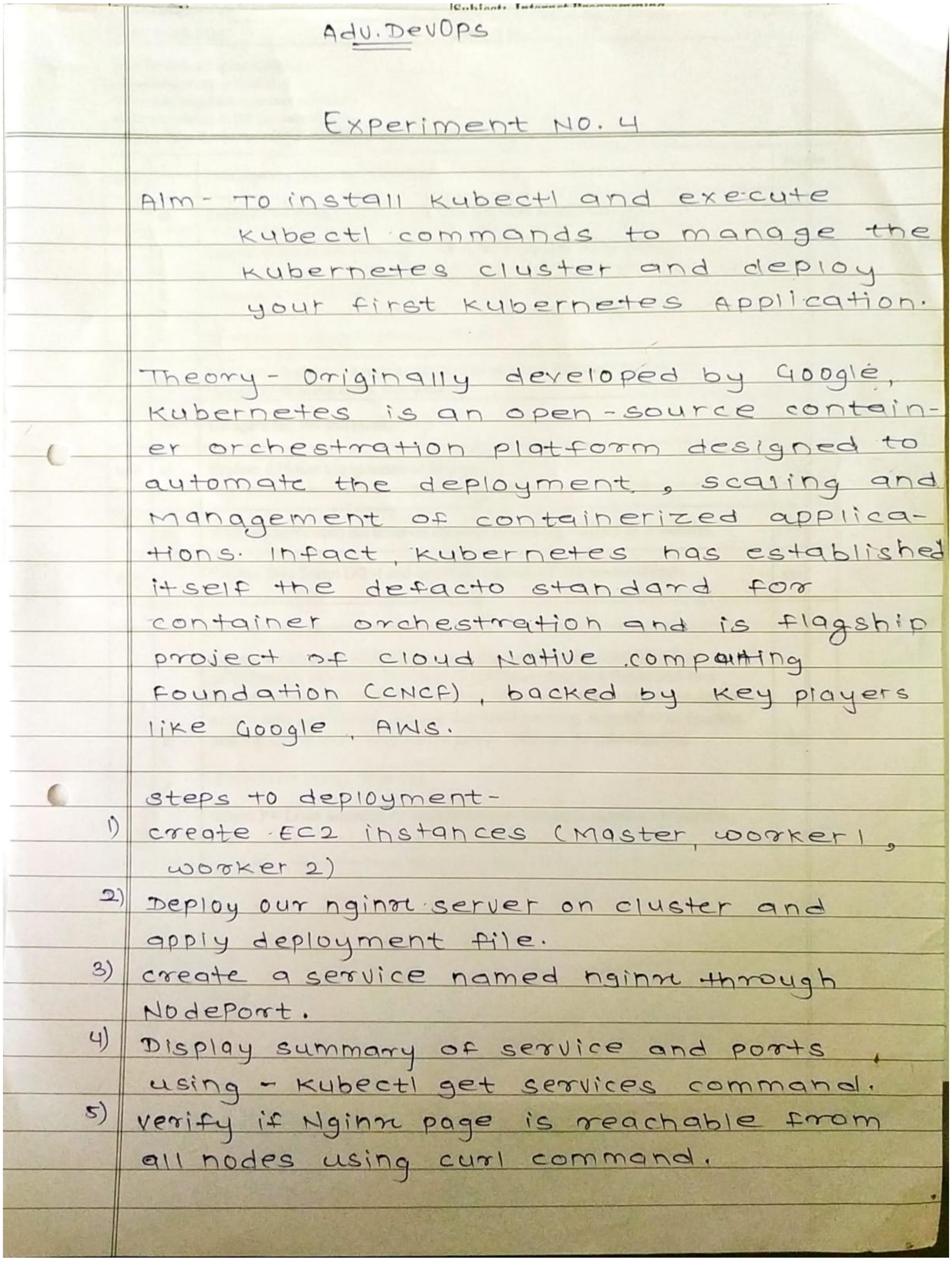
Adv.DevOps Exp 04

Name- Ishika Devare RollNo- 14 Batch- A



Adv.DevOps

Experiment No. 4

Aim - To install kubectl and execute kubectl commands to manage the kubernetes cluster and deploy your first kubernetes Application.

Theory - Originally developed by Google, Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling and management of containerized applications. Infact, Kubernetes has established itself the defacto standard for container orchestration and is flagship project of cloud Native computing Foundation (CNCF), backed by key players like Google, AWS.

steps to deployment -

1) create EC2 instances (Master, worker1, worker 2)
2) Deploy our nginx server on cluster and apply deployment file.
3) create a service named nginx through NodePort.
4) Display summary of service and ports using - kubectl get services command.
5) verify if Nginx page is reachable from all nodes using curl command.

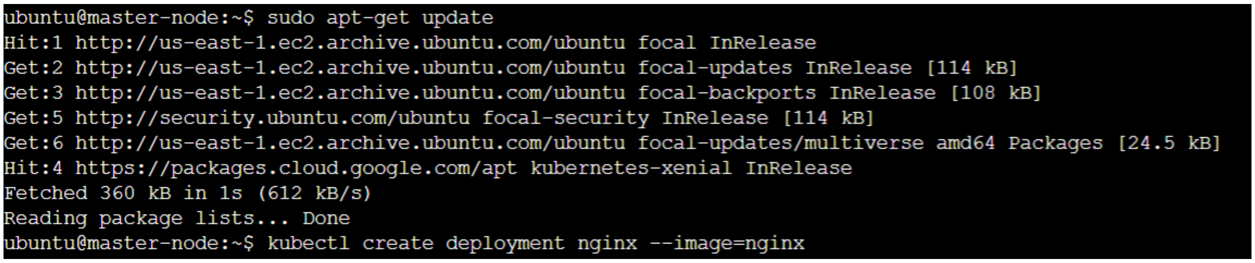
Implementation:

****Note: As we have created master and worker nodes and created a kubernetes cluster as well in our previous experiment 3 so will use that only.****

Running An Application on the Cluster

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.

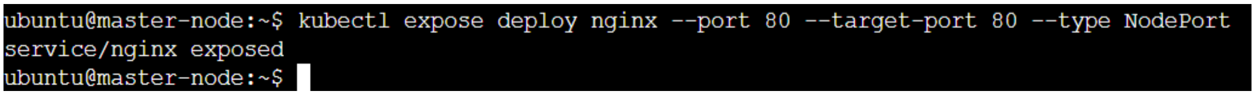
kubernetes-master:~$kubectl create deployment nginx --image=nginx



```
ubuntu@master-node:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [24.5 kB]
Hit:4 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Fetched 360 kB in 1s (612 kB/s)
Reading package lists... Done
ubuntu@master-node:~$ kubectl create deployment nginx --image=nginx
```

Step 2: 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.

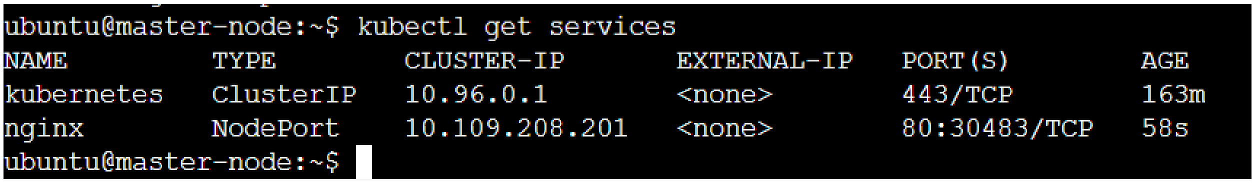
kubernetes-master:~$kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort



```
ubuntu@master-node:~$ kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort
service/nginx exposed
ubuntu@master-node:~$
```

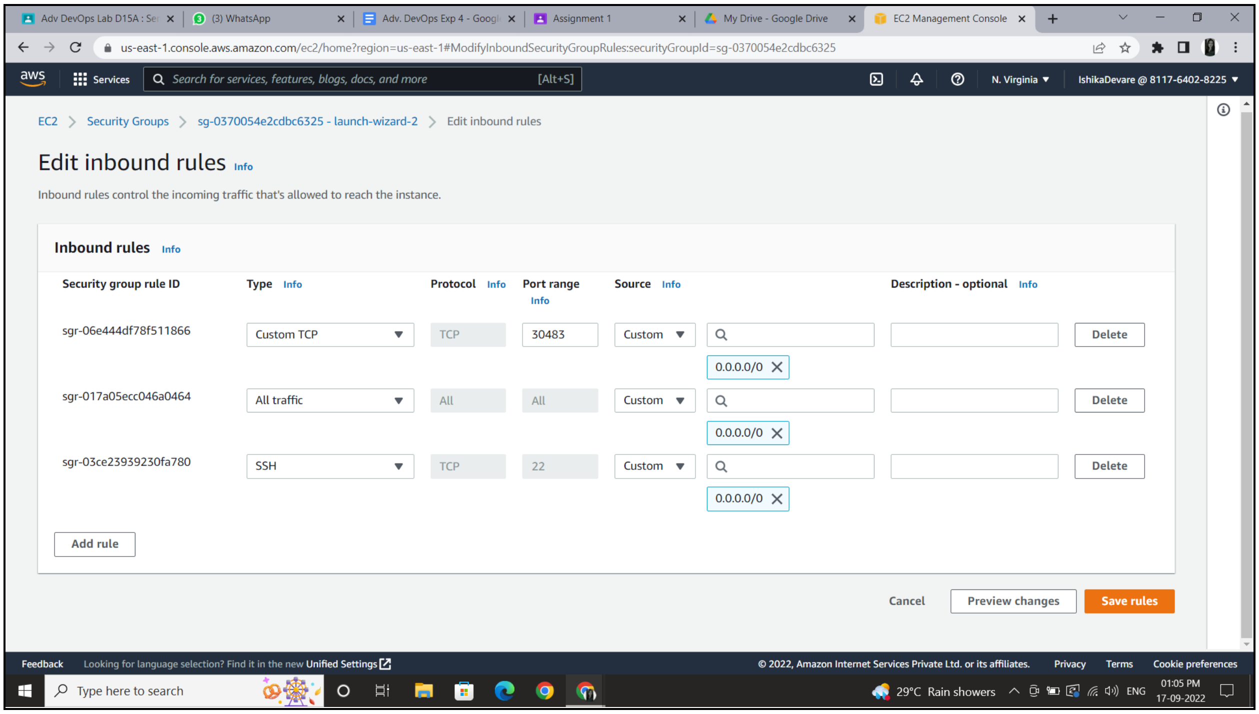
Step 3: Run this command to see a summary of the service and the ports exposed.

kubernetes-master:~$kubectl get services



```
ubuntu@master-node:~$ kubectl get services
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP        163m
nginx        NodePort    10.109.208.201  <none>        80:30483/TCP   58s
ubuntu@master-node:~$
```

Step 4: Add the port which is displayed i.e. 30483 (in my case) in the inbound rules of the security group.



EC2 > Security Groups > sg-0370054e2cdbc6325 - launch-wizard-2 > Edit inbound rules

Edit inbound rules Info

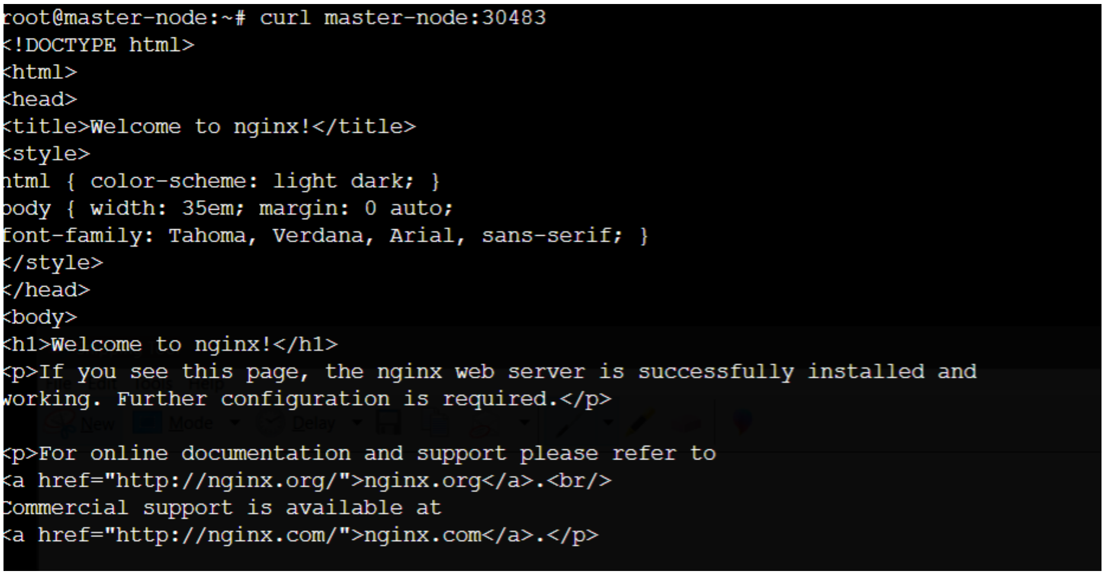
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

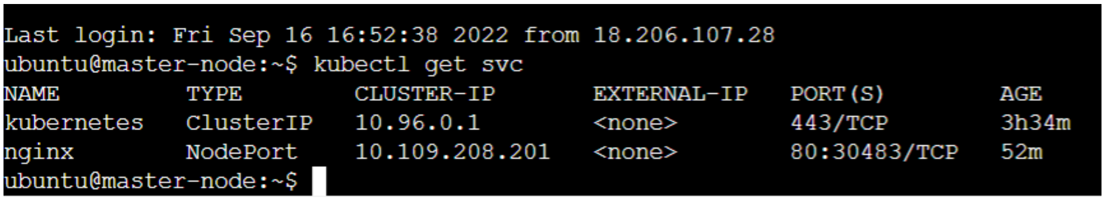
Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-06e444df78f511866	Custom TCP	TCP	30483	Custom	0.0.0.0/0		Delete
sgr-017a05ecc046a0464	All traffic	All	All	Custom	0.0.0.0/0		Delete
sgr-03ce23939230fa780	SSH	TCP	22	Custom	0.0.0.0/0		Delete

Step 5: Now you can verify that the Nginx page is reachable on all nodes using the curl command.

Master

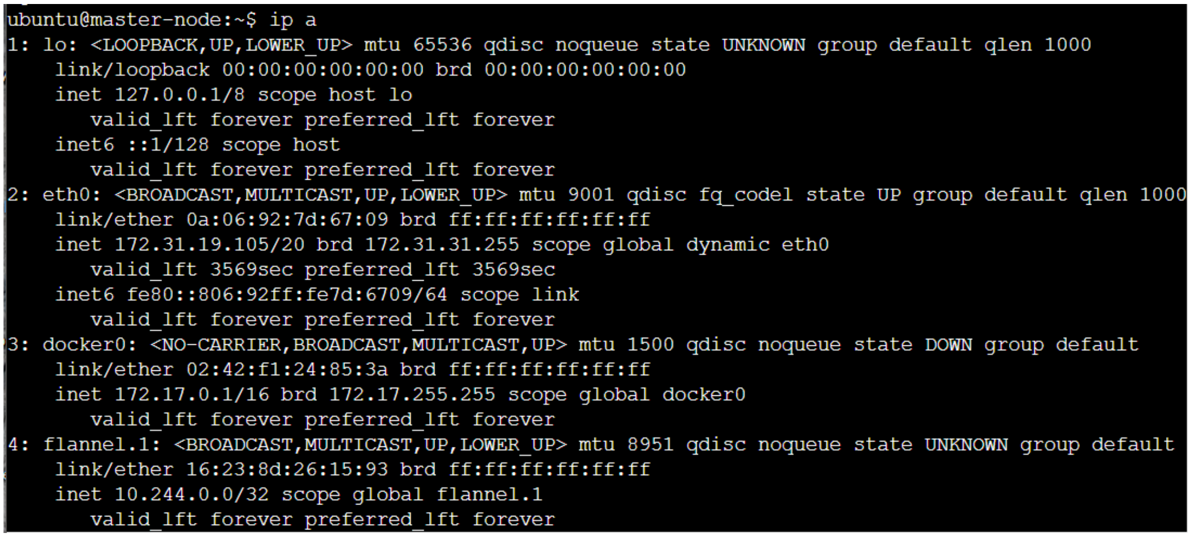


```
root@master-node:~# curl master-node:30483
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you 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>
```



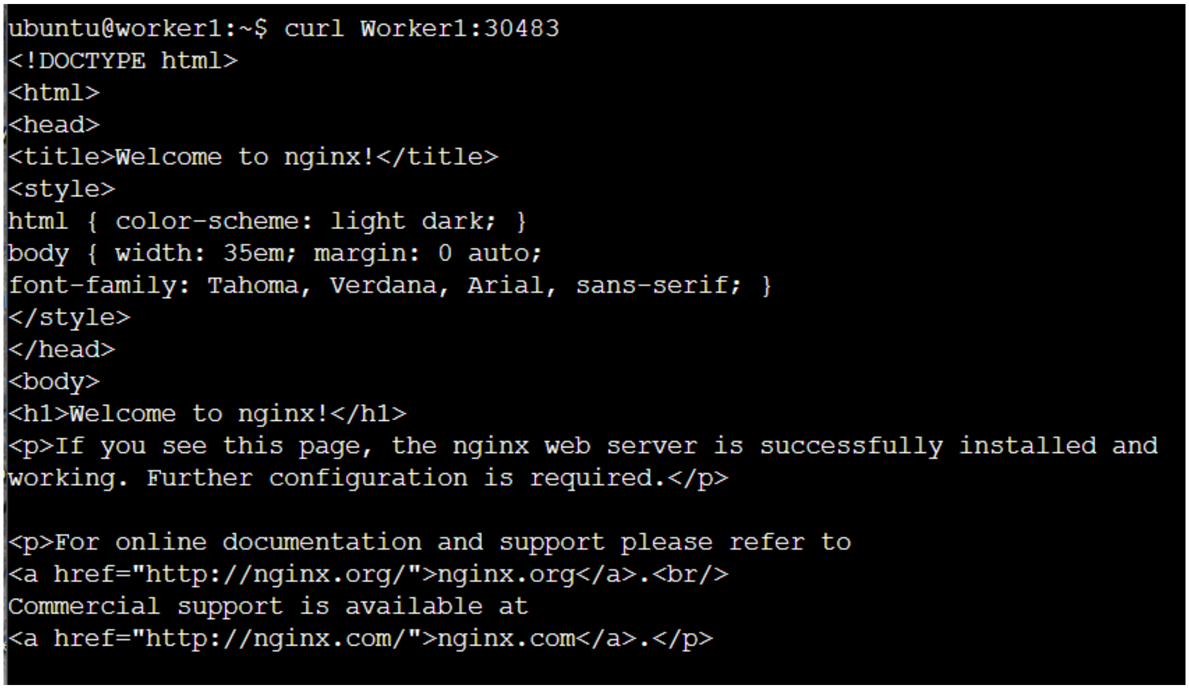
```
Last login: Fri Sep 16 16:52:38 2022 from 18.206.107.28
ubuntu@master-node:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP        3h34m
nginx        NodePort    10.109.208.201   <none>        80:30483/TCP   52m
ubuntu@master-node:~$
```

Ip a



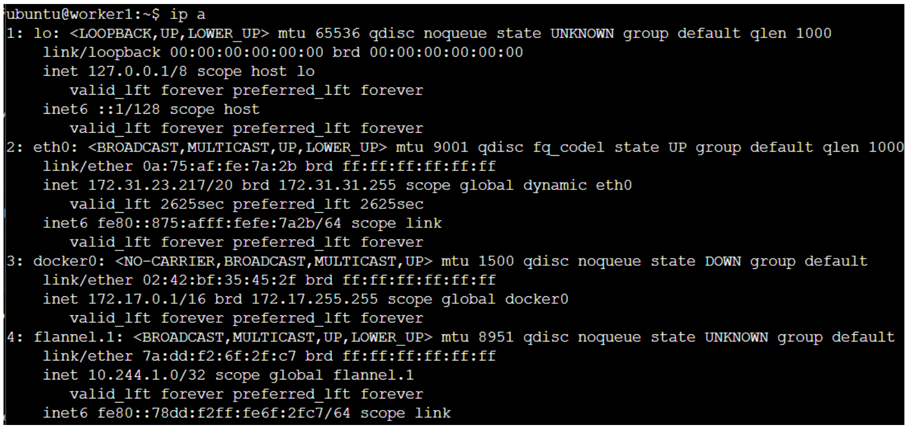
```
ubuntu@master-node:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 0a:06:92:7d:67:09 brd ff:ff:ff:ff:ff:ff
    inet 172.31.19.105/20 brd 172.31.31.255 scope global dynamic eth0
       valid_lft 3569sec preferred_lft 3569sec
    inet6 fe80::806:92ff:fe7d:6709/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:f1:24:85:3a brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
4: flannel.1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8951 qdisc noqueue state UNKNOWN group default
    link/ether 16:23:8d:26:15:93 brd ff:ff:ff:ff:ff:ff
    inet 10.244.0.0/32 scope global flannel.1
       valid_lft forever preferred_lft forever
```

Worker 1



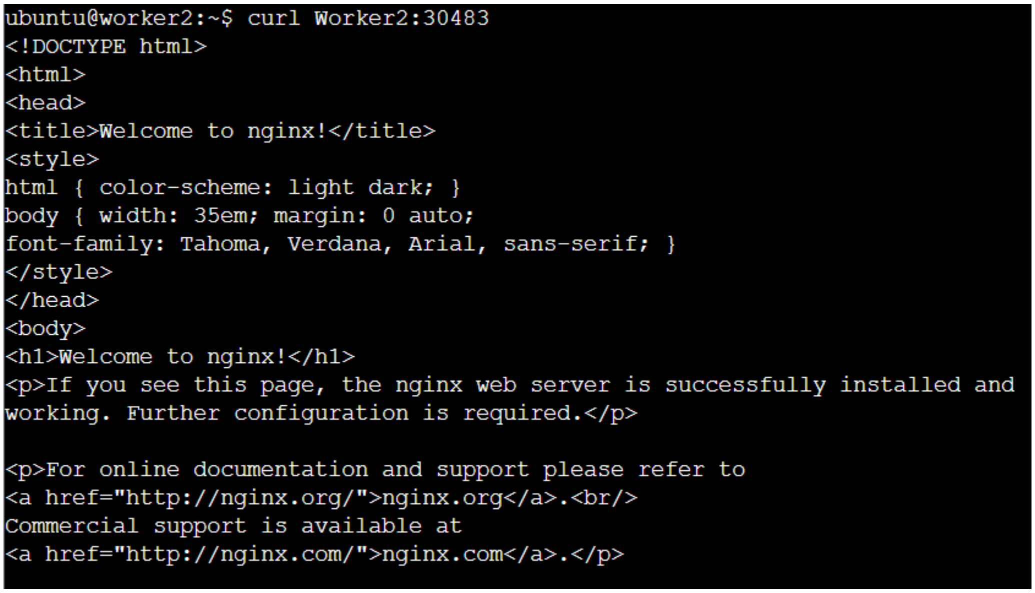
```
ubuntu@worker1:~$ curl Worker1:30483
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this 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>
```

Ip a



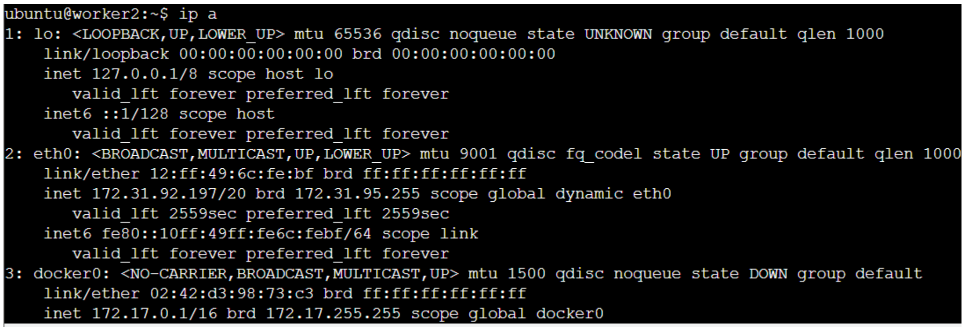
```
ubuntu@worker1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 0a:75:af:fe:7a:2b brd ff:ff:ff:ff:ff:ff
    inet 172.31.23.217/20 brd 172.31.31.255 scope global dynamic eth0
       valid_lft 2625sec preferred_lft 2625sec
    inet6 fe80::875:afff:fefe:7a2b/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:bf:35:45:2f brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
4: flannel.1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8951 qdisc noqueue state UNKNOWN group default
    link/ether 7a:dd:f2:6f:2f:c7 brd ff:ff:ff:ff:ff:ff
    inet 10.244.1.0/32 scope global flannel.1
       valid_lft forever preferred_lft forever
    inet6 fe80::78dd:f2ff:fe6f:2fc7/64 scope link
```

Worker 2



```
ubuntu@worker2:~$ curl Worker2:30483
<!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>
```

Ip a

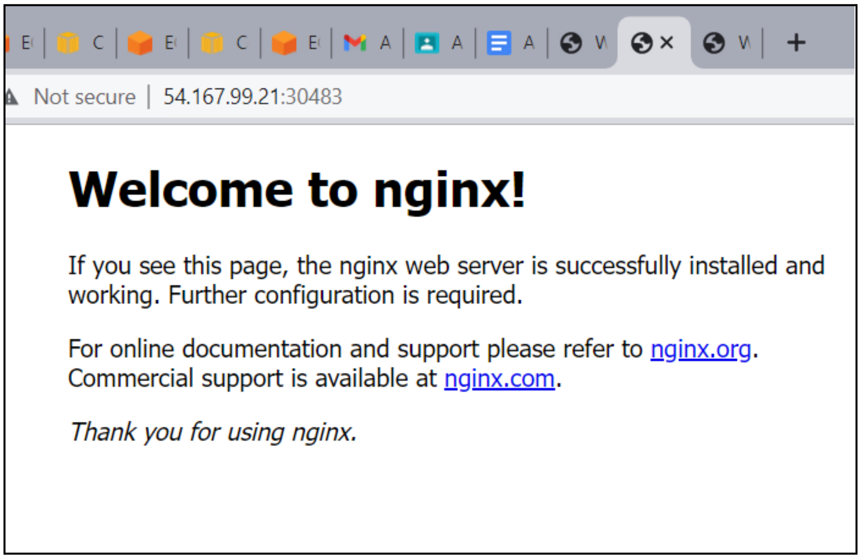


```
ubuntu@worker2:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 12:ff:49:6c:fe:bf brd ff:ff:ff:ff:ff:ff
    inet 172.31.92.197/20 brd 172.31.95.255 scope global dynamic eth0
       valid_lft 2559sec preferred_lft 2559sec
    inet6 fe80::10ff:49ff:fe6c:febf/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:d3:98:73:c3 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
```

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

Step 6: 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.

Worker 1: http://54.167.99.21:30483



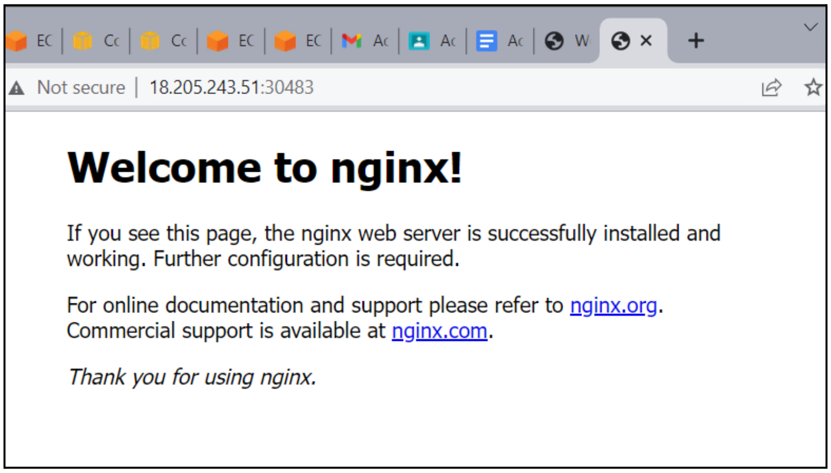
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Worker 1: http://18.205.243.51:30483



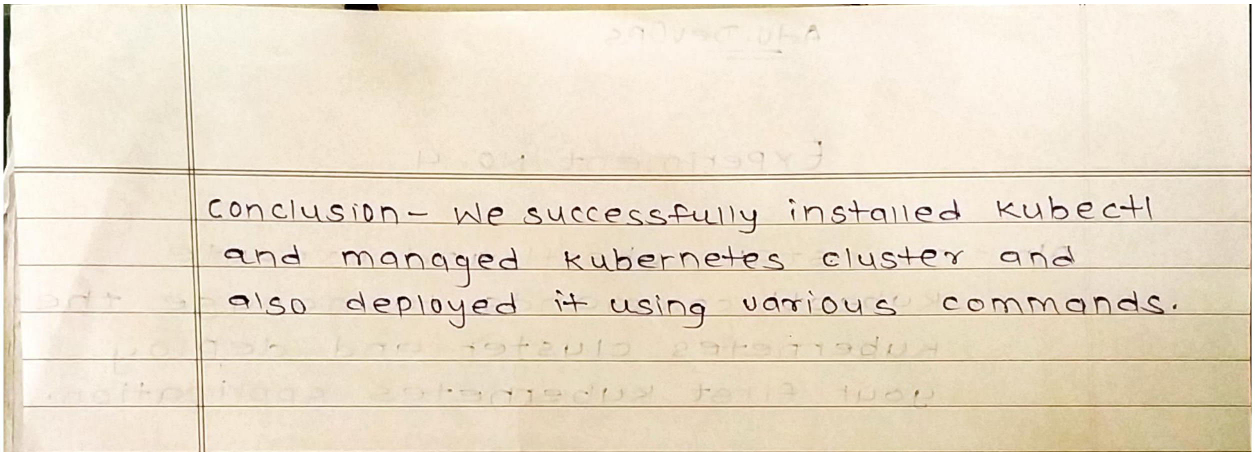
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

****Note: IPs of the machines will change again and again as soon as you stop the instance and start it again.****



Conclusion- We successfully installed kubectl and managed kubernetes cluster and also deployed it using various commands.