**Kubernetes Assignments**

**[**

1. **Assignment one**
2. **Deploy a Kubernetes Cluster for 3 nodes**
3. **Create a nginx deployment fo 3 replicas**

Master node configuration

Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1031-aws x86\_64)

\* Documentation: https://help.ubuntu.com

\* Management: https://landscape.canonical.com

\* Support: https://ubuntu.com/advantage

System information as of Mon Apr 10 04:24:21 UTC 2023

System load: 0.13671875 Processes: 112

Usage of /: 20.2% of 7.57GB Users logged in: 0

Memory usage: 6% IPv4 address for eth0: 172.31.18.114

Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.

See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.

To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;

the exact distribution terms for each program are described in the

individual files in /usr/share/doc/\*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by

applicable law.

To run a command as administrator (user "root"), use "sudo <command>".

See "man sudo\_root" for details.

ubuntu@ip-172-31-18-114:~$ sudo apt update -y

Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease

Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]

Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]

Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]

Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]

Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]

Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]

Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]

Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]

Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]

Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [990 kB]

Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [210 kB]

Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [13.9 kB]

Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [744 kB]

Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [115 kB]

Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [576 B]

Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [899 kB]

Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [180 kB]

Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [18.6 kB]

Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [24.1 kB]

Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [6312 B]

Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [444 B]

Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [40.6 kB]

Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [9800 B]

Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]

Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]

Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [20.3 kB]

Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [14.4 kB]

Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [480 B]

Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]

Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [728 kB]

Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [147 kB]

Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [9020 B]

Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [701 kB]

Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [109 kB]

Get:36 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [576 B]

Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [716 kB]

Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [118 kB]

Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [14.2 kB]

Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [19.4 kB]

Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [4068 B]

Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [228 B]

Fetched 26.6 MB in 4s (6721 kB/s)

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

12 packages can be upgraded. Run 'apt list --upgradable' to see them.

ubuntu@ip-172-31-18-114:~$ **sudo apt install docker.io -y**

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-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 12 not upgraded.

Need to get 72.4 MB of archives.

After this operation, 287 MB of additional disk space will be used.

Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]

Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]

Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.4-0ubuntu1~22.04.1 [4241 kB]

Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.6.12-0ubuntu1~22.04.1 [34.4 MB]

Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 dns-root-data all 2021011101 [5256 B]

Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dnsmasq-base amd64 2.86-1.1ubuntu0.2 [354 kB]

Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io amd64 20.10.21-0ubuntu1~22.04.2 [33.2 MB]

Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]

Fetched 72.4 MB in 2s (45.4 MB/s)

Preconfiguring packages ...

Selecting previously unselected package pigz.

(Reading database ... 63657 files and directories currently installed.)

Preparing to unpack .../0-pigz\_2.6-1\_amd64.deb ...

Unpacking pigz (2.6-1) ...

Selecting previously unselected package bridge-utils.

Preparing to unpack .../1-bridge-utils\_1.7-1ubuntu3\_amd64.deb ...

Unpacking bridge-utils (1.7-1ubuntu3) ...

Selecting previously unselected package runc.

Preparing to unpack .../2-runc\_1.1.4-0ubuntu1~22.04.1\_amd64.deb ...

Unpacking runc (1.1.4-0ubuntu1~22.04.1) ...

Selecting previously unselected package containerd.

Preparing to unpack .../3-containerd\_1.6.12-0ubuntu1~22.04.1\_amd64.deb ...

Unpacking containerd (1.6.12-0ubuntu1~22.04.1) ...

Selecting previously unselected package dns-root-data.

Preparing to unpack .../4-dns-root-data\_2021011101\_all.deb ...

Unpacking dns-root-data (2021011101) ...

Selecting previously unselected package dnsmasq-base.

Preparing to unpack .../5-dnsmasq-base\_2.86-1.1ubuntu0.2\_amd64.deb ...

Unpacking dnsmasq-base (2.86-1.1ubuntu0.2) ...

Selecting previously unselected package docker.io.

Preparing to unpack .../6-docker.io\_20.10.21-0ubuntu1~22.04.2\_amd64.deb ...

Unpacking docker.io (20.10.21-0ubuntu1~22.04.2) ...

Selecting previously unselected package ubuntu-fan.

Preparing to unpack .../7-ubuntu-fan\_0.12.16\_all.deb ...

Unpacking ubuntu-fan (0.12.16) ...

Setting up dnsmasq-base (2.86-1.1ubuntu0.2) ...

Setting up runc (1.1.4-0ubuntu1~22.04.1) ...

Setting up dns-root-data (2021011101) ...

Setting up bridge-utils (1.7-1ubuntu3) ...

Setting up pigz (2.6-1) ...

Setting up containerd (1.6.12-0ubuntu1~22.04.1) ...

Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.

Setting up ubuntu-fan (0.12.16) ...

Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.

Setting up docker.io (20.10.21-0ubuntu1~22.04.2) ...

Adding group `docker' (GID 122) ...

Done.

Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.

Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.

Processing triggers for dbus (1.12.20-2ubuntu4.1) ...

Processing triggers for man-db (2.10.2-1) ...

Scanning processes...

Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

ubuntu@ip-172-31-18-114:~$ **sudo systemctl start docker**

ubuntu@ip-172-31-18-114:~$ **sudo systemctl enable docker**

ubuntu@ip-172-31-18-114:~$ ^[[200~sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg~

sudo: command not found

ubuntu@ip-172-31-18-114:~$ **sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg**

ubuntu@ip-172-31-18-114:~$ **echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list**

deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main

ubuntu@ip-172-31-18-114:~$ ^C

ubuntu@ip-172-31-18-114:~$

ubuntu@ip-172-31-18-114:~$ **sudo apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y**

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

No apt package "kubeadm", but there is a snap with that name.

Try "snap install kubeadm"

No apt package "kubectl", but there is a snap with that name.

Try "snap install kubectl"

No apt package "kubelet", but there is a snap with that name.

Try "snap install kubelet"

E: Unable to locate package kubeadm

E: Unable to locate package kubectl

E: Unable to locate package kubelet

ubuntu@ip-172-31-18-114:~$ **sudo apt update -y**

Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease

Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease

Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease

Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease

Get:5 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [8993 B]

Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [64.5 kB]

Fetched 73.5 kB in 1s (138 kB/s)

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

12 packages can be upgraded. Run 'apt list --upgradable' to see them.

ubuntu@ip-172-31-18-114:~$ **sudo apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y**

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

The following additional packages will be installed:

conntrack cri-tools ebtables kubernetes-cni socat

The following NEW packages will be installed:

conntrack cri-tools ebtables kubeadm kubectl kubelet kubernetes-cni socat

0 upgraded, 8 newly installed, 0 to remove and 12 not upgraded.

Need to get 81.5 MB of archives.

After this operation, 318 MB of additional disk space will be used.

Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 conntrack amd64 1:1.4.6-2build2 [33.5 kB]

Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 ebtables amd64 2.0.11-4build2 [84.9 kB]

Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 socat amd64 1.7.4.1-3ubuntu4 [349 kB]

Get:4 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 cri-tools amd64 1.26.0-00 [18.9 MB]

Get:5 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubernetes-cni amd64 1.2.0-00 [27.6 MB]

Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubelet amd64 1.20.0-00 [18.8 MB]

Get:7 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubectl amd64 1.20.0-00 [7942 kB]

Get:8 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 kubeadm amd64 1.20.0-00 [7707 kB]

Fetched 81.5 MB in 2s (33.7 MB/s)

Selecting previously unselected package conntrack.

(Reading database ... 64012 files and directories currently installed.)

Preparing to unpack .../0-conntrack\_1%3a1.4.6-2build2\_amd64.deb ...

Unpacking conntrack (1:1.4.6-2build2) ...

Selecting previously unselected package cri-tools.

Preparing to unpack .../1-cri-tools\_1.26.0-00\_amd64.deb ...

Unpacking cri-tools (1.26.0-00) ...

Selecting previously unselected package ebtables.

Preparing to unpack .../2-ebtables\_2.0.11-4build2\_amd64.deb ...

Unpacking ebtables (2.0.11-4build2) ...

Selecting previously unselected package kubernetes-cni.

Preparing to unpack .../3-kubernetes-cni\_1.2.0-00\_amd64.deb ...

Unpacking kubernetes-cni (1.2.0-00) ...

Selecting previously unselected package socat.

Preparing to unpack .../4-socat\_1.7.4.1-3ubuntu4\_amd64.deb ...

Unpacking socat (1.7.4.1-3ubuntu4) ...

Selecting previously unselected package kubelet.

Preparing to unpack .../5-kubelet\_1.20.0-00\_amd64.deb ...

Unpacking kubelet (1.20.0-00) ...

Selecting previously unselected package kubectl.

Preparing to unpack .../6-kubectl\_1.20.0-00\_amd64.deb ...

Unpacking kubectl (1.20.0-00) ...

Selecting previously unselected package kubeadm.

Preparing to unpack .../7-kubeadm\_1.20.0-00\_amd64.deb ...

Unpacking kubeadm (1.20.0-00) ...

Setting up conntrack (1:1.4.6-2build2) ...

Setting up kubectl (1.20.0-00) ...

Setting up ebtables (2.0.11-4build2) ...

Setting up socat (1.7.4.1-3ubuntu4) ...

Setting up cri-tools (1.26.0-00) ...

Setting up kubernetes-cni (1.2.0-00) ...

Setting up kubelet (1.20.0-00) ...

Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /lib/systemd/system/kubelet.service.

Setting up kubeadm (1.20.0-00) ...

Processing triggers for man-db (2.10.2-1) ...

Scanning processes...

Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host

ubuntu@ip-172-31-18-114:~$ **sudo su**

root@ip-172-31-18-114:/home/ubuntu# **hostnamectl set-hostname master**

root@ip-172-31-18-114:/home/ubuntu# exec bash

**root@master:/home/ubuntu# kubeadm init**

I0410 05:20:04.814619 5694 version.go:251] remote version is much newer: v1.26.3; falling back to: stable-1.20

[init] Using Kubernetes version: v1.20.15

[preflight] Running pre-flight checks

[WARNING SystemVerification]: this Docker version is not on the list of validated versions: 20.10.21. Latest validated version: 19.03

[preflight] Pulling images required for setting up a Kubernetes cluster

[preflight] This might take a minute or two, depending on the speed of your internet connection

[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'

[certs] Using certificateDir folder "/etc/kubernetes/pki"

[certs] Generating "ca" certificate and key

[certs] Generating "apiserver" certificate and key

[certs] apiserver serving cert is signed for DNS names [kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local master] and IPs [10.96.0.1 172.31.18.114]

[certs] Generating "apiserver-kubelet-client" certificate and key

[certs] Generating "front-proxy-ca" certificate and key

[certs] Generating "front-proxy-client" certificate and key

[certs] Generating "etcd/ca" certificate and key

[certs] Generating "etcd/server" certificate and key

[certs] etcd/server serving cert is signed for DNS names [localhost master] and IPs [172.31.18.114 127.0.0.1 ::1]

[certs] Generating "etcd/peer" certificate and key

[certs] etcd/peer serving cert is signed for DNS names [localhost master] and IPs [172.31.18.114 127.0.0.1 ::1]

[certs] Generating "etcd/healthcheck-client" certificate and key

[certs] Generating "apiserver-etcd-client" certificate and key

[certs] Generating "sa" key and public key

[kubeconfig] Using kubeconfig folder "/etc/kubernetes"

[kubeconfig] Writing "admin.conf" kubeconfig file

[kubeconfig] Writing "kubelet.conf" kubeconfig file

[kubeconfig] Writing "controller-manager.conf" kubeconfig file

[kubeconfig] Writing "scheduler.conf" kubeconfig file

[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"

[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"

[kubelet-start] Starting the kubelet

[control-plane] Using manifest folder "/etc/kubernetes/manifests"

[control-plane] Creating static Pod manifest for "kube-apiserver"

[control-plane] Creating static Pod manifest for "kube-controller-manager"

[control-plane] Creating static Pod manifest for "kube-scheduler"

[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"

[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can take up to 4m0s

[apiclient] All control plane components are healthy after 11.503257 seconds

[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace

[kubelet] Creating a ConfigMap "kubelet-config-1.20" in namespace kube-system with the configuration for the kubelets in the cluster

[upload-certs] Skipping phase. Please see --upload-certs

[mark-control-plane] Marking the node master as control-plane by adding the labels "node-role.kubernetes.io/master=''" and "node-role.kubernetes.io/control-plane='' (deprecated)"

[mark-control-plane] Marking the node master as control-plane by adding the taints [node-role.kubernetes.io/master:NoSchedule]

[bootstrap-token] Using token: q67ewe.116cupfeda2vy21x

[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles

[bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to get nodes

[bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials

[bootstrap-token] configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token

[bootstrap-token] configured RBAC rules to allow certificate rotation for all node client certificates in the cluster

[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace

[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable 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 "kubectl 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 172.31.18.114:6443 --token q67ewe.116cupfeda2vy21x \

--discovery-token-ca-cert-hash sha256:18598e9b74c911a3383d72d0894b95d6a9ecdb168c36d9ecb0b1b5997570f801

root@master:/home/ubuntu#

root@master:/home/ubuntu# mkdir -p $HOME/.kube

root@master:/home/ubuntu# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

root@master:/home/ubuntu# sudo chown $(id -u):$(id -g) $HOME/.kube/config

root@master:/home/ubuntu# kubectl apply -f https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset-k8s.yaml

serviceaccount/weave-net created

clusterrole.rbac.authorization.k8s.io/weave-net created

clusterrolebinding.rbac.authorization.k8s.io/weave-net created

role.rbac.authorization.k8s.io/weave-net created

rolebinding.rbac.authorization.k8s.io/weave-net created

daemonset.apps/weave-net created

root@master:/home/ubuntu# kubectl get nodes -o wide

NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION CONTAINER-RUNTIME

ip-172-31-23-95 Ready <none> 10m v1.20.0 172.31.23.95 <none> Ubuntu 22.04.2 LTS 5.15.0-1031-aws docker://20.10.21

ip-172-31-29-171 Ready <none> 10m v1.20.0 172.31.29.171 <none> Ubuntu 22.04.2 LTS 5.15.0-1031-aws docker://20.10.21

master Ready control-plane,master 13m v1.20.0 172.31.18.114 <none> Ubuntu 22.04.2 LTS 5.15.0-1031-aws docker://20.10.21

root@master:/home/ubuntu# exit

exit

ubuntu@ip-172-31-18-114:~$ sudo nano nginxdeployment.yml

ubuntu@ip-172-31-18-114:~$ kubectl create -f nginxdeployment.yml

The connection to the server localhost:8080 was refused - did you specify the right host or port?

ubuntu@ip-172-31-18-114:~$ sudo nano nginxdeployment.yml

ubuntu@ip-172-31-18-114:~$ kubectl create -f nginxdeployment.yml

The connection to the server localhost:8080 was refused - did you specify the right host or port?

ubuntu@ip-172-31-18-114:~$ kubectl create deployment nginx --image=nginx --port=80

error: failed to create deployment: Post "http://localhost:8080/apis/apps/v1/namespaces/default/deployments?fieldManager=kubectl-create": dial tcp 127.0.0.1:8080: connect: connection refused

ubuntu@ip-172-31-18-114:~$ sudo su

root@master:/home/ubuntu# exit

exit

ubuntu@ip-172-31-18-114:~$ sudo kubectl create -f nginxdeployment.yml

deployment.apps/nginx-deployment created

ubuntu@ip-172-31-18-114:~$ kubectl get pods -o wide

The connection to the server localhost:8080 was refused - did you specify the right host or port?

ubuntu@ip-172-31-18-114:~$ sudo kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

nginx-deployment-7848d4b86f-952d9 1/1 Running 0 98s 10.46.0.3 ip-172-31-29-171 <none> <none>

nginx-deployment-7848d4b86f-gwjhn 1/1 Running 0 98s 10.40.0.2 ip-172-31-23-95 <none> <none>

nginx-deployment-7848d4b86f-v6nlz 1/1 Running 0 98s 10.46.0.2 ip-172-31-29-171 <none> <none>

ubuntu@ip-172-31-18-114:~$

ubuntu@ip-172-31-18-114:~$ **curl 10.46.0.3**

<!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>

1. **Kuber-Slave 1 node configuration**

ubuntu@ip-172-31-29-171:~$ history

1 sudo apt update -y

2 sudo apt install docker.io -y

3 sudo systemctl start docker

4 sudo systemctl enable docker

5 sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg

6 echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

7 sudo apt update -y

8 sudo apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y

ubuntu@ip-172-31-29-171:~$ sudo su

root@ip-172-31-29-171:/home/ubuntu# kubeadm join 172.31.18.114:6443 --token q67ewe.116cupfeda2vy21x \

--discovery-token-ca-cert-hash sha256:18598e9b74c911a3383d72d0894b95d6a9ecdb168c36d9ecb0b1b5997570f801

[preflight] Running pre-flight checks

[WARNING SystemVerification]: this Docker version is not on the list of validated versions: 20.10.21. Latest validated version: 19.03

[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-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:

\* Certificate signing request was sent to apiserver 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.

1. **Kuber-Slave 2 node configuration**

ubuntu@ip-172-31-23-95:~$ history

1 sudo apt update -y

2 sudo apt install docker.io -y

3 sudo systemctl start docker

4 sudo systemctl enable docker

5 sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg

6 echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

7 sudo apt update -y

8 sudo apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y

ubuntu@ip-172-31-29-171:~$ sudo su

root@ip-172-31-29-171:/home/ubuntu# kubeadm join 172.31.18.114:6443 --token q67ewe.116cupfeda2vy21x \

--discovery-token-ca-cert-hash sha256:18598e9b74c911a3383d72d0894b95d6a9ecdb168c36d9ecb0b1b5997570f801

[preflight] Running pre-flight checks

[WARNING SystemVerification]: this Docker version is not on the list of validated versions: 20.10.21. Latest validated version: 19.03

[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-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:

\* Certificate signing request was sent to apiserver 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.

**END** **]**

1. **Assignment 2**

* **(Use the previous deployment**
* **(Create a service type of Node Port for nginx deployment**
* **(Check the nodeport service on a browser to verify**

**On Master node**

ubuntu@ip-172-31-18-114:~$ sudo nano nginxservice.yml

ubuntu@ip-172-31-18-114:~$ kubectl create -f nginxservice.yml

The connection to the server localhost:8080 was refused - did you specify the right host or port?

ubuntu@ip-172-31-18-114:~$ sudo kubectl create -f nginxservice.yml

service/my-nginx-deployment created

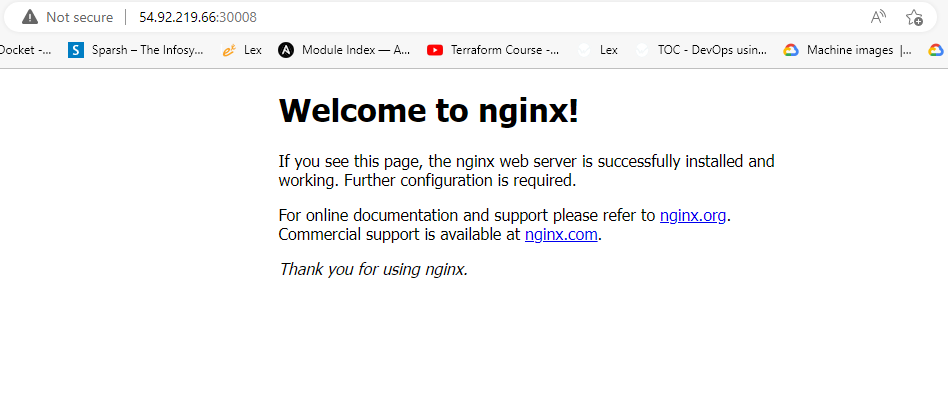
ubuntu@ip-172-31-18-114:~$ sudo kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

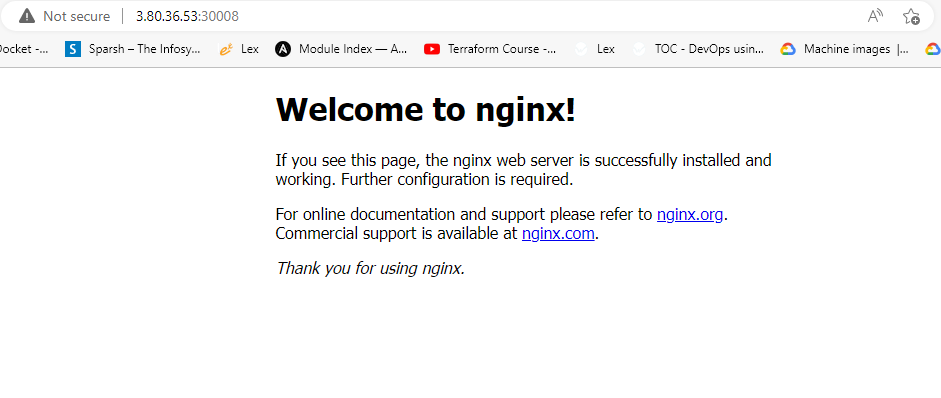
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 81m

my-nginx-deployment NodePort 10.111.80.160 <none> 80:30008/TCP 72s

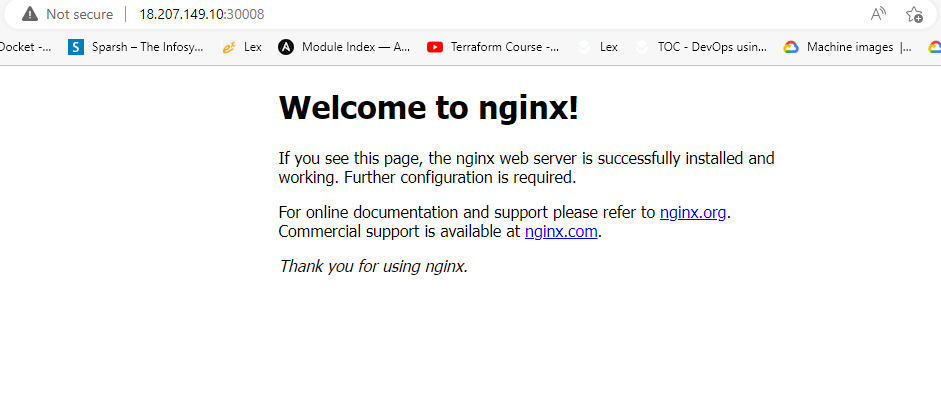
**Webpage on master page**

****

**Webpage on slave1 page**

****

**Webpage on slave2 page**

****

**ubuntu@ip-172-31-18-114:~$ sudo cat nginxservice.yml**

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: my-nginx-deployment**

**spec:**

**type: NodePort**

**ports:**

**- targetPort: 80**

**port: 80**

**nodePort: 30008**

**selector:**

**app: nginx**

1. **Assignment 3**

* **(Use the previous deployment**
* **(Change the replicas to 5 for the deployment**

**On master node**

**ubuntu@master:~$ ls -l**

**total 8**

**-rw-r--r-- 1 root root 334 Apr 10 05:54 nginxdeployment.yml**

**-rw-r--r-- 1 root root 185 Apr 10 06:38 nginxservice.yml**

**ubuntu@master:~$ sudo kubectl edit deploy**

**deployment.apps/nginx-deployment edited**

**ubuntu@master:~$ sudo kubectl get deploy**

**NAME READY UP-TO-DATE AVAILABLE AGE**

**nginx-deployment 5/5 5 5 82m**

**ubuntu@master:~$**

1. **Assignment 4**

* **(Use the previous deployment**
* **(Change the service type to ClusterIP**

**On master node**

**buntu@master:~$ sudo kubectl get svc**

**NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE**

**kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 127m**

**my-nginx-deployment NodePort 10.111.80.160 <none> 80:30008/TCP 47m**

**ubuntu@master:~$ sudo kubectl edit svc**

**service/kubernetes skipped**

**service/my-nginx-deployment edited**

**ubuntu@master:~$ sudo kubectl get svc**

**NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE**

**kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 133m**

**my-nginx-deployment ClusterIP 10.111.80.160 <none> 80/TCP 53m**

**ubuntu@master:~$**

1. **Assignment 5**

* Use the previous deployment
* Deploy an nginx deployment of 3 replicas
* Create an nginx service of type clusterip
* Create an ingress service /apache to apache service /nginx to nginx service

**ubuntu@ip-172-31-19-119:~$ sudo apt 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-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 12 not upgraded.**

**Need to get 72.4 MB of archives.**

**After this operation, 287 MB of additional disk space will be used.**

**Do you want to continue? [Y/n] y**

**Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]**

**Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]**

**Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.4-0ubuntu1~22.04.1 [4241 kB]**

**Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.6.12-0ubuntu1~22.04.1 [34.4 MB]**

**Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 dns-root-data all 2021011101 [5256 B]**

**Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dnsmasq-base amd64 2.86-1.1ubuntu0.2 [354 kB]**

**Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io amd64 20.10.21-0ubuntu1~22.04.2 [33.2 MB]**

**Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]**

**Fetched 72.4 MB in 1s (49.8 MB/s)**

**Preconfiguring packages ...**

**Selecting previously unselected package pigz.**

**(Reading database ... 63657 files and directories currently installed.)**

**Preparing to unpack .../0-pigz\_2.6-1\_amd64.deb ...**

**Unpacking pigz (2.6-1) ...**

**Selecting previously unselected package bridge-utils.**

**Preparing to unpack .../1-bridge-utils\_1.7-1ubuntu3\_amd64.deb ...**

**Unpacking bridge-utils (1.7-1ubuntu3) ...**

**Selecting previously unselected package runc.**

**Preparing to unpack .../2-runc\_1.1.4-0ubuntu1~22.04.1\_amd64.deb ...**

**Unpacking runc (1.1.4-0ubuntu1~22.04.1) ...**

**Selecting previously unselected package containerd.**

**Preparing to unpack .../3-containerd\_1.6.12-0ubuntu1~22.04.1\_amd64.deb ...**

**Unpacking containerd (1.6.12-0ubuntu1~22.04.1) ...**

**Selecting previously unselected package dns-root-data.**

**Preparing to unpack .../4-dns-root-data\_2021011101\_all.deb ...**

**Unpacking dns-root-data (2021011101) ...**

**Selecting previously unselected package dnsmasq-base.**

**Preparing to unpack .../5-dnsmasq-base\_2.86-1.1ubuntu0.2\_amd64.deb ...**

**Unpacking dnsmasq-base (2.86-1.1ubuntu0.2) ...**

**Selecting previously unselected package docker.io.**

**Preparing to unpack .../6-docker.io\_20.10.21-0ubuntu1~22.04.2\_amd64.deb ...**

**Unpacking docker.io (20.10.21-0ubuntu1~22.04.2) ...**

**Selecting previously unselected package ubuntu-fan.**

**Preparing to unpack .../7-ubuntu-fan\_0.12.16\_all.deb ...**

**Unpacking ubuntu-fan (0.12.16) ...**

**Setting up dnsmasq-base (2.86-1.1ubuntu0.2) ...**

**Setting up runc (1.1.4-0ubuntu1~22.04.1) ...**

**Setting up dns-root-data (2021011101) ...**

**Setting up bridge-utils (1.7-1ubuntu3) ...**

**Setting up pigz (2.6-1) ...**

**Setting up containerd (1.6.12-0ubuntu1~22.04.1) ...**

**Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.**

**Setting up ubuntu-fan (0.12.16) ...**

**Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.**

**Setting up docker.io (20.10.21-0ubuntu1~22.04.2) ...**

**Adding group `docker' (GID 122) ...**

**Done.**

**Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.**

**Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.**

**Processing triggers for dbus (1.12.20-2ubuntu4.1) ...**

**Processing triggers for man-db (2.10.2-1) ...**

**Scanning processes...**

**Scanning linux images...**

**Running kernel seems to be up-to-date.**

**No services need to be restarted.**

**No containers need to be restarted.**

**No user sessions are running outdated binaries.**

**No VM guests are running outdated hypervisor (qemu) binaries on this host.**

**ubuntu@ip-172-31-19-119:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube\_latest\_amd64.deb**

**% Total % Received % Xferd Average Speed Time Time Time Current**

**Dload Upload Total Spent Left Speed**

**100 27.9M 100 27.9M 0 0 38.6M 0 --:--:-- --:--:-- --:--:-- 38.6M**

**ubuntu@ip-172-31-19-119:~$ sudo dpkg -i minikube\_latest\_amd64.deb**

**Selecting previously unselected package minikube.**

**(Reading database ... 64012 files and directories currently installed.)**

**Preparing to unpack minikube\_latest\_amd64.deb ...**

**Unpacking minikube (1.30.1-0) ...**

**Setting up minikube (1.30.1-0) ...**

**ubuntu@ip-172-31-19-119:~$ sudo chmod 777 /var/run/docker.sock**

**ubuntu@ip-172-31-19-119:~$ minikube start**

**\* minikube v1.30.1 on Ubuntu 22.04 (xen/amd64)**

**\* Automatically selected the docker driver. Other choices: ssh, none**

**\* Using Docker driver with root privileges**

**\* Starting control plane node minikube in cluster minikube**

**\* Pulling base image ...**

**\* Downloading Kubernetes v1.26.3 preload ...**

**> preloaded-images-k8s-v18-v1...: 397.02 MiB / 397.02 MiB 100.00% 64.41 M**

**> gcr.io/k8s-minikube/kicbase...: 373.53 MiB / 373.53 MiB 100.00% 44.05 M**

**\* Creating docker container (CPUs=2, Memory=2200MB) ...**

**\* Preparing Kubernetes v1.26.3 on Docker 23.0.2 ...**

**- Generating certificates and keys ...**

**- Booting up control plane ...**

**- Configuring RBAC rules ...**

**\* Configuring bridge CNI (Container Networking Interface) ...**

**- Using image gcr.io/k8s-minikube/storage-provisioner:v5**

**\* Verifying Kubernetes components...**

**\* Enabled addons: storage-provisioner, default-storageclass**

**\* kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'**

**\* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default**

**ubuntu@ip-172-31-19-119:~$ sudo snap install kubectl --classic**

**kubectl 1.26.3 from Canonical✓ installed**

**ubuntu@ip-172-31-19-119:~$ minikube addons enable ingress**

**\* ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.**

**You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS**

**- Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20230312-helm-chart-4.5.2-28-g66a760794**

**- Using image registry.k8s.io/ingress-nginx/controller:v1.7.0**

**- Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20230312-helm-chart-4.5.2-28-g66a760794**

**\* Verifying ingress addon...**

**\* The 'ingress' addon is enabled**

**ubuntu@ip-172-31-19-119:~$ kubectl create deployment nginx --image=nginx --port=80**

**deployment.apps/nginx created**

**ubuntu@ip-172-31-19-119:~$ kubectl expose deploy nginx --type NodePort**

**service/nginx exposed**

**ubuntu@ip-172-31-19-119:~$ nano ingress.yaml**

**ubuntu@ip-172-31-19-119:~$ kubectl apply -f ingress.yaml**

**ingress.networking.k8s.io/ingress created**

**ubuntu@ip-172-31-19-119:~$ kubectl get svc**

**NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE**

**kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 8m52s**

**nginx NodePort 10.100.184.135 <none> 80:32174/TCP 107s**

**ubuntu@ip-172-31-19-119:~$ kubectl port-forward service/ingress-nginx-controller -n ingress-nginx --address 0.0.0.0 :443**

**Forwarding from 0.0.0.0:44867 -> 443**

**Handling connection for 44867**

**Handling connection for 44867**

**Handling connection for 44867**

**Handling connection for 44867**

**Handling connection for 44867**

**Handling connection for 44867**

**Handling connection for 44867**

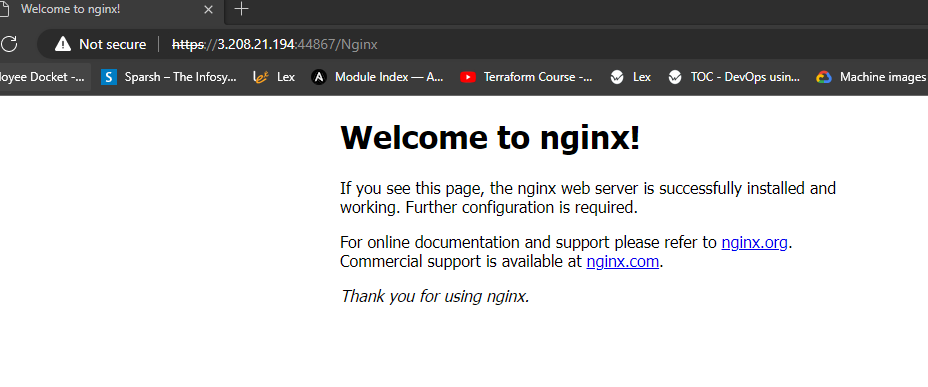
**Handling connection for 44867**

**Handling connection for 44867**

**Handling connection for 44867**

**Handling connection for 44867**

**Handling connection for 44867**

****

**1 clear**

**2 sudo apt update**

**3 clear**

**4 sudo apt install docker.io**

**5 curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube\_latest\_amd64.deb**

**6 sudo dpkg -i minikube\_latest\_amd64.deb**

**7 sudo chmod 777 /var/run/docker.sock**

**8 minikube start**

**9 sudo snap install kubectl --classic**

**10 minikube addons enable ingress**

**11 kubectl create deployment nginx --image=nginx --port=80**

**12 kubectl expose deploy nginx --type NodePort**

**13 nano ingress.yaml**

**14 kubectl apply -f ingress.yaml**

**15 kubectl get svc**

**16 kubectl port-forward service/ingress-nginx-controller -n ingress-nginx --address 0.0.0.0 :443**

**17 history**