

ADVANCE DEVOPS EXP 4

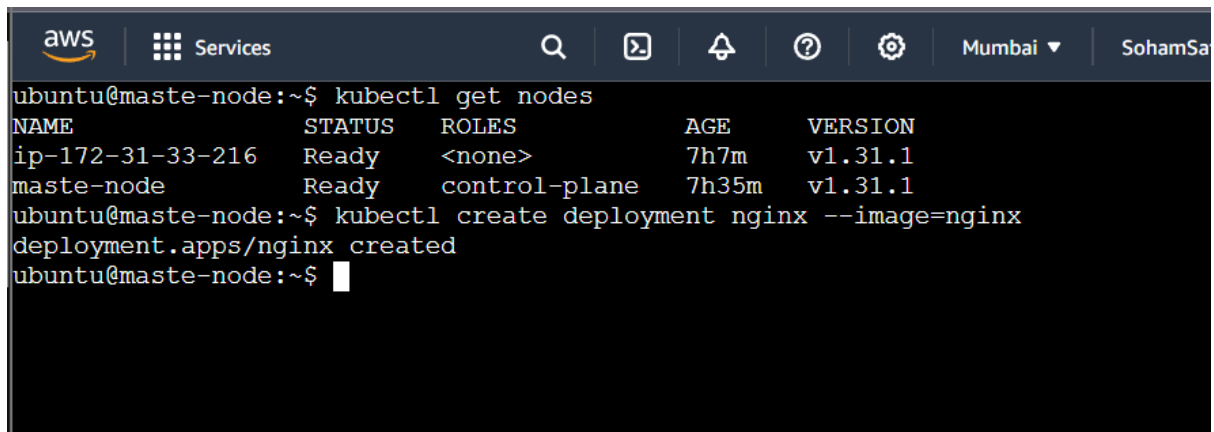
Name :- Soham Satpute

Roll no :- 52

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.

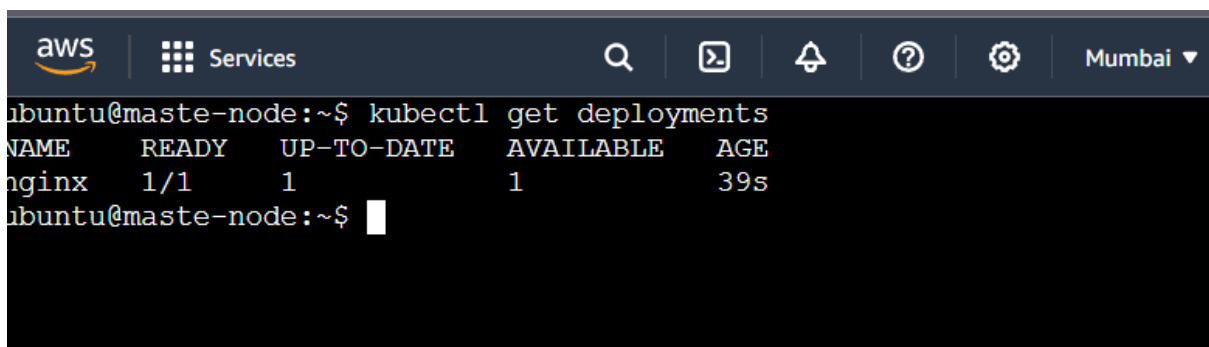
\$kubectl create deployment nginx --image=nginx



```
aws | Services | Search | Copy | Paste | Help | Settings | Mumbai | SohamSa
ubuntu@maste-node:~$ kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
ip-172-31-33-216    Ready    <none>   7h7m  v1.31.1
maste-node          Ready    control-plane  7h35m  v1.31.1
ubuntu@maste-node:~$ kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
ubuntu@maste-node:~$
```

Step 2: Verify the deployment using the command:

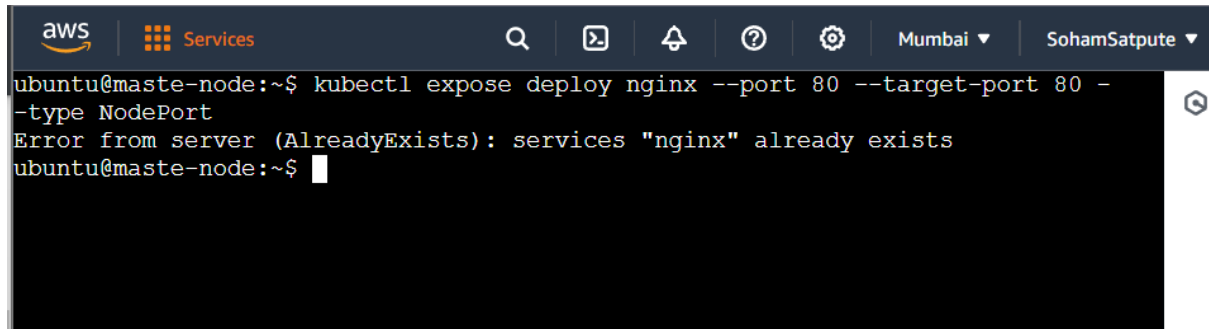
\$kubectl get deployments



```
aws | Services | Search | Copy | Paste | Help | Settings | Mumbai
ubuntu@maste-node:~$ kubectl get deployments
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nginx   1/1     1             1           39s
ubuntu@maste-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 | View | Alert | Help | Settings | Mumbai | SohamSatpute
ubuntu@maste-node:~$ kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort
Error from server (AlreadyExists): services "nginx" already exists
ubuntu@maste-node:~$
```

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

```
$kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	7h49m
nginx	NodePort	10.107.91.218	<none>	80:32756/TCP	46s

```
ubuntu@maste-node:~$
```

Step 5: Add the port which is displayed i.e. 32756(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. As you can see, the “**WELCOME TO NGINX!**” page can be reached.

```
aws | Services | Q | [ ] | [ ] | [ ] | [ ] | Mumbai | SohamSa
ubuntu@maste-node:~$ sudo -i
root@maste-node:~# curl maste-node:32756
<!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@maste-node:~#
```

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://13.127.63.136:32756/>

