Labels:      app=httpd
              pod-template-hash=7b64654c86
  Containers:
    httpd:
      Image:      httpd:2.4
      Port:       <none>
      Host Port:  <none>
      Environment:  <none>
      Mounts:       <none>
  Volumes:       <none>
  Node-Selectors:  <none>
  Tolerations:    <none>
```

Updating images

Step: 1

`kubectl set image deployment/httpd-deploy httpd=httpd:2.4.64`

```
controlplane $ kubectl set image deployment/httpd-deploy httpd=httpd:2.4.64
deployment.apps/httpd-deploy image updated
```

Step: 2

To verify the check of version

```
controlplane $ kubectl describe deployment/httpd-deploy | grep Image
Image:      httpd:2.4.64
```

Step: 3

To kubectl rollout history and status

```
^Ccontrolplane $ kubectl rollout history deployment/httpd-deploy
deployment.apps/httpd-deploy
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
```

Step: 4

To rollback

kubectl rollout undo deployment/httpd-deploy --to-revision=1

```
controlplane $ kubectl rollout undo deployment/httpd-deploy --to-revision=1
deployment.apps/httpd-deploy rolled back
```

To verify the version

```
controlplane $ kubectl describe deployments/httpd-deploy |grep Image
Image:          httpd:2.4
```

To get rollback to older version

kubectl rollout undo deployment httpd-deploy --to-revision=3

```
controlplane $ kubectl describe deployments/httpd-deploy |grep Image
Image:          httpd:2.4
controlplane $ kubectl rollout undo deployment httpd-deploy --to-revision=4
deployment.apps/httpd-deploy rolled back
controlplane $ kubectl describe deployments/httpd-deploy |grep Image
Image:          httpd:2.4.64
```

Scaling Process

To verify the status

```
controlplane $ kubectl scale deployment httpd-deploy --replicas=5
deployment.apps/httpd-deploy scaled
controlplane $ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
httpd-deploy-7b64654c86-9ndxt	1/1	Running	0	3m58s
httpd-deploy-7b64654c86-bq95g	1/1	Running	0	3m58s
httpd-deploy-7b64654c86-d4lk9	1/1	Running	0	3s
httpd-deploy-7b64654c86-dzn8h	1/1	Running	0	3m58s
httpd-deploy-7b64654c86-l2p2m	1/1	Running	0	3s

Task

Update Deployment Image

Update the nginx image version in the "nginx-deploy" Deployment to nginx:1.17.9.

Step:1

vi deploy.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deploy
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.17.9
```


Step: 2

To create the pods

```
controlplane $ kubectl create -f deploy.yaml
deployment.apps/nginx-deploy created
```

Step:3

To check the pods

Command is kubectl get pods or kubectl get deployments.apps

```
controlplane $ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deploy-7f98bf575b-62hf9       1/1     Running   0           11s
nginx-deploy-7f98bf575b-jgzqc       1/1     Running   0           11s
controlplane $ kubectl get deployments.apps
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deploy  2/2     2             2           2m42s
```

Step: 4

To verify the details

```
controlplane $ kubectl describe deployments.apps nginx-deploy
Name:          nginx-deploy
Namespace:     default
CreationTimestamp: Sat, 27 Jul 2024 02:49:37 +0000
Labels:        <none>
Annotations:   deployment.kubernetes.io/revision: 1
Selector:      app=nginx
Replicas:      2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
    nginx:
      Image:      nginx:1.17.9
      Port:       <none>
      Host Port:  <none>
      Environment: <none>
      Mounts:      <none>
  Volumes:      <none>
  Node-Selectors: <none>
  Tolerations:  <none>
```

Conditions:

Type	Status	Reason
Available	True	MinimumReplicasAvailable
Progressing	True	NewReplicaSetAvailable

OldReplicaSets: <none>

NewReplicaSet: nginx-deploy-7f98bf575b (2/2 replicas created)

Events:

Type	Reason	Age	From	Message
Normal	ScalingReplicaSet	3m15s	deployment-controller	Scaled up replica set nginx-deploy-7f98bf575b to 2