Lesson 04 Demo 08

Deploying Image Versions Using Rollout

Objective: To demonstrate the process of creating Kubernetes deployments, upgrading image versions, and reverting to previous versions, facilitating effective application deployment and version control

Tools required: kubeadm, kubectl, kubelet, and containerd

Prerequisites: A Kubernetes cluster should already be set up (refer to the steps provided in Lesson 02, Demo 01 for guidance).

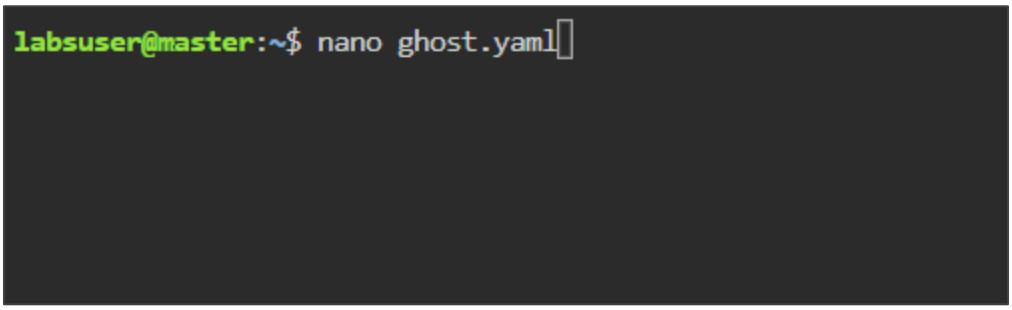
Steps to be followed:

1. Create the Kubernetes deployment
2. Upgrade the image version
3. Switch back to the old version

Step 1: Create the Kubernetes deployment

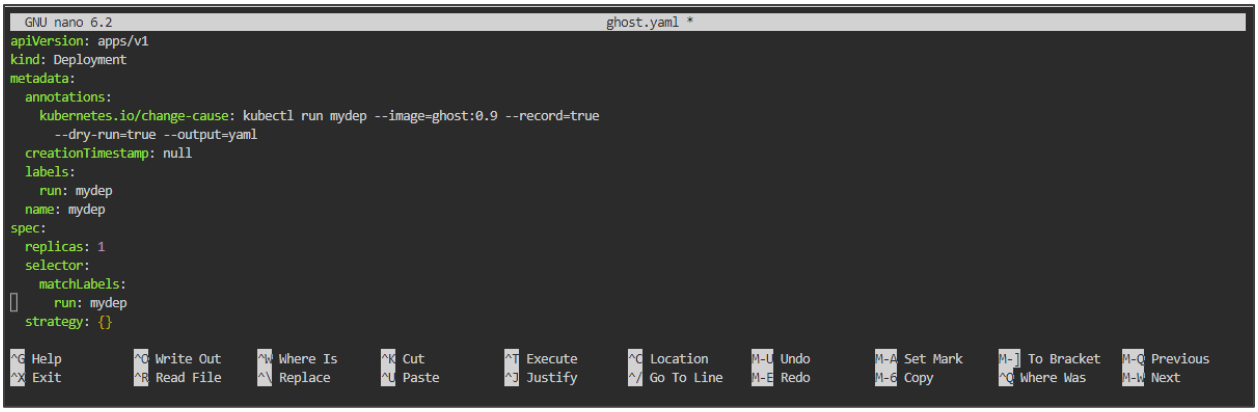
1.1 Create the YAML file by using the following command:

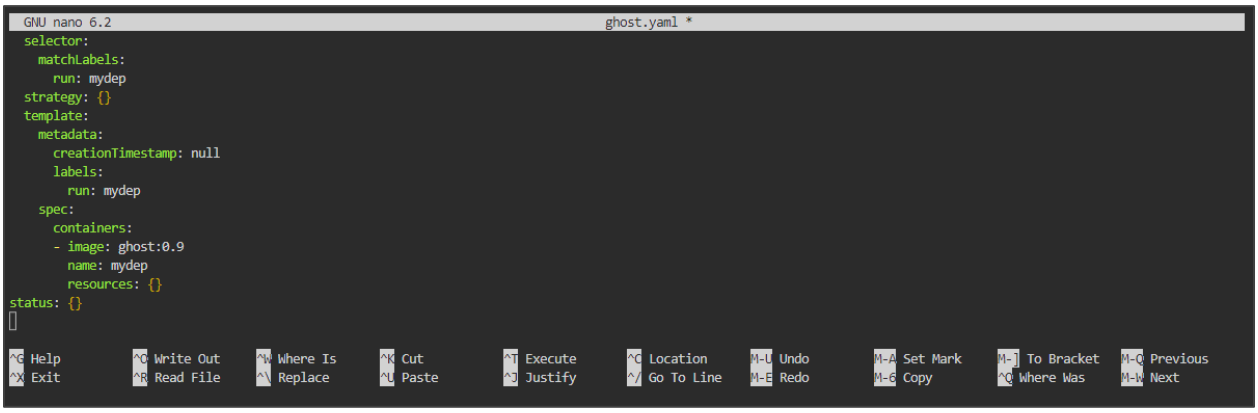
nano ghost.yaml



1.2 Add the following code to the **ghost.yaml** file:

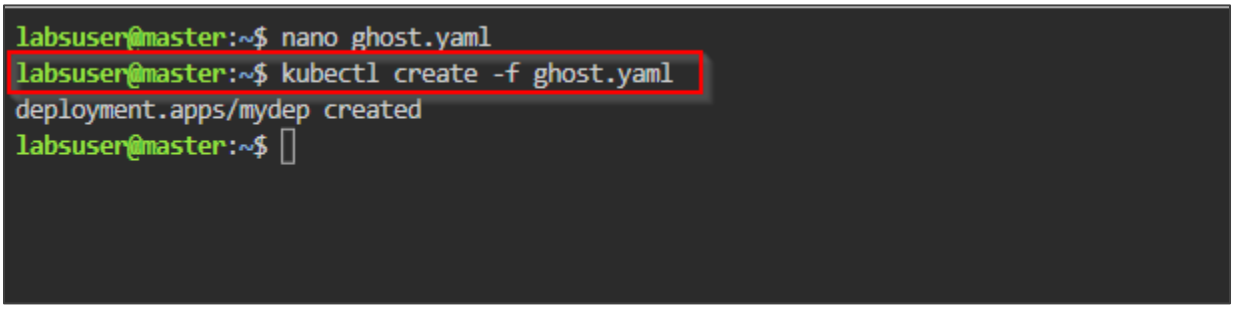
```
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    kubernetes.io/change-cause: kubectl run mydep --image=ghost:0.9 --record=true
      --dry-run=true --output=yaml
  creationTimestamp: null
  labels:
    run: mydep
  name: mydep
spec:
  replicas: 1
  selector:
    matchLabels:
      run: mydep
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        run: mydep
    spec:
      containers:
      - image: ghost:0.9
        name: mydep
        resources: {}
status: {}
```





1.3 Create the deployment resource by using the following command:

kubectl create -f ghost.yaml



1.4 Verify the deployment by using the following command:

kubectl get deployment

```
labsuser@master:~$ nano ghost.yaml
labsuser@master:~$ kubectl create -f ghost.yaml
deployment.apps/mydep created
labsuser@master:~$ kubectl get deployment
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
mydep   0/1     1            0           2m9s
labsuser@master:~$
```

The deployment has been successfully created.

Step 2: Upgrade the image version

2.1 Verify the deployment rollout history by using the following command:

kubectl rollout history deployment/mydep

```
labsuser@master:~$ kubectl get deployment
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
mydep   0/1     1            0           2m9s
labsuser@master:~$ kubectl rollout history deployment/mydep
deployment.apps/mydep
REVISION  CHANGE-CAUSE
1         kubectl run mydep --image=ghost:0.9 --record=true --dry-run=true --output=yaml

labsuser@master:~$
```

2.2 Upgrade the deployment image version to **0.10** by using the following command:

kubectl set image deployment/mydep mydep=ghost:0.10 --record

```
labsuser@master:~$ kubectl rollout history deployment/mydep
deployment.apps/mydep
REVISION  CHANGE-CAUSE
1         kubectl run mydep --image=ghost:0.9 --record=true --dry-run=true --output=yaml

labsuser@master:~$ kubectl set image deployment/mydep mydep=ghost:0.10 -record
error: all resources must be specified before image changes: -record
labsuser@master:~$ kubectl set image deployment/mydep mydep=ghost:0.10 --record
Flag --record has been deprecated, --record will be removed in the future
deployment.apps/mydep image updated
labsuser@master:~$
```

2.3 Verify the deployment rollout history by using the following command:

kubectl rollout history deployment/mydep

```
labsuser@master:~$ kubectl set image deployment/mydep mydep=ghost:0.10 --record
Flag --record has been deprecated, --record will be removed in the future
deployment.apps/mydep image updated
labsuser@master:~$ kubectl rollout history deployment/mydep
deployment.apps/mydep
REVISION  CHANGE-CAUSE
1         kubectl run mydep --image=ghost:0.9 --record=true --dry-run=true --output=yaml
2         kubectl set image deployment/mydep mydep=ghost:0.10 --record=true

labsuser@master:~$
```

The image version of the deployment has been upgraded to **0.10**.

Step 3: Switch back to the old version

3.1 Execute the following command to revert to the initial version of deployment:

kubectl rollout undo deployment/mydep --to-revision=1

```
labsuser@master:~$ kubectl rollout history deployment/mydep
deployment.apps/mydep
REVISION  CHANGE-CAUSE
1         kubectl run mydep --image=ghost:0.9 --record=true --dry-run=true --output=yaml
2         kubectl set image deployment/mydep mydep=ghost:0.10 --record=true

labsuser@master:~$ kubectl rollout undo deployment/mydep --to-revision=1
deployment.apps/mydep rolled back
labsuser@master:~$
```

3.2 Verify the deployment rollout history by using the following command:

kubectl rollout history deployment/mydep

```
labsuser@master:~$ kubectl rollout history deployment/mydep
deployment.apps/mydep
REVISION  CHANGE-CAUSE
1         kubectl run mydep --image=ghost:0.9 --record=true --dry-run=true --output=yaml
2         kubectl set image deployment/mydep mydep=ghost:0.10 --record=true

labsuser@master:~$ kubectl rollout undo deployment/mydep --to-revision=1
deployment.apps/mydep rolled back
labsuser@master:~$ kubectl rollout history deployment/mydep
deployment.apps/mydep
REVISION  CHANGE-CAUSE
2         kubectl set image deployment/mydep mydep=ghost:0.10 --record=true
3         kubectl run mydep --image=ghost:0.9 --record=true --dry-run=true --output=yaml

labsuser@master:~$
```

The deployment image version has been returned to its original state.

By following these steps, you have successfully completed the process of creating Kubernetes deployments and version control within Kubernetes clusters.