1. SSH to your AWS Workstation.

Replace **<your-name>** with your **name** throughout the Lab.

|  |
| --- |
| $ sudo su # vim dotnet\_app-<your-name>.yaml |

**2. Paste the below script in the dotnet\_app.yaml**

Update the image: **lovescloud/docker.net.demo:latest** and replace <yourname> with your namein the below script with your dockerhub image name that you uploaded to docker hub in docker lab Pushing images to docker Hub for dotnet.**Save and exit by pressing the ESC key and type wq to save and quit by pressing enter**

|  |
| --- |
| apiVersion: apps/v1 kind: Deployment metadata:  name: dotnetapp-<your-name> spec:  selector:  matchLabels:  run: dotnetapp-<your-name>  replicas: 2  template:  metadata:  labels:  run: dotnetapp-<your-name>  spec:  containers:  - name: dotnetapp-<your-name>  image: lovescloud/docker.net.demo:latest  ports:  - name: port80  containerPort: 80 --- apiVersion: v1 kind: Service metadata:  name: dotnetapp-<your-name>  labels:  run: dotnetapp-<your-name> spec:  type: NodePort  ports:  - name: port80  port: 80  protocol: TCP  selector:  run: dotnetapp-<your-name> |

3. Run the below commands to deploy the .NET application on your Kubernetes Cluster

|  |
| --- |
| # kubectl apply -f dotnet\_app.yaml |

4. Get the POD deployment details by running the below command.

|  |
| --- |
| # kubectl get po -o wide |

**Run the below command to bind POD port to your Workstation Port. Replace <your-name> with your name.**

|  |
| --- |
| **# kubectl port-forward --address 0.0.0.0 svc/dotnetapp-<your-name> -n default 8080:80 &** |

**Access the application now on the public ip of your AWS Workstation on port 8080.**

**Example**

[**http://34.239.177.176:8080/**](http://34.239.177.176:8080/)