Lab: Blue-Green Deployment

Introduction:

Blue/Green deployments are a form of progressive delivery where a new version of the application is deployed while the old version still exists. The two versions coexist for a brief period of time while user traffic is routed to the new version, before the old version is discarded.

Objectives:

- **Create blue-deployment**
- **Create blue-deployment service**
- **Create green-deployment**
- **Create green-deployment service**
- **Cleanup**

Ensure that you have logged-in as **root** user on **eoc-controller** node.

1. Creating Blue Deployment

1.1 Let's **view** the yaml manifest file.

```
# cat -n ~/kubernetes/deployment-bluegreen-blue.yml
```

Output:

```
[root@eoc-controller ~]#cat -n ~/kubernetes/deployment-bluegreen-blue.yml
     1  apiVersion: apps/v1
     2  kind: Deployment
     3  metadata:
     4   name: web-app-blue
     5   labels:
     6     app: webserver
     7  spec:
     8   replicas: 4
     9   selector:
    10     matchLabels:
    11       app: webserver
    12       ver: blue
    13   template:
    14     metadata:
    15       labels:
    16         app: webserver
    17         ver: blue
    18     spec:
    19       containers:
    20       - name: webserver-container
    21         image: eyesoncloud/web-app:v1
```

1.2 Let's **create** the deployment using yaml manifest file by executing the below file.

```
# kubectl apply -f ~/kubernetes/deployment-bluegreen-blue.yml
```

Output:

```
[root@eoc-controller ~]#kubectl apply -f ~/kubernetes/deployment-bluegreen-blue.yml
deployment.apps/web-app-blue created
```

1.3 Let's **list** the deployment.

```
# kubectl get deployment
```

Output:

```
[root@eoc-controller ~]#kubectl get deploy
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
web-app-blue   4/4     4            4           2m38s
```

1.4 Let's **describe** the details of Deployment by executing the below command.

```
# kubectl describe deployment web-app
```

Output:

```
[root@eoc-controller ~]#kubectl describe deployment web-app-blue
Name:                   web-app-blue
Namespace:              default
CreationTimestamp:      Wed, 06 Sep 2023 08:09:53 -0400
Labels:                 app=webserver
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=webserver,ver=blue
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=webserver
           ver=blue
  Containers:
   webserver-container:
    Image:        eyesoncloud/web-app:v1
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
```

2. Create blue-deployment service

2.1 Let's view the blue-deployment service yaml by executing below command.

```
# cat -n ~/kubernetes/deployment-bluegreen-service-blue.yml
```

Output:

```
[root@eoc-controller ~]#cat -n ~/kubernetes/deployment-bluegreen-service-blue.yml
     1  apiVersion: v1
     2  kind: Service
     3  metadata:
     4    name: np-service
     5  spec:
     6    type: NodePort
     7    selector:
     8      app: webserver
     9      ver: blue
    10    ports:
    11    - protocol: TCP
    12      nodePort: 30010
    13      port: 80
    14      targetPort: 80
```

2.2 Let's **create** the blue-deployment service by executing below command

```
# kubectl apply -f ~/kubernetes/deployment-bluegreen-service-blue.yml
```

Output:

```
[root@eoc-controller ~]#kubectl apply -f ~/kubernetes/deployment-bluegreen-service-blue.yml
service/np-service created
```

2.3 Let's **list** the service by executing below command.

```
# kubectl get svc np-service
```

Output:

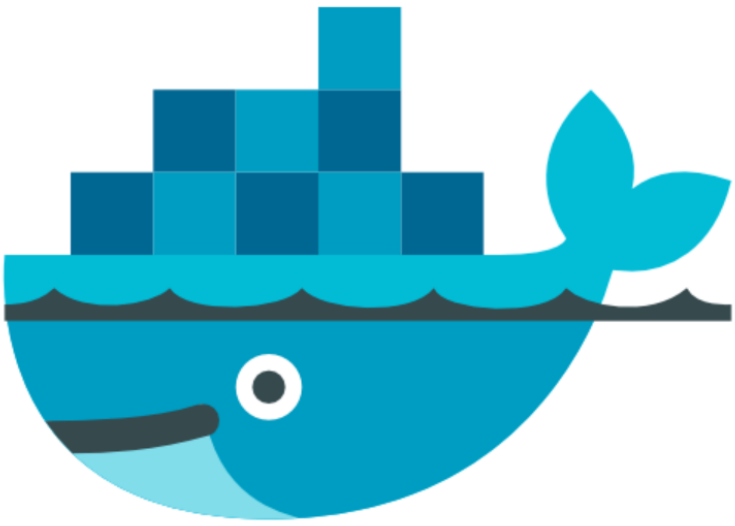
```
[root@eoc-controller ~]#kubectl get svc np-service
NAME         TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
np-service   NodePort   10.110.64.6    <none>        80:30010/TCP     92s
```

2.4 Let's access from the browser.

```
# http://PublicIP:30010
```

Output:

Hello Blue Whale



3. Creating Green Deployment

3.1 Let's **view** the yaml manifest file.

```
# cat -n ~/kubernetes/deployment-bluegreen-green.yml
```

Output:

```
[root@eoc-controller ~]#cat -n ~/kubernetes/deployment-bluegreen-green.yml
     1  apiVersion: apps/v1
     2  kind: Deployment
     3  metadata:
     4   name: web-app-green
     5   labels:
     6     app: webserver
     7  spec:
     8   replicas: 4
     9   selector:
    10     matchLabels:
    11       app: webserver
    12       ver: green
    13   template:
    14     metadata:
    15       labels:
    16         app: webserver
    17         ver: green
    18     spec:
    19       containers:
    20       - name: webserver-container
    21         image: eyesoncloud/web-app:v2
```

3.2 Let's **create** the deployment using yaml manifest file by executing the below file.

```
# kubectl apply -f ~/kubernetes/deployment-bluegreen-green.yml
```

Output:

```
[root@eoc-controller ~]#kubectl apply -f ~/kubernetes/deployment-bluegreen-green.yml
deployment.apps/web-app-green created
```

3.3 Let's **describe** the details of Deployment by executing the below command.

```
# kubectl describe deployment web-app-green
```

Output:

```
[root@eoc-controller ~]#kubectl describe deployment web-app-green
Name:                   web-app-green
Namespace:              default
CreationTimestamp:      Wed, 06 Sep 2023 08:32:56 -0400
Labels:                 app=webserver
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=webserver,ver=green
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=webserver
           ver=green
  Containers:
   webserver-container:
    Image:        eyesoncloud/web-app:v2
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
```

4. Create green-deployment service

4.1 Let's view the green-deployment service yaml by executing below command.

```
# cat -n ~/kubernetes/deployment-bluegreen-service-green.yml
```

Output:

```
[root@eoc-controller ~]#cat -n ~/kubernetes/deployment-bluegreen-service-green.yml
     1  apiVersion: v1
     2  kind: Service
     3  metadata:
     4    name: np-service
     5  spec:
     6    type: NodePort
     7    selector:
     8      app: webserver
     9      ver: green
    10    ports:
    11    - protocol: TCP
    12      nodePort: 30010
    13      port: 80
    14      targetPort: 80
```

4.2 Let's **Create** the green-deployment service by executing below command.

```
# kubectl apply -f ~/kubernetes/deployment-bluegreen-service-
green.yml
```

Output:

```
[root@eoc-controller ~]#kubectl apply -f ~/kubernetes/deployment-bluegreen-service-green.yml
service/np-service configured
```

4.3 Let's **list** the service by executing below command.

```
# kubectl get svc np-service
```

Output:

```
[root@eoc-controller ~]#kubectl get svc np-service
NAME         TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
np-service   NodePort   10.110.64.6    <none>        80:30010/TCP   22m
```

4.4 Let's access from the browser.

```
# http://PublicIP:30010
```

Output:

5. Cleanup.

5.1 Let's **delete** the blue deployment and notice that it deletes the pods also.

```
# kubectl delete -f ~/kubernetes/deployment-bluegreen-blue.yml
```

Output:

```
[root@eoc-controller ~]#kubectl delete -f ~/kubernetes/deployment-bluegreen-blue.yml
deployment.apps "web-app-blue" deleted
```

5.2 Let's **delete** the green deployment and notice that it deletes the pods also

```
# kubectl delete -f ~/kubernetes/deployment-bluegreen-
green.yml
```

Output:

```
[root@eoc-controller ~]#kubectl delete -f ~/kubernetes/deployment-bluegreen-green.yml
deployment.apps "web-app-green" deleted
```

5.3 Let's **delete** the service by executing below command.

```
# kubectl delete svc np-service
```

Output:

```
[root@eoc-controller ~]#kubectl delete svc np-service
service "np-service" deleted
```