

# **LAPORAN PRAKTIKUM VIRTUALISASI KOMPUTER**

## **INSTALASI MINIKUBE DAN SETUP KUBERNETES CLUSTER**



**Agus Pranata Marpaung**

**13323033**

**DIII TEKNOLOGI KOMPUTER**

**INSTITUT TEKNOLOGI DEL  
FAKULTAS VOKASI**

## Judul Praktikum

---

<b>Minggu/Sesi</b>	:	XII/3
<b>Kode Mata Kuliah</b>	:	4332103
<b>Nama Mata Kuliah</b>	:	VIRTUALISASI KOMPUTER
<b>Setoran</b>	:	Jawaban dalam bentuk <i>softcopy</i>
<b>Batas Waktu Setoran</b>	:	<i>Senin, 25 November jam 21:30</i>
<b>Tujuan</b>	:	1. Mahasiswa mampu melakukan instalasi dan pembuatan cluster Kubernetes menggunakan Minikube pada sistem operasi Ubuntu.

## Petunjuk

## Teori

### A. Apa itu Kubernetes?



# kubernetes

(Sumber: [kubernetes.io](https://kubernetes.io))

**Kubernetes** adalah *system orchestration container*. Dengan kubernetes, Anda bisa membuat container di server yang berbeda, baik fisik maupun virtual, dan semua itu dilakukan secara otomatis. Kubernetes menangani distribusi beban di beberapa server yang memungkinkan Anda untuk menggunakan *resource* secara efisien dan menghindari penggunaan yang kurang atau berlebihan *resource*. Selain itu, Kubernetes juga menangani pemantauan dan pemeriksaan container. Dan jika ada container yang gagal karena alasan tertentu, Kubernetes dapat secara otomatis mengganti container yang gagal.

### B. Apa itu Minikube?



# minikube

(Sumber: [alxibra.medium.com](https://alxibra.medium.com))

**Minikube** adalah sebuah *tools* yang bersifat open source yang memungkinkan Anda menjalankan Kubernetes cluster secara local pada computer/laptop Anda. Minikube dirancang untuk memfasilitasi pengembangan dan pengujian di lingkungan local.

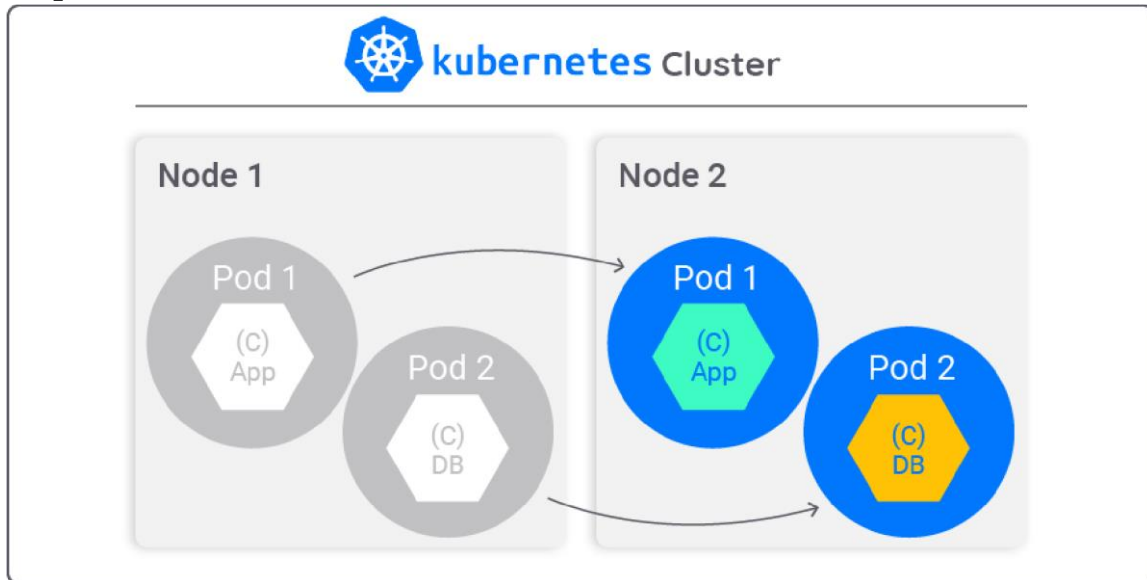
### C. Apa itu Pods?



(Sumber: [www.weave.works](https://www.weave.works))

Dalam dunia Docker, unit terkecilnya adalah container. Sama halnya dengan dunia Kubernetes, pods adalah unit terkecil di dunia Kubernetes. Dan container dibuat didalam pods, baik satu atau bahkan beberapa container. Semua container yang berada di dalam pods yang sama akan saling berbagi volume dan IP Address.

#### D. Apa itu Kubernetes Cluster dan Nodes?



(Sumber: [cloudify.co](https://cloudify.co))

Kubernetes cluster terdiri dari node. Node merupakan server baik itu server bare metal dan server virtual. Node tidak akan secara otomatis membentuk cluster tanpa Anda konfigurasi terlebih dahulu. Setelah Anda konfigurasi, semuanya akan serba otomatis.

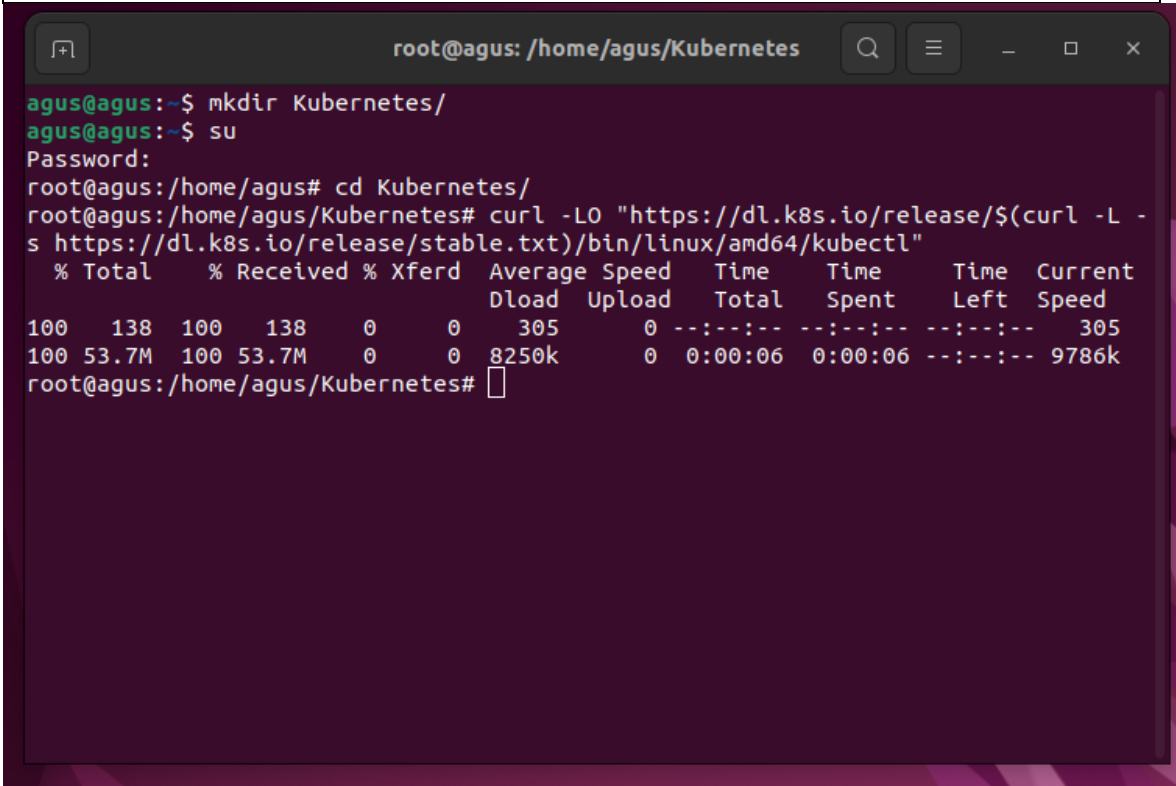
#### E. Apa itu kubectl?

**Kubectl** adalah sebuah *tools command line* yang memungkinkan Anda untuk terhubung ke Kubernetes cluster dan mengelolanya dari jarak jauh. Dengan menggunakan kubectl, Anda bisa terhubung ke layanan API pada sebuah node master melalui https.

## Praktikum

1. Sebelum dilakukan instalasi Minikube, Anda pastikan bahwa system Anda memenuhi persyaratan berikut:
  - a. CPU : 2 atau lebih
  - b. Memory : 2GB atau lebih
  - c. Harddisk : 40GB atau lebih
  - d. Koneksi internet
2. Kemudian pastikan bahwa pada host OS Anda sudah terinstal Docker. Jika tidak ada, Anda bisa melihat instalasi Docker pada praktikum sebelumnya.
3. Setelah Anda instal package yang missing secara manual, Anda downloadlah kubectl dengan menjalankan *command* berikut.

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
```

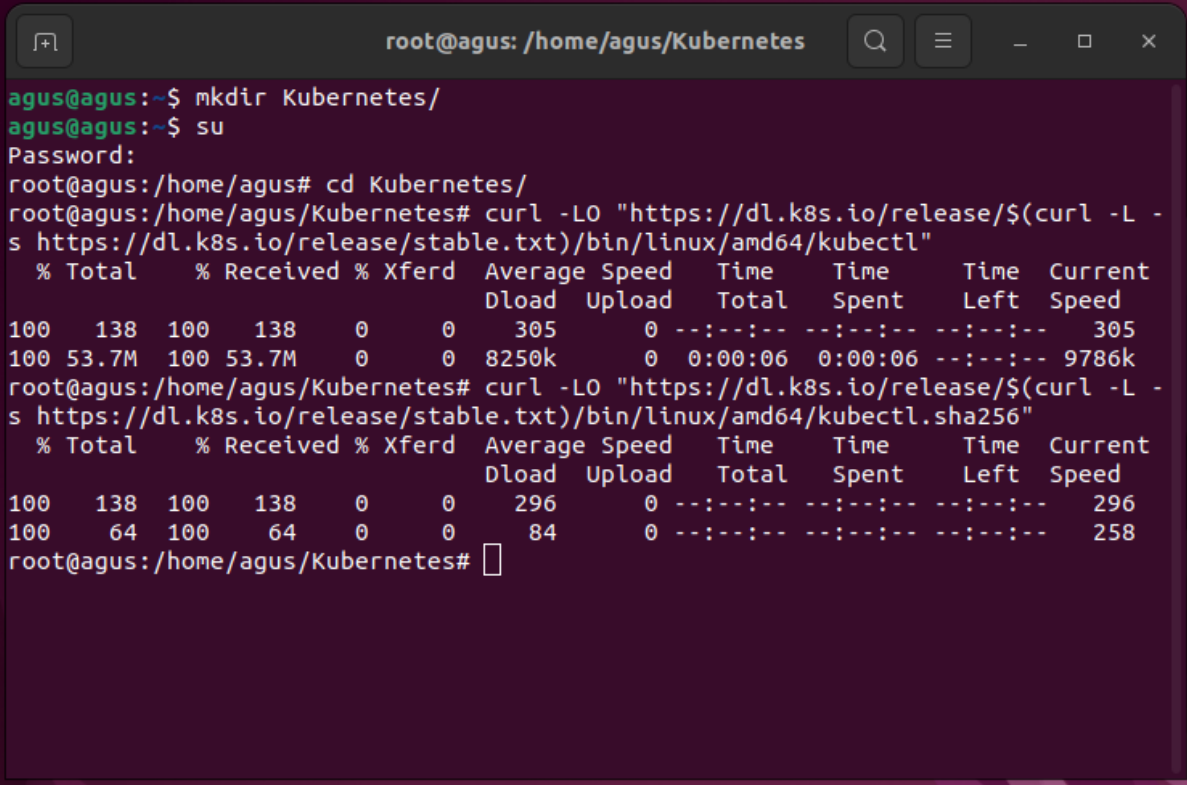


```
root@agus: /home/agus/Kubernetes

agus@agus:~$ mkdir Kubernetes/
agus@agus:~$ su
Password:
root@agus:/home/agus# cd Kubernetes/
root@agus:/home/agus/Kubernetes# curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/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   305      0  --:--:-- --:--:-- --:--:--   305
100 53.7M  100 53.7M    0    0 8250k      0  0:00:06  0:00:06 --:--:-- 9786k
root@agus:/home/agus/Kubernetes#
```

4. Lalu Anda bisa mendownload file checksum kubectl dengan menjalankan *command* berikut.

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256"
```

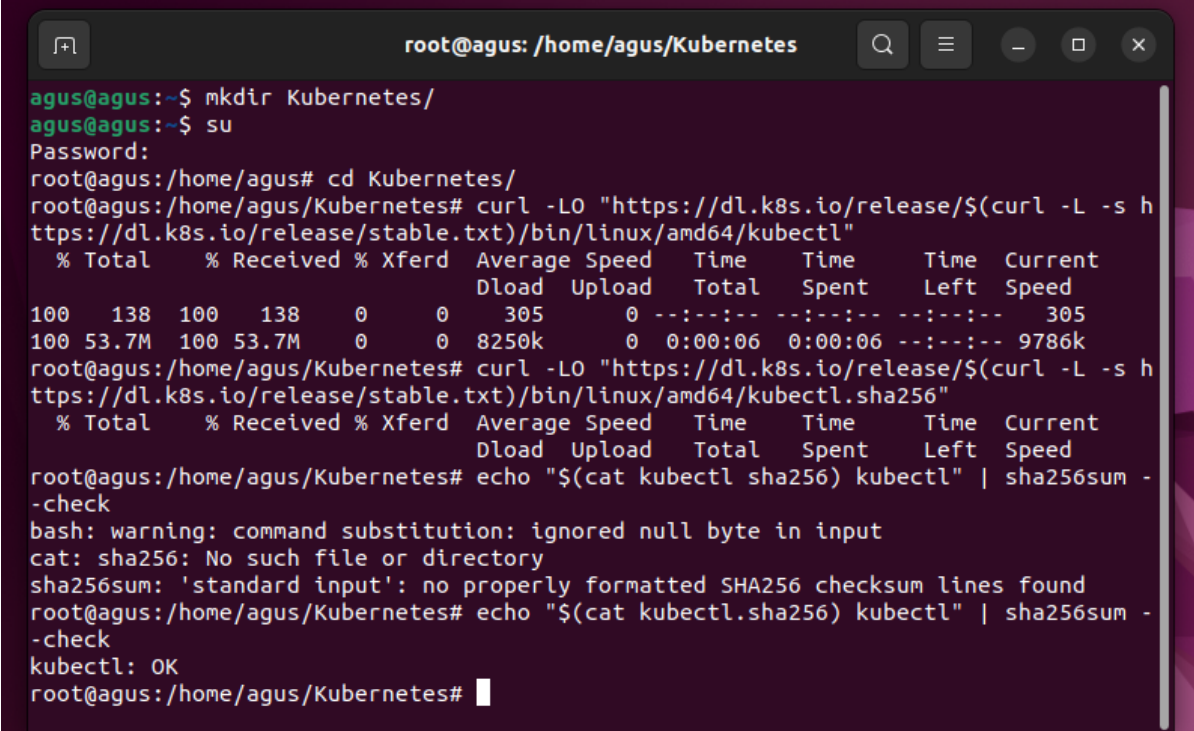


```
root@agus: /home/agus/Kubernetes

agus@agus:~$ mkdir Kubernetes/
agus@agus:~$ su
Password:
root@agus:/home/agus# cd Kubernetes/
root@agus:/home/agus/Kubernetes# curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 138 100 138 0 0 305 0 --:--:-- --:--:-- --:--:-- 305
100 53.7M 100 53.7M 0 0 8250k 0 0:00:06 0:00:06 --:--:-- 9786k
root@agus:/home/agus/Kubernetes# curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256"
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 138 100 138 0 0 296 0 --:--:-- --:--:-- --:--:-- 296
100 64 100 64 0 0 84 0 --:--:-- --:--:-- --:--:-- 258
root@agus:/home/agus/Kubernetes#
```

5. Kemudian Anda bisa melakukan validasi kubectl terhadap file checksum dengan menjalankan *command* berikut.

```
echo "$(cat kubectl.sha256) kubectl" | sha256sum --check
```



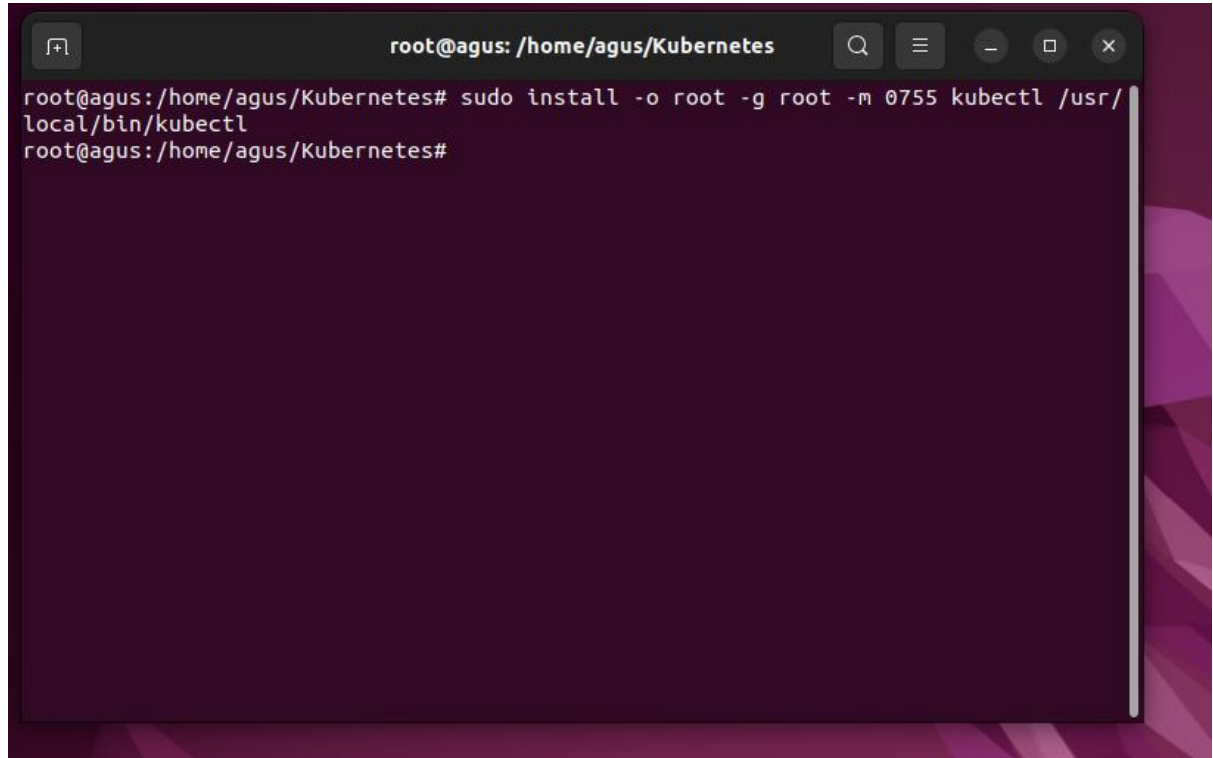
```
root@agus: /home/agus/Kubernetes

agus@agus:~$ mkdir Kubernetes/
agus@agus:~$ su
Password:
root@agus:/home/agus# cd Kubernetes/
root@agus:/home/agus/Kubernetes# curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 138 100 138 0 0 305 0 --:--:-- --:--:-- --:--:-- 305
100 53.7M 100 53.7M 0 0 8250k 0 0:00:06 0:00:06 --:--:-- 9786k
root@agus:/home/agus/Kubernetes# curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256"
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
root@agus:/home/agus/Kubernetes# echo "$(cat kubectl sha256) kubectl" | sha256sum --check
bash: warning: command substitution: ignored null byte in input
cat: sha256: No such file or directory
sha256sum: 'standard input': no properly formatted SHA256 checksum lines found
root@agus:/home/agus/Kubernetes# echo "$(cat kubectl.sha256) kubectl" | sha256sum --check
kubectl: OK
root@agus:/home/agus/Kubernetes#
```

Pada gambar diatas menunjukkan bahwa kubectl sudah valid.

- Setelah didownload dan sudah valid, Anda bisa lakukan instalasi kubectl dengan menjalankan *command* berikut.

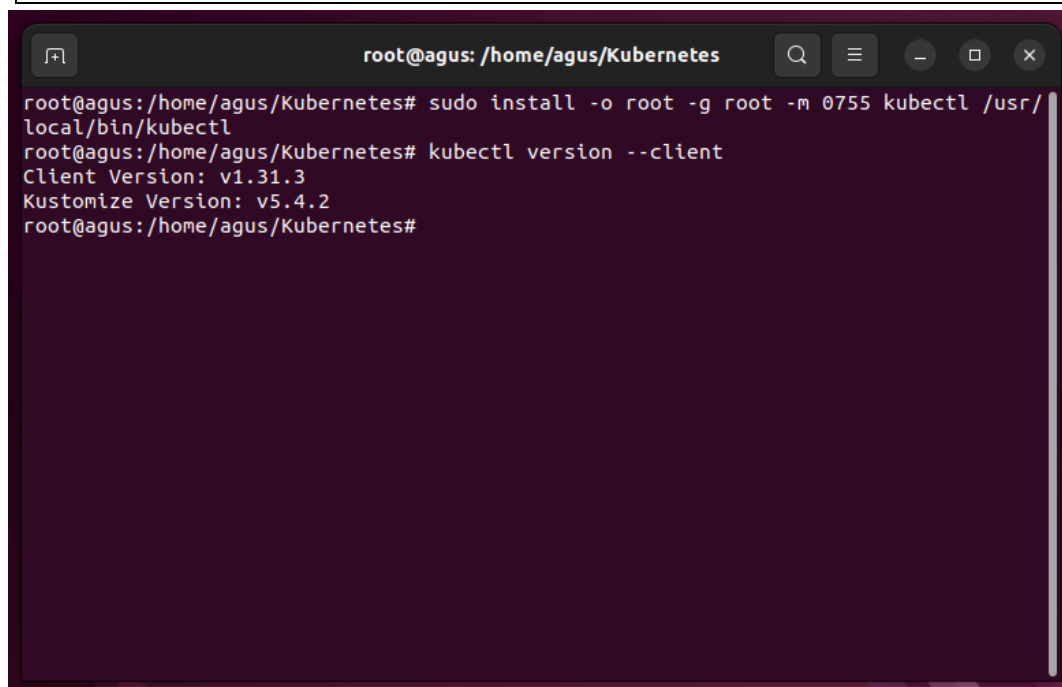
```
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

A terminal window titled 'root@agus: /home/agus/Kubernetes' with search, menu, and window control icons. The terminal shows the command 'sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl' being executed. The output shows the command being run and then the prompt returning to 'root@agus: /home/agus/Kubernetes#'.

```
root@agus: /home/agus/Kubernetes# sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
root@agus: /home/agus/Kubernetes#
```

- Setelah terinstal Anda bisa memastikan versi kubectl yang Anda instal dengan menjalankan *command* berikut.

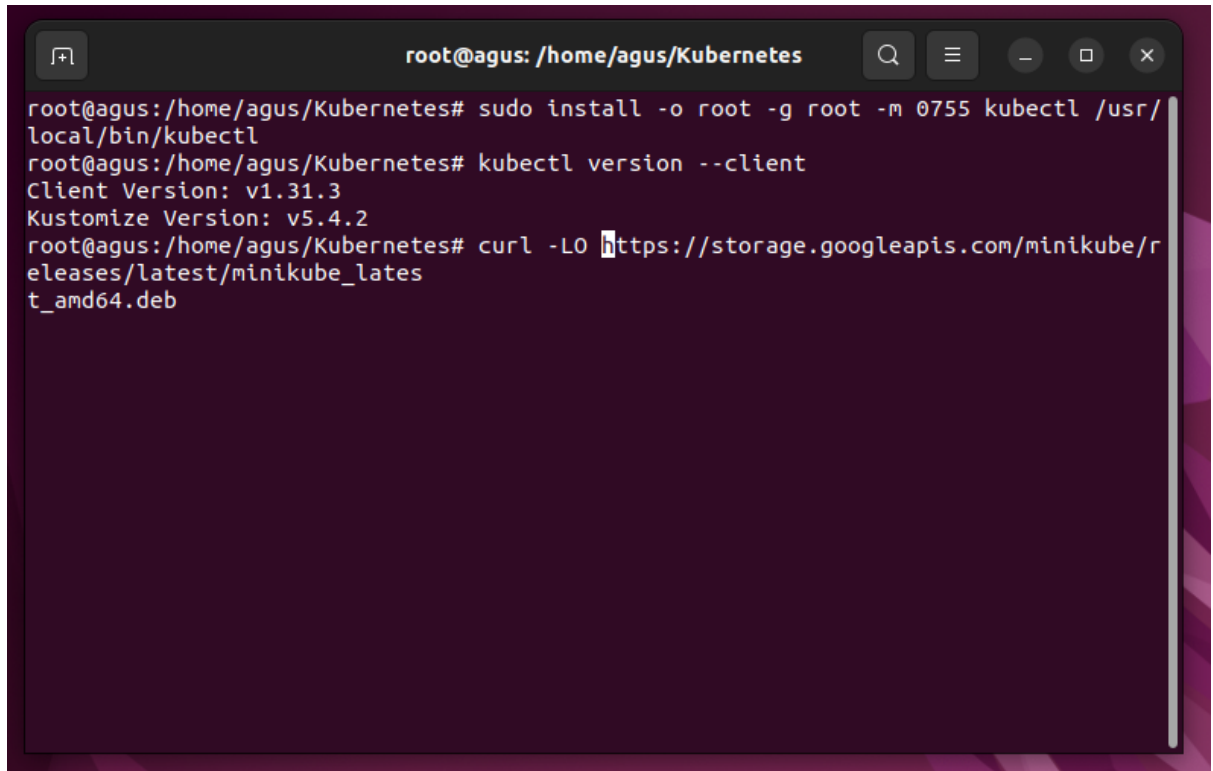
```
kubectl version --client
```

A terminal window titled 'root@agus: /home/agus/Kubernetes' with search, menu, and window control icons. The terminal shows the command 'sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl' being executed, followed by the command 'kubectl version --client'. The output shows 'Client Version: v1.31.3' and 'Kustomize Version: v5.4.2', followed by the prompt returning to 'root@agus: /home/agus/Kubernetes#'.

```
root@agus: /home/agus/Kubernetes# sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
root@agus: /home/agus/Kubernetes# kubectl version --client
Client Version: v1.31.3
Kustomize Version: v5.4.2
root@agus: /home/agus/Kubernetes#
```

8. Setelah itu, Anda sudah bisa melakukan download Minikube dengan menjalankan *command* berikut.

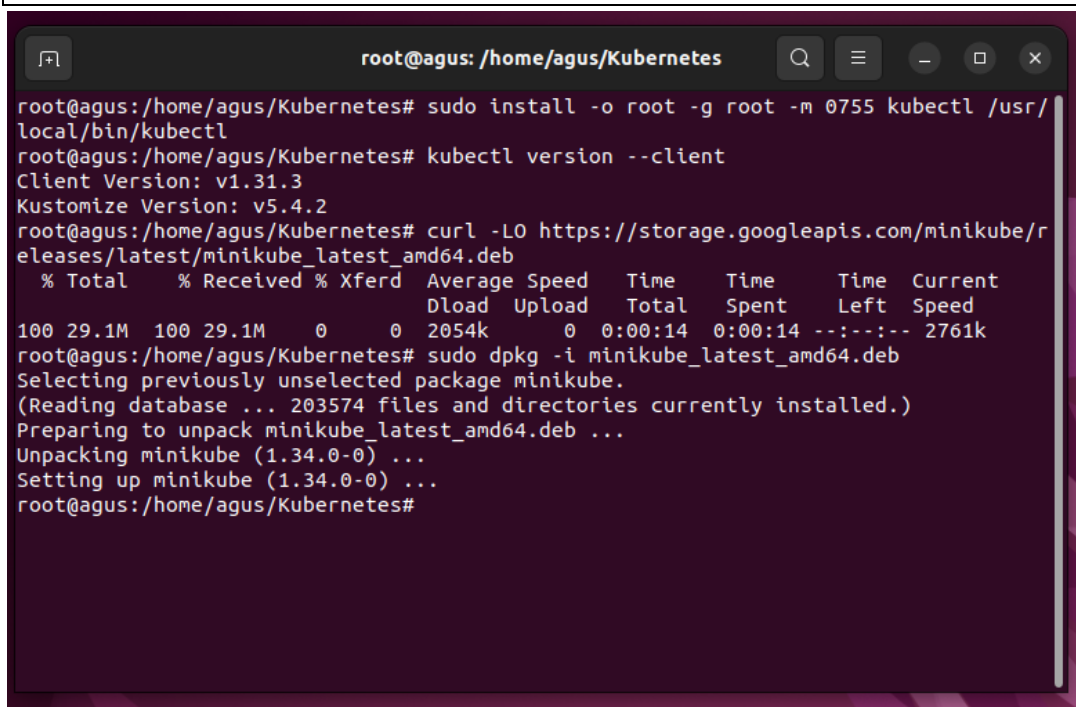
```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
```



```
root@agus: /home/agus/Kubernetes
root@agus:/home/agus/Kubernetes# sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
root@agus:/home/agus/Kubernetes# kubectl version --client
Client Version: v1.31.3
Kustomize Version: v5.4.2
root@agus:/home/agus/Kubernetes# curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
```

9. Setelah didownload, Anda bisa lakukan instalasi minikube menggunakan package Debian dengan menjalankan *command* berikut.

```
sudo dpkg -i minikube_latest_amd64.deb
```

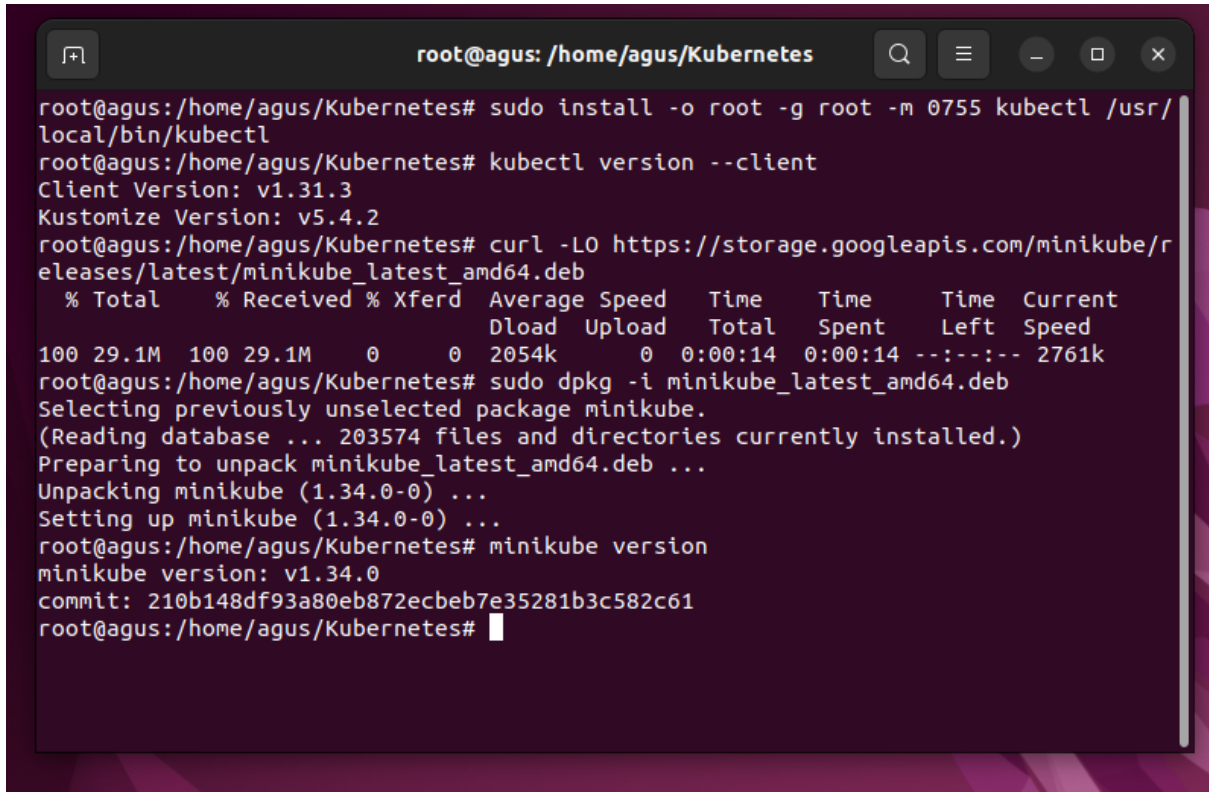


```
root@agus:/home/agus/Kubernetes# sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
root@agus:/home/agus/Kubernetes# kubectl version --client
Client Version: v1.31.3
Kustomize Version: v5.4.2
root@agus:/home/agus/Kubernetes# curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left    Speed
100 29.1M  100 29.1M    0     0 2054k      0  0:00:14  0:00:14 --:--:-- 2761k
root@agus:/home/agus/Kubernetes# sudo dpkg -i minikube_latest_amd64.deb
Selecting previously unselected package minikube.
(Reading database ... 203574 files and directories currently installed.)
Preparing to unpack minikube_latest_amd64.deb ...
Unpacking minikube (1.34.0-0) ...
Setting up minikube (1.34.0-0) ...
root@agus:/home/agus/Kubernetes#
```



10. Kemudian Anda bisa memastikan versinya dengan menjalankan *command* berikut.

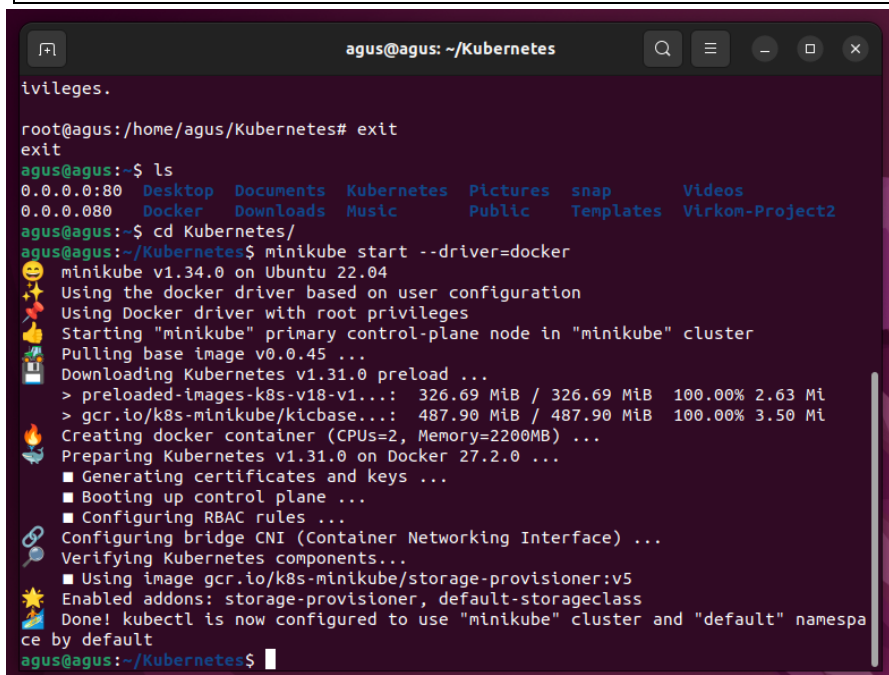
```
minikube version
```



```
root@agus:/home/agus/Kubernetes# sudo install -o root -g root -m 0755 kubectrl /usr/local/bin/kubectrl
root@agus:/home/agus/Kubernetes# kubectl version --client
Client Version: v1.31.3
Kustomize Version: v5.4.2
root@agus:/home/agus/Kubernetes# 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 29.1M  100 29.1M    0     0  2054k      0  0:00:14  0:00:14 --:--:-- 2761k
root@agus:/home/agus/Kubernetes# sudo dpkg -i minikube_latest_amd64.deb
Selecting previously unselected package minikube.
(Reading database ... 203574 files and directories currently installed.)
Preparing to unpack minikube_latest_amd64.deb ...
Unpacking minikube (1.34.0-0) ...
Setting up minikube (1.34.0-0) ...
root@agus:/home/agus/Kubernetes# minikube version
minikube version: v1.34.0
commit: 210b148df93a80eb872ecbeb7e35281b3c582c61
root@agus:/home/agus/Kubernetes#
```

11. Setelah itu Anda akan mencoba membuat sebuah cluster Kubernetes dengan mengetikkan *command* berikut.

```
minikube start --driver=docker
```



```
agus@agus: ~/Kubernetes

ivileges.

root@agus:/home/agus/Kubernetes# exit
exit
agus@agus:~$ ls
0.0.0.0:80 Desktop Documents Kubernetes Pictures snap Videos
0.0.0.080 Docker Downloads Music Public Templates Virkom-Project2
agus@agus:~$ cd Kubernetes/
agus@agus:~/Kubernetes$ minikube start --driver=docker
🐳 minikube v1.34.0 on Ubuntu 22.04
🌟 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% 2.63 Mi
> gcr.io/k8s-minikube/kicbase...: 487.90 MiB / 487.90 MiB 100.00% 3.50 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 default
agus@agus:~/Kubernetes$
```

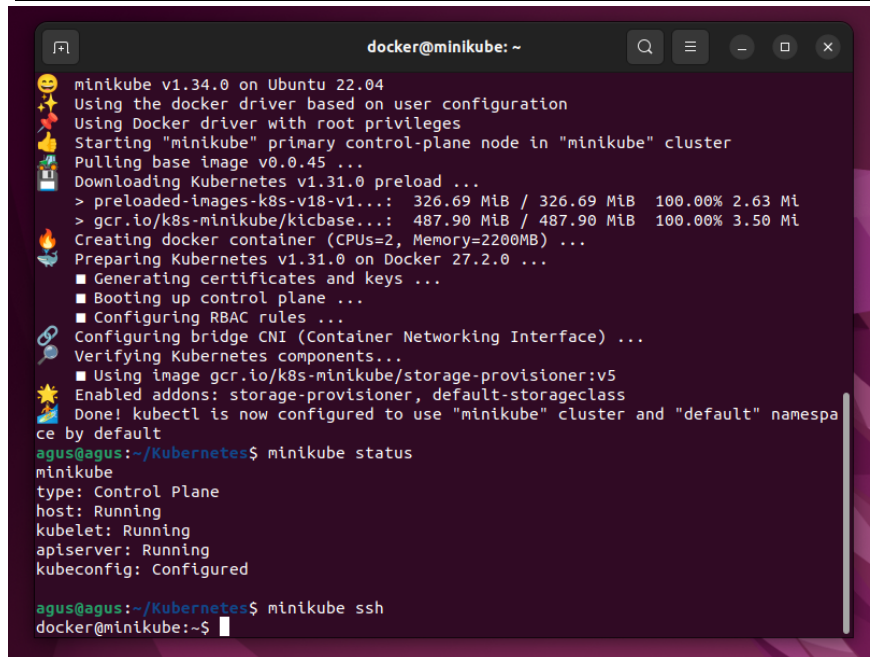
12. Lalu Anda bisa melihat status dari minikube itu sendiri dengan menjalankan *command* berikut.

```
minikube status
```

```
agus@agus:~/Kubernetes$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

13. Setelah itu Anda bisa melakukan remote menggunakan protocol ssh dengan menjalankan *command* berikut.

```
minikube ssh
```



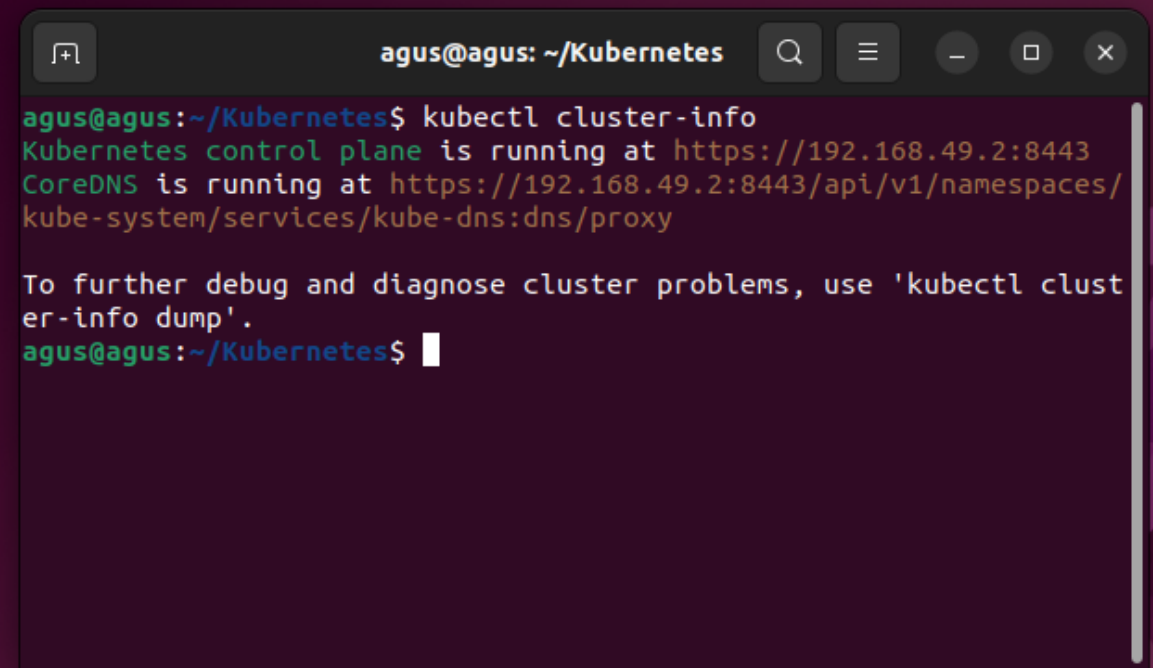
```
docker@minikube: ~
minikube v1.34.0 on Ubuntu 22.04
Using the docker driver based on user configuration
Using Docker driver with root privileges
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.45 ...
Downloading Kubernetes v1.31.0 preload ...
> preloaded-images-k8s-v18-v1...: 326.69 MiB / 326.69 MiB 100.00% 2.63 Mi
> gcr.io/k8s-minikube/kicbase...: 487.90 MiB / 487.90 MiB 100.00% 3.50 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 default
agus@agus:~/Kubernetes$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
agus@agus:~/Kubernetes$ minikube ssh
docker@minikube: ~
```

14. Setelah berhasil diremote, sekarang Anda bisa memverifikasi container apa saja yang dibuat didalam node yang baru Anda buat dengan menjalankan *command* berikut.

```
agus@agus:~/Kubernetes$ minikube ssh
docker@minikube:~$ docker ps
CONTAINER ID   IMAGE                                     COMMAND                  CREATED        STATUS        P
ORTS          NAMES
68fb67b518a8   6e38f40d628d                            "/storage-provisioner"   4 minutes ago Up 4 minutes
k8s_storage-provisioner_storage-provisioner_kube-system_5e59e60e-7c31-4fce-8b88-7afee3496a5d_0
a3b40f3cf073   registry.k8s.io/pause:3.10              "/pause"                4 minutes ago Up 4 minutes
k8s_POD_storage-provisioner_kube-system_5e59e60e-7c31-4fce-8b88-7afee3496a5d_0
4b7bceb74470   cbb01a7bd410                            "/coredns -conf /etc..." 4 minutes ago Up 4 minutes
k8s_coredns_coredns-6f6b679f8f-56m9j_kube-system_cd95dd50-4aa7-4eb6-9f0b-393764084896_0
d8ccadb5f77c   registry.k8s.io/pause:3.10              "/pause"                4 minutes ago Up 4 minutes
k8s_POD_coredns-6f6b679f8f-56m9j_kube-system_cd95dd50-4aa7-4eb6-9f0b-393764084896_0
760994cd9d13   ad83b2ca7b09                            "/usr/local/bin/kube..." 4 minutes ago Up 4 minutes
k8s_kube-proxy_kube-proxy-tmq5s_kube-system_5514bc28-8726-44d9-acec-a75edfcd218_0
b6d8219201a5   registry.k8s.io/pause:3.10              "/pause"                4 minutes ago Up 4 minutes
k8s_POD_kube-proxy-tmq5s_kube-system_5514bc28-8726-44d9-acec-a75edfcd218_0
1374060a848d   1766f54c897f                            "kube-scheduler --au..." 5 minutes ago Up 5 minutes
k8s_kube-scheduler_kube-scheduler-minikube_kube-system_e039200acb850c82bb901653cc38ff6e_0
c6bff400ef21   604f5db92eaa                            "kube-apiserver --ad..." 5 minutes ago Up 5 minutes
k8s_kube-apiserver_kube-apiserver-minikube_kube-system_9e315b3a91fa9f6f7463439d9daci56_0
39ca9d6e6e3e   045733566833                            "kube-controller-man..." 5 minutes ago Up 5 minutes
k8s_kube-controller-manager_kube-controller-manager-minikube_kube-system_40f5f661ab65f2e4bfe
41ac2993c01de_0
1b9be4550822   2e96e5913fc0                            "etcd --advertise-cl..." 5 minutes ago Up 5 minutes
k8s_etcd_etcd-minikube_kube-system_a5363f4f31e043bdae3c93aca4991903_0
4bd1e054f9ba   registry.k8s.io/pause:3.10              "/pause"                5 minutes ago Up 5 minutes
k8s_POD_kube-scheduler-minikube_kube-system_e039200acb850c82bb901653cc38ff6e_0
a73390138692   registry.k8s.io/pause:3.10              "/pause"                5 minutes ago Up 5 minutes
k8s_POD_kube-controller-manager-minikube_kube-system_40f5f661ab65f2e4bfe41ac2993c01de_0
a75b87bd4f29   registry.k8s.io/pause:3.10              "/pause"                5 minutes ago Up 5 minutes
k8s_POD_kube-apiserver-minikube_kube-system_9e315b3a91fa9f6f7463439d9daci56_0
5172628d6efa   registry.k8s.io/pause:3.10              "/pause"                5 minutes ago Up 5 minutes
k8s_POD_etcd-minikube_kube-system_a5363f4f31e043bdae3c93aca4991903_0
docker@minikube:~$
```

15. Kemudian Anda bisa melihat informasi yang tersedia pada Kubernetes cluster yang baru saja Anda buat dengan menjalankan command berikut

```
kubectl cluster-info
```



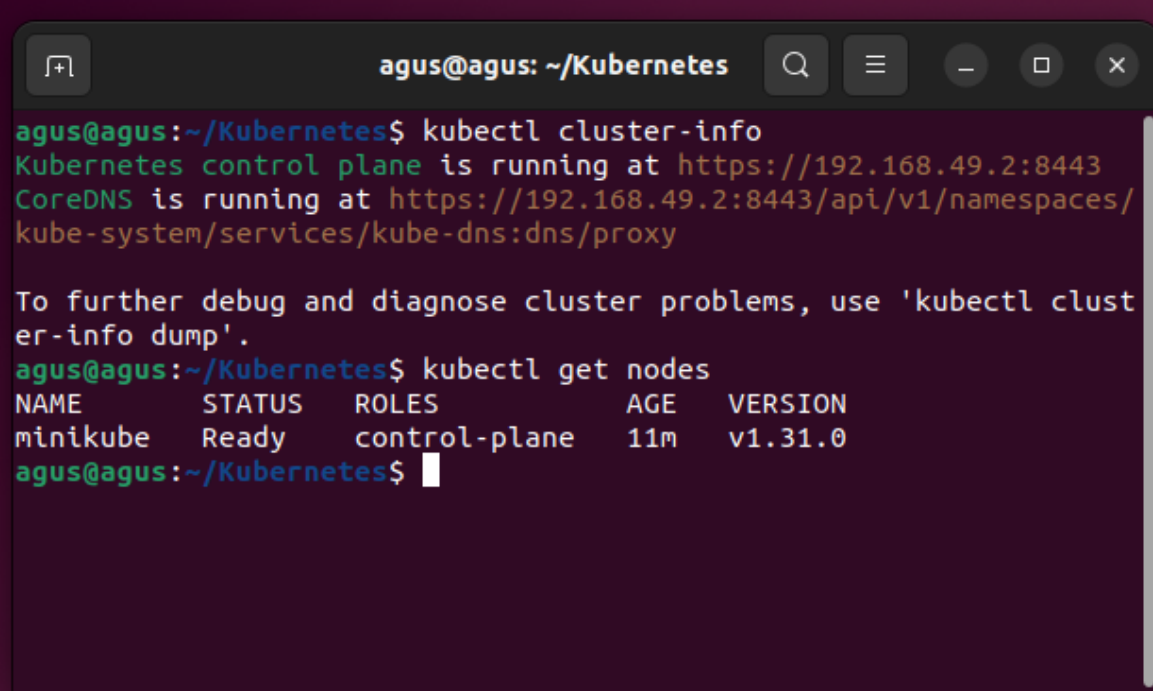
A terminal window titled 'agus@agus: ~/Kubernetes' showing the output of the 'kubectl cluster-info' command. The output indicates that the Kubernetes control plane is running at 'https://192.168.49.2:8443' and CoreDNS is running at 'https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy'. It also provides a tip to use 'kubectl cluster-info dump' for further debugging. The prompt 'agus@agus:~/Kubernetes\$' is visible at the bottom.

```
agus@agus:~/Kubernetes$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/
kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl clust
er-info dump'.
agus@agus:~/Kubernetes$
```

16. Lalu Anda lihatlah daftar node yang tersedia di Kubernetes cluster Anda dengan menjalankan *command* berikut.

```
kubectl get nodes
```



A terminal window titled 'agus@agus: ~/Kubernetes' showing the output of the 'kubectl get nodes' command. The output displays a table with one node named 'minikube' in a 'Ready' status, with the role 'control-plane', an age of '11m', and version 'v1.31.0'. The prompt 'agus@agus:~/Kubernetes\$' is visible at the bottom.

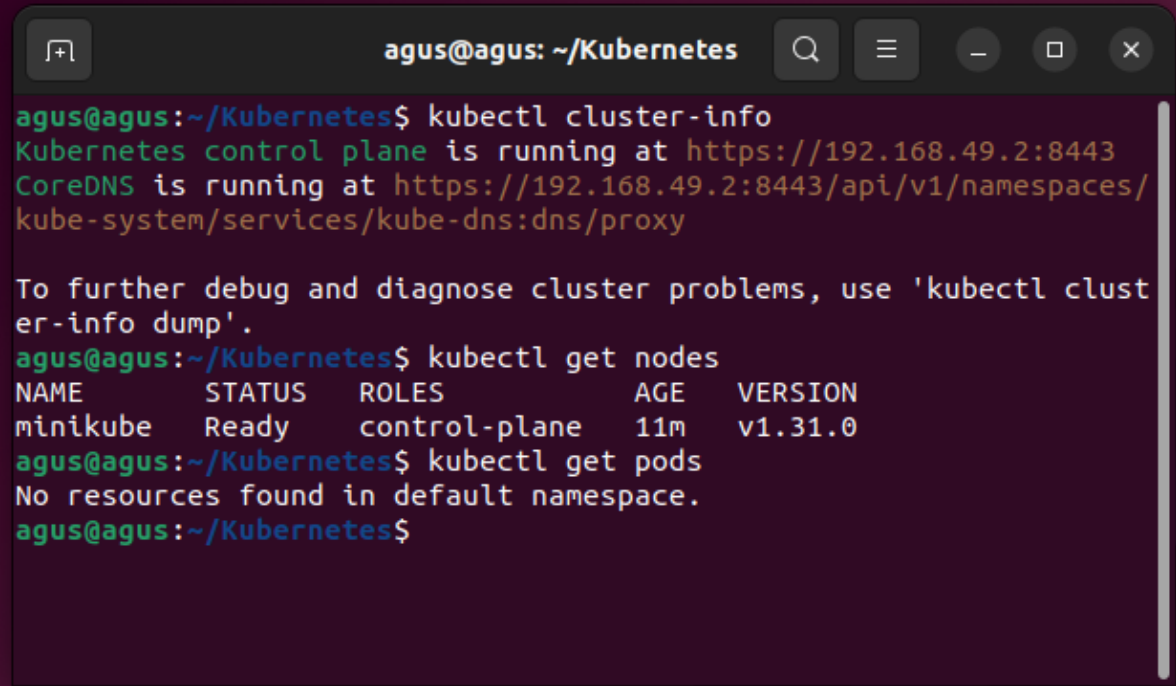
```
agus@agus:~/Kubernetes$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/
kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl clust
er-info dump'.
agus@agus:~/Kubernetes$ kubectl get nodes
NAME        STATUS    ROLES    AGE   VERSION
minikube    Ready    control-plane   11m   v1.31.0
agus@agus:~/Kubernetes$
```

Disini terlihat bahwa hanya satu node dengan nama minikube yang tersedia, karena minicube membuat cluster node tunggal.

17. Kemudian, Anda bisa melihat pods yang tersedia didalam namespace dengan menjalankan *command* berikut.

```
kubectl get pods
```

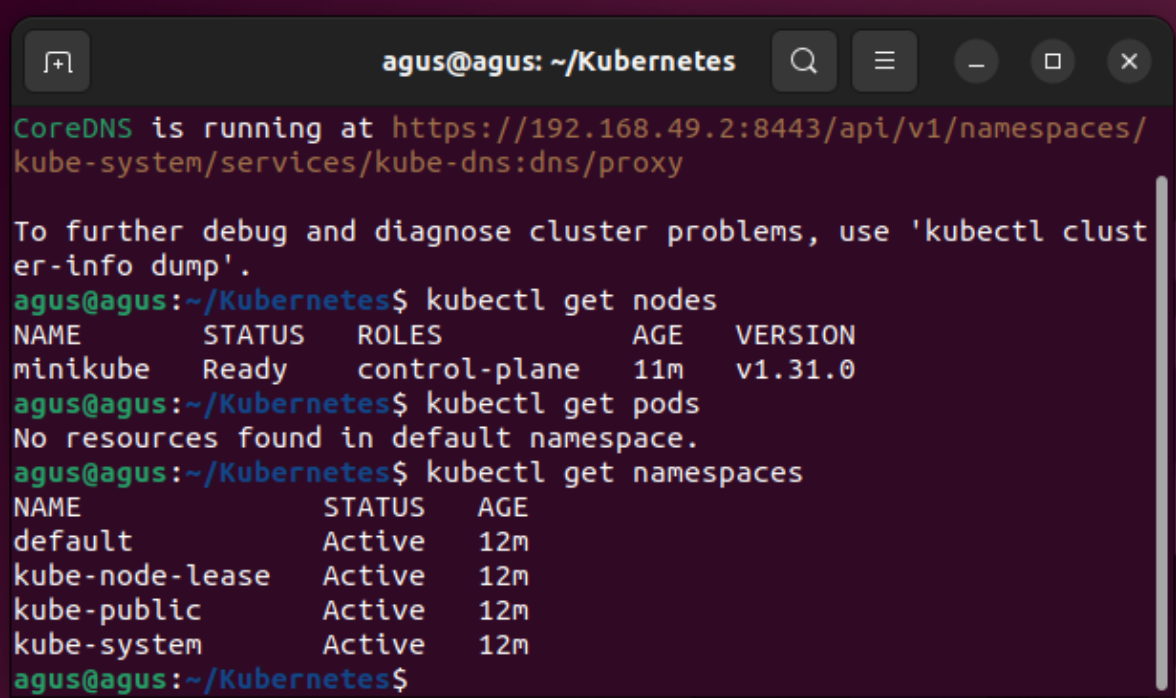


```
agus@agus: ~/Kubernetes
agus@agus:~/Kubernetes$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/
kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl clust
er-info dump'.
agus@agus:~/Kubernetes$ kubectl get nodes
NAME          STATUS    ROLES          AGE    VERSION
minikube      Ready     control-plane   11m    v1.31.0
agus@agus:~/Kubernetes$ kubectl get pods
No resources found in default namespace.
agus@agus:~/Kubernetes$
```

18. Lalu Anda bisa melihat daftar namespace yang tersedia saat ini pada Kubernetes cluster Anda dengan menjalankan *command* berikut.

```
kubectl get namespaces
```



```
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/
kube-system/services/kube-dns:dns/proxy

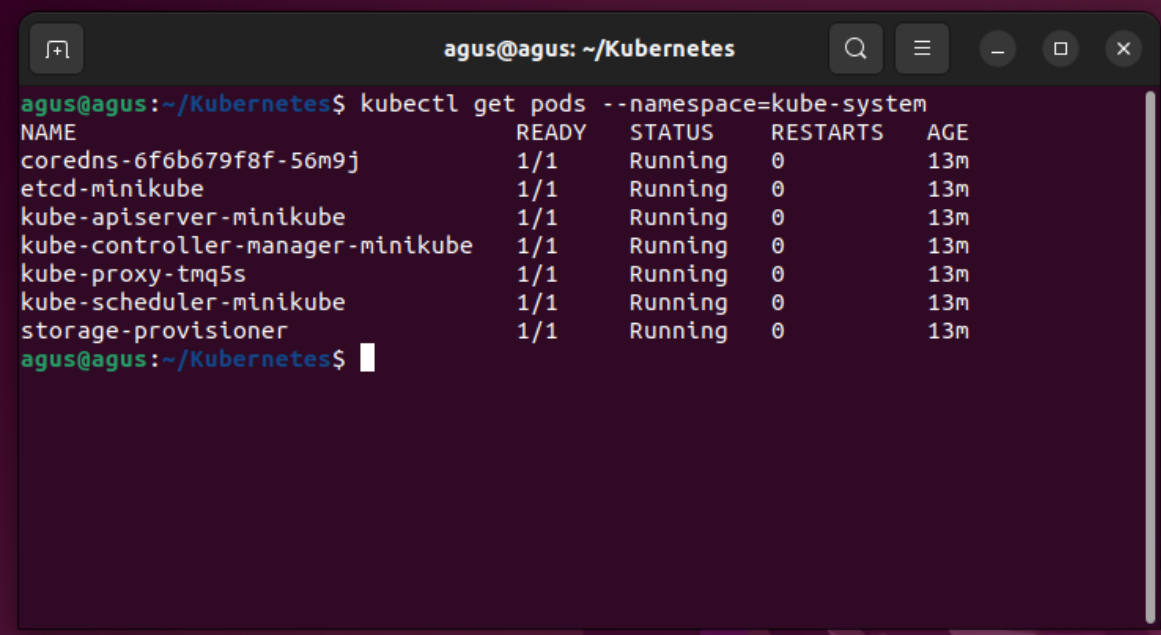
To further debug and diagnose cluster problems, use 'kubectl clust
er-info dump'.
agus@agus:~/Kubernetes$ kubectl get nodes
NAME          STATUS    ROLES          AGE    VERSION
minikube      Ready     control-plane   11m    v1.31.0
agus@agus:~/Kubernetes$ kubectl get pods
No resources found in default namespace.
agus@agus:~/Kubernetes$ kubectl get namespaces
NAME          STATUS    AGE
default       Active    12m
kube-node-lease Active    12m
kube-public   Active    12m
kube-system   Active    12m
agus@agus:~/Kubernetes$
```

**Note:**

Namespaces digunakan untuk mengelompokkan *resource* yang berbeda dan objek konfigurasi.

19. Jika Anda ingin mengetahui pods mana yang berjalan didalam namespace tertentu, Anda bisa menjalankan *command* berikut.

```
kubectl get pods --namespace=kube-system
```



The image shows a terminal window titled 'agus@agus: ~/Kubernetes'. The command 'kubectl get pods --namespace=kube-system' has been executed, resulting in a table of pods. The table has five columns: NAME, READY, STATUS, RESTARTS, and AGE. The pods listed are coredns-6f6b679f8f-56m9j, etcd-minikube, kube-apiserver-minikube, kube-controller-manager-minikube, kube-proxy-tmq5s, kube-scheduler-minikube, and storage-provisioner. All pods are in a 'Running' state with 0 restarts and have been running for 13 minutes.

NAME	READY	STATUS	RESTARTS	AGE
coredns-6f6b679f8f-56m9j	1/1	Running	0	13m
etcd-minikube	1/1	Running	0	13m
kube-apiserver-minikube	1/1	Running	0	13m
kube-controller-manager-minikube	1/1	Running	0	13m
kube-proxy-tmq5s	1/1	Running	0	13m
kube-scheduler-minikube	1/1	Running	0	13m
storage-provisioner	1/1	Running	0	13m

Disini terlihat bahwa beberapa pods yang berjalan pada namespace **kube-system**.

20. Selesai!