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

1. SSH to the your **AWS WorkStation**.

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

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

Update the image: **lovescloud/docker-python-merc: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 python.

|  |
| --- |
| apiVersion: apps/v1  kind: Deployment  metadata:  name: pythonapp-<your-name>  spec:  selector:  matchLabels:  run: pythonapp-<your-name>  replicas: 2  template:  metadata:  labels:  run: pythonapp-<your-name>  spec:  containers:  - name: pythonapp-<your-name>  Image: lovescloud/docker-python-merc:latest  ports:  - name: port4000  containerPort: 4000  ---  apiVersion: v1  kind: Service  metadata:  name: pythonapp-<your-name>  labels:  run: pythonapp-<your-name>  spec:  type: NodePort  ports:  - name: port4000  port: 4000  protocol: TCP  selector:  run: pythonapp-<your-name> |

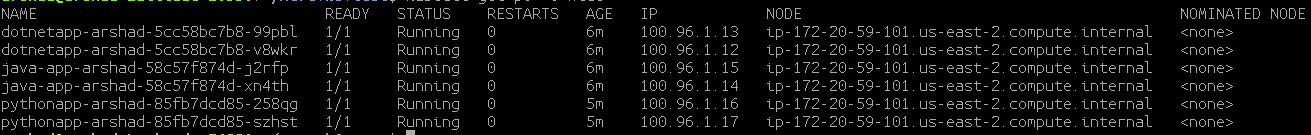
**Save and exit by pressing the ESC key and type wq to save and quit by pressing enter**

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

|  |
| --- |
| # kubectl apply -f python\_app-<your-name>.yaml |

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

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



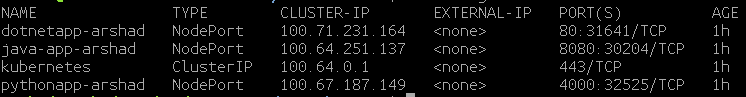
In this example the app has been deployed to the NODE **ip-172-20-59-101.us-east-2.compute.internal**

5. Check the NODEPORT of the application

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

**Example**

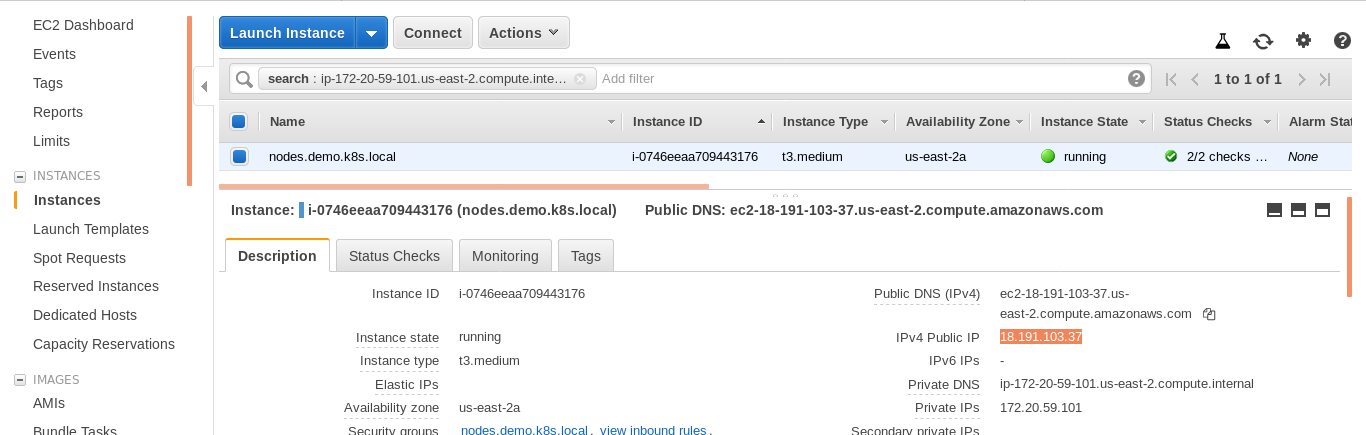
**I**n this example the Python application has been exposed on port **32525** as shown in the below screenshot.

****

7. Login to the **AWS** portal and check the public IP of the NODE (**ip-172-20-59-101.us-east-2.compute.internal**) to access the java application web page from the NODE Public IP address and Node Port on which it is exposed at.

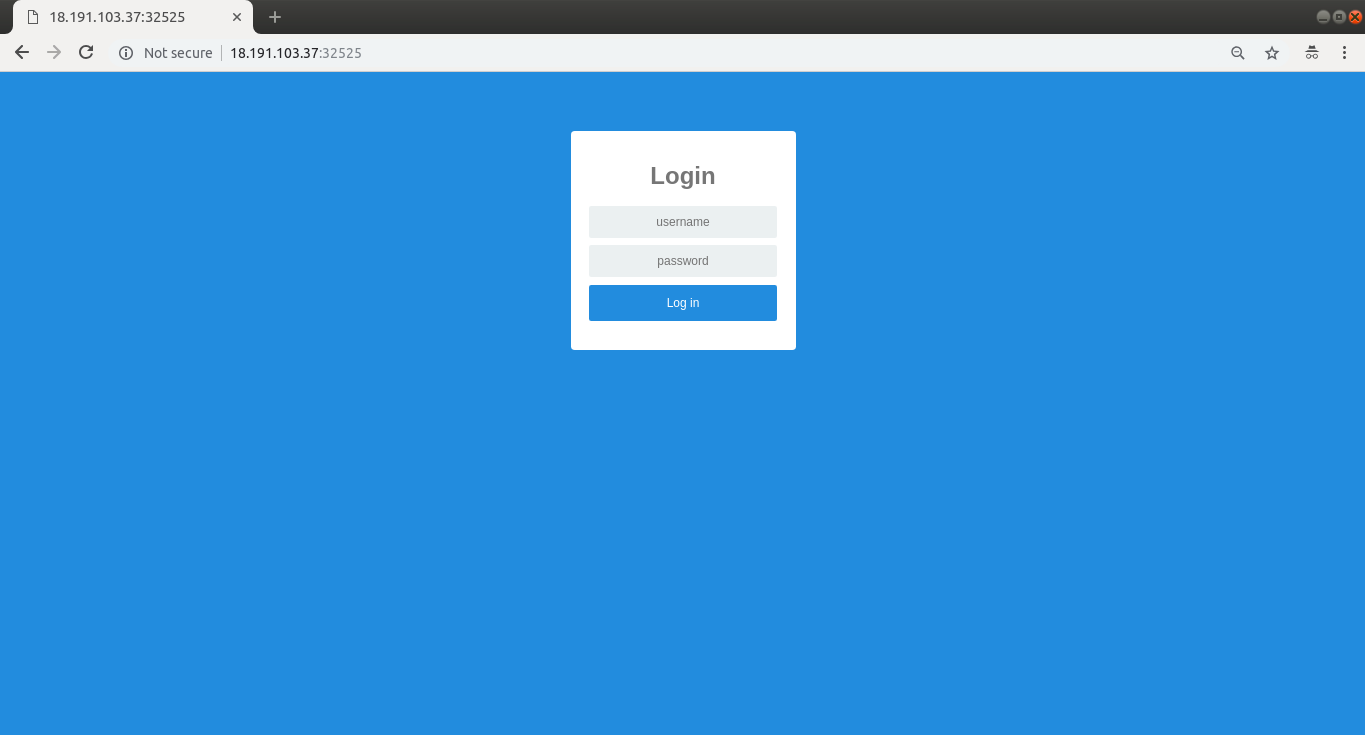
**http://<NODE-PUBLIC-IP>:NODEPORT**

**Search for the NODE where your application has been deployed on the AWS EC2 Dashboard as shown below**

**Access the application from the public IP of the NODE and the NodePort as shown below**

Example.

[**http://18.191.103.37:32525/**](http://18.191.103.37:32525/)



Login with the below credentials.

Username - admin

Password - password

