INSTITUT TEKNOLOGI DEL MATERI

PRAKTIKUM

SEMESTER V TAHUN AJARAN 2025/2026

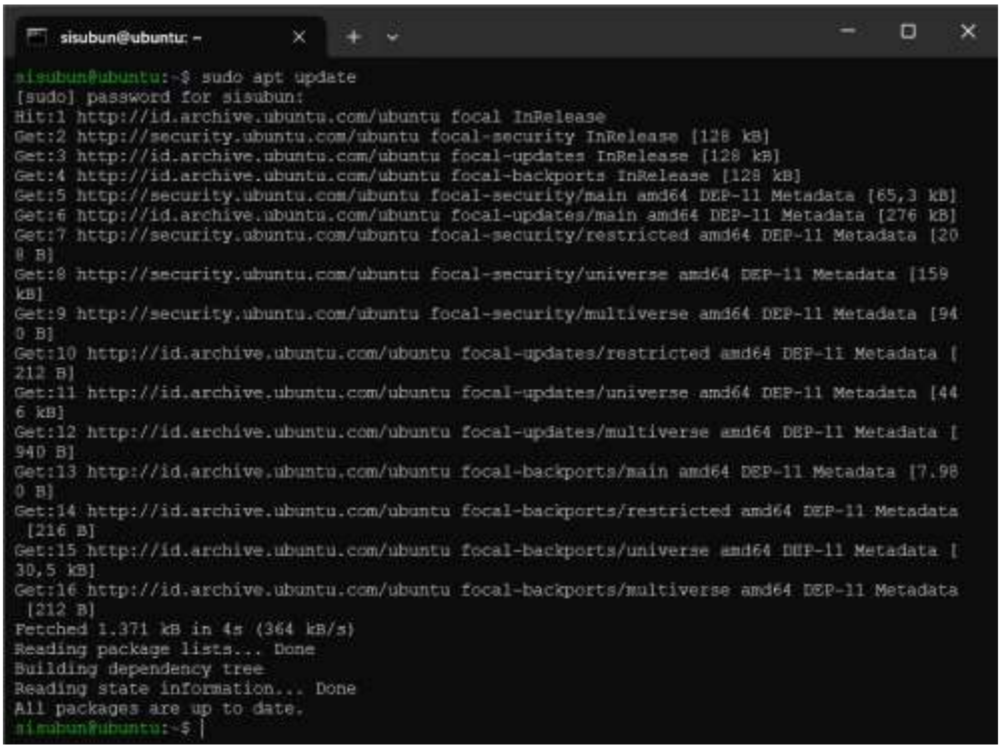
Session Date	: Selasa, 28 Oktober 2025
Semester	: 5
Courses	: Sistem Komputasi Awan
Week/Session	: 10/2
Key Topics	: Manual Minikube Deployment and Automation OpenTofu Deployment
Duration	: Sesi Praktikum
Delivery	: Laporan (Softcopy)
Deadline of delivery	: Senin, 03 November 2025
Place of Delivery	: E-Course
Purpose	: Mahasiswa mampu melakukan set up pada Minikube dan OpenTofu serta mampu melakukan deployment secara manual dan otomatis menggunakan OpenTofu

Praktikum

A. Setting Up Minikube, Kubernetes CLI, and OpenTofu

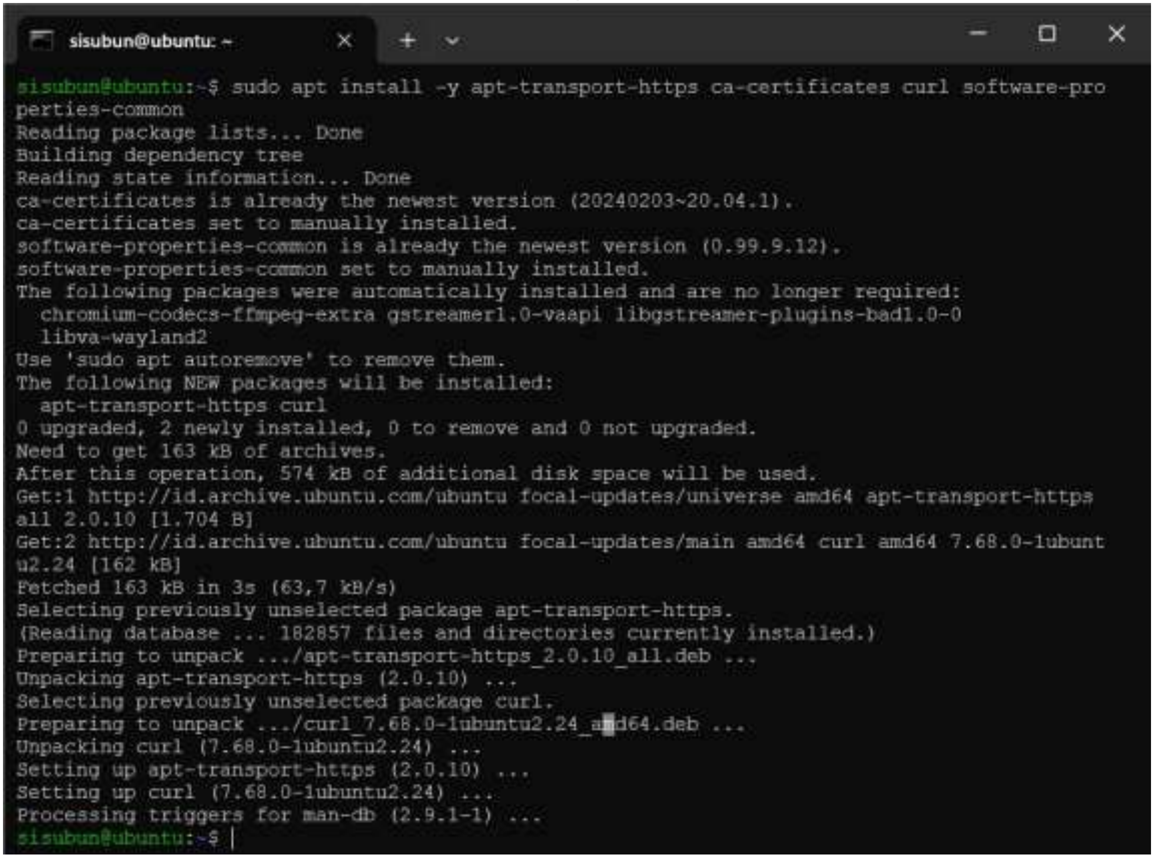
1. Sebelum menginstal Docker, lakukan update terlebih dahulu pada repository Ubuntu Anda dengan command berikut.

```
sudo apt update
```



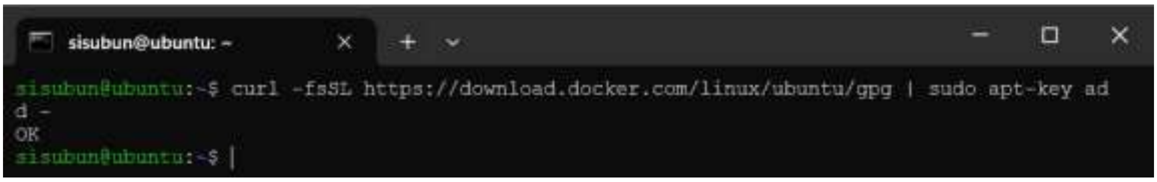
```
sisubun@ubuntu:~$ sudo apt update
[sudo] password for sisubun:
Hit:1 http://id.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Get:3 http://id.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:4 http://id.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [65,3 kB]
Get:6 http://id.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [276 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 DEP-11 Metadata [20
8 B]
Get:8 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [159
kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [94
0 B]
Get:10 http://id.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 DEP-11 Metadata [
212 B]
Get:11 http://id.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [44
6 kB]
Get:12 http://id.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [
940 B]
Get:13 http://id.archive.ubuntu.com/ubuntu focal-backports/main amd64 DEP-11 Metadata [7.98
0 B]
Get:14 http://id.archive.ubuntu.com/ubuntu focal-backports/restricted amd64 DEP-11 Metadata
 [216 B]
Get:15 http://id.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [
30,5 kB]
Get:16 http://id.archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 DEP-11 Metadata
 [212 B]
Fetched 1.371 kB in 4s (364 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
sisubun@ubuntu:~$ |
```

2. Kemudian lakukan update CA certificate dengan command berikut.



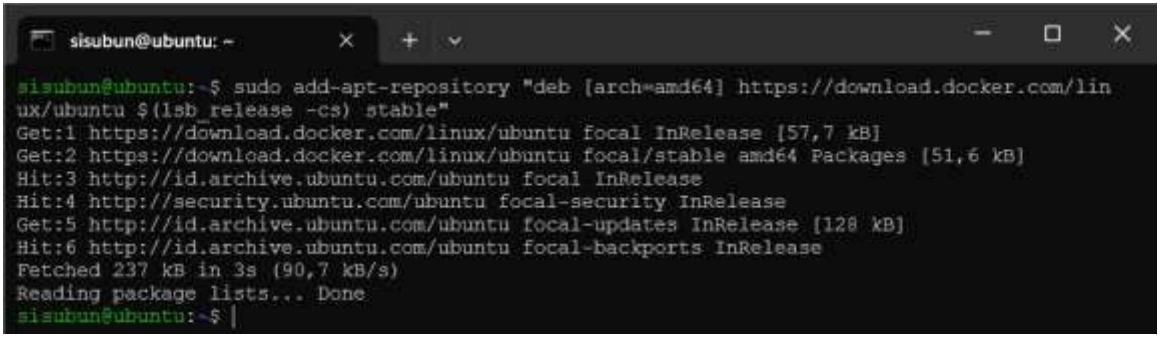
```
sisubun@ubuntu:~$ sudo apt install -y apt-transport-https ca-certificates curl software-pro
perties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20240203~20.04.1).
ca-certificates set to manually installed.
software-properties-common is already the newest version (0.99.9.12).
software-properties-common set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0
  libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  apt-transport-https curl
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 163 kB of archives.
After this operation, 574 kB of additional disk space will be used.
Get:1 http://id.archive.ubuntu.com/ubuntu focal-updates/universe amd64 apt-transport-https
all 2.0.10 [1.704 B]
Get:2 http://id.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7.68.0-1ubunt
u2.24 [162 kB]
Fetched 163 kB in 3s (63,7 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 182857 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.0.10_all.deb ...
Unpacking apt-transport-https (2.0.10) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.68.0-1ubuntu2.24_amd64.deb ...
Unpacking curl (7.68.0-1ubuntu2.24) ...
Setting up apt-transport-https (2.0.10) ...
Setting up curl (7.68.0-1ubuntu2.24) ...
Processing triggers for man-db (2.9.1-1) ...
sisubun@ubuntu:~$ |
```

3. Lalu tambahkan GPG key dengan command berikut.



```
sisubun@ubuntu:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key ad
d -
OK
sisubun@ubuntu:~$ |
```

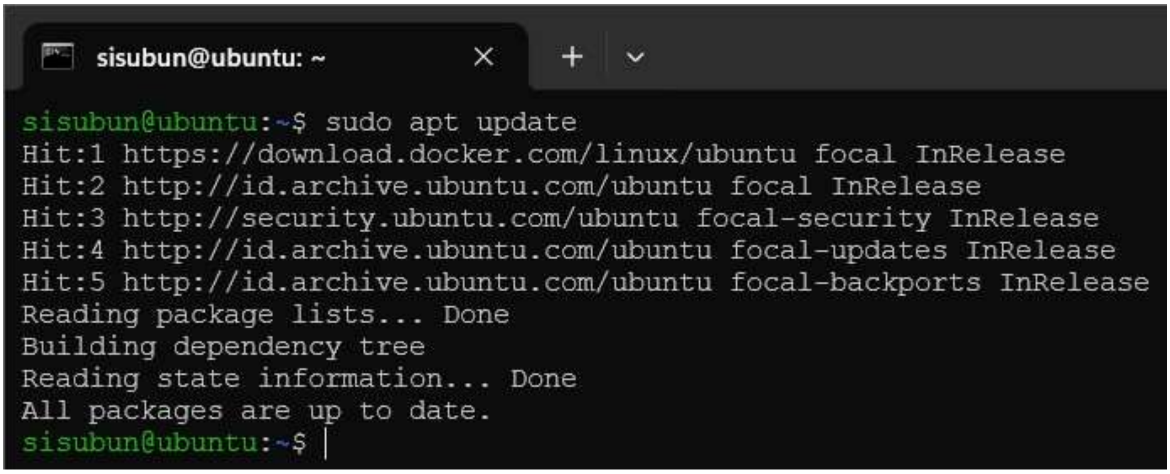
4. Lalu tambahkan repo docker dengan command berikut.



```
sisubun@ubuntu:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/lin
ux/ubuntu $(lsb_release -cs) stable"
Get:1 https://download.docker.com/linux/ubuntu focal InRelease [57,7 kB]
Get:2 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [51,6 kB]
Hit:3 http://id.archive.ubuntu.com/ubuntu focal InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:5 http://id.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Hit:6 http://id.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 237 kB in 3s (90,7 kB/s)
Reading package lists... Done
sisubun@ubuntu:~$ |
```

5. Kemudian silahkan lakukan update kembali dengan command berikut.

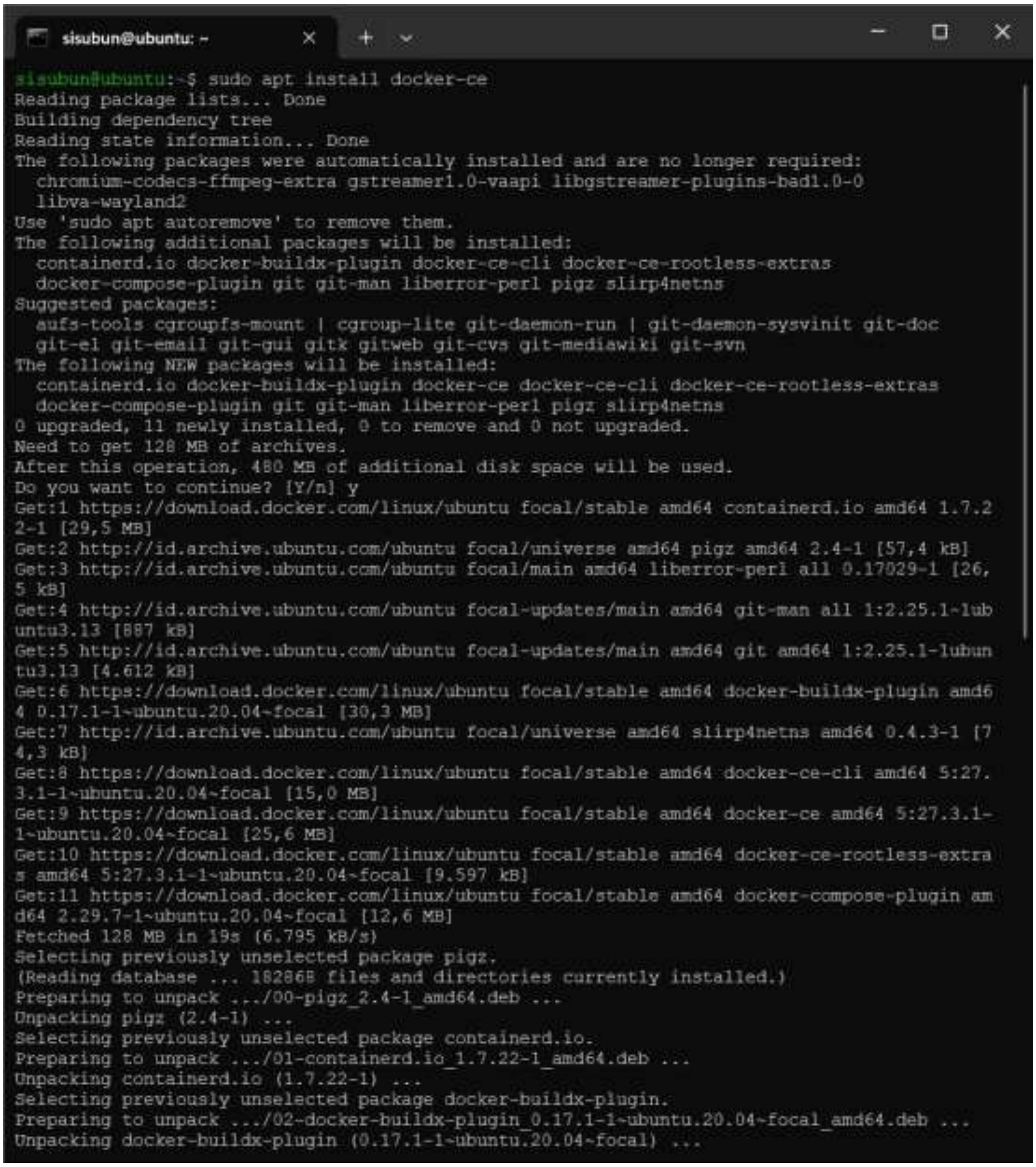
```
sudo apt update
```



```
sisubun@ubuntu:~$ sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://id.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
sisubun@ubuntu:~$ |
```

6. Sekarang Anda bisa menginstal docker dengan menjalankan command berikut.

```
sudo apt install docker-ce
```



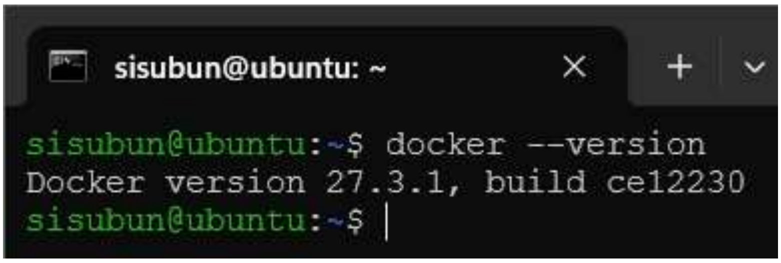
```
sisubun@ubuntu:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0
  libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin git git-man liberror-perl pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc
  git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin git git-man liberror-perl pigz slirp4netns
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 128 MB of archives.
After this operation, 480 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io amd64 1.7.2
2-1 [29,5 MB]
Get:2 http://id.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57,4 kB]
Get:3 http://id.archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.17029-1 [26,
5 kB]
Get:4 http://id.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all 1:2.25.1-1ub
untu3.13 [887 kB]
Get:5 http://id.archive.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubun
tu3.13 [4.612 kB]
Get:6 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-buildx-plugin amd6
4 0.17.1-1~ubuntu.20.04~focal [30,3 MB]
Get:7 http://id.archive.ubuntu.com/ubuntu focal/universe amd64 slirp4netns amd64 0.4.3-1 [7
4,3 kB]
Get:8 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-cli amd64 5:27.
3.1-1~ubuntu.20.04~focal [15,0 MB]
Get:9 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce amd64 5:27.3.1-
1~ubuntu.20.04~focal [25,6 MB]
Get:10 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-rootless-extra
s amd64 5:27.3.1-1~ubuntu.20.04~focal [9.597 kB]
Get:11 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-compose-plugin am
d64 2.29.7-1~ubuntu.20.04~focal [12,6 MB]
Fetched 128 MB in 19s (6.795 kB/s)
Selecting previously unselected package pigz.
(Reading database ... 182868 files and directories currently installed.)
Preparing to unpack .../00-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../01-containerd.io_1.7.22-1_amd64.deb ...
Unpacking containerd.io (1.7.22-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../02-docker-buildx-plugin_0.17.1-1~ubuntu.20.04~focal_amd64.deb ...
Unpacking docker-buildx-plugin (0.17.1-1~ubuntu.20.04~focal) ...
```

7. Anda bisa menjalankan command berikut apabila Anda ingin menggunakan docker tanpa mengetikkan "sudo" dengan menambahkan user ke docker group.

```
sudo usermod -aG docker ${USER}
```

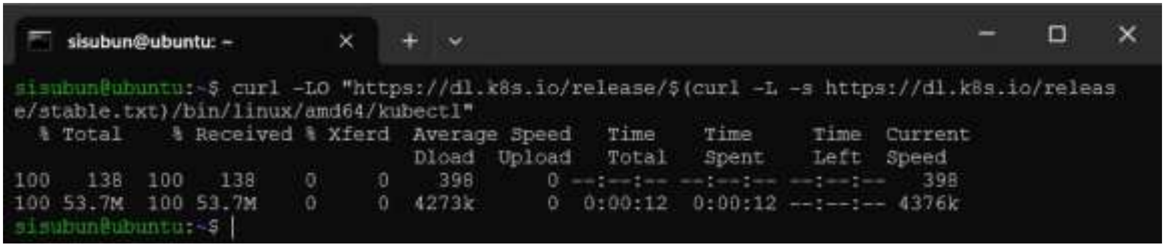
8. Lalu pastikan apakah docker benar-benar sudah berhasil diinstal dengan command berikut.

```
docker --version
```



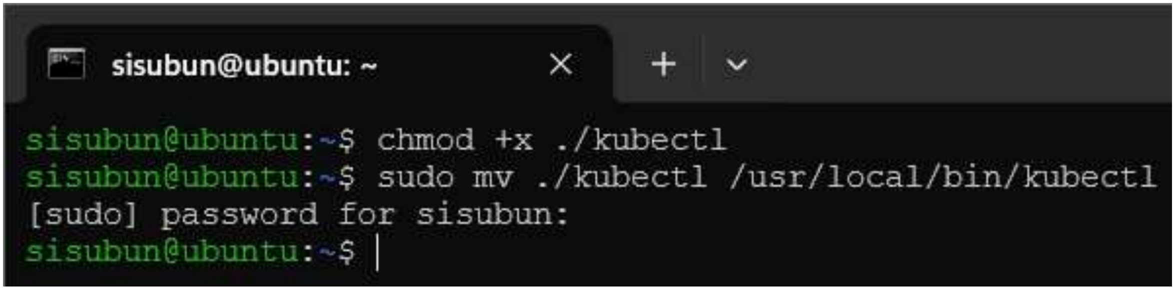
```
sisubun@ubuntu:~$ docker --version
Docker version 27.3.1, build ce12230
sisubun@ubuntu:~$
```

9. Setelah Docker sudah Anda terinstal, Anda bisa mendownload terlebih dahulu kubectl sebelum menginstal Minikube dengan menjalankan command berikut.



```
sisubun@ubuntu:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/releas
e/stable.txt)/bin/linux/amd64/kubectl"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0    398      0 --:--:-- --:--:-- --:--:--   398
100 53.7M  100 53.7M    0     0  4273k      0  0:00:12  0:00:12 --:--:-- 4376k
sisubun@ubuntu:~$ |
```

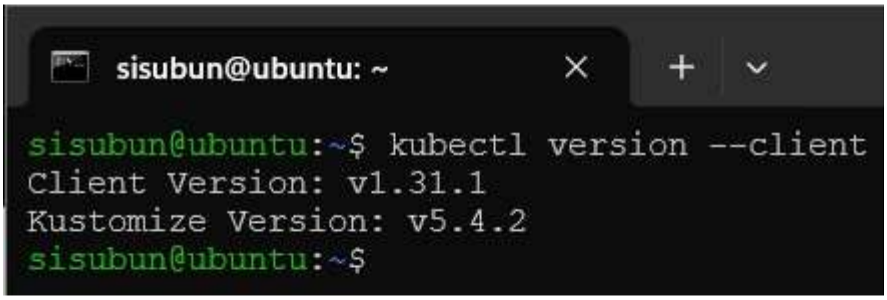
10. Lalu installah kubectl dengan command berikut.



```
sisubun@ubuntu:~$ chmod +x ./kubectl
sisubun@ubuntu:~$ sudo mv ./kubectl /usr/local/bin/kubectl
[sudo] password for sisubun:
sisubun@ubuntu:~$ |
```

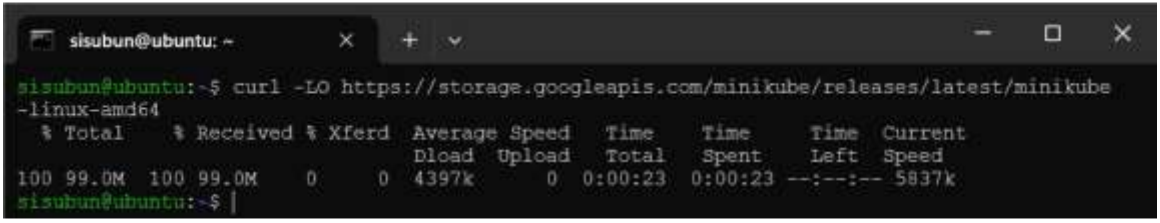
11. Pastikanlah kubectl berhasil diinstal dengan command berikut.

```
kubectl version --client
```



```
sisubun@ubuntu:~$ kubectl version --client
Client Version: v1.31.1
Kustomize Version: v5.4.2
sisubun@ubuntu:~$
```

12. Setelah itu, Anda bisa mendownload minikube dengan menjalankan command berikut.



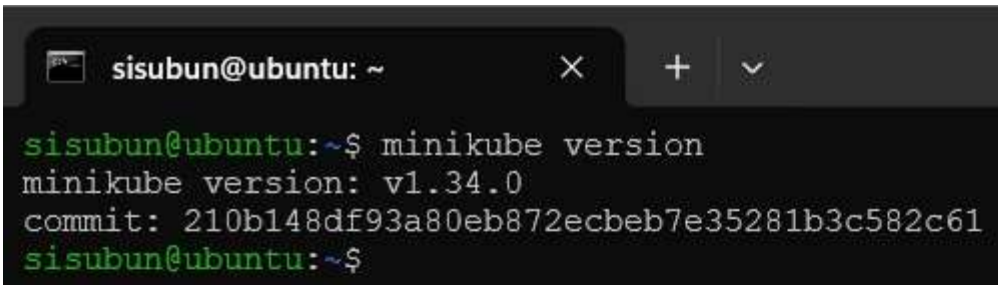
```
sisubun@ubuntu:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube
-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 99.0M  100 99.0M    0     0  4397k      0  0:00:23  0:00:23 --:--:-- 5837k
sisubun@ubuntu:~$ |
```

13. Setelah didownload, Anda bisa menginstal minikube dengan command berikut.

```
sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-
linux-amd64
```

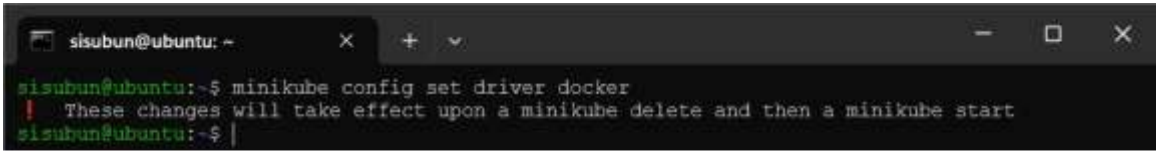
14. Pastikan bahwa minikube sudah berhasil diinstal dengan command berikut.

```
minikube version
```



```
sisubun@ubuntu:~$ minikube version
minikube version: v1.34.0
commit: 210b148df93a80eb872ecbeb7e35281b3c582c61
sisubun@ubuntu:~$
```

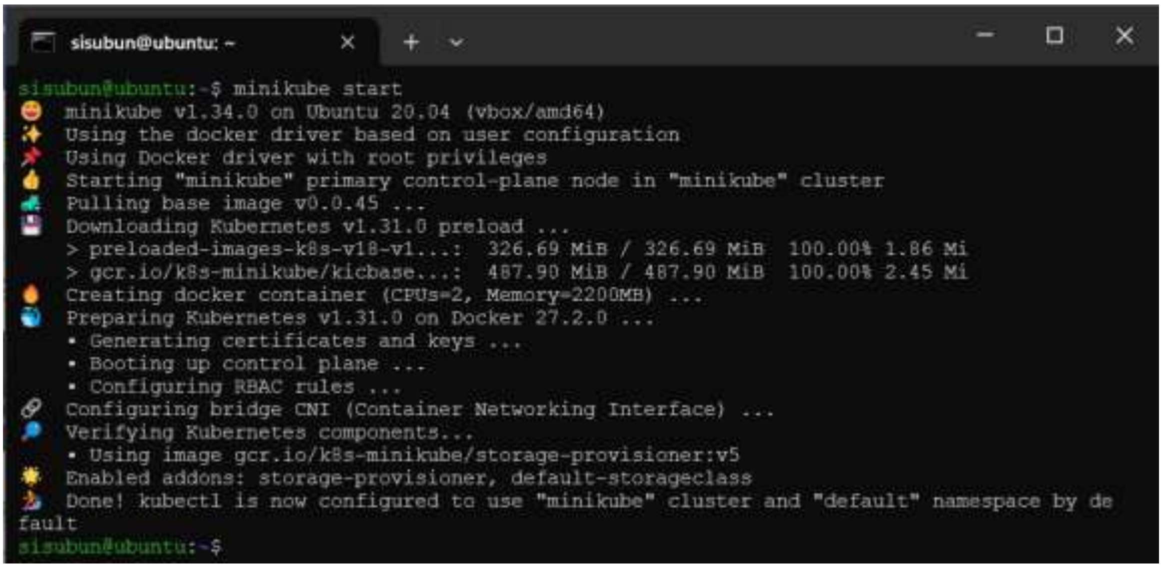
15. Jalankan command berikut jika Anda ingin membuat docker sebagai default driver.



```
sisubun@ubuntu:~$ minikube config set driver docker
❗  These changes will take effect upon a minikube delete and then a minikube start
sisubun@ubuntu:~$ |
```

16. Lalu coba jalankan minikube yang sudah diinstal dengan command berikut.

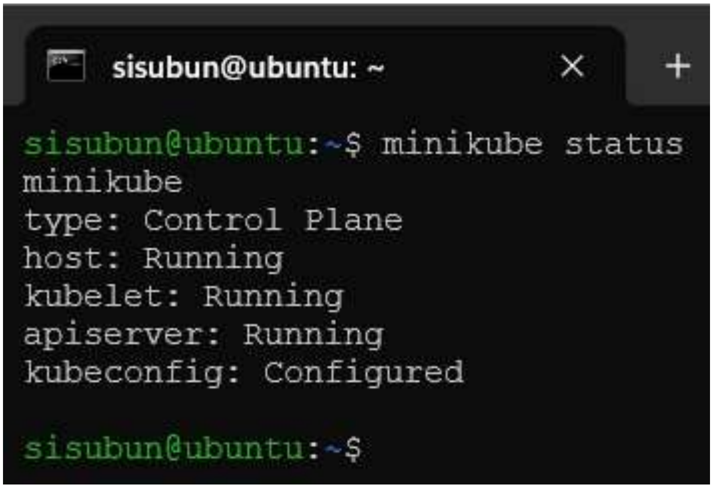
```
minikube start
```



```
sisubun@ubuntu:~$ minikube start
😄  minikube v1.34.0 on Ubuntu 20.04 (vbox/amd64)
✨  Using the docker driver based on user configuration
📌  Using Docker driver with root privileges
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.45 ...
💾  Downloading Kubernetes v1.31.0 preload ...
    > preloaded-images-k8s-v18-v1...:  326.69 MiB / 326.69 MiB  100.00% 1.86 Mi
    > gcr.io/k8s-minikube/kicbase...:  487.90 MiB / 487.90 MiB  100.00% 2.45 Mi
🔥  Creating docker container (CPUs=2, Memory=2200MB) ...
🐳  Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by de
fault
sisubun@ubuntu:~$
```

17. Pastikan kalau Minikube berhasil berjalan atau tidak.

```
minikube status
```



```
sisubun@ubuntu:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

sisubun@ubuntu:~$
```

18. Setelah Minikube sudah terinstal, Anda bisa melanjutkan untuk menginstal OpenTofu. Pertama-tama Anda akan menambahkan GPG key dari OpenTofu. Ini memverifikasi bahwa paket Anda memang telah dibuat menggunakan pipeline resmi dan belum dirusak.

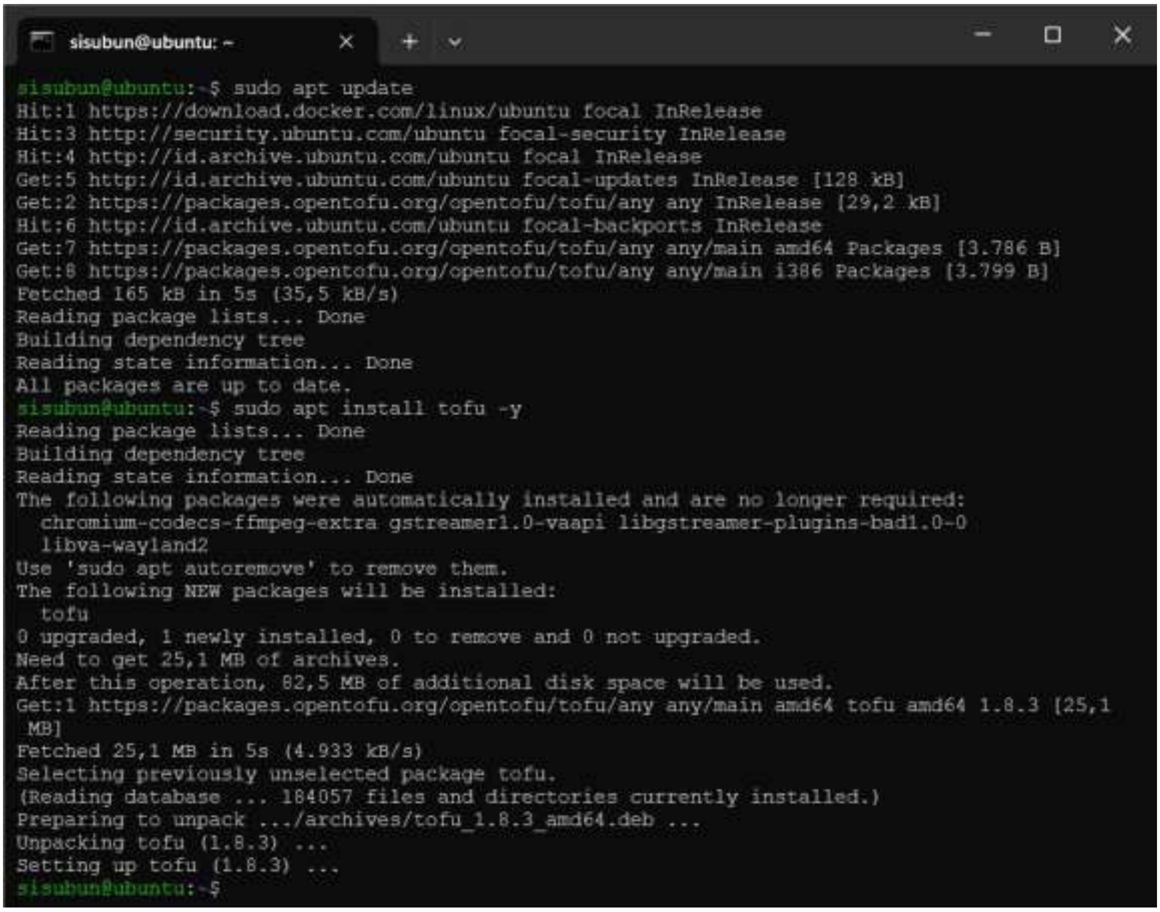
```
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://get.opentofu.org/opentofu.gpg | sudo tee
/etc/apt/keyrings/opentofu.gpg         >/dev/null         curl         -fsSL
https://packages.opentofu.org/opentofu/tofu/gpgkey | sudo gpg --no-tty --
batch --dearmor -o /etc/apt/keyrings/opentofu-repo.gpg
>/dev/null
sudo          chmod          a+r          /etc/apt/keyrings/opentofu.gpg

/etc/apt/keyrings/opentofu-repo.gpg
```

19. Anda bisa membuat source list dari OpenTofu dengan command berikut.

```
echo \
  "deb                                                              [signed-
by=/etc/apt/keyrings/opentofu.gpg,/etc/apt/keyrings/opentofu-repo.gpg]
https://packages.opentofu.org/opentofu/tofu/any/ any main
deb-src                                                           [signed-
by=/etc/apt/keyrings/opentofu.gpg,/etc/apt/keyrings/opentofu-repo.gpg]
https://packages.opentofu.org/opentofu/tofu/any/ any main" | \   sudo tee
/etc/apt/sources.list.d/opentofu.list > /dev/null sudo chmod a+r
/etc/apt/sources.list.d/opentofu.list
```

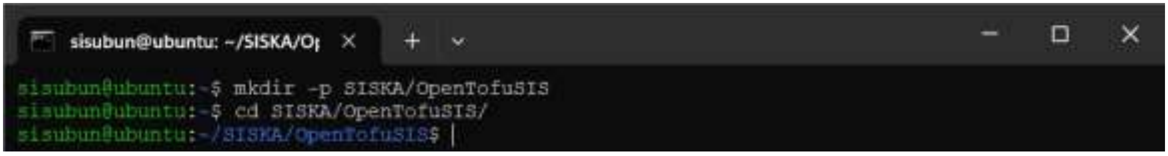
20. Sekarang Anda bisa menginstal OpenTofu dengan command berikut.

```
sudo apt-get update sudo
apt-get install -y tofu
```



```
sisubun@ubuntu:~$ sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu focal InRelease
Get:5 http://id.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:2 https://packages.opentofu.org/opentofu/tofu/any any InRelease [29,2 kB]
Hit:6 http://id.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:7 https://packages.opentofu.org/opentofu/tofu/any any/main amd64 Packages [3.786 B]
Get:8 https://packages.opentofu.org/opentofu/tofu/any any/main i386 Packages [3.799 B]
Fetched 165 kB in 5s (35,5 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
sisubun@ubuntu:~$ sudo apt install tofu -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0
  libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  tofu
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 25,1 MB of archives.
After this operation, 82,5 MB of additional disk space will be used.
Get:1 https://packages.opentofu.org/opentofu/tofu/any any/main amd64 tofu amd64 1.8.3 [25,1
 MB]
Fetched 25,1 MB in 5s (4.933 kB/s)
Selecting previously unselected package tofu.
(Reading database ... 184057 files and directories currently installed.)
Preparing to unpack .../archives/tofu_1.8.3_amd64.deb ...
Unpacking tofu (1.8.3) ...
Setting up tofu (1.8.3) ...
sisubun@ubuntu:~$
```

21. Setelah itu Anda membuat direktori baru, sesuaikan penamaan direktorinya sesuai yang Anda inginkan.



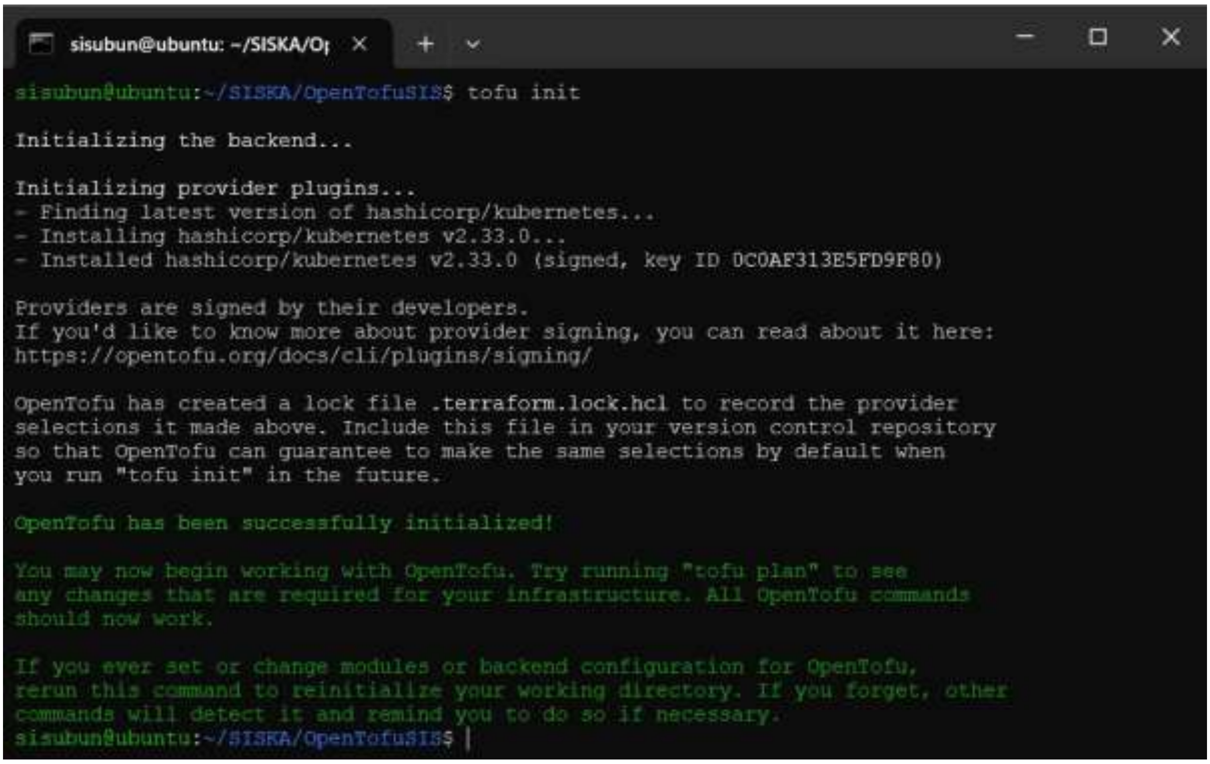
```
sisubun@ubuntu:~$ mkdir -p SISKA/OpenTofuSIS
sisubun@ubuntu:~$ cd SISKA/OpenTofuSIS/
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ |
```

22. Disini Anda akan menerapkan suatu web server Nginx secara otomatis dengan OpenTofu. Oleh karena itu, Anda perlu membuat file konfigurasi OpenTofu dengan ekstensi .tf untuk menerapkan secara otomatis web server Nginx berikut.

```
provider "kubernetes" {
  config_path = "~/.kube/config"
}
resource "kubernetes_deployment" "nginx" {
metadata {
    name = "nginx-deployment"
  }
spec {
    replicas = 1

    selector {
match_labels = {
app = "nginx"
      }
    }      template {      metadata {        labels
= {           app = "nginx"
        }
      }
      spec {           container {            name  = "nginx"
image = "nginx:latest"          port {
            container_port = 80
          }
        }
      }
    }
  }
}
resource "kubernetes_service" "nginx" {   metadata {
    name = "nginx-service"
  }    spec {     selector = {       app =
"nginx"
    }
    type = "NodePort"
     port {       port = 80       target_port = 80
    }   }
}
```

23. Setelah Anda membuat file konfigurasinya, Anda terlebih dahulu menginisialisasi OpenTofu di dalam direktori tersebut dengan command berikut.

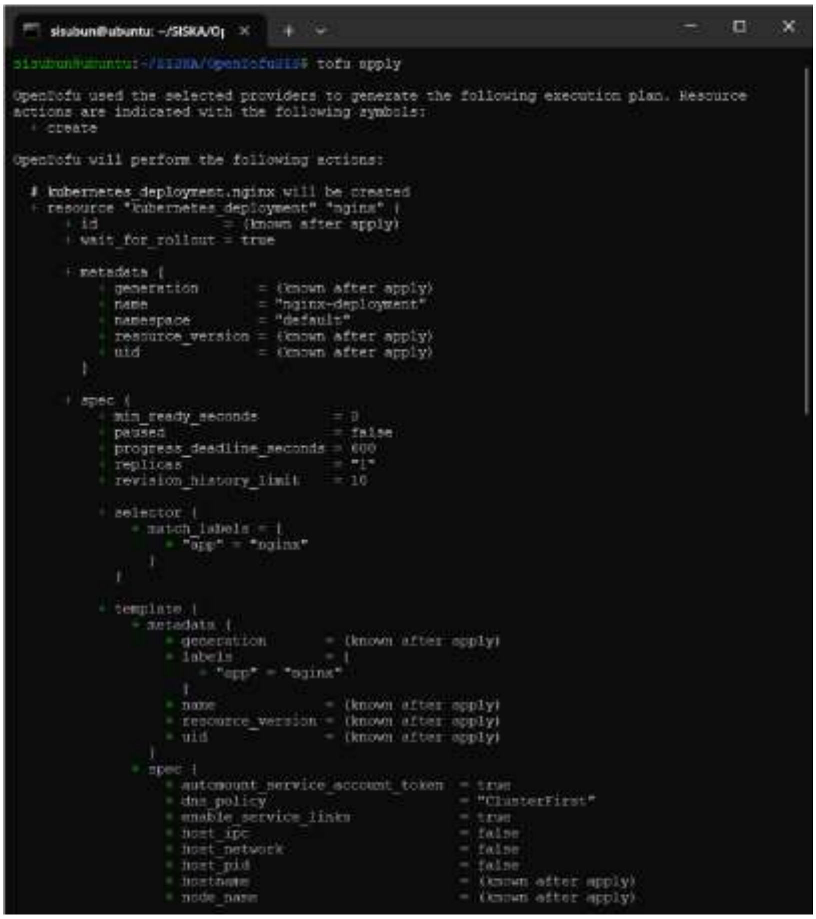
```
tofu init
```



```
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ tofu init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/kubernetes...
- Installing hashicorp/kubernetes v2.33.0...
- Installed hashicorp/kubernetes v2.33.0 (signed, key ID 0C0AF313E5FD9F80)

Providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://opentofu.org/docs/cli/plugins/signing/

OpenTofu has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that OpenTofu can guarantee to make the same selections by default when
you run "tofu init" in the future.

OpenTofu has been successfully initialized!

You may now begin working with OpenTofu. Try running "tofu plan" to see
any changes that are required for your infrastructure. All OpenTofu commands
should now work.

If you ever set or change modules or backend configuration for OpenTofu,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ |
```

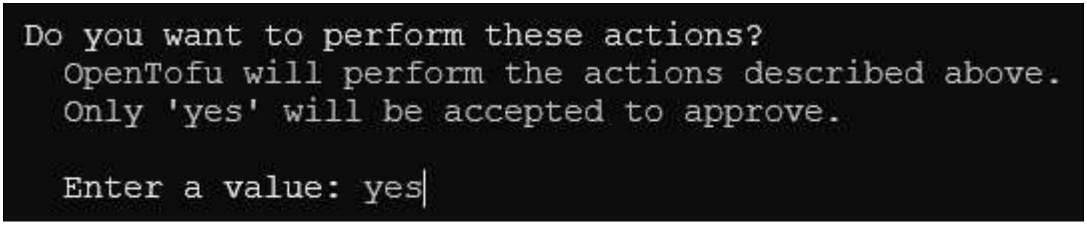
24. Lalu sekarang Anda bisa meng-apply file konfigurasi yang sudah dibuat dengan command berikut.

```
tofu apply
```



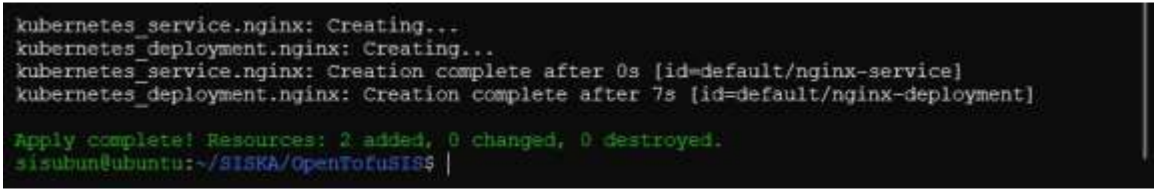
```
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ tofu apply

OpenTofu used the selected providers to generate the following execution plan. Resource
actions are indicated with the following symbols:
  + create

OpenTofu will perform the following actions:

  # kubernetes_deployment.nginx will be created
  + resource "kubernetes_deployment" "nginx" {
      + id               = (known after apply)
      + wait_for_rollout = true

      + metadata {
          + generation       = (known after apply)
          + name             = "nginx-deployment"
          + namespace        = "default"
          + resource_version = (known after apply)
          + uid              = (known after apply)
        }

      + spec {
          + min_ready_seconds         = 0
          + paused                    = false
          + progress_deadline_seconds = 600
          + replicas                  = "1"
          + revision_history_limit    = 10

          + selector {
              + match_labels = {
                  + "app" = "nginx"
                }
            }

          + template {
              + metadata {
                  + generation       = (known after apply)
                  + labels           = {
                      + "app" = "nginx"
                    }
                  + name             = (known after apply)
                  + resource_version = (known after apply)
                  + uid              = (known after apply)
                }
              + spec {
                  + automount_service_account_token  = true
                  + dns_policy                       = "ClusterFirst"
                  + enable_service_links             = true
                  + host_ipc                         = false
                  + host_network                     = false
                  + host_pid                         = false
                  + hostname                         = (known after apply)
                  + node_name                        = (known after apply)
```

Nanti akan muncul verifikasi berikut, Anda bisa mengetikkan **yes** lalu tekan Enter.



```
Do you want to perform these actions?
  OpenTofu will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes
```

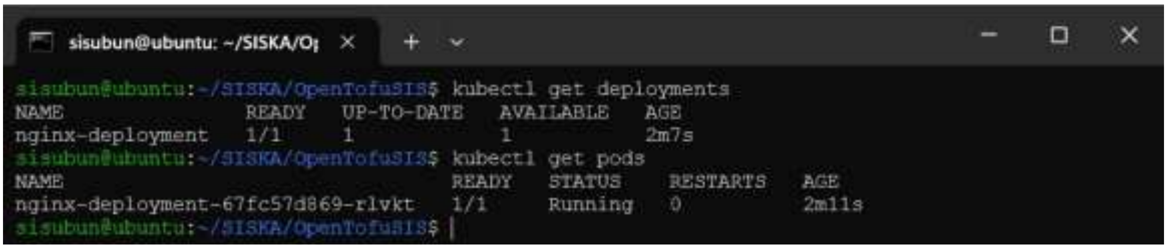
Maka hasilnya akan seperti ini:



```
kubernetes_service.nginx: Creating...
kubernetes_deployment.nginx: Creating...
kubernetes_service.nginx: Creation complete after 0s [id=default/nginx-service]
kubernetes_deployment.nginx: Creation complete after 7s [id=default/nginx-deployment]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ |
```

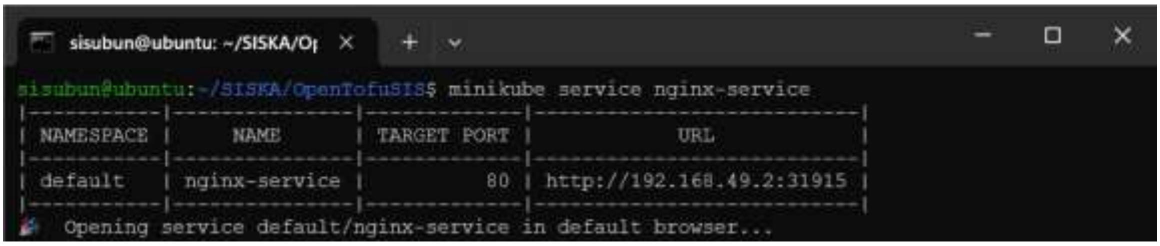
Maka nginx-deployment dan nginx-service akan terbuat.

25. Anda juga bisa memeriksa apakah penerapan web server Nginx berhasil atau tidak dengan command berikut.



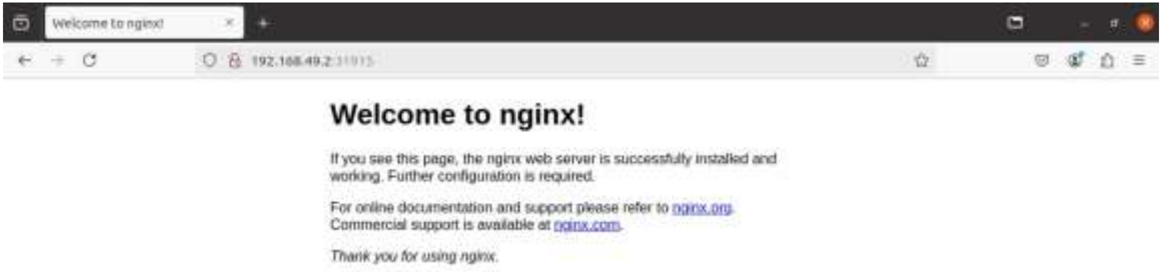
```
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   1/1     1            1           2m7s
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-67fc57d869-rlvkt   1/1     Running   0          2m11s
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ |
```

26. Sekarang Anda sudah bisa mengakses layanan dari Nginx dengan menjalankan command berikut.



```
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ minikube service nginx-service
|-----------|---------------|-------------|---------------------------|
| NAMESPACE |     NAME      | TARGET PORT |            URL            |
|-----------|---------------|-------------|---------------------------|
| default   | nginx-service |          80 | http://192.168.49.2:31915 |
|-----------|---------------|-------------|---------------------------|
🎉  Opening service default/nginx-service in default browser...
```

27. Jika Anda membuka URL yang diberikan pada web browser, maka tampilannya akan seperti ini.



Welcome to nginx!

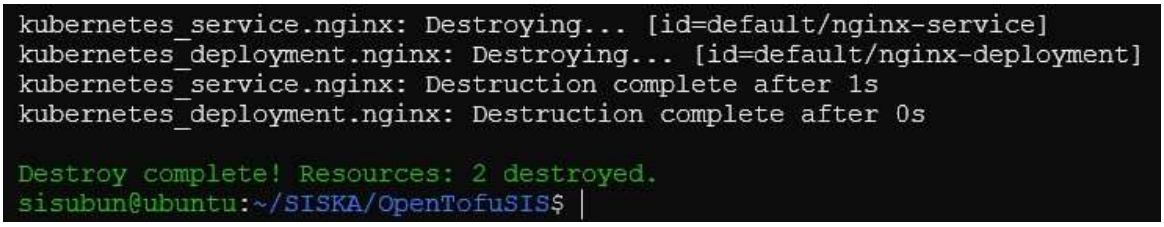
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

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

Thank you for using nginx.

28. Jika sudah selesai, Anda bisa membersihkan atau menghapusnya dengan command berikut.

```
tofu destroy
```

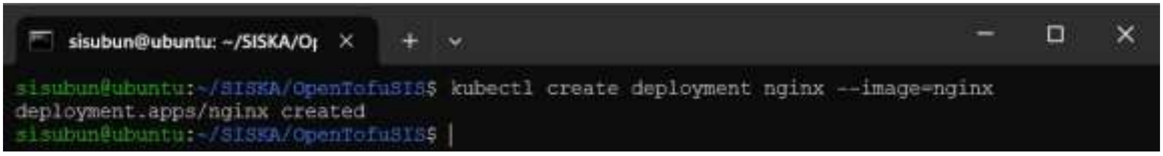


```
kubernetes_service.nginx: Destroying... [id=default/nginx-service]
kubernetes_deployment.nginx: Destroying... [id=default/nginx-deployment]
kubernetes_service.nginx: Destruction complete after 1s
kubernetes_deployment.nginx: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
sisubun@ubuntu:~/SISKA/OpenTofuSIS$
```

29. Disini Anda sudah berhasil melakukan set up kubectl, minikube, dan mendeploy web server Nginx menggunakan OpenTofu.

B. Manual Kubernetes Deployment and OpenTofu Automation for Nginx

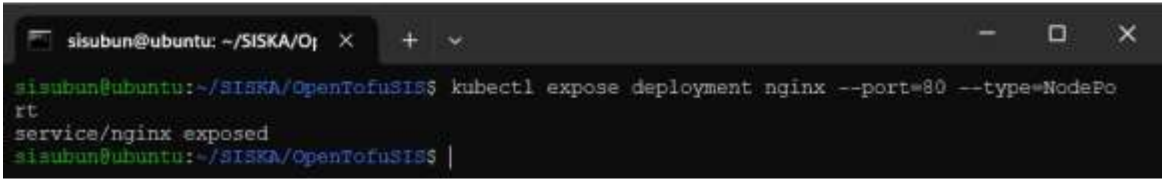
1. Selanjutnya disini Anda akan mencoba untuk mendeploy web server Nginx secara manual menggunakan kubectl.



```
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ |
```

Maka deployment dengan nama nginx akan terbentuk menggunakan docker image dari nginx.

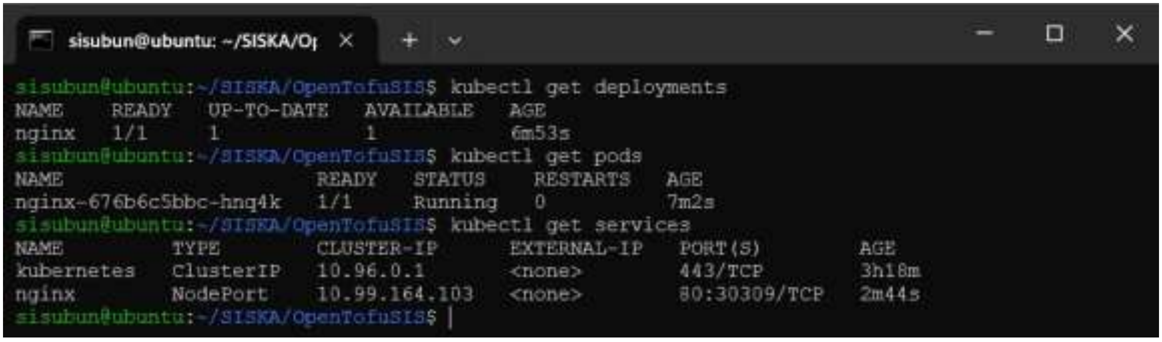
2. Setelah itu Anda akan mengekspos layanan dari nginx agar dapat diakses dengan menjalankan command berikut.



```
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ kubectl expose deployment nginx --port=80 --type=NodePo
rt
service/nginx exposed
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ |
```

Maka terbentuk layanan yang mengekspos deployment dari nginx pada port 80.

3. Lalu ceklah apakah deployment dan layanan dari Nginx berhasil dibuat atau tidak dengan menjalankan command berikut.

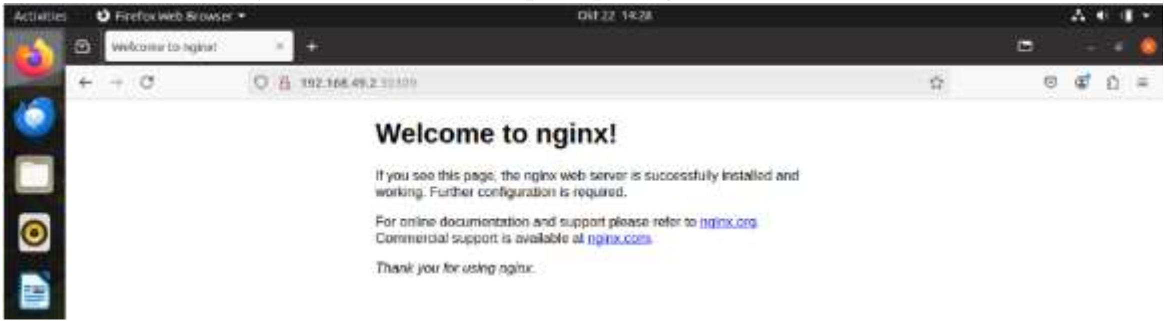


```
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ kubectl get deployments
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nginx   1/1     1            1           6m53s
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ kubectl get pods
NAME                       READY   STATUS    RESTARTS   AGE
nginx-676b6c5bbc-hnq4k     1/1     Running   0          7m2s
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ kubectl get services
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP        3h18m
nginx        NodePort    10.99.164.103   <none>        80:30309/TCP   2m44s
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ |
```

4. Untuk mengakses web server Nginx, Anda bisa menjalankan command berikut untuk mendapatkan URL dari layanan Nginx tersebut.

```
minikube service nginx
```

5. Lalu akseslah URLnya di web browser seperti pada gambar berikut.



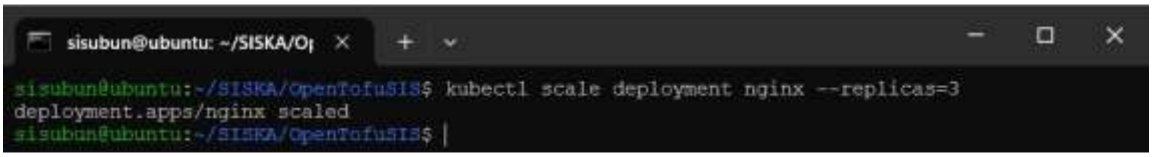
Welcome to nginx!

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.

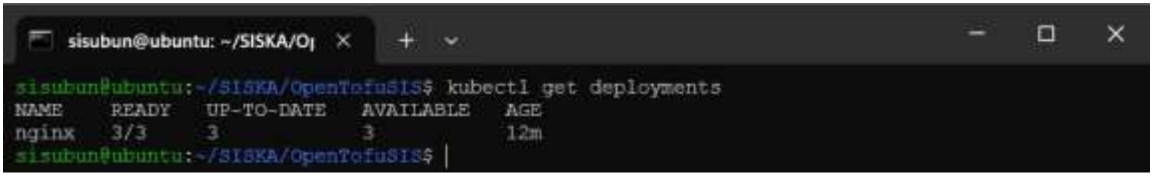
Thank you for using nginx.

6. Anda juga bisa melakukan scale pada Nginx sampai berjumlah 3 secara manual dengan menjalankan command berikut.



```
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ kubectl scale deployment nginx --replicas=3
deployment.apps/nginx scaled
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ |
```

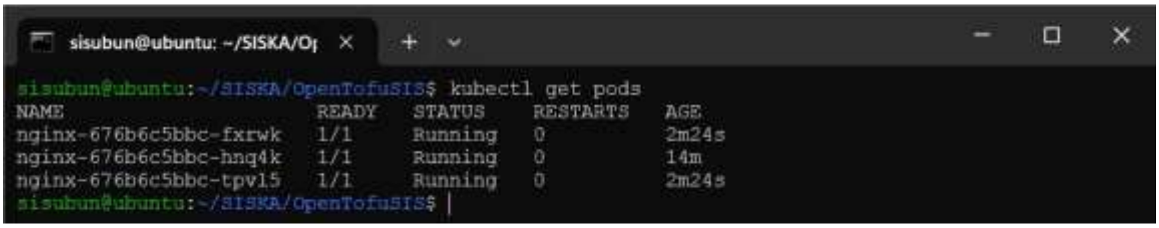
7. Lalu ceklah apakah deployment dari Nginx berhasil di-scale dengan command berikut.



```
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ kubectl get deployments
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nginx   3/3     3            3           12m
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ |
```

Terbukti bahwa deployment dari Nginx memiliki 3 replika.

8. Anda juga bisa memastikan bahwa pods dari nginx juga akan berjumlah 3 dengan command berikut.



```
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
nginx-676b6c5bbc-fxrwk   1/1     Running   0          2m24s
nginx-676b6c5bbc-hnq4k   1/1     Running   0          14m
nginx-676b6c5bbc-tpvl5   1/1     Running   0          2m24s
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ |
```

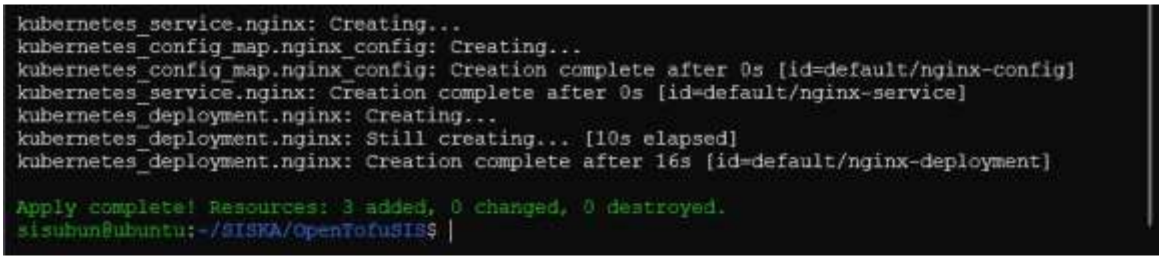
9. Lalu disini Anda akan men-deploy web server Nginx termasuk dengan konfigurasi custom menggunakan Kubernetes ConfigMap secara otomatis menggunakan OpenTofu. Anda bisa membuat file konfigurasi OpenTofu dengan ekstensi **.tf** berada pada folder yang sudah Anda buat sebelumnya.

```
provider "kubernetes" {
  config_path = "~/.kube/config"
}

# Create a ConfigMap for custom NGINX configuration
resource "kubernetes_config_map" "nginx_config" {
metadata {
    name = "nginx-config"
  }
  data = {
    "default.conf" = <<EOF
server {     listen 80;
    server_name localhost;

    location / {
        root /usr/share/nginx/html;
index index.html index.htm;
    }
}
EOF
  }
}

# Create an NGINX deployment using the ConfigMap
resource "kubernetes_deployment" "nginx" {
metadata {
    name = "nginx-deployment"
}
```

```
   spec {
replicas = 2

    selector {
match_labels = {
app = "nginx"
      }
    }      template {
metadata {
labels = {
app = "nginx"
        }
      }
      spec {          container {
name   = "nginx"            image
= "nginx:latest"

          volume_mount {
            name      = "nginx-config-volume"
mount_path = "/etc/nginx/conf.d"
          }

          port {
            container_port = 80
          }
        }

        volume {
          name = "nginx-config-volume"

          config_map {
            name = kubernetes_config_map.nginx_config.metadata[0].name
          }
        }
      }
    }
  }
}

# Expose the NGINX deployment as a service
resource "kubernetes_service" "nginx" {
metadata {
    name = "nginx-service"
  }    spec {
selector = {
app = "nginx"
    }
    type = "NodePort"
     port {       port
= 80      target_port
= 80
    }
  }
}
```

10. Lalu sekarang Anda bisa meng-apply file konfigurasi yang sudah dibuat dengan command berikut.

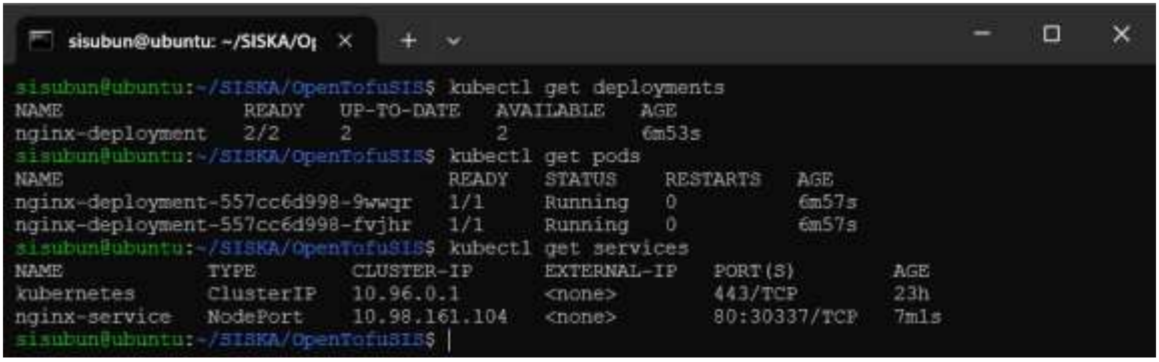
```
tofu apply
```



```
kubernetes_service.nginx: Creating...
kubernetes_config_map.nginx_config: Creating...
kubernetes_config_map.nginx_config: Creation complete after 0s [id=default/nginx-config]
kubernetes_service.nginx: Creation complete after 0s [id=default/nginx-service]
kubernetes_deployment.nginx: Creating...
kubernetes_deployment.nginx: Still creating... [10s elapsed]
kubernetes_deployment.nginx: Creation complete after 16s [id=default/nginx-deployment]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ |
```

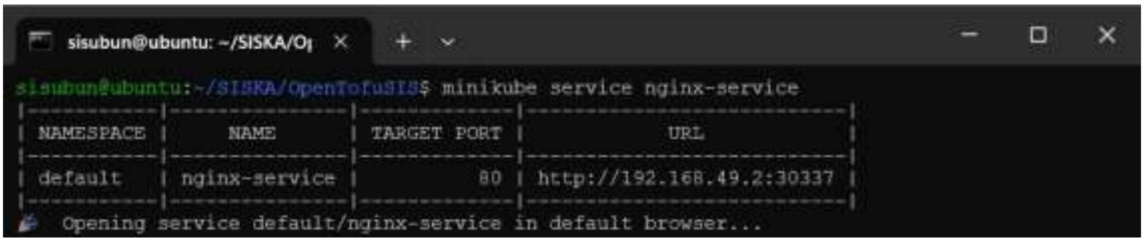
Maka akan terbentuk Nginx dengan 2 replika menggunakan custom configuration dan mengeksposnya layanannya pada port 80.

11. Lalu pastikan apakah deployment dan layanan sudah berhasil dibuat atau belum.



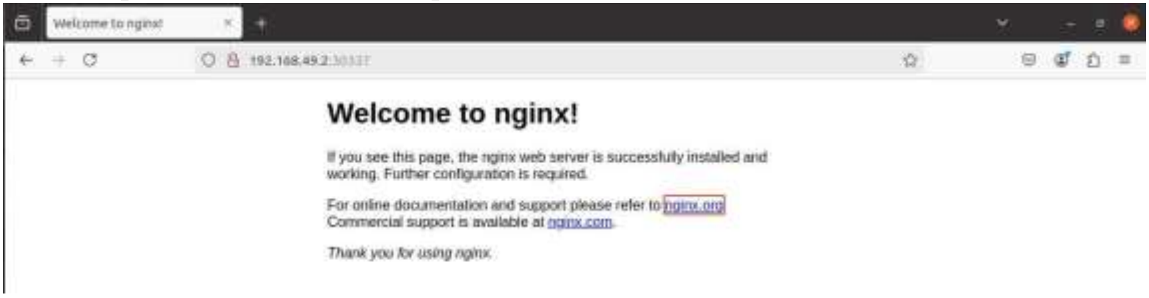
```
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           6m53s
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-557cc6d998-9wwqr   1/1     Running   0          6m57s
nginx-deployment-557cc6d998-fvjhr   1/1     Running   0          6m57s
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ kubectl get services
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP   10.96.0.1       <none>        443/TCP        23h
nginx-service   NodePort    10.98.161.104   <none>        80:30337/TCP   7m1s
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ |
```

12. Sekarang Anda sudah bisa mengakses layanan dari Nginx dengan menjalankan command berikut.



```
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ minikube service nginx-service
|-----------|---------------|-------------|---------------------------|
| NAMESPACE |     NAME      | TARGET PORT |            URL            |
|-----------|---------------|-------------|---------------------------|
| default   | nginx-service |          80 | http://192.168.49.2:30337 |
|-----------|---------------|-------------|---------------------------|
🎉  Opening service default/nginx-service in default browser...
```

13. Maka tampilan di browser akan seperti ini.



Welcome to nginx!

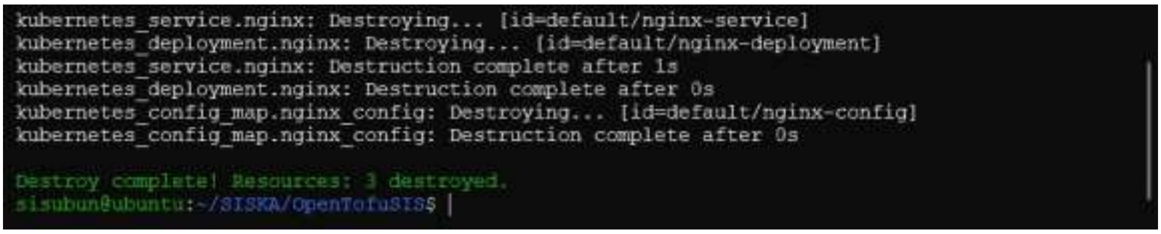
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

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

Thank you for using nginx.

14. Jika sudah selesai, Anda bisa membersihkan atau menghapusnya dengan command berikut.

```
tofu destroy
```



```
kubernetes_service.nginx: Destroying... [id=default/nginx-service]
kubernetes_deployment.nginx: Destroying... [id=default/nginx-deployment]
kubernetes_service.nginx: Destruction complete after 1s
kubernetes_deployment.nginx: Destruction complete after 0s
kubernetes_config_map.nginx_config: Destroying... [id=default/nginx-config]
kubernetes_config_map.nginx_config: Destruction complete after 0s

Destroy complete! Resources: 3 destroyed.
sisubun@ubuntu:~/SISKA/OpenTofuSIS$ |
```

15. Selesai.