NAMA : FIDELIA PING
NIM : 245410012
KELAS : INFORMATIKA 1

MODUL 10
Simulasi Arsitektur Client-Server Menggunakan Docker di AWS EC2

DASAR TEORI

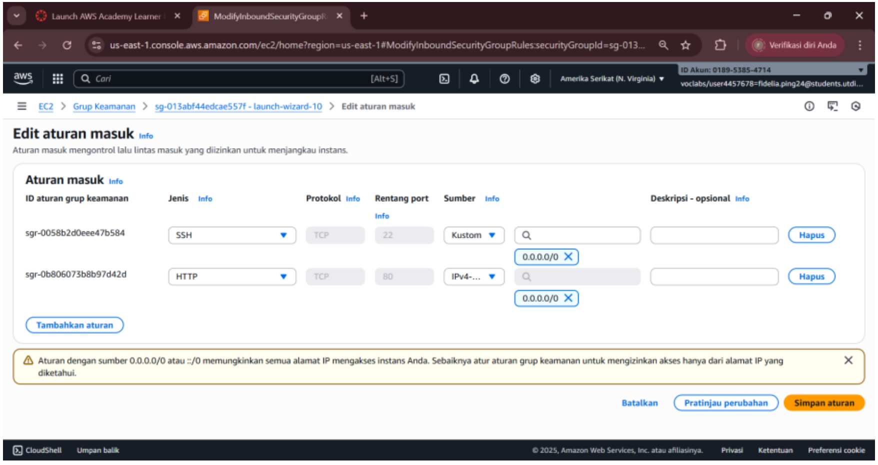
Simulasi Arsitektur Client-Server menggunakan Docker di AWS EC2 adalah proses membangun lingkungan percobaan di mana server dan client dijalankan dalam container Docker pada instance EC2 untuk mempelajari cara kerja komunikasi jaringan, pengelolaan layanan, isolasi aplikasi, serta pengujian performa dalam sistem terdistribusi; Docker digunakan untuk membuat lingkungan yang konsisten dan terisolasi, sementara EC2 menyediakan infrastruktur cloud berskala sehingga model client-server dapat diuji secara realistis seperti pada deployment sebenarnya.

PRAKTIK

1. Konfigurasi VM 1 (Server Application)

A. Konfigurasi Security Group (Inbound Rules

Type: SSH | **Port:** 22 | **Source:** Anywhere-IPv4 (0.0.0.0/0)
Type: HTTP | **Port:** 80 | **Source:** Anywhere-IPv4 (0.0.0.0/0)


Launch AWS Academy Learner
ModifyInboundSecurityGroupR
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-013...
Verifikasi diri Anda
aws
Cari
[Alt+S]
Amerika Serikat (N. Virginia)
ID Akun: 0189-5385-4714
voclabs/user4457678=fidelia.ping24@students.utdi...
EC2 > Grup Keamanan > sg-013abf44edcae557f - launch-wizard-10 > Edit aturan masuk

Edit aturan masuk Info

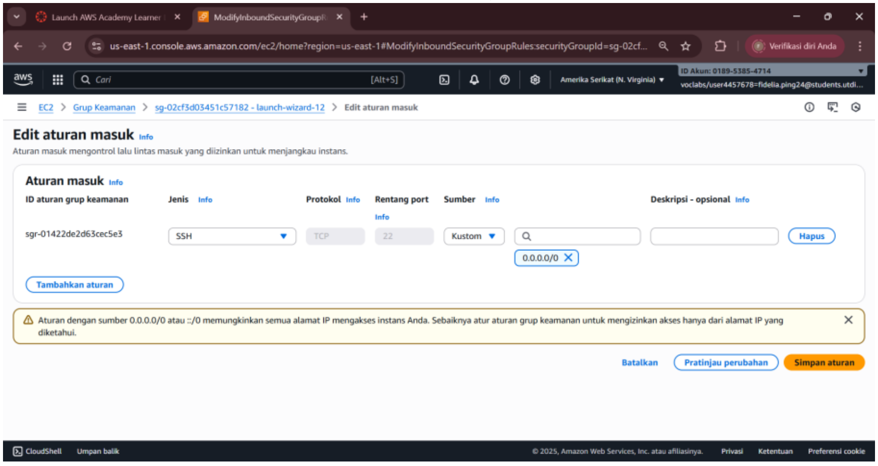
Aturan masuk mengontrol lalu lintas masuk yang diizinkan untuk menjangkau instans.

Aturan masuk Info

ID aturan grup keamanan	Jenis Info	Protokol Info	Rentang port Info	Sumber Info		Deskripsi - opsional Info
sgr-0058b2d0eee47b584	SSH	TCP	22	Kustom	0.0.0.0/0	
sgr-0b806073b8b97d42d	HTTP	TCP	80	IPv4-...	0.0.0.0/0	

Aturan dengan sumber 0.0.0.0/0 atau ::/0 memungkinkan semua alamat IP mengakses instans Anda. Sebaiknya atur aturan grup keamanan untuk mengizinkan akses hanya dari alamat IP yang diketahui.

VM 2 (Client) default cukup ssh saja


Launch AWS Academy Learner
ModifyInboundSecurityGroupR
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-02cf...
Verifikasi diri Anda
aws
Cari
[Alt+S]
Amerika Serikat (N. Virginia)
ID Akun: 0189-5385-4714
voclabs/user4457678=fidelia.ping24@students.utdi...
EC2 > Grup Keamanan > sg-02cf3d03451c57182 - launch-wizard-12 > Edit aturan masuk

Edit aturan masuk Info

Aturan masuk mengontrol lalu lintas masuk yang diizinkan untuk menjangkau instans.

Aturan masuk Info

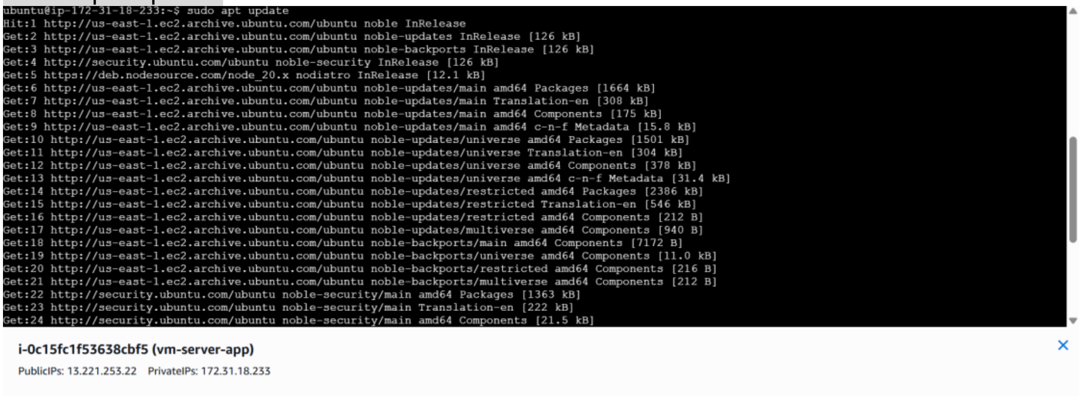
ID aturan grup keamanan	Jenis Info	Protokol Info	Rentang port Info	Sumber Info		Deskripsi - opsional Info
sgr-01422de2d63cec5e3	SSH	TCP	22	Kustom	0.0.0.0/0	

Aturan dengan sumber 0.0.0.0/0 atau ::/0 memungkinkan semua alamat IP mengakses instans Anda. Sebaiknya atur aturan grup keamanan untuk mengizinkan akses hanya dari alamat IP yang diketahui.

B. Instalasi Docker dan Git

# Update repository

```
sudo apt update
```



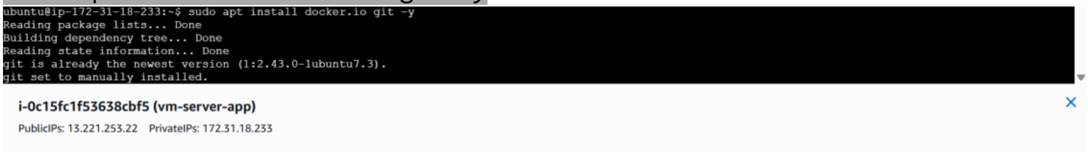
```
ubuntu@ip-172-31-18-233:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 https://deb.nodesource.com/node_20.x nodistro InRelease [12.1 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1664 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [308 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.8 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1501 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [304 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [378 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.4 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2386 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [546 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7172 B]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:22 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1363 kB]
Get:23 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [222 kB]
Get:24 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
```

i-0c15fc1f53638cbf5 (vm-server-app)

PublicIPs: 13.221.253.22 PrivateIPs: 172.31.18.233

# Install Docker dan Git

```
sudo apt install docker.io git -y
```



```
ubuntu@ip-172-31-18-233:~$ sudo apt install docker.io git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.3).
git set to manually installed.
```

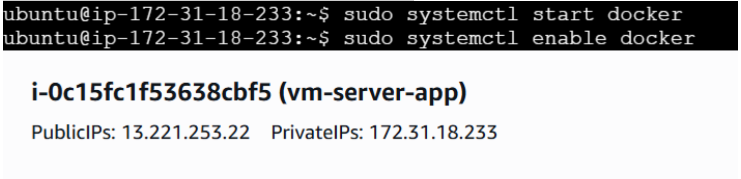
i-0c15fc1f53638cbf5 (vm-server-app)

PublicIPs: 13.221.253.22 PrivateIPs: 172.31.18.233

C. Menjalankan Service Docker

# Start Docker dan set agar otomatis jalan saat restart

```
sudo systemctl start docker
sudo systemctl enable docker
```



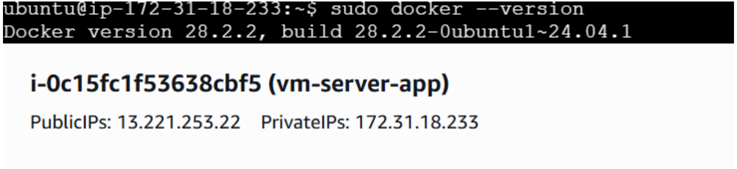
```
ubuntu@ip-172-31-18-233:~$ sudo systemctl start docker
ubuntu@ip-172-31-18-233:~$ sudo systemctl enable docker
```

i-0c15fc1f53638cbf5 (vm-server-app)

PublicIPs: 13.221.253.22 PrivateIPs: 172.31.18.233

# Cek apakah docker sudah jalan

```
sudo docker --version
```



```
ubuntu@ip-172-31-18-233:~$ sudo docker --version
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1
```

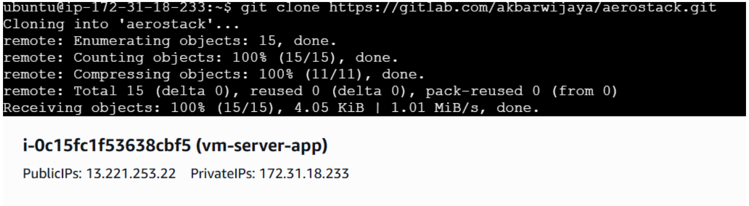
i-0c15fc1f53638cbf5 (vm-server-app)

PublicIPs: 13.221.253.22 PrivateIPs: 172.31.18.233

D. Deployment Aplikasi (AeroStack)

# Clone repo gitlab dibawah ini

```
git clone https://gitlab.com/akbarwijaya/aerostack.git
```



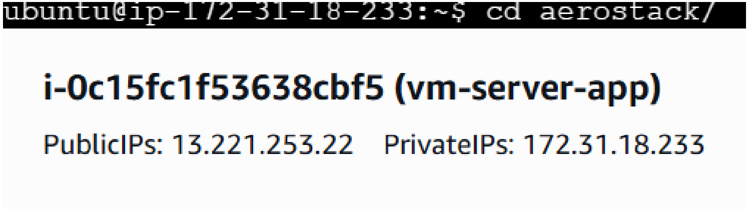
```
ubuntu@ip-172-31-18-233:~$ git clone https://gitlab.com/akbarwijaya/aerostack.git
Cloning into 'aerostack'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 15 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (15/15), 4.05 KiB | 1.01 MiB/s, done.
```

i-0c15fc1f53638cbf5 (vm-server-app)

PublicIPs: 13.221.253.22 PrivateIPs: 172.31.18.233

# Masuk ke folder repo

```
cd aerostack
```



```
ubuntu@ip-172-31-18-233:~$ cd aerostack/
```

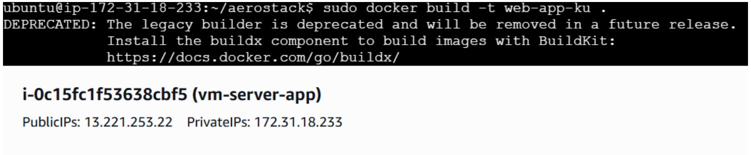
i-0c15fc1f53638cbf5 (vm-server-app)

PublicIPs: 13.221.253.22 PrivateIPs: 172.31.18.233

# Build Docker Image (Asumsi ada Dockerfile di dalamnya)

# Tanda titik (.) berarti build dari direktori saat ini

```
sudo docker build -t web-app-ku .
```



```
ubuntu@ip-172-31-18-233:~/aerostack$ sudo docker build -t web-app-ku .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/
```

i-0c15fc1f53638cbf5 (vm-server-app)

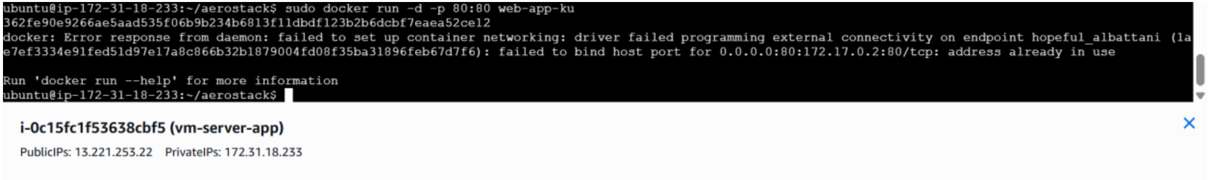
PublicIPs: 13.221.253.22 PrivateIPs: 172.31.18.233

# Jalankan Container

# -d: detach (jalan di background)

# -p 80:XXX : Mapping port 80 di VM ke port aplikasi di container

```
sudo docker run -d -p 80:80 web-app-ku
```



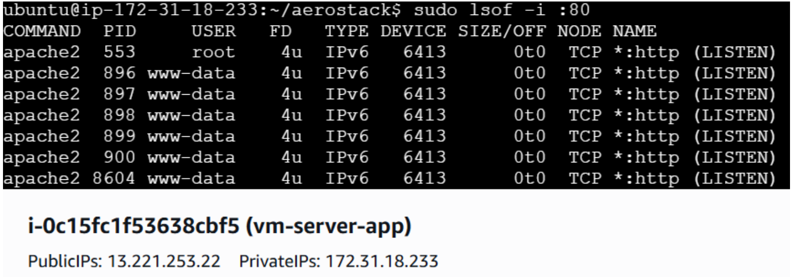
```
ubuntu@ip-172-31-18-233:~/aerostack$ sudo docker run -d -p 80:80 web-app-ku
362fe90e9266ae5aad535f06b9b234b6813f11dbdf123b2b6dcbf7eaea52ce12
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint hopeful_albattani (1a
e7ef3334e91fed51d97e17a8c866b32b1879004fd08f35ba31896feb67d7f6): failed to bind host port for 0.0.0.0:80:172.17.0.2:80/tcp: address already in use

Run 'docker run --help' for more information
ubuntu@ip-172-31-18-233:~/aerostack$
```

i-0c15fc1f53638cbf5 (vm-server-app)

PublicIPs: 13.221.253.22 PrivateIPs: 172.31.18.233

```
sudo lsof -i :80
```



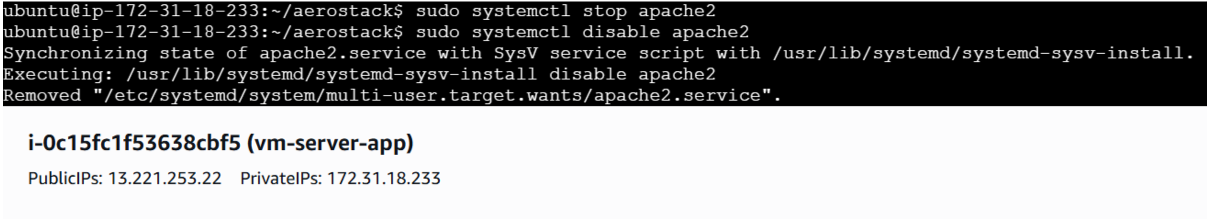
```
ubuntu@ip-172-31-18-233:~/aerostack$ sudo lsof -i :80
COMMAND  PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
apache2  553     root    4u  IPv6   6413      0t0  TCP *:http (LISTEN)
apache2  896 www-data    4u  IPv6   6413      0t0  TCP *:http (LISTEN)
apache2  897 www-data    4u  IPv6   6413      0t0  TCP *:http (LISTEN)
apache2  898 www-data    4u  IPv6   6413      0t0  TCP *:http (LISTEN)
apache2  899 www-data    4u  IPv6   6413      0t0  TCP *:http (LISTEN)
apache2  900 www-data    4u  IPv6   6413      0t0  TCP *:http (LISTEN)
apache2 8604 www-data    4u  IPv6   6413      0t0  TCP *:http (LISTEN)
```

i-0c15fc1f53638cbf5 (vm-server-app)

PublicIPs: 13.221.253.22 PrivateIPs: 172.31.18.233

jika apache2 jalankan

```
sudo systemctl stop apache2
sudo systemctl disable apache2
```



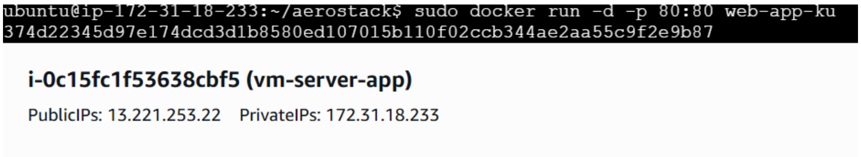
```
ubuntu@ip-172-31-18-233:~/aerostack$ sudo systemctl stop apache2
ubuntu@ip-172-31-18-233:~/aerostack$ sudo systemctl disable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install disable apache2
Removed "/etc/systemd/system/multi-user.target.wants/apache2.service".
```

i-0c15fc1f53638cbf5 (vm-server-app)

PublicIPs: 13.221.253.22 PrivateIPs: 172.31.18.233

#jalani ulang

```
sudo docker run -d -p 80:80 web-app-ku
```



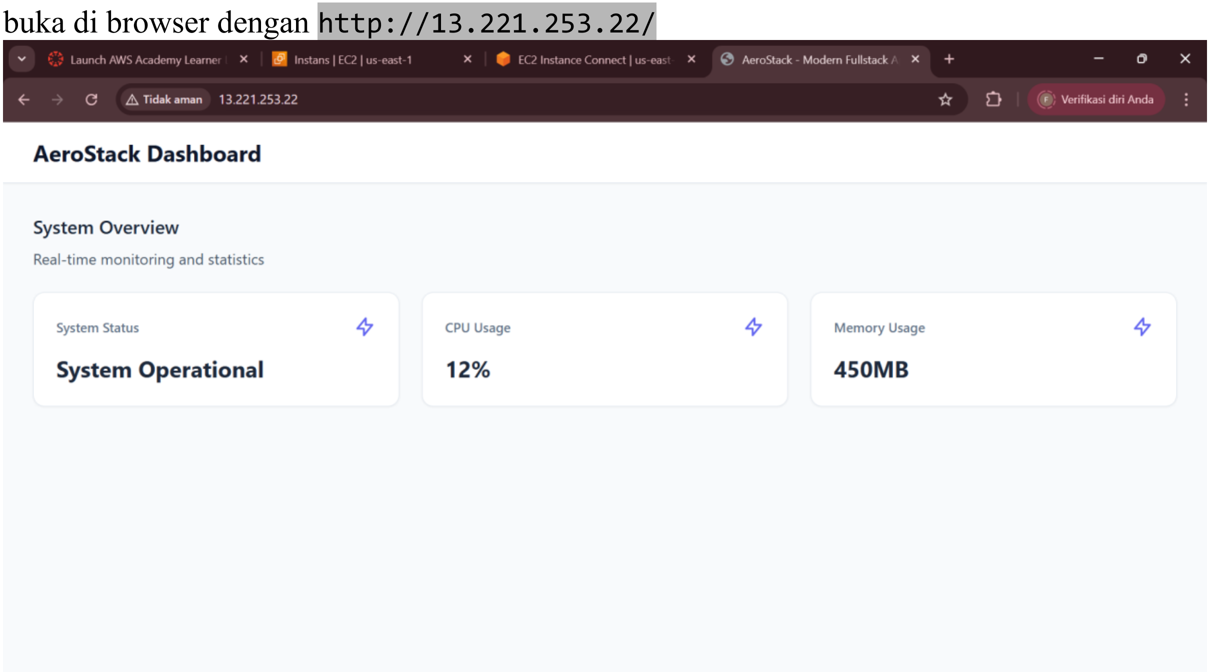
```
ubuntu@ip-172-31-18-233:~/aerostack$ sudo docker run -d -p 80:80 web-app-ku
374d22345d97e174dcd3d1b8580ed107015b110f02ccb344ae2aa55c9f2e9b87
```

i-0c15fc1f53638cbf5 (vm-server-app)

PublicIPs: 13.221.253.22   PrivateIPs: 172.31.18.233

E. Pengujian via Browser

buka di browser dengan `http://13.221.253.22/`



Launch AWS Academy Learner
Instans | EC2 | us-east-1
EC2 Instance Connect | us-east
AeroStack - Modern Fullstack A
Tidak aman
13.221.253.22
Verifikasi diri Anda

AeroStack Dashboard

System Overview

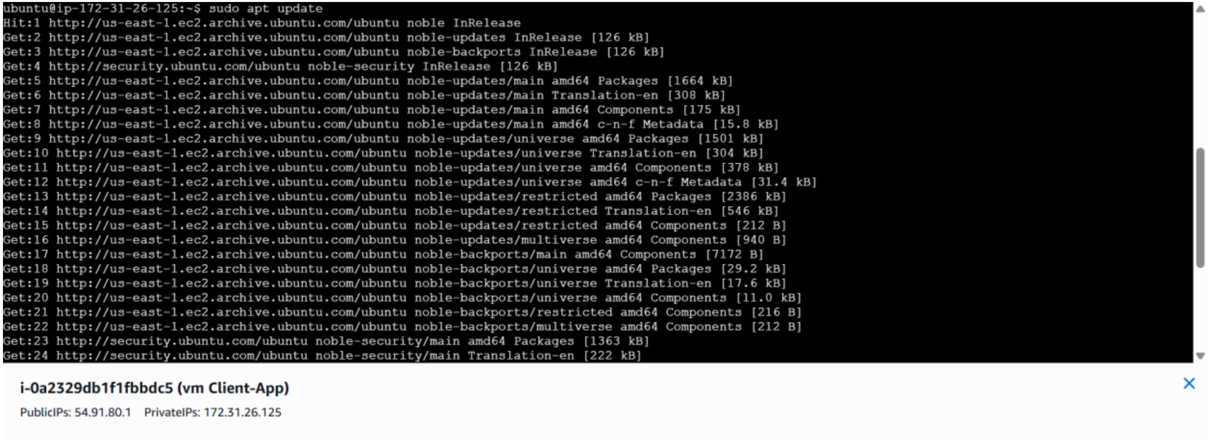
Real-time monitoring and statistics

System Status	CPU Usage	Memory Usage
System Operational	12%	450MB

2. Konfigurasi VM 2 (Client Testing)

A. Instalasi Tools Testing

`sudo apt update`

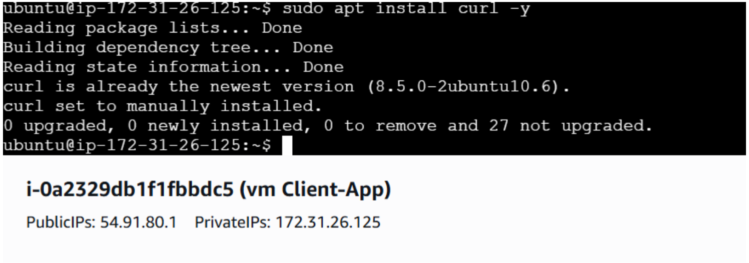


```
ubuntu@ip-172-31-26-125:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1664 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [308 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.8 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1501 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [304 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [378 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.4 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2386 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [546 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7172 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [29.2 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [17.6 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11.0 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:23 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1363 kB]
Get:24 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [222 kB]
```

i-0a2329db1f1fbbdc5 (vm Client-App)

PublicIPs: 54.91.80.1   PrivateIPs: 172.31.26.125

`sudo apt install curl -y`



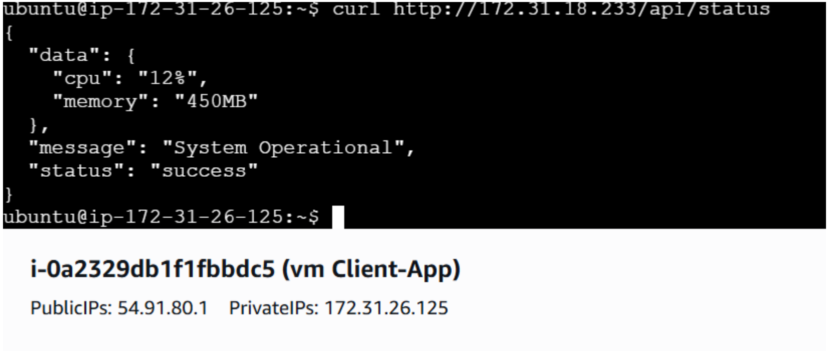
```
ubuntu@ip-172-31-26-125:~$ sudo apt install curl -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.5.0-2ubuntu10.6).
curl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
ubuntu@ip-172-31-26-125:~$
```

i-0a2329db1f1fbbdc5 (vm Client-App)

PublicIPs: 54.91.80.1   PrivateIPs: 172.31.26.125

B. Pengujian Koneksi (API Request)

```
curl http://172.31.18.233/api/status
```



```
ubuntu@ip-172-31-26-125:~$ curl http://172.31.18.233/api/status
{
  "data": {
    "cpu": "12%",
    "memory": "450MB"
  },
  "message": "System Operational",
  "status": "success"
}
ubuntu@ip-172-31-26-125:~$
```

i-0a2329db1f1fbbdc5 (vm Client-App)

PublicIPs: 54.91.80.1   PrivateIPs: 172.31.26.125

KESIMPULAN

Dari praktikum simulasi arsitektur client-server menggunakan Docker di AWS EC2, saya dapat menyimpulkan bahwa Docker sangat membantu saya dalam membuat lingkungan server dan client yang terisolasi, konsisten, dan mudah dijalankan tanpa pengaturan sistem yang rumit. Dengan menjalankan container di instance EC2, saya bisa memahami bagaimana sebuah server menerima request dari client melalui jaringan cloud, cara kerja port mapping, serta bagaimana EC2 mengatur akses melalui security group. Praktikum ini membuat saya mampu melihat langsung bagaimana layanan client-server berjalan, bagaimana komunikasi terjadi, dan bagaimana deployment berbasis container membuat proses pengujian serta pemeliharaan menjadi jauh lebih sederhana dan efisien bagi saya.