

Deployment

Introduction:

A deployment is a resource object in Kubernetes that provides declarative updates to applications. It allows us to describe an application's life cycle, such as which images to use for the app, the number of pods there should be, and the way in which they should be updated.

Objectives:

1. Create a deployment using Imperative command
2. Create a deployment using YAML file
3. Increase the number of pods in a deployment
4. Decrease the number of pods in a deployment
5. Set the image in a deployment
6. Delete a deployment

1. Create a deployment using Imperative command:

If we create a deployment, by default it will create only one pod unless we provide the number of replicas.

Use the below command to create a deployment.

```
kubectl create deploy prod-deploy --image nginx --replicas=4
```

The above command will create a deployment named prod-deploy having 4 pods with **nginx** image. Check the output below.

```
kubectl get deploy
```

Below we can see that our deployment has been created with 4 pods.

```
root@master:~# kubectl get deploy
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
prod-deploy   4/4     4            4           3m33s
```

```
root@master:~# kubectl create deploy prod-deploy --image nginx --replicas=4
deployment.apps/prod-deploy created
root@master:~#
root@master:~# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
prod-deploy-5b9d84c685-28572        1/1     Running   0           8s
prod-deploy-5b9d84c685-fc886        1/1     Running   0           8s
prod-deploy-5b9d84c685-ph2c7        1/1     Running   0           8s
prod-deploy-5b9d84c685-rbj9x        1/1     Running   0           8s
```

Describe the deployment to get more details.

```
kubectl describe deploy prod-deploy
```

```
root@master:~# kubectl describe deploy prod-deploy
Name:                prod-deploy
Namespace:           default
CreationTimestamp:    Wed, 11 Jan 2023 11:13:06 +0000
Labels:              app=prod-deploy
Annotations:         deployment.kubernetes.io/revision: 1
Selector:             app=prod-deploy
Replicas:            4 desired | 4 updated | 4 total | 4 available | 0 unavailable
StrategyType:        RollingUpdate
MinReadySeconds:     0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=prod-deploy
  Containers:
    nginx:
      Image:      nginx
      Port:       <none>
      Host Port:  <none>
      Environment: <none>
      Mounts:     <none>
      Volumes:    <none>
  Conditions:
    Type           Status  Reason
    ----           -
    Available       True    MinimumReplicasAvailable
    Progressing     True    NewReplicaSetAvailable
  OldReplicaSets:  <none>
  NewReplicaSet:   prod-deploy-5b9d84c685 (4/4 replicas created)
  Events:
    Type           Reason             Age           From              Message
    ---           -
    Normal         ScalingReplicaSet   5m49s        deployment-controller Scaled up replica set prod-deploy-5b9d84c685 to 4
```

```
kubectl get all
```

Above command will show us what resources we have after creating this deployment.

We can see below that the single imperative command for deployment creation has created a deployment, a replicaset and 4 pods.

```
root@master:~# kubectl get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/prod-deploy-5b9d84c685-28572        1/1     Running   0           14m
pod/prod-deploy-5b9d84c685-fc886        1/1     Running   0           14m
pod/prod-deploy-5b9d84c685-ph2c7        1/1     Running   0           14m
pod/prod-deploy-5b9d84c685-rbj9x        1/1     Running   0           14m

NAME                                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes                  ClusterIP     10.96.0.1    <none>        443/TCP    3d1h

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/prod-deploy         4/4     4             4           14m

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/prod-deploy-5b9d84c685 4         4         4       14m
```

2. Create a deployment using YAML file:

We can either create a deployment template using the imperative command or we can use a YAML definition file to create a deployment. Use the below command to create a deployment template.

```
kubectl create deploy dev-deploy --image httpd --replicas=3 --dry-run=client -o yaml > dev-deploy.yaml
```

Above command will not implement the deployment instead it will give the output in yaml format and will save it in dev-deploy.yaml. This file can be used to make further changes to create a deployment.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: dev-deploy
    env: development
  name: dev-deploy
spec:
  replicas: 3
  selector:
    matchLabels:
      app: dev-deploy
  template:
    metadata:
      labels:
        app: dev-deploy
    spec:
      containers:
        - image: httpd
          name: httpd-container
```

above we have created a Yaml file and we can use this file to create a deployment having 3 pods with httpd image.

Use the apply command and mention the definition file to create the deployment.

```
kubectl apply -f dev-deploy.yaml
```

See the output below, now we have 2 deployments in our cluster.

```

root@master:~# kubectl get deploy
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
dev-deploy    3/3     3             3           77s
prod-deploy   4/4     4             4           30m
root@master:~#
root@master:~#
root@master:~# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
dev-deploy-57b86d5ccc-4gwbc        1/1     Running   0          83s
dev-deploy-57b86d5ccc-cmcl2        1/1     Running   0          83s
dev-deploy-57b86d5ccc-cv7m5        1/1     Running   0          83s
prod-deploy-5b9d84c685-28572        1/1     Running   0          31m
prod-deploy-5b9d84c685-fc886        1/1     Running   0          31m
prod-deploy-5b9d84c685-ph2c7        1/1     Running   0          31m
prod-deploy-5b9d84c685-rbj9x        1/1     Running   0          31m

```

3. Increase the number of pods in a deployment:

Use the below command to increase the number of pods.

```
kubectl scale deploy prod-deploy --replicas=6
```

See the output below. The number of pods have reached to 6 now.

```

root@master:~# kubectl scale deploy prod-deploy --replicas=6
deployment.apps/prod-deploy scaled
root@master:~# kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
dev-deploy    3/3     3             3           9m55s
prod-deploy   6/6     6             6           39m
root@master:~# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
dev-deploy-57b86d5ccc-4gwbc        1/1     Running   0          10m
dev-deploy-57b86d5ccc-cmcl2        1/1     Running   0          10m
dev-deploy-57b86d5ccc-cv7m5        1/1     Running   0          10m
prod-deploy-5b9d84c685-28572        1/1     Running   0          39m
prod-deploy-5b9d84c685-5r9tt        1/1     Running   0          24s
prod-deploy-5b9d84c685-fc886        1/1     Running   0          39m
prod-deploy-5b9d84c685-ph2c7        1/1     Running   0          39m
prod-deploy-5b9d84c685-rbj9x        1/1     Running   0          39m
prod-deploy-5b9d84c685-wnqd5        1/1     Running   0          24s

```

Or we can use the edit command to increase the number of pods.

```
kubectl edit deploy prod-deploy
```

Below we have changed the number of replicas to 6 and as soon as we save and exit the file, it will create 2 more pods.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: "2023-01-11T11:13:06Z"
  generation: 3
  labels:
    app: prod-deploy
  name: prod-deploy
  namespace: default
  resourceVersion: "111057"
  uid: dc471493-1ae7-41ea-bcf7-cd483134f8bf
spec:
  progressDeadlineSeconds: 600
  replicas: 6
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: prod-deploy
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: prod-deploy
    spec:
      containers:
      - image: nginx
        imagePullPolicy: Always
        name: nginx
        resources: {}
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler

```

4. Decrease the number of pods in a deployment:

Here also we can use the scale command and the edit command to get the job done.

```
kubectl scale deploy prod-deploy --replicas=3
```

Below we can see the pods are terminating and finally numbers came down to 3. Same way we can use the edit command to decrease the number of pods.

```
root@master:~# kubectl get deploy
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
dev-deploy    3/3     3            3           21m
prod-deploy   6/6     6            6           51m
root@master:~#
root@master:~# kubectl scale deploy prod-deploy --replicas=3
deployment.apps/prod-deploy scaled
root@master:~# kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
dev-deploy-57b86d5ccc-4gwbc         1/1     Running            0          22m
dev-deploy-57b86d5ccc-cmcl2         1/1     Running            0          22m
dev-deploy-57b86d5ccc-cv7m5         1/1     Running            0          22m
prod-deploy-5b9d84c685-28572        1/1     Running            0          51m
prod-deploy-5b9d84c685-dhhc4        0/1     Terminating      0          5m20s
prod-deploy-5b9d84c685-fc886        1/1     Running            0          51m
prod-deploy-5b9d84c685-ph2c7        1/1     Running            0          51m
prod-deploy-5b9d84c685-rbj9x        0/1     Terminating      0          51m
root@master:~# kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
dev-deploy-57b86d5ccc-4gwbc         1/1     Running            0          22m
dev-deploy-57b86d5ccc-cmcl2         1/1     Running            0          22m
dev-deploy-57b86d5ccc-cv7m5         1/1     Running            0          22m
prod-deploy-5b9d84c685-28572        1/1     Running            0          52m
prod-deploy-5b9d84c685-fc886        1/1     Running            0          52m
prod-deploy-5b9d84c685-ph2c7        1/1     Running            0          52m
root@master:~#
root@master:~# kubectl get deploy
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
dev-deploy    3/3     3            3           22m
prod-deploy   3/3     3            3           52m
```

5. Set the image in a deployment:

We can use set image command or edit command to change the image of the pods which are a part of a deployment.

Use the below command to apply a specific image.

```
kubectl set image deploy prod-deploy nginx=nginx:1.23.3
```

Below we can see the image has been updated as soon as we execute the above command.

We can also edit the deployment using the below mentioned command and make the changes in the image field to get the result.

```
kubectl edit deploy prod-deploy
```

```

root@master:~# kubectl set image deploy prod-deploy nginx=nginx:1.23.3
deployment.apps/prod-deploy image updated
root@master:~# ^C
root@master:~# kubectl describe deployment prod-deploy
Name:                prod-deploy
Namespace:            default
CreationTimestamp:    Wed, 11 Jan 2023 11:13:06 +0000
Labels:               app=prod-deploy
Annotations:          deployment.kubernetes.io/revision: 2
Selector:             app=prod-deploy
Replicas:             3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:         RollingUpdate
MinReadySeconds:      0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=prod-deploy
  Containers:
    nginx:
      Image:   nginx:1.23.3
      Port:    <none>
      Host Port: <none>
      Environment: <none>
      Mounts:    <none>
      Volumes:   <none>
  Conditions:
    Type           Status  Reason
    ----           -
    Available       True    MinimumReplicasAvailable
    Progressing     True    NewReplicaSetAvailable
  OldReplicaSets:  <none>
  NewReplicaSet:   prod-deploy-86d66bfd5b (3/3 replicas created)
Events:
  Type           Reason             Age             From              Message
  ----           -
  Normal         ScalingReplicaSet   59m             deployment-controller Scaled up replica set prod-deploy-5b9d84c685 to 4
  Normal         ScalingReplicaSet   19m             deployment-controller Scaled down replica set prod-deploy-5b9d84c685 to 4
  Normal         ScalingReplicaSet   12m (x2 over 20m) deployment-controller Scaled up replica set prod-deploy-5b9d84c685 to 6
  Normal         ScalingReplicaSet   7m34s           deployment-controller Scaled down replica set prod-deploy-5b9d84c685 to 3
  Normal         ScalingReplicaSet   48s             deployment-controller Scaled up replica set prod-deploy-86d66bfd5b to 1
  Normal         ScalingReplicaSet   46s             deployment-controller Scaled down replica set prod-deploy-5b9d84c685 to 2
  Normal         ScalingReplicaSet   46s             deployment-controller Scaled up replica set prod-deploy-86d66bfd5b to 2
  Normal         ScalingReplicaSet   43s             deployment-controller Scaled down replica set prod-deploy-5b9d84c685 to 1

```

6. Delete a deployment:

Let's first delete a pod which is a part of a deployment and we will see if the pod is deleted or not.

```
kubectl delete pod prod-deploy-86d66bfd5b-djbdr
```

Using the above command, we are deleting a pod which is a part of prod-deploy deployment. The pod will be deleted but a new pod will be recreated to maintain the number of replicas we have specified in **prod-deploy** deployment

See the output below.

```

root@master:~# kubectl get deploy
NAME          READY    UP-TO-DATE    AVAILABLE    AGE
dev-deploy    3/3      3              3             37m
prod-deploy    3/3      3              3             67m
root@master:~#
root@master:~# kubectl get pods
NAME                                READY    STATUS    RESTARTS    AGE
dev-deploy-57b86d5ccc-4gwbc        1/1      Running   0            37m
dev-deploy-57b86d5ccc-cmcl2        1/1      Running   0            37m
dev-deploy-57b86d5ccc-cv7m5        1/1      Running   0            37m
prod-deploy-86d66bfd5b-djbdr        1/1      Running   0            8m59s
prod-deploy-86d66bfd5b-mgtx9        1/1      Running   0            8m57s
prod-deploy-86d66bfd5b-spgcw        1/1      Running   0            8m54s
root@master:~#
root@master:~# kubectl delete pod prod-deploy-86d66bfd5b-djbdr
pod "prod-deploy-86d66bfd5b-djbdr" deleted
root@master:~#
root@master:~# kubectl get pods
NAME                                READY    STATUS    RESTARTS    AGE
dev-deploy-57b86d5ccc-4gwbc        1/1      Running   0            38m
dev-deploy-57b86d5ccc-cmcl2        1/1      Running   0            38m
dev-deploy-57b86d5ccc-cv7m5        1/1      Running   0            38m
prod-deploy-86d66bfd5b-kkdjl        1/1      Running   0            10s
prod-deploy-86d66bfd5b-mgtx9        1/1      Running   0            9m23s
prod-deploy-86d66bfd5b-spgcw        1/1      Running   0            9m20s

```

Let's delete the deployment now and we will see the pods and replicaset will also be deleted.

Use the below command to delete the deployment.

```

kubectl delete deploy prod-deploy
kubectl delete deploy dev-deploy

```

See the output below.


```

root@master:~# kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/dev-deploy-57b86d5ccc-4gwbc     1/1     Running   0           46m
pod/dev-deploy-57b86d5ccc-cmcl2     1/1     Running   0           46m
pod/dev-deploy-57b86d5ccc-cv7m5     1/1     Running   0           46m
pod/prod-deploy-86d66bfd5b-kkdjl    1/1     Running   0           7m47s
pod/prod-deploy-86d66bfd5b-mgtx9    1/1     Running   0           17m
pod/prod-deploy-86d66bfd5b-spgcw    1/1     Running   0           16m

NAME                                TYPE                      CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes                  ClusterIP       10.96.0.1    <none>        443/TCP    3d2h

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/dev-deploy          3/3     3             3           46m
deployment.apps/prod-deploy         3/3     3             3           75m

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/dev-deploy-57b86d5ccc 3         3         3       46m
replicaset.apps/prod-deploy-5b9d84c685 0         0         0       75m
replicaset.apps/prod-deploy-86d66bfd5b 3         3         3       17m
root@master:~#
root@master:~#
root@master:~# kubectl delete deploy prod-deploy
deployment.apps "prod-deploy" deleted
root@master:~# kubectl delete deploy dev-deploy
deployment.apps "dev-deploy" deleted
root@master:~#
root@master:~# kubectl get deploy
No resources found in default namespace.
root@master:~# kubectl get pods
No resources found in default namespace.
root@master:~#
root@master:~# kubectl get rs
No resources found in default namespace.

```