Lesson 02 Demo 05

Using Basic Commands of Kubernetes

Objective: To demonstrate the fundamental Kubernetes commands for managing resources, including deployment creation, namespace management, scaling, deployment deletion, and enhancing cluster management

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).

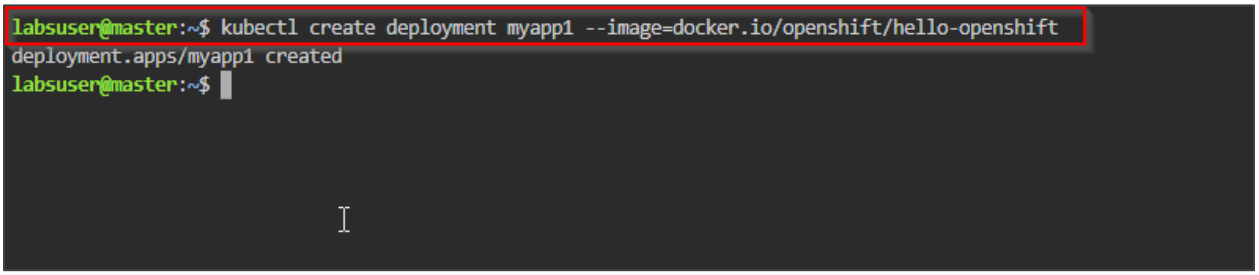
Steps to be followed:

1. Create the deployment
2. Create the namespaces
3. Scale and delete the deployment

Step 1: Create the deployment

1.1 Create the deployment by using the following command:

kubectl create deployment myapp1 --image=docker.io/openshift/hello-openshift



1.2 Verify the deployment and pod status by using the following command:

kubectl get pods

```
labsuser@master:~$ kubectl create deployment myapp1 --image=docker.io/openshift/hello-openshift
deployment.apps/myapp1 created
labsuser@master:~$ kubectl get pods
NAME                      READY   STATUS    RESTARTS      AGE
apache2                   1/1     Running   1 (63m ago)   163m
apache3                   1/1     Running   1 (63m ago)   154m
myapp1-57bb57dd79-dz8dg   1/1     Running   0             88s
mypod1                    1/1     Running   0             48m
mypod2                    1/1     Running   0             43m
labsuser@master:~$
```

1.3 Copy the **name** of the pod you created

```
labsuser@master:~$ kubectl create deployment myapp1 --image=docker.io/openshift/hello-openshift
deployment.apps/myapp1 created
labsuser@master:~$ kubectl get pods
NAME                      READY   STATUS    RESTARTS      AGE
apache2                   1/1     Running   1 (63m ago)   163m
apache3                   1/1     Running   1 (63m ago)   154m
myapp1-57bb57dd79-dz8dg   1/1     Running   0             88s
mypod1                    1/1     Running   0             48m
mypod2                    1/1     Running   0             43m
```

1.4 Describe the pod content by using the following command:

kubectl describe pod myapp1-57bb57dd79-dz8dg

```
labsuser@master:~$ kubectl describe pod myapp1-57bb57dd79-dz8dg
Name:             myapp1-57bb57dd79-dz8dg
Namespace:        default
Priority:         0
Service Account:  default
Node:             worker-node-1.example.com/172.31.14.131
Start Time:       Thu, 05 Oct 2023 10:27:54 +0000
Labels:           app=myapp1
                  pod-template-hash=57bb57dd79
Annotations:      cni.projectcalico.org/containerID: 5fd57c2fa17a6b791d4977642c80f134726187a66fa89b3b5fbe46af55dcbb55
                  cni.projectcalico.org/podIP: 192.168.47.132/32
                  cni.projectcalico.org/podIPs: 192.168.47.132/32
Status:           Running
IP:               192.168.47.132
IPs:
  IP:           192.168.47.132
Controlled By:  ReplicaSet/myapp1-57bb57dd79
Containers:
  hello-openshift:
    Container ID:   containerd://106bb7cdf11881bac72b3fcdfb71d2f1ac2005cbd99b99d98042ba072c8cefdd
```

```
Volumes:
  kube-api-access-wzw9p:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  5m7s  default-scheduler  Successfully assigned default/myapp1-57bb57dd79-dz8dg to worker-node-1.example.com
  Normal  Pulling    5m6s  kubelet            Pulling image "docker.io/openshift/hello-openshift"
  Normal  Pulled     5m6s  kubelet            Successfully pulled image "docker.io/openshift/hello-openshift" in 587ms (587ms including waiting)
  Normal  Created    5m5s  kubelet            Created container hello-openshift
  Normal  Started    5m5s  kubelet            Started container hello-openshift
labsuser@master:~$
```

1.5 Check the running deployment and describe its content by using the following command:

kubectl get deployment

```
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  5m7s  default-scheduler  Successfully assigned default/myapp1-57bb57dd79-dz8dg to worker-node-1.example.com
  Normal  Pulling    5m6s  kubelet            Pulling image "docker.io/openshift/hello-openshift"
  Normal  Pulled     5m6s  kubelet            Successfully pulled image "docker.io/openshift/hello-openshift" in 587ms (587ms including waiting)
  Normal  Created    5m5s  kubelet            Created container hello-openshift
  Normal  Started    5m5s  kubelet            Started container hello-openshift
labsuser@master:~$ kubectl get deployment
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
myapp1   1/1     1            1           7m57s
labsuser@master:~$
```

1.6 Replace **<deploymentName>** with your deployment's name and describe it by using the following command, as shown in the screenshot below:

kubectl describe deployment <deploymentName>

```
labsuser@master:~$ kubectl get deployment
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
myapp1   1/1     1            1           7m57s
labsuser@master:~$ kubectl describe deployment myapp1
Name:                   myapp1
Namespace:              default
CreationTimestamp:      Thu, 05 Oct 2023 10:27:54 +0000
Labels:                 app=myapp1
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=myapp1
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=myapp1
  Containers:
   hello-openshift:
    Image:        docker.io/openshift/hello-openshift
    Port:         <none>
```

1.7 Create a YAML file by using the following command:

kubectl create deployment myhttpd --image=docker.io/httpd --dry-run=client -o yaml > myapp1.yaml

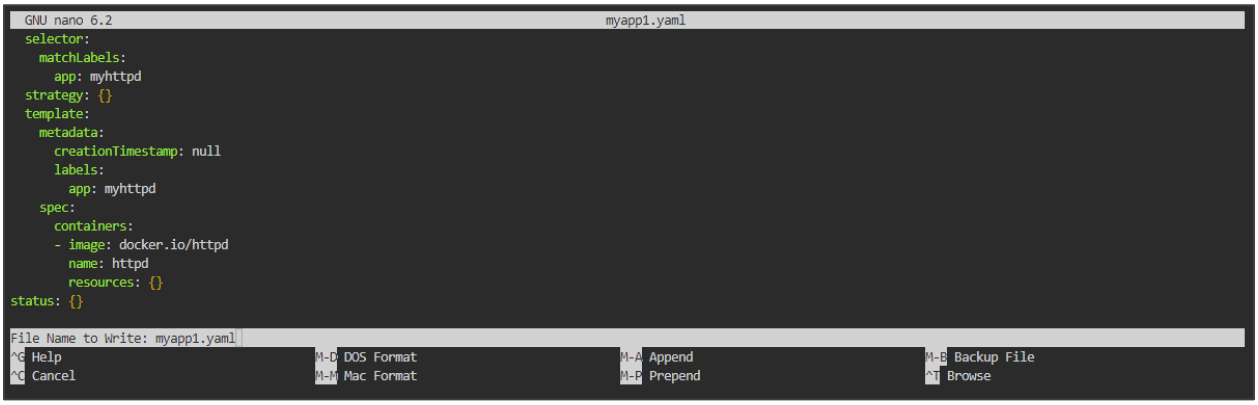
```
   hello-openshift:
    Image:        docker.io/openshift/hello-openshift
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   myapp1-57bb57dd79 (1/1 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  16m   deployment-controller  Scaled up replica set myapp1-57bb57dd79 to 1
labsuser@master:~$ kubectl create deployment myhttpd --image=docker.io/httpd --dry-run=client -o yaml > myapp1.yaml
labsuser@master:~$
```

1.8 Edit the deployment file by using the following command:

nano myapp1.yaml

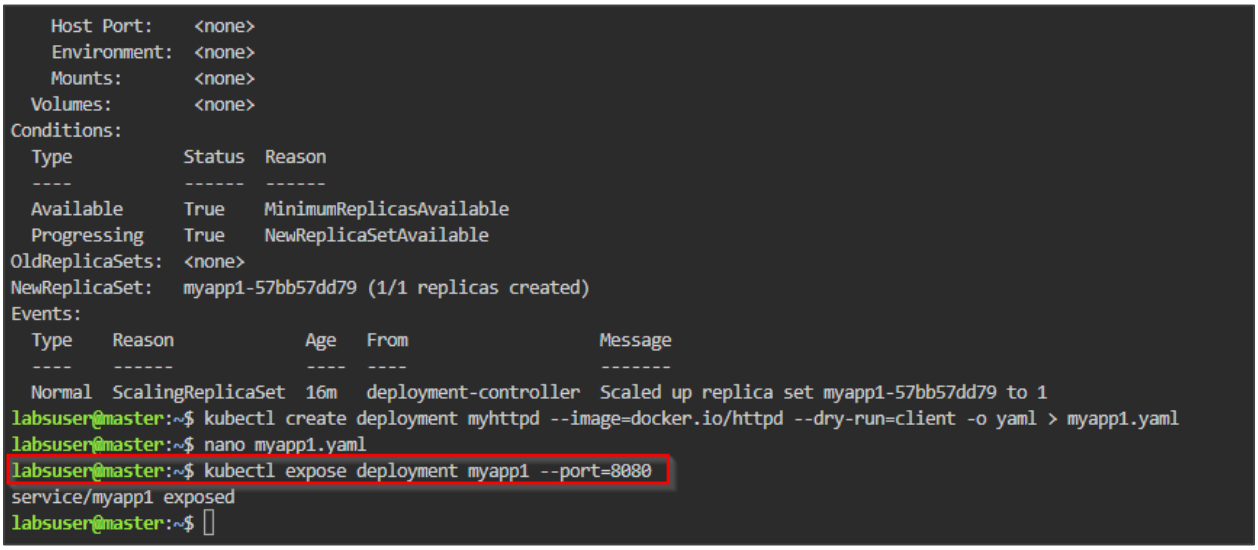
```
   hello-openshift:
    Image:        docker.io/openshift/hello-openshift
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   myapp1-57bb57dd79 (1/1 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  16m   deployment-controller  Scaled up replica set myapp1-57bb57dd79 to 1
labsuser@master:~$ kubectl create deployment myhttpd --image=docker.io/httpd --dry-run=client -o yaml > myapp1.yaml
labsuser@master:~$ nano myapp1.yaml
```

1.9 Press the **ctrl + o** keys to write, and then press the **enter** key; press the **ctrl + x** keys to exit the editor.



1.10 Expose the deployment to create a service by using the following command:

kubectl expose deployment myapp1 --port=8080



1.11 Verify the created services by using the following command:

kubectl get svc

```
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   myapp1-57bb57dd79 (1/1 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  16m   deployment-controller  Scaled up replica set myapp1-57bb57dd79 to 1
labsuser@master:~$ kubectl create deployment myhttpd --image=docker.io/httpd --dry-run=client -o yaml > myapp1.yaml
labsuser@master:~$ nano myapp1.yaml
labsuser@master:~$ kubectl expose deployment myapp1 --port=8080
service/myapp1 exposed
labsuser@master:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP    3h43m
myapp1       ClusterIP   10.96.82.231    <none>        8080/TCP   2m56s
myservice    ClusterIP   10.101.183.1    <none>        8081/TCP   145m
labsuser@master:~$
```

1.12 Describe the service by using the following command:

kubectl describe svc myapp1

```
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP    3h43m
myapp1       ClusterIP   10.96.82.231    <none>        8080/TCP   2m56s
myservice    ClusterIP   10.101.183.1    <none>        8081/TCP   145m
labsuser@master:~$ kubectl describe svc myapp1
Name:              myapp1
Namespace:         default
Labels:            app=myapp1
Annotations:       <none>
Selector:          app=myapp1
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.96.82.231
IPs:               10.96.82.231
Port:              <unset>  8080/TCP
TargetPort:        8080/TCP
Endpoints:         192.168.47.132:8080
Session Affinity:  None
Events:            <none>
labsuser@master:~$
```

Step 2: Create the namespaces

2.1 Create specific namespaces by using the following command:

kubectl create namespace mynamespace

```
myservice    ClusterIP   10.101.183.1   <none>        8081/TCP   145m
labsuser@master:~$ kubectl describe svc myapp1
Name:              myapp1
Namespace:         default
Labels:            app=myapp1
Annotations:       <none>
Selector:          app=myapp1
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.96.82.231
IPs:               10.96.82.231
Port:              <unset>  8080/TCP
TargetPort:        8080/TCP
Endpoints:         192.168.47.132:8080
Session Affinity:  None
Events:            <none>
labsuser@master:~$ kubectl create namespace mynamespace
namespace/mynamespace created
labsuser@master:~$
```

2.2 Verify the namespace by using the following command:

kubectl get namespace

```
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.96.82.231
IPs:               10.96.82.231
Port:              <unset>  8080/TCP
TargetPort:        8080/TCP
Endpoints:         192.168.47.132:8080
Session Affinity:  None
Events:            <none>
labsuser@master:~$ kubectl create namespace mynamespace
namespace/mynamespace created
labsuser@master:~$ kubectl get namespace
NAME              STATUS   AGE
default           Active   3h50m
kube-node-lease   Active   3h50m
kube-public       Active   3h50m
kube-system       Active   3h50m
mynamespace       Active   78s
labsuser@master:~$
```

Step 3: Scale and delete the deployment

3.1 Create a deployment in a specific namespace and verify it by using the following commands:

kubectl create deployment myapp1 --image=docker.io/httpd -n mynamespace

kubectl get deployment -n mynamespace

kubectl get pods -n mynamespace

```
labsuser@master:~$ kubectl create namespace mynamespace
namespace/mynamespace created
labsuser@master:~$ kubectl get namespace
NAME              STATUS   AGE
default           Active   3h50m
kube-node-lease   Active   3h50m
kube-public       Active   3h50m
kube-system       Active   3h50m
mynamespace       Active   78s
labsuser@master:~$ kubectl create deployment myapp1 --image=docker.io/httpd -n mynamespace
deployment.apps/myapp1 created
labsuser@master:~$ kubectl get deployment -n namespace
No resources found in namespace namespace.
labsuser@master:~$ kubectl get deployment -n mynamespace
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
myapp1   1/1     1            1           84s
labsuser@master:~$ kubectl get pods -n mynamespace
NAME                      READY   STATUS    RESTARTS   AGE
myapp1-56f676576b-zqh5v   1/1     Running   0          100s
labsuser@master:~$
```

3.2 Scale and verify the deployment by using the following commands:

kubectl scale --replicas=3 deployment myapp1 -n mynamespace

kubectl get deployment -n mynamespace

```
labsuser@master:~$ kubectl get deployment -n mynamespace
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
myapp1   1/1     1            1           84s
labsuser@master:~$ kubectl get pods -n mynamespace
NAME                      READY   STATUS    RESTARTS   AGE
myapp1-56f676576b-zqh5v   1/1     Running   0          100s
labsuser@master:~$ kubectl scale --replicas=3 deployment mydep -n  mynamespace
error: no objects passed to scale
labsuser@master:~$ kubectl scale --replicas=3 deployment myapp1 -n  mynamespace
deployment.apps/myapp1 scaled
labsuser@master:~$ kubectl get deployment -n mynamespace
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
myapp1   3/3     3            3           9m
labsuser@master:~$
```

3.3 Retrieve the endpoints by using the following commands:

kubectl get endpoints

kubectl describe endpoints

```
labsuser@master:~$ kubectl get endpoints
NAME         ENDPOINTS                                AGE
kubernetes   172.31.25.147:6443                       4h6m
myapp1       192.168.47.132:8080                      25m
myservice    192.168.232.194:80,192.168.47.130:80     168m
labsuser@master:~$ kubectl describe endpoints
Name:         kubernetes
Namespace:    default
Labels:       endpointslice.kubernetes.io/skip-mirror=true
Annotations:  <none>
Subsets:
  Addresses:          172.31.25.147
  NotReadyAddresses:  <none>
  Ports:
    Name   Port  Protocol
    ----   ----  --------
    https  6443  TCP

Events:  <none>
```

3.4 Delete the deployment by using the following command:

kubectl delete deployment myapp1

```
Events:  <none>


Name:         myservice
Namespace:    default
Labels:       <none>
Annotations:  endpoints.kubernetes.io/last-change-trigger-time: 2023-10-05T09:26:37Z
Subsets:
  Addresses:          192.168.232.194,192.168.47.130
  NotReadyAddresses:  <none>
  Ports:
    Name     Port  Protocol
    ----     ----  --------
    <unset>  80    TCP

Events:  <none>
labsuser@master:~$ kubectl delete deployment myapp1
deployment.apps "myapp1" deleted
labsuser@master:~$
```

3.5 Delete the service by using the following command:

kubectl delete svc myapp1

```
Name:         myservice
Namespace:    default
Labels:       <none>
Annotations:  endpoints.kubernetes.io/last-change-trigger-time: 2023-10-05T09:26:37Z
Subsets:
  Addresses:          192.168.232.194,192.168.47.130
  NotReadyAddresses:  <none>
  Ports:
    Name     Port  Protocol
    ----     ----  --------
    <unset>  80    TCP

Events:  <none>
labsuser@master:~$ kubectl delete deployment myapp1
deployment.apps "myapp1" deleted
labsuser@master:~$ kubectl delete svc myapp1
service "myapp1" deleted
```

3.6 Verify the deleted service by using the following command:

kubectl get svc

```
labsuser@master:~$ kubectl delete deployment myapp1
deployment.apps "myapp1" deleted
labsuser@master:~$ kubectl delete svc myapp1
service "myapp1" deleted
labsuser@master:~$ kubectl get svc
NAME          TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
kubernetes    ClusterIP   10.96.0.1      <none>        443/TCP    4h13m
myservice     ClusterIP   10.101.183.1   <none>        8081/TCP   175m
labsuser@master:~$
```

3.7 Delete the namespace by using the following command:

kubectl delete namespace mynamespace

```
labsuser@master:~$ kubectl delete deployment myapp1
deployment.apps "myapp1" deleted
labsuser@master:~$ kubectl delete svc myapp1
service "myapp1" deleted
labsuser@master:~$ kubectl get svc
NAME          TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
kubernetes    ClusterIP   10.96.0.1      <none>        443/TCP    4h13m
myservice     ClusterIP   10.101.183.1   <none>        8081/TCP   175m
labsuser@master:~$ kubectl delete namespace mynamespace
namespace "mynamespace" deleted
```

3.8 Verify the events by using the following command:

kubectl get events

```
labsuser@master:~$ kubectl delete deployment myapp1
deployment.apps "myapp1" deleted
labsuser@master:~$ kubectl delete svc myapp1
service "myapp1" deleted
labsuser@master:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP    4h13m
myservice    ClusterIP   10.101.183.1   <none>        8081/TCP   175m
labsuser@master:~$ kubectl delete namespace mynamespace
namespace "mynamespace" deleted
labsuser@master:~$ kubectl get events
LAST SEEN   TYPE     REASON    OBJECT                        MESSAGE
6m43s       Normal   Killing   pod/myapp1-57bb57dd79-dz8dg   Stopping container hello-openshift
labsuser@master:~$
```

3.9 Verify the nodes state by using the following command:

kubectl get nodes

```
labsuser@master:~$ kubectl delete deployment myapp1
deployment.apps "myapp1" deleted
labsuser@master:~$ kubectl delete svc myapp1
service "myapp1" deleted
labsuser@master:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP    4h13m
myservice    ClusterIP   10.101.183.1   <none>        8081/TCP   175m
labsuser@master:~$ kubectl delete namespace mynamespace
namespace "mynamespace" deleted
labsuser@master:~$ kubectl get events
LAST SEEN   TYPE     REASON    OBJECT                        MESSAGE
6m43s       Normal   Killing   pod/myapp1-57bb57dd79-dz8dg   Stopping container hello-openshift
labsuser@master:~$ kubectl get nodes
NAME                       STATUS   ROLES           AGE     VERSION
master.example.com         Ready    control-plane   4h20m   v1.28.2
worker-node-1.example.com  Ready    <none>          4h2m    v1.28.2
worker-node-2.example.com  Ready    <none>          4h2m    v1.28.2
labsuser@master:~$
```

3.10 Describe the configuration of the node by using the following command:

kubectl describe node worker-node-1.example.com

```
labsuser@master:~$ kubectl describe node worker-node-1.example.com
Name:               worker-node-1.example.com
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=worker-node-1.example.com
                    kubernetes.io/os=linux
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: unix:///var/run/containerd/containerd.sock
                    node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: 172.31.14.131/20
                    projectcalico.org/IPv4IPIPTunnelAddr: 192.168.47.128
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Thu, 05 Oct 2023 07:30:08 +0000
Taints:             <none>
Unschedulable:      false
Lease:
  HolderIdentity:  worker-node-1.example.com
  AcquireTime:     <unset>
  RenewTime:       Thu, 05 Oct 2023 11:37:28 +0000
Conditions:
  Type                 Status  LastHeartbeatTime                 LastTransitionTime                Reason                       Message
  ----                 ------  -----------------                 ------------------                ------                       -------
  NetworkUnavailable   False   Thu, 05 Oct 2023 09:25:55 +0000   Thu, 05 Oct 2023 09:25:55 +0000   CalicoIsUp                   Calico is running on this node
  MemoryPressure       False   Thu, 05 Oct 2023 11:33:51 +0000   Thu, 05 Oct 2023 07:30:08 +0000   KubeletHasSufficientMemory   kubelet has sufficient memory available
  DiskPressure         False   Thu, 05 Oct 2023 11:33:51 +0000   Thu, 05 Oct 2023 07:30:08 +0000   KubeletHasNoDiskPressure     kubelet has no disk pressure
```

```
  Boot ID:                    0813ac4f-bf2b-4423-9198-f448596eaf84
  Kernel Version:             6.2.0-1012-aws
  OS Image:                   Ubuntu 22.04.3 LTS
  Operating System:           linux
  Architecture:               amd64
  Container Runtime Version:  containerd://1.6.8
  Kubelet Version:            v1.28.2
  Kube-Proxy Version:         v1.28.2
Non-terminated Pods:          (4 in total)
  Namespace                   Name                CPU Requests  CPU Limits  Memory Requests  Memory Limits  Age
  ---------                   ----                ------------  ----------  ---------------  -------------  ---
  default                     apache2             0 (0%)        0 (0%)      0 (0%)           0 (0%)         4h23m
  default                     mypod2              0 (0%)        0 (0%)      0 (0%)           0 (0%)         143m
  kube-system                 calico-node-t97m4   250m (12%)    0 (0%)      0 (0%)           0 (0%)         4h39m
  kube-system                 kube-proxy-zd5vc    0 (0%)        0 (0%)      0 (0%)           0 (0%)         4h39m
Allocated resources:
  (Total limits may be over 100 percent, i.e., overcommitted.)
  Resource           Requests    Limits
  --------           --------    ------
  cpu                250m (12%)  0 (0%)
  memory             0 (0%)      0 (0%)
  ephemeral-storage  0 (0%)      0 (0%)
  hugepages-1Gi      0 (0%)      0 (0%)
  hugepages-2Mi      0 (0%)      0 (0%)
Events:              <none>
labsuser@master:~$
```

By following these steps, you have successfully completed the deployment creation, namespace management, scaling, and deletion, improving Kubernetes cluster management.