1. SSH to your AWS Workstation.

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

|  |
| --- |
| $ sudo su  # vim java-app-<your-name>.yaml |

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

Update the image: **lovescloud/java-docker:latest** and replace <yourname> with your namein the below script with your docker hub image.

|  |
| --- |
| apiVersion: apps/v1  kind: Deployment  metadata:  name: java-app-<your-name>  spec:  selector:  matchLabels:  run: java-app-<your-name>  replicas: 2  template:  metadata:  labels:  run: java-app-<your-name>  spec:  containers:  - name: java-app-<your-name>  image: lovescloud/java-docker:latest  ports:  - name: port8080  containerPort: 8080  ---  apiVersion: v1  kind: Service  metadata:  name: java-app-<your-name>  labels:  run: java-app-<your-name>  spec:  type: NodePort  ports:  - name: port8080  port: 8080  protocol: TCP  selector:  run: java-app-<your-name> |

3. Run the below commands to deploy the JAVA application on your EKS K8s Cluster

|  |
| --- |
| # kubectl apply -f java-app-<your-name>.yaml |

4. Check the **NODE** where your app has been deployed.

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

5. Check the NODEPORT on which the application has been exposed

|  |
| --- |
| # kubectl get svc |

**6. 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/java-app-<your-name> -n default 80:8080 &** |

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

**Example**

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