Lesson 03 Demo 07

Deploying a Voting Application

Objective: To deploy a voting application using Kubernetes pods

Tools required: kubeadm, kubectl, kubelet, and containerd

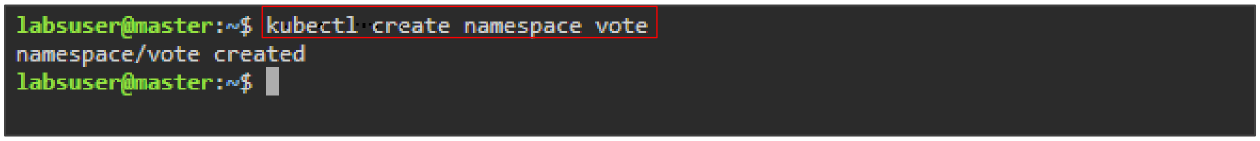
Prerequisites: A Kubernetes cluster (refer to Demo 01 from Lesson 01 for setting up a cluster)

Steps to be followed:

1. Create a namespace
2. Create an application for deployment
3. Verify the deployment of the application

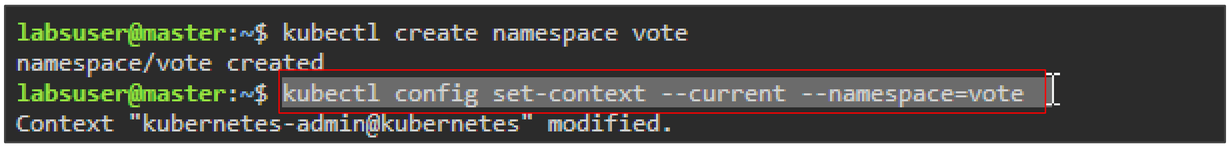
Step 1: Create a namespace

1.1 Create a namespace named **vote** in the master node using the following command:
kubectl create namespace vote



```
labsuser@master:~$ kubectl create namespace vote
namespace/vote created
labsuser@master:~$ 
```

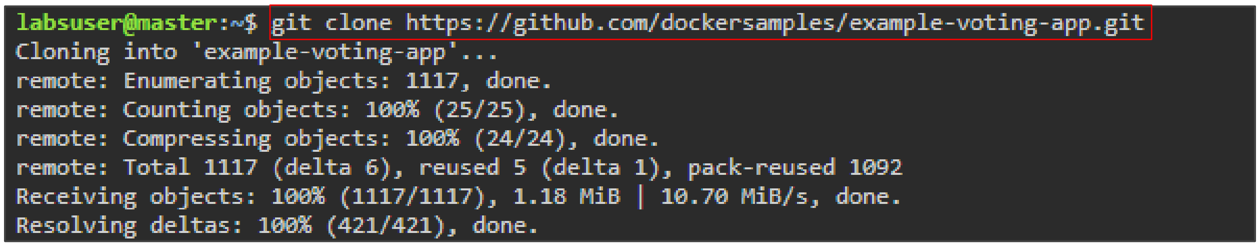
1.2 Execute the following command to set the **vote** namespace as the current context:
kubectl config set-context --current --namespace=vote



```
labsuser@master:~$ kubectl create namespace vote
namespace/vote created
labsuser@master:~$ kubectl config set-context --current --namespace=vote
Context "kubernetes-admin@kubernetes" modified.
```

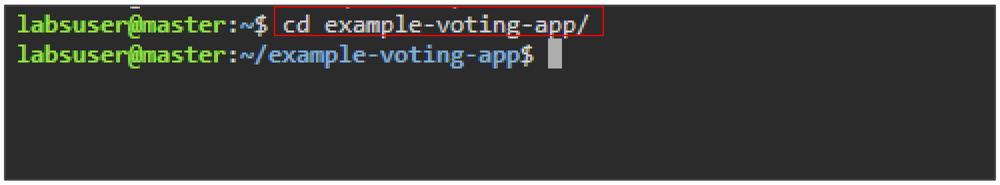
Step 2: Create an application for deployment

2.1 Execute the following command to clone the repository that contains the voting application:
git clone https://github.com/dockersamples/example-voting-app.git



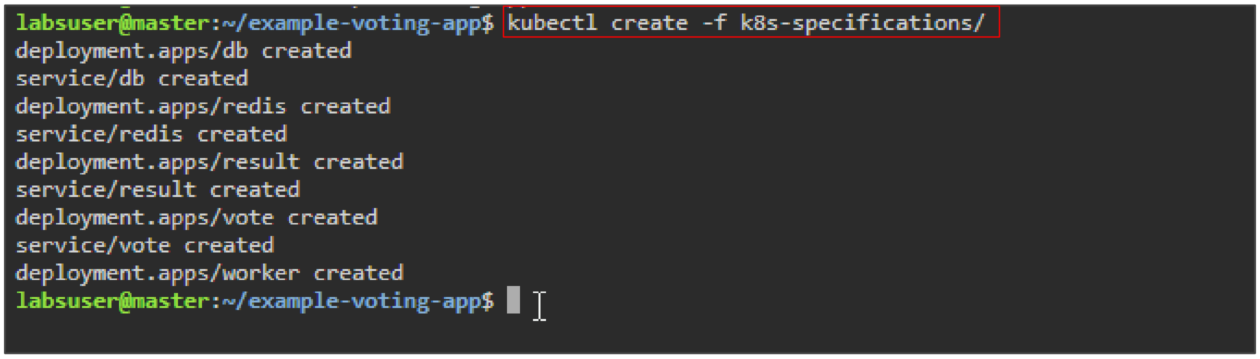
```
labsuser@master:~$ git clone https://github.com/dockersamples/example-voting-app.git
Cloning into 'example-voting-app'...
remote: Enumerating objects: 1117, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 1117 (delta 6), reused 5 (delta 1), pack-reused 1092
Receiving objects: 100% (1117/1117), 1.18 MiB | 10.70 MiB/s, done.
Resolving deltas: 100% (421/421), done.
```

2.2 Navigate to the cloned directory using the following command:
cd example-voting-app/



```
labsuser@master:~$ cd example-voting-app/
labsuser@master:~/example-voting-app$
```

2.3 Execute the following command to deploy the resources defined in the configuration files located in the **k8s-specifications** directory:
kubectl create -f k8s-specifications/

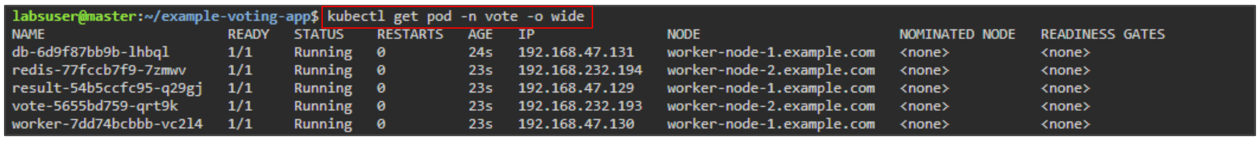


```
labsuser@master:~/example-voting-app$ kubectl create -f k8s-specifications/
deployment.apps/db created
service/db created
deployment.apps/redis created
service/redis created
deployment.apps/result created
service/result created
deployment.apps/vote created
service/vote created
deployment.apps/worker created
labsuser@master:~/example-voting-app$
```

Step 3: Verify the deployment of the application

3.1 Verify the created Kubernetes pod state using the following command:

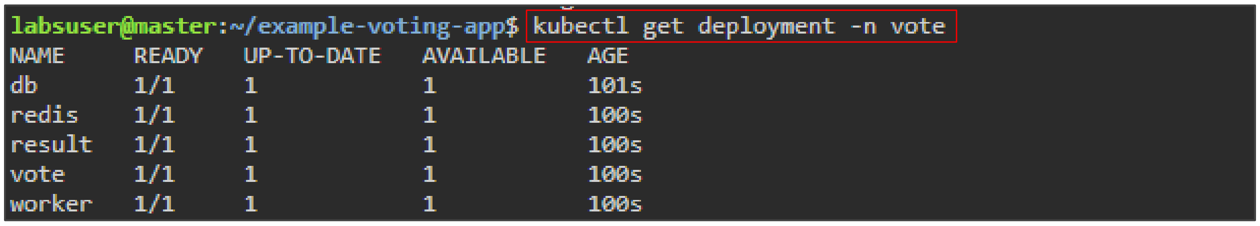
kubectl get pod -n vote -o wide



```
labsuser@master:~/example-voting-app$ kubectl get pod -n vote -o wide
NAME                        READY   STATUS    RESTARTS   AGE   IP                NODE                       NOMINATED NODE   READINESS GATES
db-6d9f87bb9b-lhbql         1/1     Running   0          24s   192.168.47.131    worker-node-1.example.com  <none>           <none>
redis-77fccb7f9-7zmwv       1/1     Running   0          23s   192.168.232.194   worker-node-2.example.com  <none>           <none>
result-54b5ccfc95-q29gj     1/1     Running   0          23s   192.168.47.129    worker-node-1.example.com  <none>           <none>
vote-5655bd759-qrt9k        1/1     Running   0          23s   192.168.232.193   worker-node-2.example.com  <none>           <none>
worker-7dd74bcbbb-vc2l4     1/1     Running   0          23s   192.168.47.130    worker-node-1.example.com  <none>           <none>
```

3.2 Execute the following command to retrieve information about deployments in the **vote** namespace:

kubectl get deployment -n vote

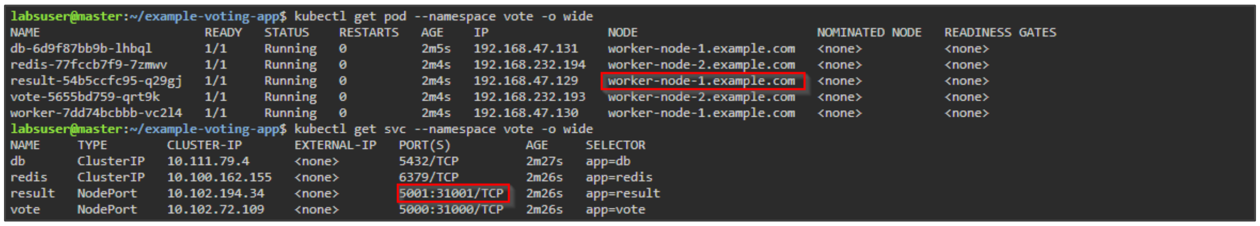


```
labsuser@master:~/example-voting-app$ kubectl get deployment -n vote
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
db       1/1     1            1           101s
redis    1/1     1            1           100s
result   1/1     1            1           100s
vote     1/1     1            1           100s
worker   1/1     1            1           100s
```

3.3 To get detailed information about the pods within the **vote** namespace, run the following commands:

kubectl get pod --namespace vote -o wide

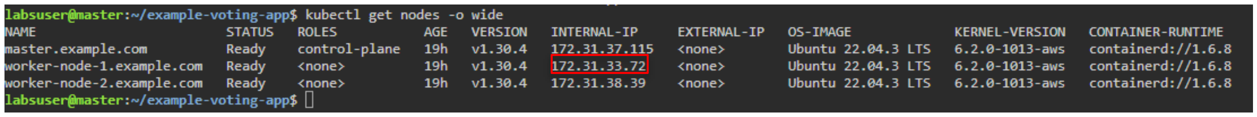
kubectl get svc --namespace vote -o wide



```
labsuser@master:~/example-voting-app$ kubectl get pod --namespace vote -o wide
NAME                        READY   STATUS    RESTARTS   AGE    IP                NODE                        NOMINATED NODE   READINESS GATES
db-6d9f87bb9b-lhbql         1/1     Running   0          2m5s   192.168.47.131    worker-node-1.example.com   <none>           <none>
redis-77fccb7f9-7zmwv       1/1     Running   0          2m4s   192.168.232.194   worker-node-2.example.com   <none>           <none>
result-54b5ccfc95-q29gj     1/1     Running   0          2m4s   192.168.47.129    worker-node-1.example.com   <none>           <none>
vote-5655bd759-qrt9k        1/1     Running   0          2m4s   192.168.232.193   worker-node-2.example.com   <none>           <none>
worker-7dd74bcbbb-vc2l4     1/1     Running   0          2m4s   192.168.47.130    worker-node-1.example.com   <none>           <none>
labsuser@master:~/example-voting-app$ kubectl get svc --namespace vote -o wide
NAME     TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE     SELECTOR
db       ClusterIP   10.111.79.4      <none>        5432/TCP         2m27s   app=db
redis    ClusterIP   10.100.162.155   <none>        6379/TCP         2m26s   app=redis
result   NodePort    10.102.194.34    <none>        5001:31001/TCP   2m26s   app=result
vote     NodePort    10.102.72.109    <none>        5000:31000/TCP   2m26s   app=vote
```

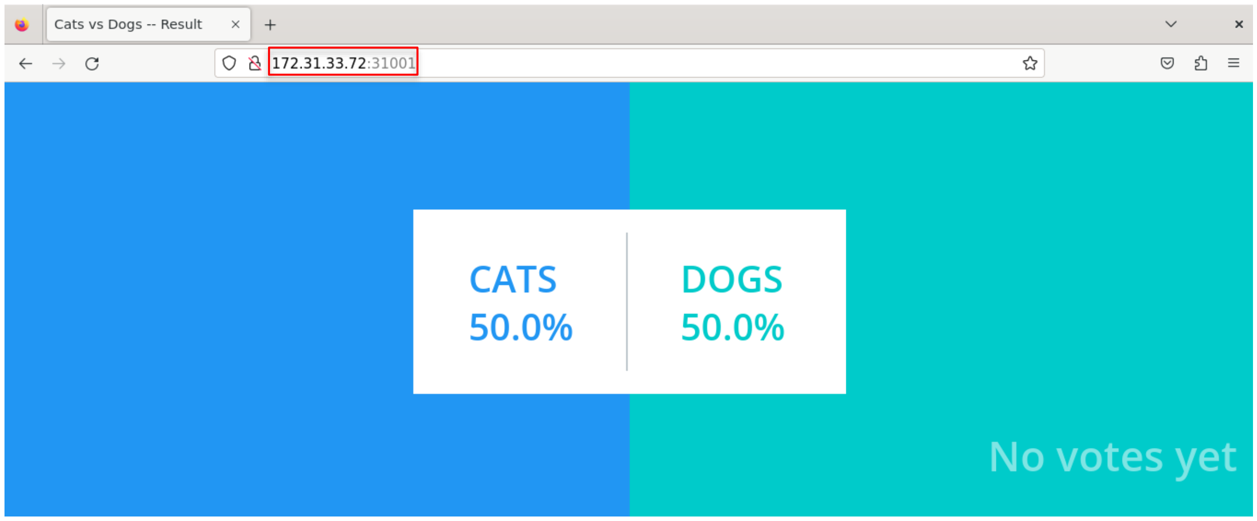
Remember to note the **NODE** and **PORT(S)** where the pod is running.

3.4 Execute the following command to get the **INTERNAL-IP** address of **worker-node-1.example.com**:
kubectl get nodes -o wide



```
labsuser@master:~/example-voting-app$ kubectl get nodes -o wide
NAME                        STATUS   ROLES           AGE   VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION    CONTAINER-RUNTIME
master.example.com          Ready    control-plane   19h   v1.30.4   172.31.37.115   <none>        Ubuntu 22.04.3 LTS   6.2.0-1013-aws    containerd://1.6.8
worker-node-1.example.com   Ready    <none>          19h   v1.30.4   172.31.33.72    <none>        Ubuntu 22.04.3 LTS   6.2.0-1013-aws    containerd://1.6.8
worker-node-2.example.com   Ready    <none>          19h   v1.30.4   172.31.38.39    <none>        Ubuntu 22.04.3 LTS   6.2.0-1013-aws    containerd://1.6.8
labsuser@master:~/example-voting-app$
```

3.5 Open the **Firefox** browser on the master node's desktop and paste the **INTERNAL-IP** address and port number



Note: Use the current IP address of **worker-node-1.example.com** and the port number where the **resulting** pod is deployed

By following these steps, you have successfully deployed a voting application using Kubernetes pods.