

ADVANCE DEVOPS EXP 4

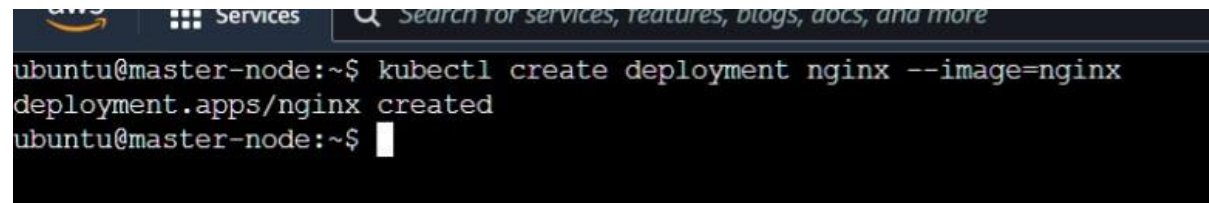
Name :- Ronit Santwani

Roll no :- 49

Aim :- To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application

Step 1: As the cluster is up and running, we can deploy our nginx server on this cluster. Apply this deployment file using this command to create a deployment.

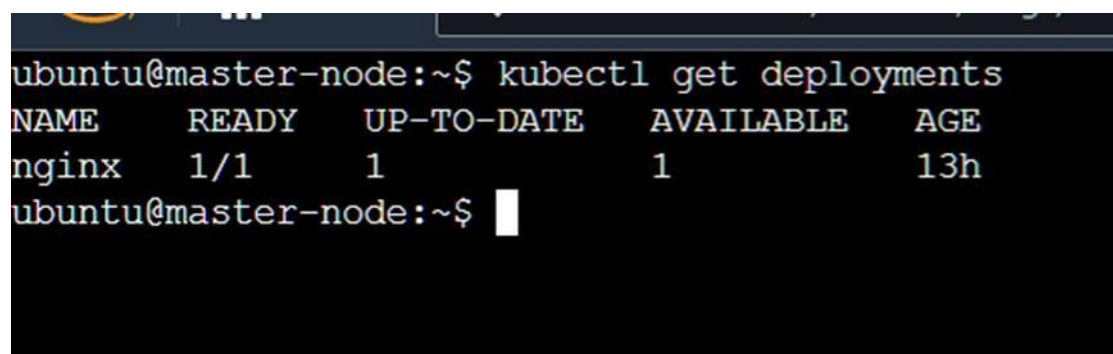
\$kubectl create deployment nginx --image=nginx

A terminal window with a dark background. The prompt is 'ubuntu@master-node:~\$'. The command entered is 'kubectl create deployment nginx --image=nginx'. The output is 'deployment.apps/nginx created'. The prompt is now 'ubuntu@master-node:~\$' with a cursor.

```
ubuntu@master-node:~$ kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
ubuntu@master-node:~$
```

Step 2: Verify the deployment using the command:

\$kubectl get deployments

A terminal window with a dark background. The prompt is 'ubuntu@master-node:~\$'. The command entered is 'kubectl get deployments'. The output is a table with 5 columns: NAME, READY, UP-TO-DATE, AVAILABLE, and AGE. The first row of data is 'nginx', '1/1', '1', '1', and '13h'. The prompt is now 'ubuntu@master-node:~\$' with a cursor.

```
ubuntu@master-node:~$ kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
nginx     1/1     1            1           13h
ubuntu@master-node:~$
```

Step 3: Next, run the following command to create a service named nginx that will expose the app publicly. It will do so through a NodePort, a scheme that will make the pod accessible through an arbitrary port opened on each node of the cluster

with this service-type, Kubernetes will assign this service on ports on the **30000+** range.

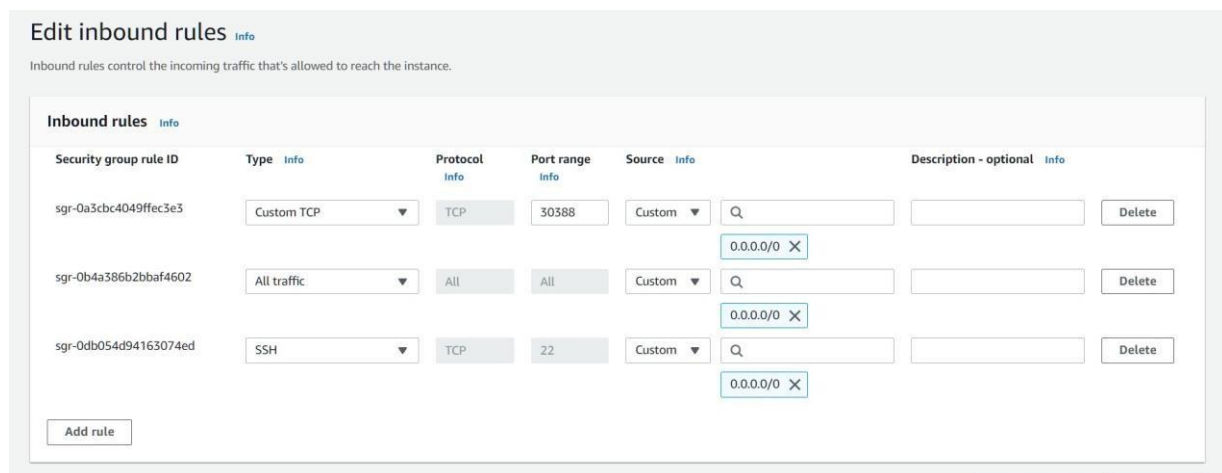
\$kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort

```
aws Services Search for services, features, blogs, docs, and more [Alt+S]
ubuntu@master-node:~$ kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort
service/nginx exposed
ubuntu@master-node:~$
```

Step 4: Run this command to see a summary of the service and the ports exposed.

\$kubectl get services

Step 5: Add the port which is displayed i.e. 30388 (in our case) in the inbound rules of the security group.



Step 6: Now you can verify that the Nginx page is reachable on all nodes using the `curl` command.

```
aws Services Search for services, features, blogs, docs, and more [Alt+S]
ubuntu@master-node:~$ kubectl get services
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes    ClusterIP     10.96.0.1     <none>         443/TCP          3d10h
nginx         NodePort      10.97.253.210 <none>         80:30388/TCP     2m3s
ubuntu@master-node:~$
```

```
ubuntu@master-node:~$ sudo -i
root@master-node:~# curl master-node:30388
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@master-node:~#
```

As you can see, the “**WELCOME TO NGINX!**” page can be reached.

Step 7: To test that everything is working, visit `http://worker_1_ip:nginx_port` or `http://worker_2_ip:nginx_port` through a browser on your local machine. You will see Nginx’s familiar welcome page.

<http://52.90.129.234:30388>

