

Experiment:4

Name: Mohit Patil Class: D15A Roll no. 37

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

What is Kubernetes?

Kubernetes, often referred to as K8s, is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. Originally developed by Google, it has become the industry standard for managing container workloads due to its flexibility and robust features.

- **Core Concepts of Kubernetes**

- Containers: These are lightweight, portable packages that include everything needed to run an application, ensuring consistency across different environments.
-
- Pods: The smallest deployable units in Kubernetes, pods can contain one or more containers that share storage and network resources.
-
- Nodes: A node is a worker machine in the Kubernetes cluster that runs at least one pod. Nodes can be either physical or virtual machines.
-
- Clusters: A cluster comprises multiple nodes that run containerized applications. The control plane manages the cluster's state.
-
- Services: Services provide stable endpoints for accessing pods and facilitate load balancing and service discovery.
-
- Deployments: A deployment manages the lifecycle of pods, allowing users to specify the number of replicas and facilitating rolling updates and rollbacks.

Role of Kubernetes

What is Kubectl?

Kubectl is the command-line interface used to interact with the Kubernetes API server. It enables users to manage resources within a Kubernetes cluster effectively.

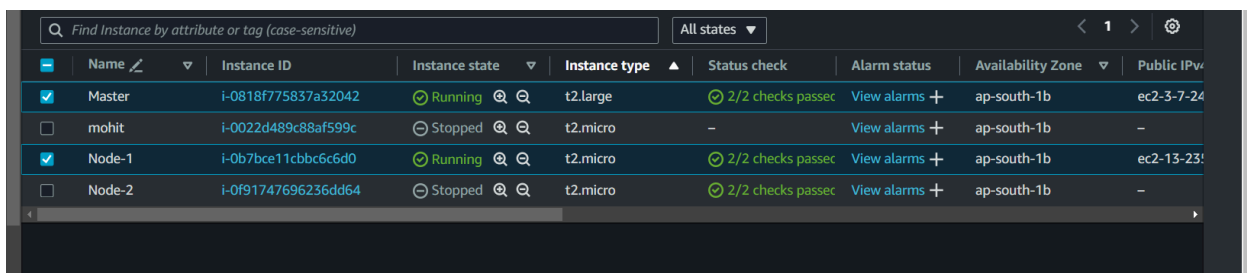
Configuration Files

Configuration files are essential for defining how resources should be created or modified within Kubernetes. Users can employ declarative configurations (using YAML/JSON files) or imperative commands directly in the terminal.

Application Deployment on Kubernetes

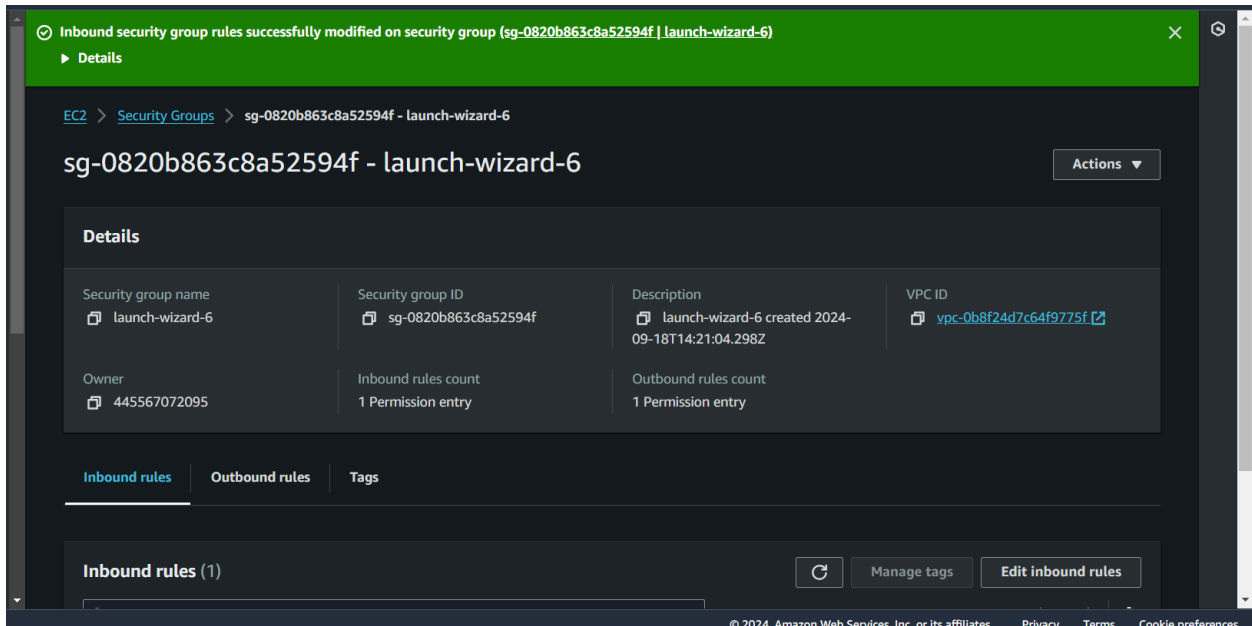
- Define Application Requirements: Identify necessary resources such as CPU, memory, storage, etc.
- Create Deployment Configurations: Write deployment manifests specifying container images, replicas for scaling, health checks, etc.
- Deploying with Kubectl: Use kubectl commands like kubectl apply to deploy applications based on these configurations.
- Monitoring and Scaling Applications: Monitor performance metrics and adjust deployments based on traffic demands.
- Updating Applications: Modify deployment configurations for updates; Kubernetes supports rolling updates by default.
- Rollback Capabilities: If an update causes issues, kubectl allows easy rollback to previous versions using commands like kubectl rollout undo.

Step 1. Creation of 2 EC2 Ubuntu Instances on AWS.

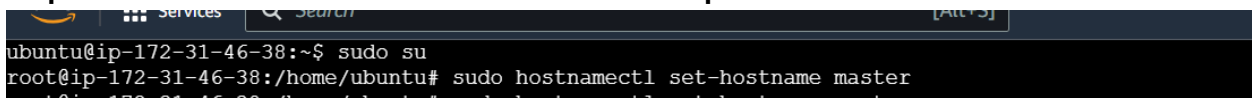


	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input checked="" type="checkbox"/>	Master	i-0818f775837a32042	Running	t2.large	2/2 checks passed	View alarms	ap-south-1b	ec2-3-7-24
<input type="checkbox"/>	mohit	i-0022d489c88af599c	Stopped	t2.micro	-	View alarms	ap-south-1b	-
<input checked="" type="checkbox"/>	Node-1	i-0b7bce11cbbc6c6d0	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1b	ec2-13-23
<input type="checkbox"/>	Node-2	i-0f91747696236dd64	Stopped	t2.micro	2/2 checks passed	View alarms	ap-south-1b	-

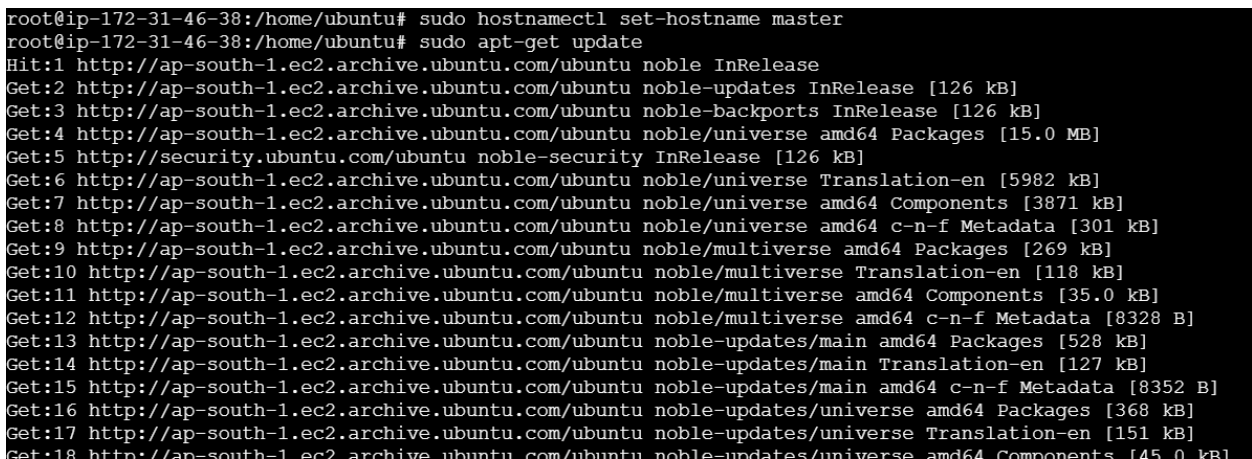
Step 2. Edit inbound rules of security group 'launch-wizard-1' and set 'All Traffic'



Step 3. Set master and worker as hostname on respective servers



Step 4. Installation of docker



```
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 133 not upgraded.
```

```
sudo: systemctl: command not found
root@ip-172-31-46-38:/home/ubuntu# sudo systemctl enable docker
root@ip-172-31-46-38:/home/ubuntu# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-09-18 19:11:49 UTC; 2min 29s ago
 TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Main PID: 2364 (dockerd)
       Tasks: 9
      Memory: 25.7M (peak: 26.0M)
         CPU: 203ms
       CGroup: /system.slice/docker.service
               └─2364 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 18 19:11:49 master systemd[1]: Starting docker.service - Docker Application Container Engine...
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.623361653Z" level=info msg="Starting up"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.623933122Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, so using"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.717378513Z" level=info msg="Loading containers: start."
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.925520997Z" level=info msg="Loading containers: done."
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.943558249Z" level=info msg="Docker daemon" commit=24.0.7-0ubuntu4.1 graphdriver=overlay2 version=24.0.7
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.943655279Z" level=info msg="Daemon has completed initialization"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.989624164Z" level=info msg="API listen on /run/docker.sock"
Sep 18 19:11:49 master systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-21/21 (END)
```

Step 5.Installation of Kubernetes-

```
sudo: systemctl: command not found
root@ip-172-31-46-38:/home/ubuntu# sudo systemctl enable docker
root@ip-172-31-46-38:/home/ubuntu# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-09-18 19:11:49 UTC; 2min 29s ago
 TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Main PID: 2364 (dockerd)
       Tasks: 9
      Memory: 25.7M (peak: 26.0M)
         CPU: 203ms
       CGroup: /system.slice/docker.service
               └─2364 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Sep 18 19:11:49 master systemd[1]: Starting docker.service - Docker Application Container Engine...
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.623361653Z" level=info msg="Starting up"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.623933122Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, so using"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.717378513Z" level=info msg="Loading containers: start."
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.925520997Z" level=info msg="Loading containers: done."
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.943558249Z" level=info msg="Docker daemon" commit=24.0.7-0ubuntu4.1 graphdriver=overlay2 version=24.0.7
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.943655279Z" level=info msg="Daemon has completed initialization"
Sep 18 19:11:49 master dockerd[2364]: time="2024-09-18T19:11:49.989624164Z" level=info msg="API listen on /run/docker.sock"
Sep 18 19:11:49 master systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-21/21 (END)
```

```

root@ip-172-31-46-38:/home/ubuntu# install ca certificate
install: cannot stat 'ca': No such file or directory
root@ip-172-31-46-38:/home/ubuntu# curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo
apt-key add -
cat << EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
usage: sudo -h | -K | -k | -V
usage: sudo -v [-ABkNns] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-ABkNns] [-g group] [-h host] [-p prompt] [-U user]
        [-u user] [command [arg ...]]
usage: sudo [-ABbEHknps] [-r role] [-t type] [-C num] [-D directory]
        [-g group] [-h host] [-p prompt] [-R directory] [-T timeout]
        [-u user] [VAR=value] [-i | -s] [command [arg ...]]
usage: sudo -e [-ABkNns] [-r role] [-t type] [-C num] [-D directory]
        [-g group] [-h host] [-p prompt] [-R directory] [-T timeout]
        [-u user] file ...
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).

```

Step.6 Kubernetes Deployment

```

E: Unable to locate package kubeadm
E: Unable to locate package kubectl
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get install -y apt-transport-https ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
The following additional packages will be installed:
  libcurl3t64-gnutls libcurl4t64
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  curl libcurl3t64-gnutls libcurl4t64
3 upgraded, 1 newly installed, 0 to remove and 130 not upgraded.
Need to get 904 kB of archives.
After this operation, 38.9 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2 [3974 B]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 curl amd64 8.5.0-2ubuntu10.4 [227 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl4t64 amd64 8.5.0-2ubuntu10.4 [341 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl3t64-gnutls amd64 8.5.0-2ubuntu10.4 [333 kB]
Fetched 904 kB in 0s (26.6 MB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.7.14build2_all.deb ...
Unpacking apt-transport-https (2.7.14build2) ...
Setting up apt-transport-https (2.7.14build2) ...

```

```

bash: https://packages.cloud.google.com/apt/doc/apt-key.gpg: No such file or directory
root@ip-172-31-46-38:/home/ubuntu# sudo curl -fsSL /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
root@ip-172-31-46-38:/home/ubuntu#

```

```

bash: /etc/apt/sources.list.d/kubernetes.list: Permission denied
root@ip-172-31-46-38:/home/ubuntu# echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main" | su
do tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main
root@ip-172-31-46-38:/home/ubuntu# sudo apt get update
E: Invalid operation get
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:5 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:6 https://packages.cloud.google.com/apt kubernetes-focal Release
      404 Not Found [IP: 142.251.42.14 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
root@ip-172-31-46-38:/home/ubuntu#

```

```

E: Unable to locate package kubect1
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:5 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:6 https://packages.cloud.google.com/apt kubernetes-focal Release
  404 Not Found [IP: 142.250.192.142 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
root@ip-172-31-46-38:/home/ubuntu# sudo apt-get install -y kubect1
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package kubect1
root@ip-172-31-46-38:/home/ubuntu#


```

Extra:

```

kubect1 1.31.1 from Canonical✓ installed
root@ip-172-31-46-38:/home/ubuntu# kubect1 version --client
Client Version: v1.31.1
Kustomize Version: v5.4.2
root@ip-172-31-46-38:/home/ubuntu#

```


Services
Q
📧
🔔
?
⚙️
N. Virginia
ShravaniAnilPatil

```

table kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubect1 apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each
as root:

kubeadm join 172.31.33.243:6443 --token ltp4ao.gzd zdpkqly6bxv nz \
--discovery-token-ca-cert-hash sha256:4423cf44f5102d477fa92160e76e03f
78f941d4e7a5836cd46bb14517dfab5ad
ubuntu@master-node:~$

```

i-0aef82b0ccd222219 (master1)
×

PublicIPs: 34.207.105.187
PrivateIPs: 172.31.33.243

Step 7. Deploy Pod Network to Cluster and Join Worker Node to Cluster

```
aws | Services | Search | [Alt+S] | N. Virginia | ShrivaniAnilPatil
ubuntu@worker-nod:~$ sudo kubeadm join 172.31.33.243:6443 --token ltp4ao.gzdmdpkyly6bxvni --discovery-token-ca-cert-hash sha256:4423cf44f5102d477fa92160e76e03f78f941d4e7a5836cd46bb14517draf5ad --ignore-preflight-errors=all
[preflight] Running pre-flight checks
[WARNING FileExisting-socat]: socat not found in system path
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 513.568999ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiservert and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@worker-nod:~$
```

Step 8. Create one file deploy.yaml

```
root@ip-172-31-46-38:/home/ubuntu# sudo nano deploy.yaml
root@ip-172-31-46-38:/home/ubuntu#
```

```

ubuntu@master-node:~$ cat deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
ubuntu@master-node:~$

```

Step 9 : Create Deployment

```

cat deploy.yaml:1:1: connection refused
The connection to the server localhost:8080 was refused - did you specify the right host or port?
root@ip-172-31-46-38:/home/ubuntu# kubectl create -f deploy.yaml

```

```

service/nginx-deployment exposed
ubuntu@master-node:~$ kubectl get svc

```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4h43m
nginx-deployment	LoadBalancer	10.101.59.94	<pending>	80:31041/TCP	4m34s

```

ubuntu@master-node:~$

```




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.