Lab guide 4:

Services

Q.1 Create a deployment using image quay.io/mayank123modi/simple-webapp named test3 After creating this expose it on ClusterIP type service by CLI method and check everything .

Q.2 create a deployment using mdhack/myapache and expose it using NodePort type Service

Q.3 Expose a deployment that named test3 service name should be kubecluster and type is NodePort by YAML creation.

Ans: 1

kubectl create deployment --image quay.io/mayank123modi/simple-webapp   test3

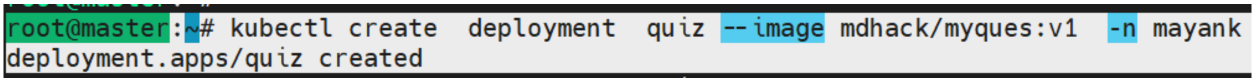
kubectl expose deployment test3 --port 8080

kubectl get svc

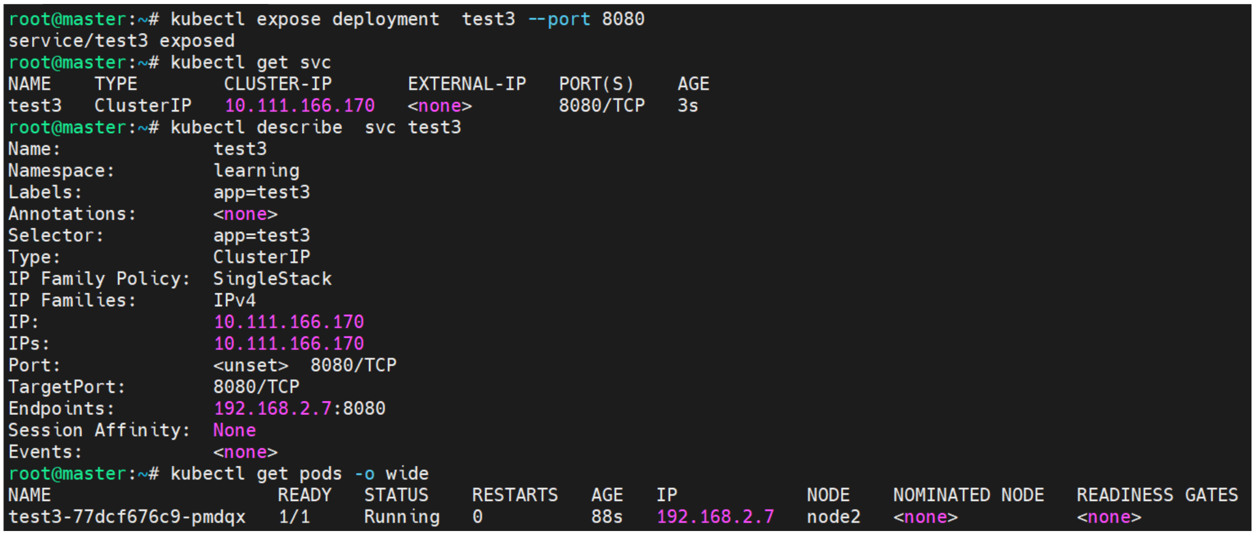
kubectl describe svc test3

kubectl get pods -o wide

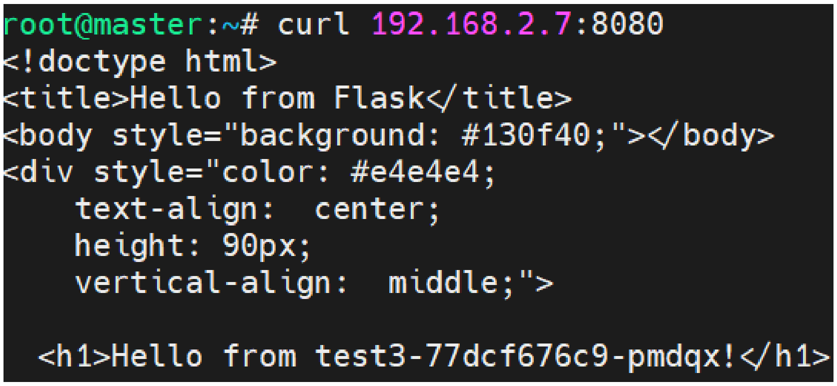
curl 192.168.2.7:8080



```
root@master:~# kubectl create  deployment  quiz --image mdhack/myques:v1  -n mayank
deployment.apps/quiz created
```



```
root@master:~# kubectl expose deployment  test3 --port 8080
service/test3 exposed
root@master:~# kubectl get svc
NAME    TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
test3   ClusterIP   10.111.166.170   <none>        8080/TCP   3s
root@master:~# kubectl describe  svc test3
Name:              test3
Namespace:         learning
Labels:            app=test3
Annotations:       <none>
Selector:          app=test3
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.111.166.170
IPs:               10.111.166.170
Port:              <unset>  8080/TCP
TargetPort:        8080/TCP
Endpoints:         192.168.2.7:8080
Session Affinity:  None
Events:            <none>
root@master:~# kubectl get pods -o wide
NAME                     READY   STATUS    RESTARTS   AGE   IP            NODE    NOMINATED NODE   READINESS GATES
test3-77dcf676c9-pmdqx   1/1     Running   0          88s   192.168.2.7   node2   <none>           <none>
```



```
root@master:~# curl 192.168.2.7:8080
<!doctype html>
<title>Hello from Flask</title>
<body style="background: #130f40;"></body>
<div style="color: #e4e4e4;
    text-align:  center;
    height: 90px;
    vertical-align:  middle;">

  <h1>Hello from test3-77dcf676c9-pmdqx!</h1>
```

Ans: 2

kubectl create deployment --image mdhack/myserver test4

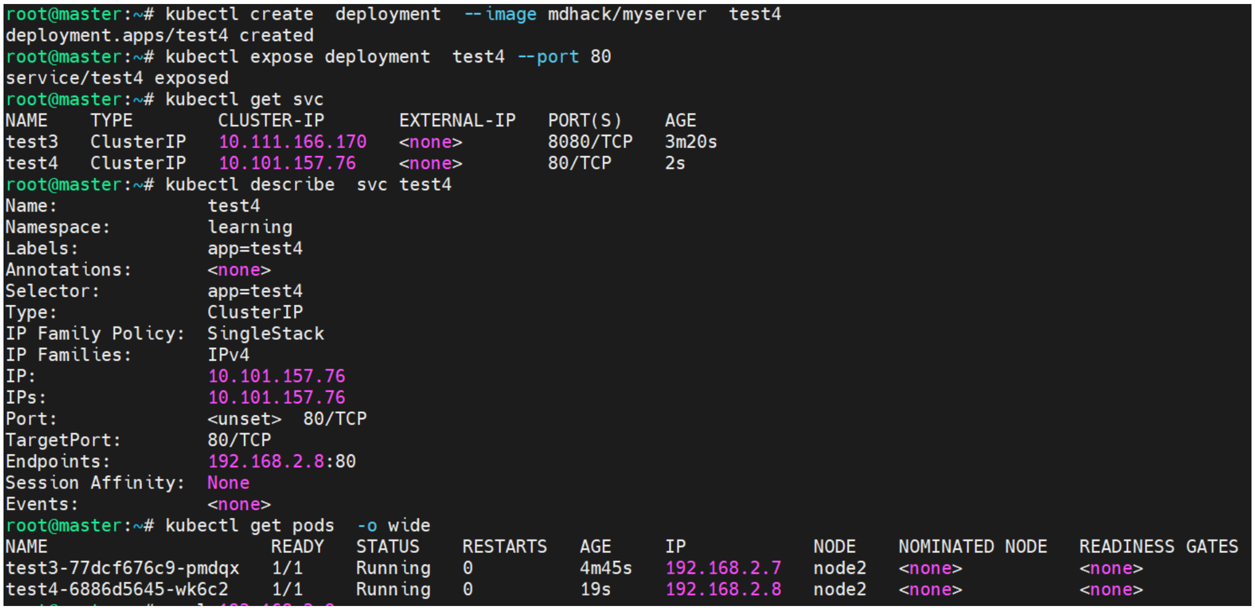
kubectl expose deployment test4 --port 80

kubectl get svc

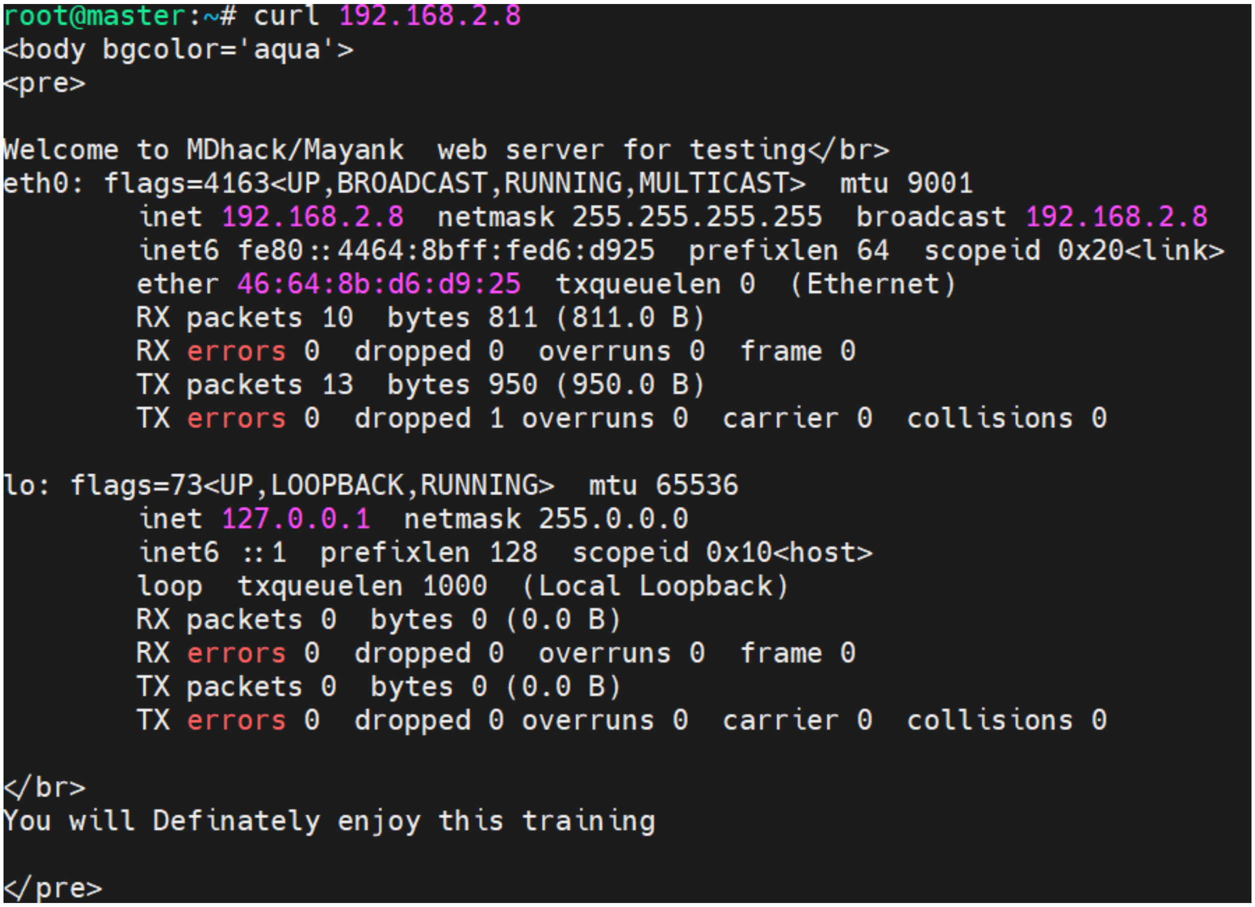
kubectl describe svc test4

kubectl get pods -o wide

curl   <your pod ip address>



```
root@master:~# kubectl create  deployment  --image mdhack/myserver  test4
deployment.apps/test4 created
root@master:~# kubectl expose deployment  test4 --port 80
service/test4 exposed
root@master:~# kubectl get svc
NAME    TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
test3   ClusterIP   10.111.166.170   <none>        8080/TCP   3m20s
test4   ClusterIP   10.101.157.76    <none>        80/TCP     2s
root@master:~# kubectl describe  svc test4
Name:              test4
Namespace:         learning
Labels:            app=test4
Annotations:       <none>
Selector:          app=test4
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.101.157.76
IPs:               10.101.157.76
Port:              <unset>  80/TCP
TargetPort:        80/TCP
Endpoints:         192.168.2.8:80
Session Affinity:  None
Events:            <none>
root@master:~# kubectl get pods  -o wide
NAME                     READY   STATUS    RESTARTS   AGE     IP            NODE    NOMINATED NODE   READINESS GATES
test3-77dcf676c9-pmdqx   1/1     Running   0          4m45s   192.168.2.7   node2   <none>           <none>
test4-6886d5645-wk6c2    1/1     Running   0          19s     192.168.2.8   node2   <none>           <none>
```



```
root@master:~# curl 192.168.2.8
<body bgcolor='aqua'>
<pre>

Welcome to MDhack/Mayank  web server for testing</br>
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 192.168.2.8  netmask 255.255.255.255  broadcast 192.168.2.8
        inet6 fe80::4464:8bff:fed6:d925  prefixlen 64  scopeid 0x20<link>
        ether 46:64:8b:d6:d9:25  txqueuelen 0  (Ethernet)
        RX packets 10  bytes 811 (811.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13  bytes 950 (950.0 B)
        TX errors 0  dropped 1 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

</br>
You will Definately enjoy this training

</pre>
```

Ans: 3

```
kubectl create  service nodeport    kubecluster  --tcp 8080  --dry-run  -o yaml  >  svc.yaml
```

```
vim svc.yaml
```

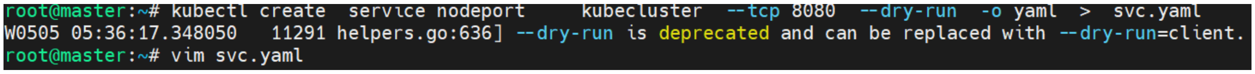
```
kubectl create  -f svc.yaml
```

```
kubectl get svc
```

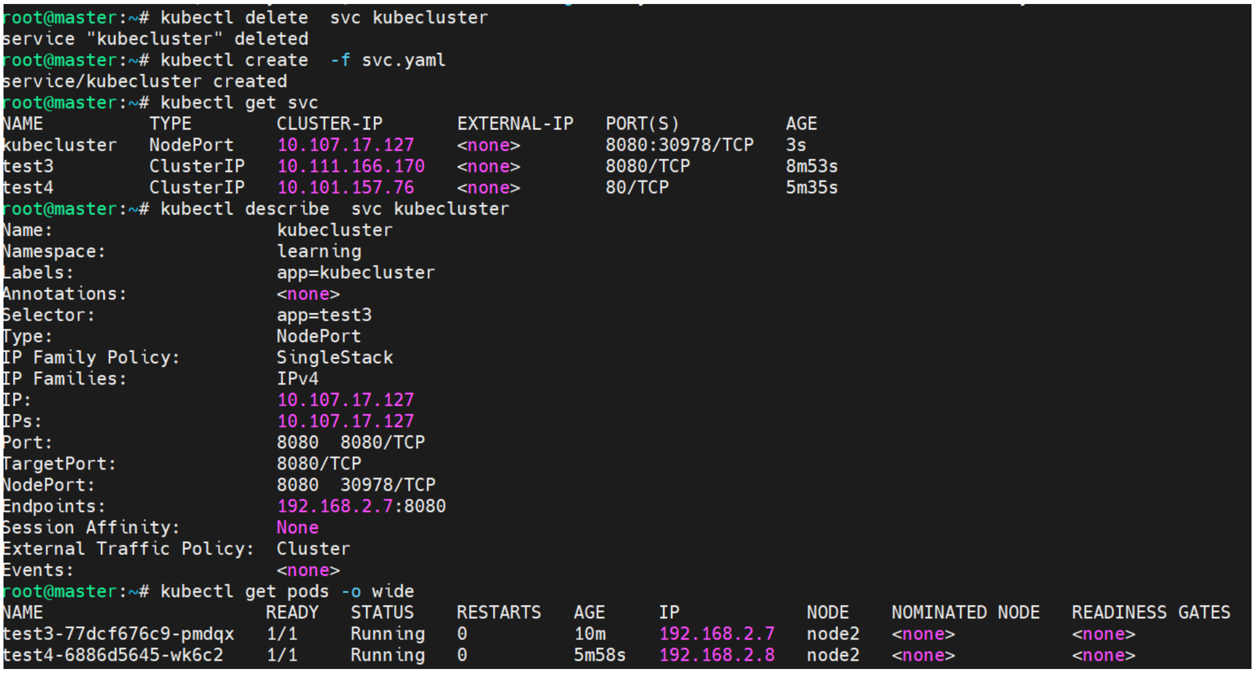
```
kubectl describe  svc kubecluster
```

```
kubectl get pods -o wide
```

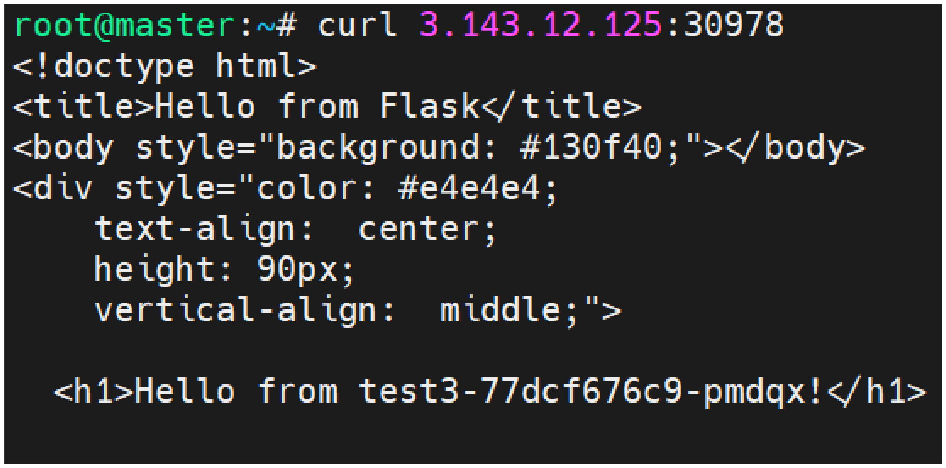
```
curl   <publicip of cluster>:<nodeport>
```



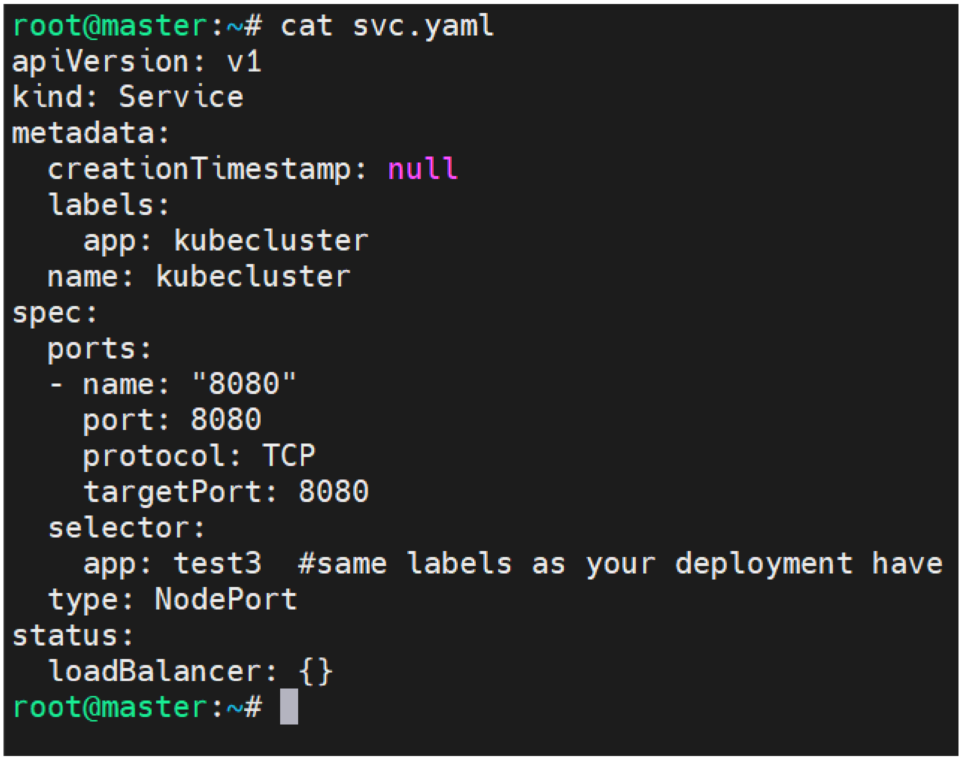
```
root@master:~# kubectl create  service nodeport    kubecluster  --tcp 8080  --dry-run  -o yaml  >  svc.yaml
W0505 05:36:17.348050   11291 helpers.go:636] --dry-run is deprecated and can be replaced with --dry-run=client.
root@master:~# vim svc.yaml
```



```
root@master:~# kubectl delete  svc kubecluster
service "kubecluster" deleted
root@master:~# kubectl create  -f svc.yaml
service/kubecluster created
root@master:~# kubectl get svc
NAME          TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubecluster   NodePort    10.107.17.127    <none>        8080:30978/TCP   3s
test3         ClusterIP   10.111.166.170   <none>        8080/TCP         8m53s
test4         ClusterIP   10.101.157.76    <none>        80/TCP           5m35s
root@master:~# kubectl describe  svc kubecluster
Name:                     kubecluster
Namespace:                learning
Labels:                   app=kubecluster
Annotations:              <none>
Selector:                 app=test3
Type:                     NodePort
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.107.17.127
IPs:                      10.107.17.127
Port:                     8080  8080/TCP
TargetPort:               8080/TCP
NodePort:                 8080  30978/TCP
Endpoints:                192.168.2.7:8080
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
root@master:~# kubectl get pods -o wide
NAME                     READY   STATUS    RESTARTS   AGE     IP            NODE    NOMINATED NODE   READINESS GATES
test3-77dcf676c9-pmdqx   1/1     Running   0          10m     192.168.2.7   node2   <none>           <none>
test4-6886d5645-wk6c2    1/1     Running   0          5m58s   192.168.2.8   node2   <none>           <none>
```



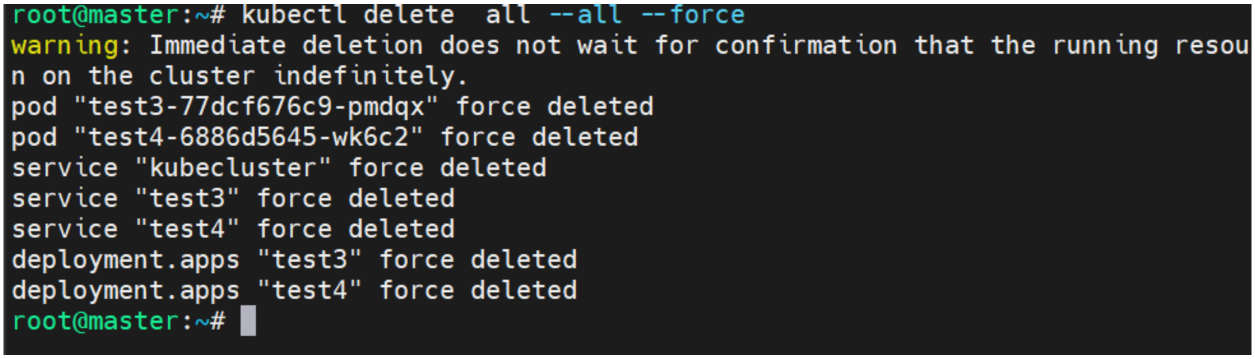
```
root@master:~# curl 3.143.12.125:30978
<!doctype html>
<title>Hello from Flask</title>
<body style="background: #130f40;"></body>
<div style="color: #e4e4e4;
    text-align:  center;
    height: 90px;
    vertical-align:  middle;">

  <h1>Hello from test3-77dcf676c9-pmdqx!</h1>
```



```
root@master:~# cat svc.yaml
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: null
  labels:
    app: kubecluster
  name: kubecluster
spec:
  ports:
  - name: "8080"
    port: 8080
    protocol: TCP
    targetPort: 8080
  selector:
    app: test3  #same labels as your deployment have
  type: NodePort
status:
  loadBalancer: {}
root@master:~#
```

<u>Clean up your environment</u>

kubectl delete all --all –force



```
root@master:~# kubectl delete  all --all --force
warning: Immediate deletion does not wait for confirmation that the running resou
n on the cluster indefinitely.
pod "test3-77dcf676c9-pmdqx" force deleted
pod "test4-6886d5645-wk6c2" force deleted
service "kubecluster" force deleted
service "test3" force deleted
service "test4" force deleted
deployment.apps "test3" force deleted
deployment.apps "test4" force deleted
root@master:~#
```