

# UTS CLOUD COMPUTING

Nama : Muhammad Fadlansyah Zikri Akhiruddin Rawosi

NRP : 152021078

Link Repository Github : <https://github.com/MFadlansy/webiste-uts-cc.git>

## Infrastruktur Dasar dengan Amazon EC2 & VPC

Membuat VPC terlebih dahulu

Tentukan blok CIDR IPv4 10.0.0.0/16.

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
fadian	vpc-0e3281a4826038a1a	Available	Off	10.0.0.0/16	-
vpc-toko	vpc-065467301106e530e	Available	Off	10.0.0.0/16	-
-	vpc-034fea5d750272b74	Available	Off	172.31.0.0/16	-
uts	vpc-08346ea6923f29558	Available	Off	10.0.0.0/16	-

**vpc-08346ea6923f29558 / uts**

**Details**

VPC ID	State	Block Public Access	DNS hostnames
vpc-08346ea6923f29558	Available	Off	Disabled
DNS resolution	Tenancy	DHCP option set	Main route table
Enabled	default	dopt-04c7c2d2d4dd5d08a	rtb-02b7b3cdcbff5580d

Selanjutnya membuat dua buah subnet (1 public, 1 private)

**VPC dashboard**

EC2 Global View

Filter by VPC

**Virtual private cloud**

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections

Pilih menu **Subnets** yang ada di side bar kiri aws



Pada pojok kanan atas tekan button **Create subnet**

Untuk public subnet CIDRnya misal 10.0.1.0/24

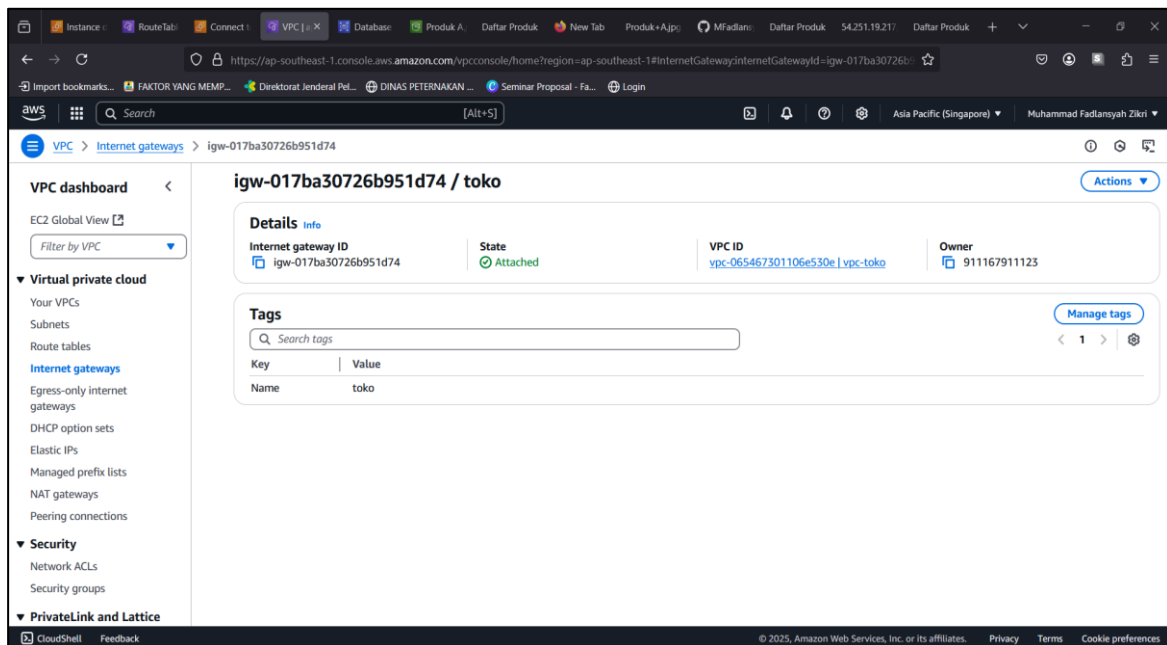


Untuk private subnet CIDRnya misal 10.0.2.0/24

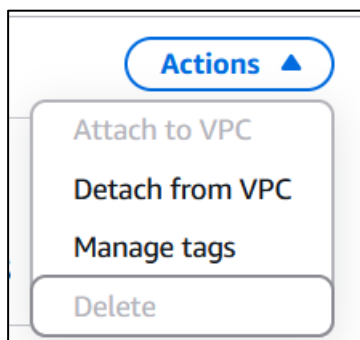


Dengan mengizinkan akses **ssh (port 22)** dan juga **http (port 80)**

Selanjutnya membuat internet gateway dan menghubungkan dengan vpc

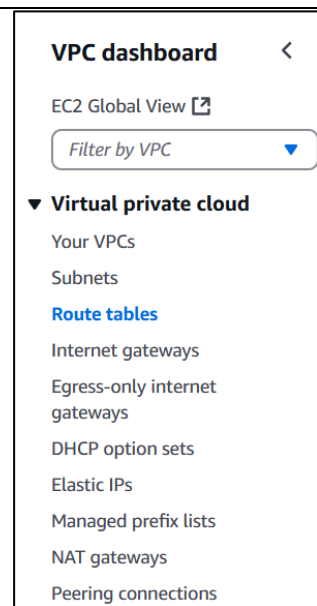


Apabila sudah dibuat, pada bagian pojok kanan atas terdapat button actions



Apabila belum di attach maka tekan **Attach to VPC** dan pilih VPC yang sudah dibuat sebelumnya

Selanjutnya membuat route table

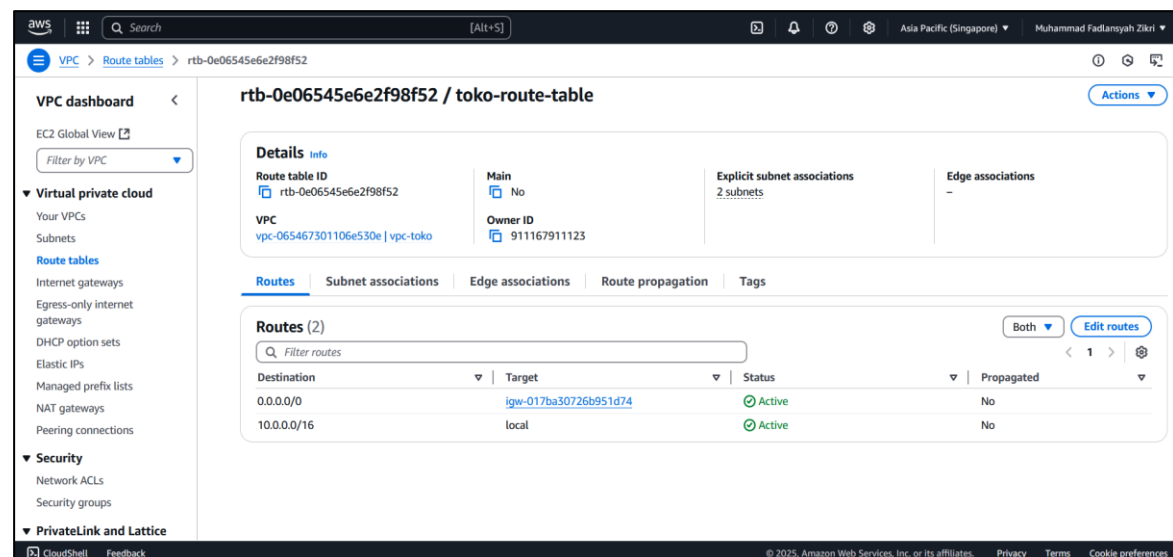


Pada sidebar di sebelah kiri pilih menu route table

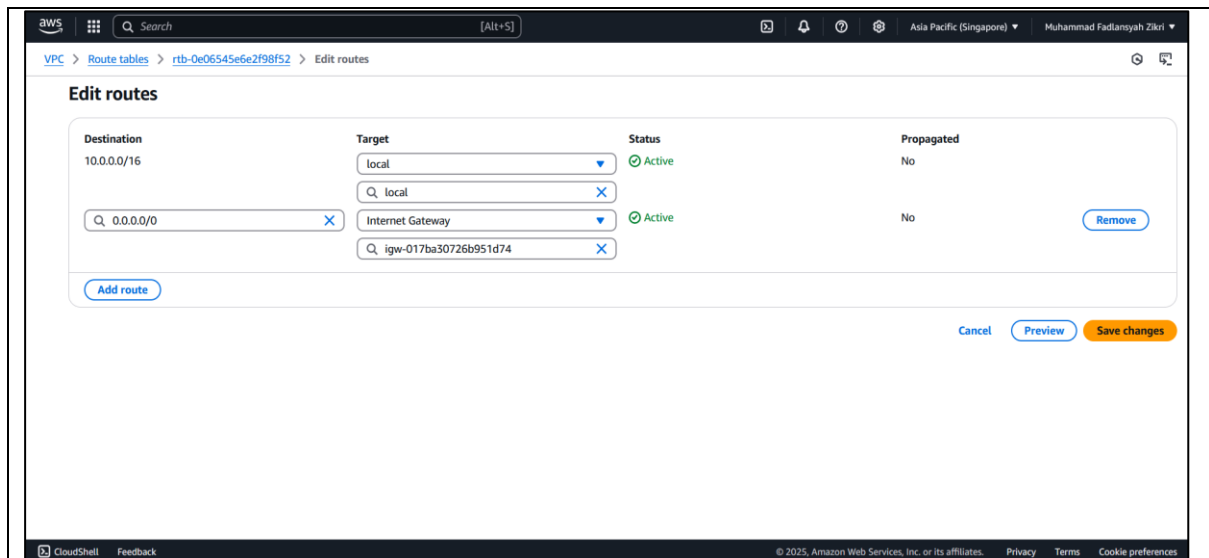


Pada pojok kanan atas tekan button **Create route table**

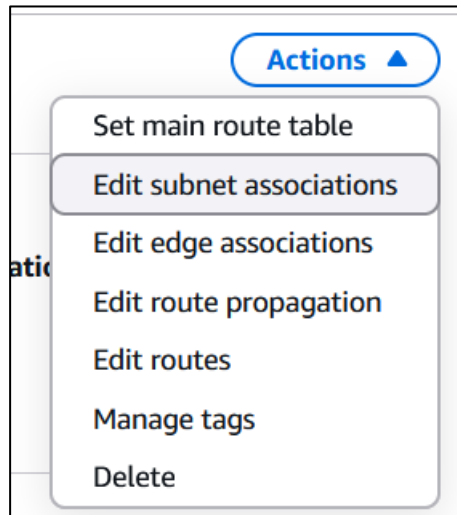
Pada menu VPC pilih VPC yang sudah dibuat sebelumnya



Selanjutnya pilih route table yang sudah dibuat, klik edit routes



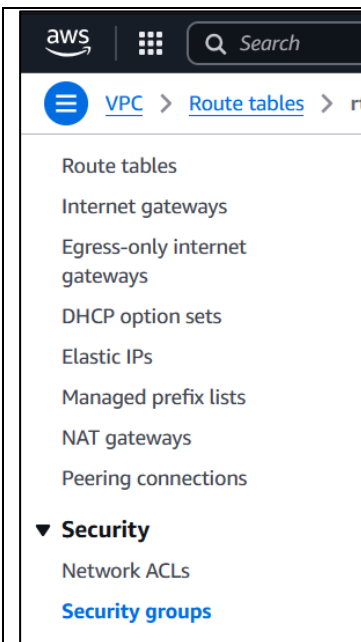
Pada bagian ini tambahkan route dengan destinasi **0.0.0.0/0** dengan target internet gateway yang sudah dibuat sebelumnya



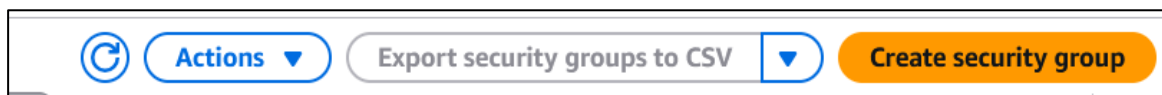
Selanjutnya pada bagian action route table nya pilih edit **subnet associations**

Tambahkan subnet public yang sudah dibuat sebelumnya lalu **save**

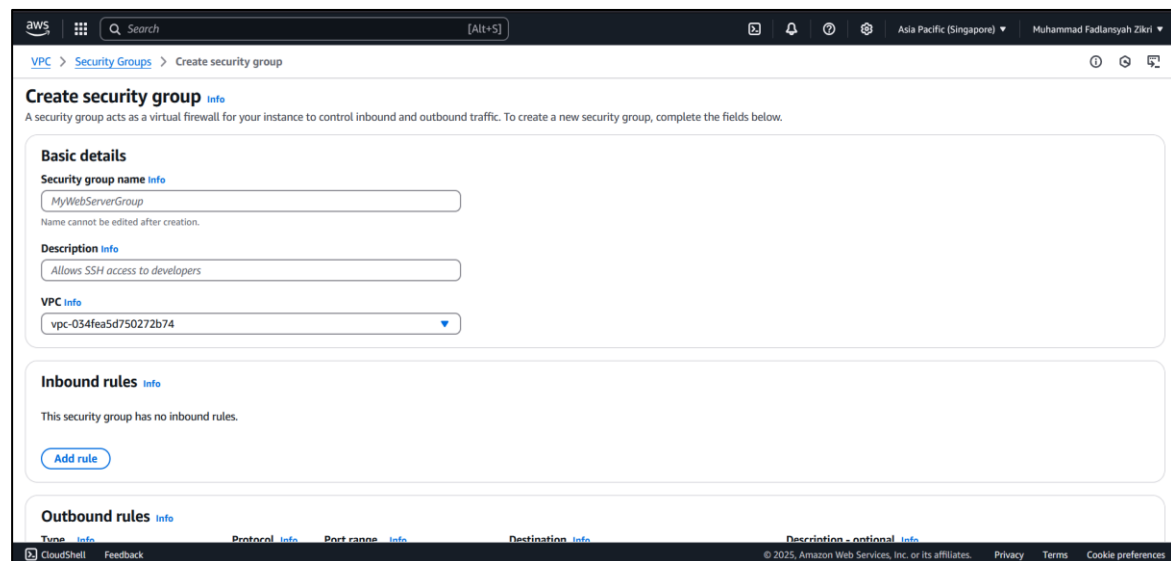
Lanjut membuat security grup



Pada sidebar di kiri, pilih menu **Security groups**



Selanjutnya pilih **Create security group** di pojok kanan atas



Buat dua security group untuk **Backend** dan juga **Frontend**

Masukkan nama security group

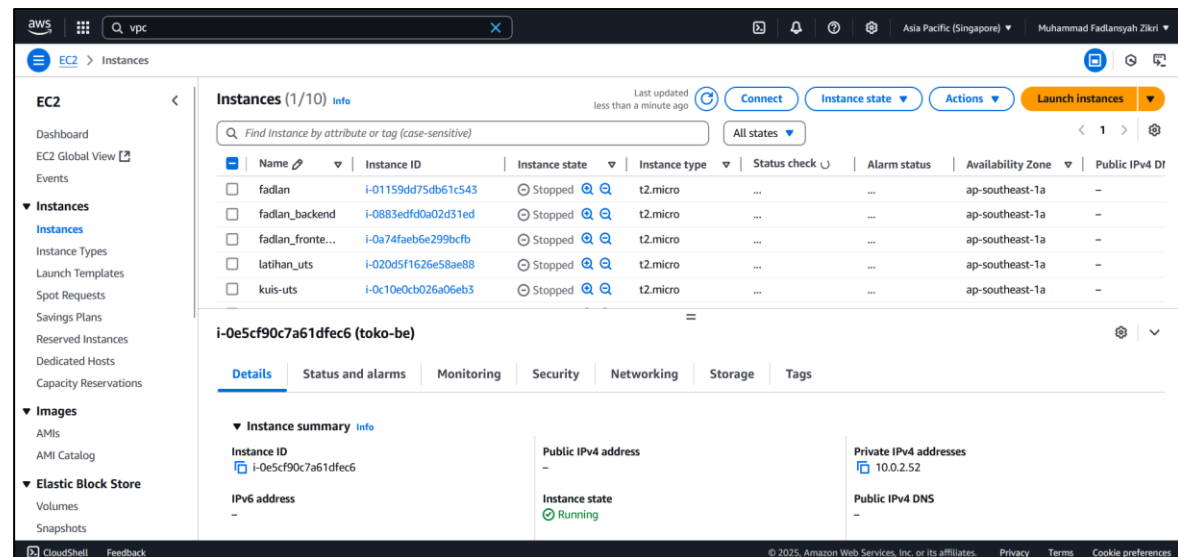
Masukkan deskripsi

Pilih VPC yang sudah dibuat sebelumnya

Tambahkan Inbound Rules **SSH (22)** dengan source **MyIP** dan **HTTP (80)** dengan source **Anywhere (0.0.0.0)**

Untuk security group Backend, SSH (22) pakai IP security group Frontend, karena nantinya kita akan melakukan ssh backend lewat EC2 Frontend

Selanjutnya yaitu membuat instance EC2



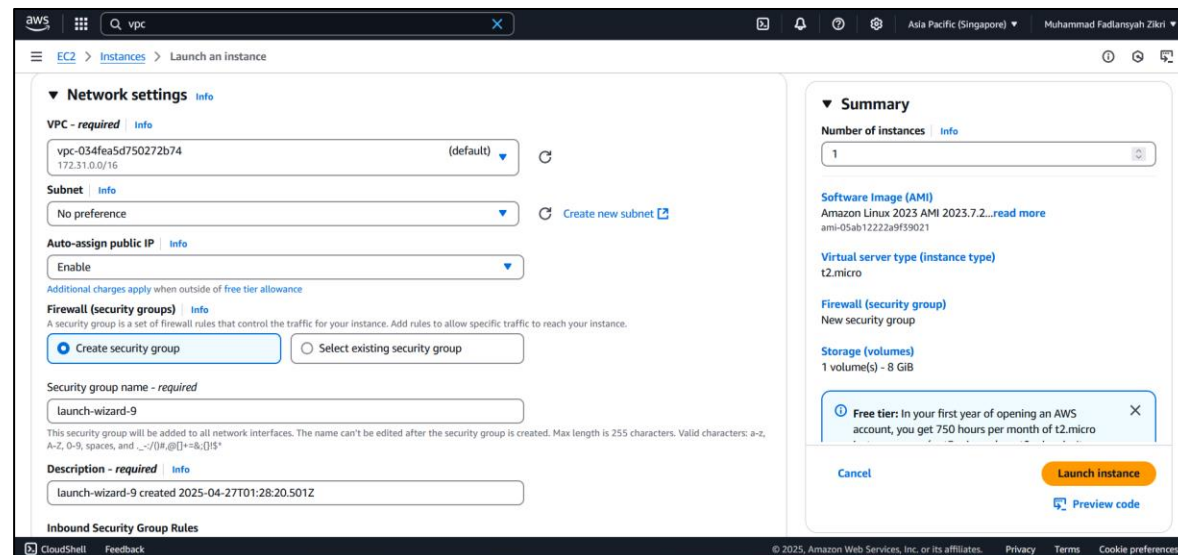
Disini kita akan membuat dua buah instance EC2 yaitu Backend dan juga Frontend



Pada pojok kanan atas tekan button Launch Instance

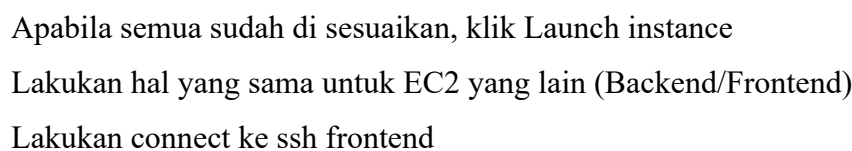
Masukkan nama instance

Buat key pair



Pada bagian ini, sesuaikan dengan VPC yang sudah dibuat

Subnet dan Security groupnya juga di perhatikan, untuk private ataupun public

[illegible]

```
sudo yum update -y
```

```
[ec2-user@ip-10-0-1-36 ~]$ sudo yum update -y
Last metadata expiration check: 11:59:37 ago on Sat Apr 26 13:38:17 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-1-36 ~]$ |
```

```
scp -i "key.pem" key.pem ec2-user@ip4 public:/home/ec2-user/
```

```
[ec2-user@ip-10-0-1-36 ~]$ ls
toko-be.pem
```

```
chmod 400 key.pem
```

Selanjutnya lakukan ssh ke EC2 Backendnya

```
[ec2-user@ip-10-0-1-36 ~]$ ssh -i "toko-be.pem" ec2-user@10.0.2.52
#_
~\  #####_      Amazon Linux 2023
~~\  #####\
~~\  ###|
~~\  \#/  _--_  https://aws.amazon.com/linux/amazon-linux-2023
~~\  V~'  '--->
   ~~~
   ~\  _
   ~\ / \
   ~/_m/'

Last login: Sat Apr 26 22:47:02 2025 from 10.0.1.36
[ec2-user@ip-10-0-2-52 ~]$
```

EC2 Frontend dan Backend berhasil untuk diakses

## Penyimpanan Data dengan (Amazon S3 & Database)

The screenshot shows the AWS Management Console for the 'Buckets' section of Amazon S3. The left sidebar contains navigation links for 'General purpose buckets', 'Directory buckets', 'Access Grants', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', and 'IAM Access Analyzer for S3'. The main content area shows a list of 'General purpose buckets (5)' in the 'Asia Pacific (Singapore) ap-southeast-1' region. The buckets listed are:

Name	AWS Region	IAM Access Analyzer	Creation date
<a href="#">bucketfadlan</a>	Asia Pacific (Singapore) ap-southeast-1	<a href="#">View analyzer for ap-southeast-1</a>	March 14, 2025, 07:35:55 (UTC+07:00)
<a href="#">cc-uts</a>	Asia Pacific (Singapore) ap-southeast-1	<a href="#">View analyzer for ap-southeast-1</a>	April 25, 2025, 19:43:18 (UTC+07:00)
<a href="#">kuis</a>	Asia Pacific (Singapore) ap-southeast-1	<a href="#">View analyzer for ap-southeast-1</a>	April 19, 2025, 13:04:52 (UTC+07:00)
<a href="#">kuis-uts</a>	Asia Pacific (Singapore) ap-southeast-1	<a href="#">View analyzer for ap-southeast-1</a>	April 19, 2025, 10:15:21 (UTC+07:00)
<a href="#">latihann</a>	Asia Pacific (Singapore) ap-southeast-1	<a href="#">View analyzer for ap-southeast-1</a>	April 18, 2025, 07:55:03 (UTC+07:00)

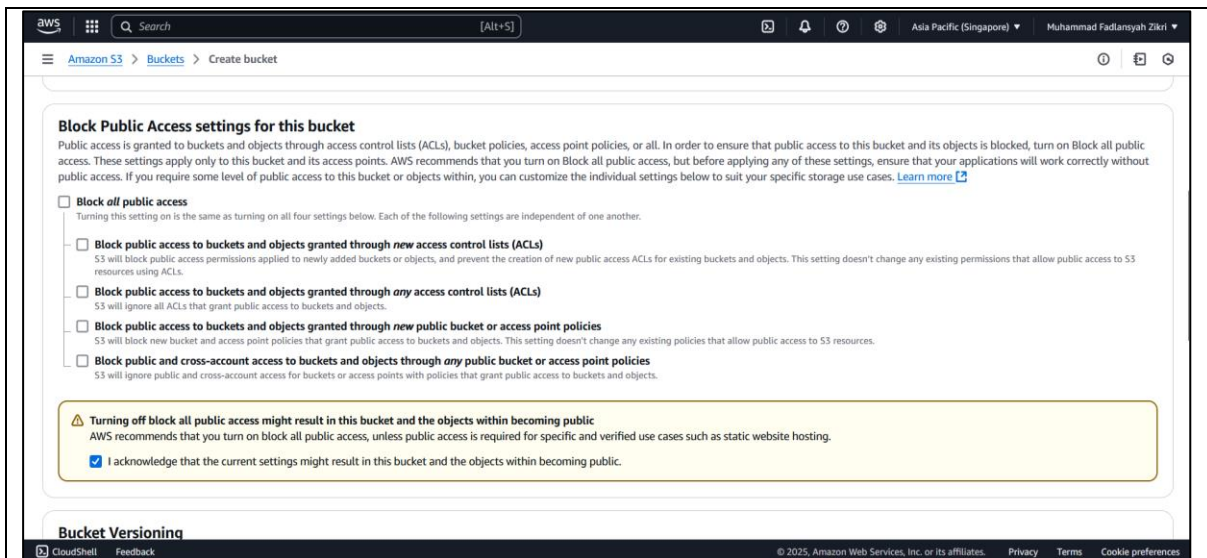
At the top of the console, there is a banner for 'Account snapshot - updated every 24 hours' and a 'View Storage Lens dashboard' button. Below the bucket list, there are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'.

Masuk ke bucket S3

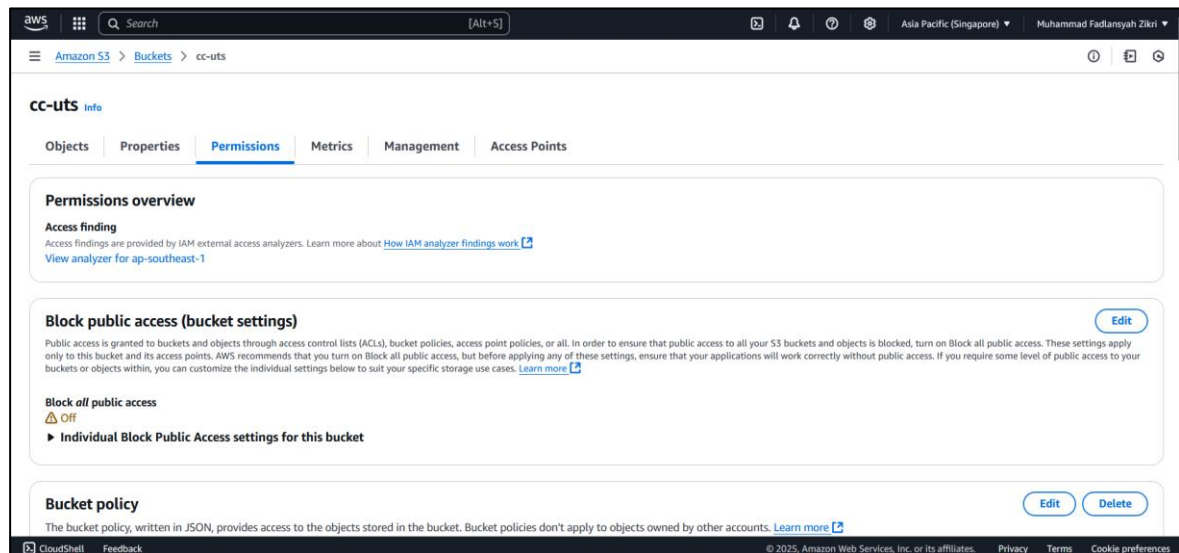
This block shows a set of controls for an S3 bucket. It includes a circular refresh icon, a 'Copy ARN' button, an 'Empty' button, a 'Delete' button, and a large orange 'Create bucket' button.

Tekan button **Create bucket**



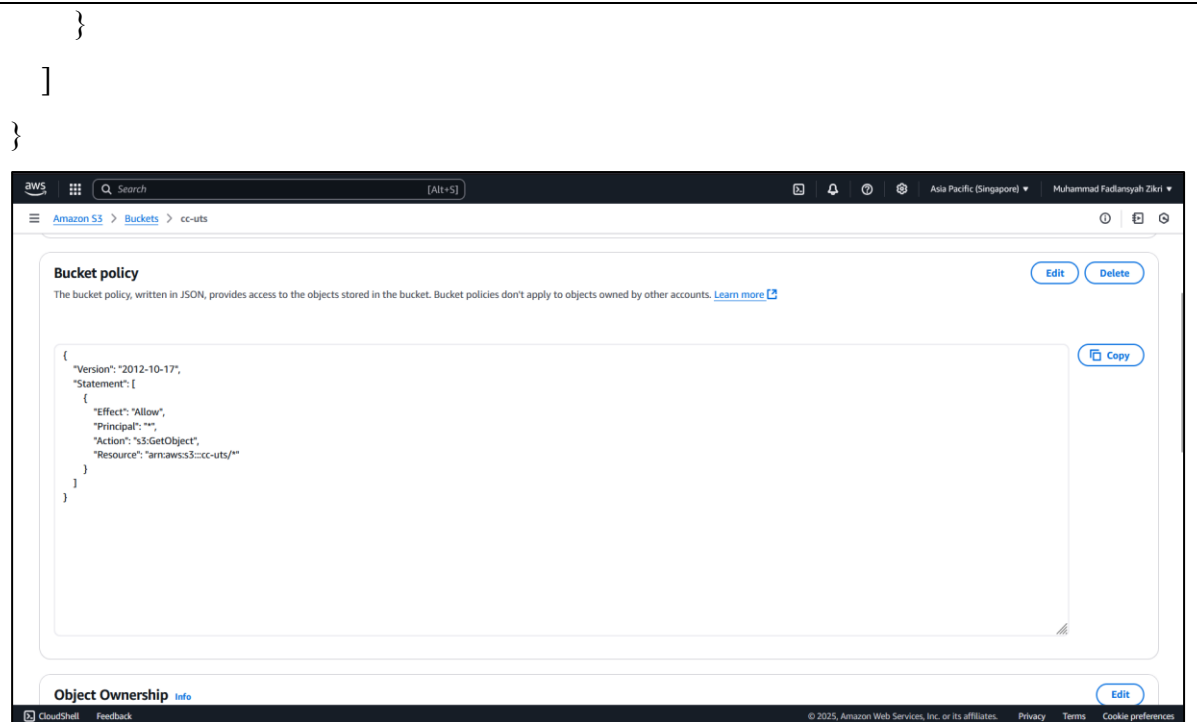


Pada bagian ini uncheck ***Block all public access*** dan jangan lupa di centang checkbox di bawahnya

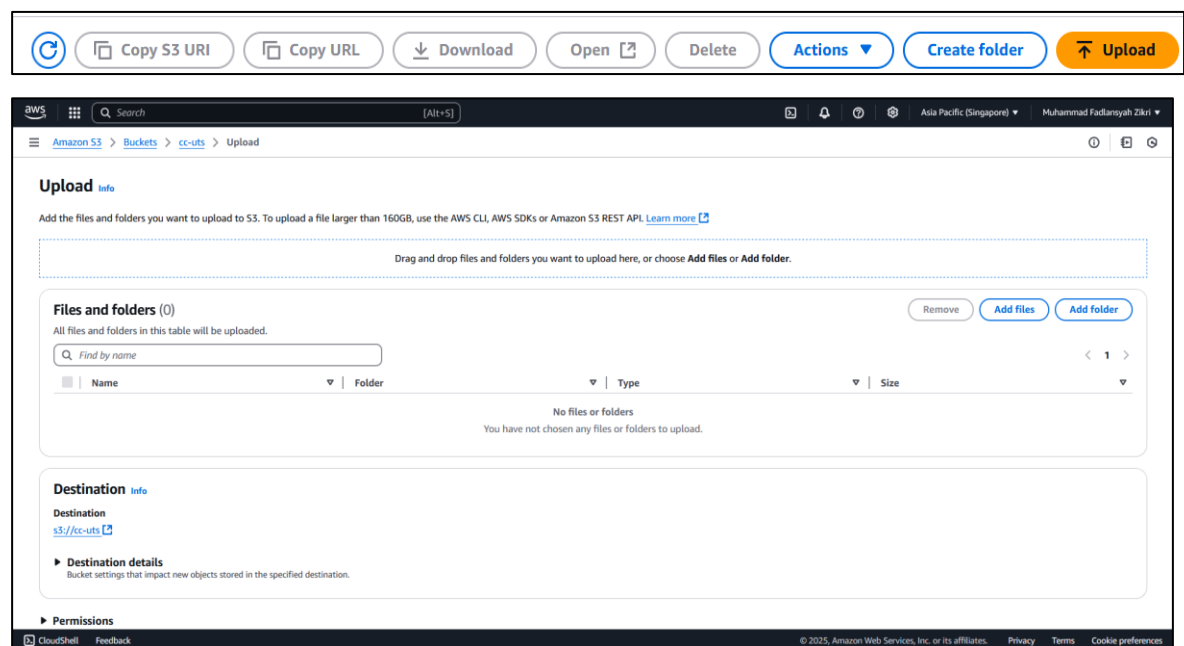


Tekan bucket s3 yang sudah dibuat sebelumnya, pada tab permissions tambahkan kode ini pada **Bucket policy**

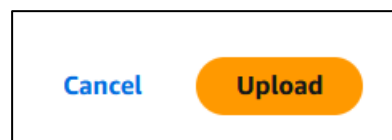
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::cc-uts/*"
    }
  ]
}
```



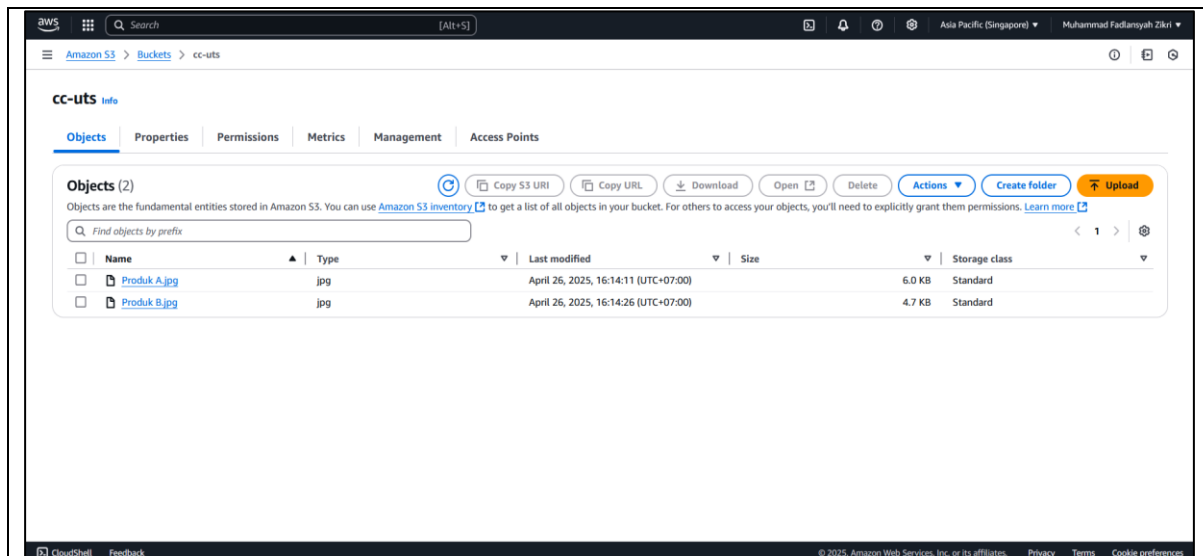
Coba upload



Pada bagian ini pilih add files

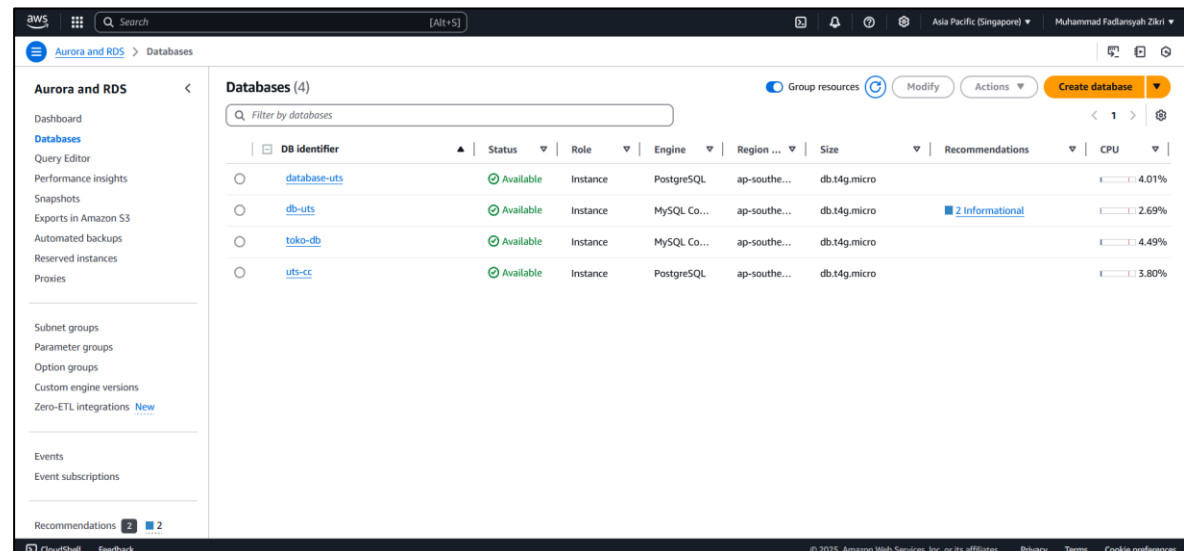


Apabila file yang mau di upload sudah dipilih, selanjutnya di pojok kanan bawah tekan button upload



Kalau berhasil di upload, file tersebut akan muncul pada halaman ini

Apabila semuanya sudah berhasil, selanjutnya kita akan membuat **RDS MySQL**



Di pojok kanan atas tekan button **Create database**


Pada bagian **engine options** pilih **MySQL**

Masukkan username dan password

Pakai VPC yang sudah dibuat sebelumnya

Gunakan security group yang sudah dibuat juga

Kalua sudah buka database client pada vscode



Connect to server

Name

Connection Name

Group

Parent/Sub

Advanced

Scope

Global

Premium Only

Note: Requires a Premium License to unlock all features. Database 1/3 , Other 0/3

Server Type

MySQL

MariaDB

PostgreSQL

SQLite

SQL Server

Db2

Oracle

Kingbase

ClickHouse

JDBC

SSH

Docker

Redis

ElasticSearch

MongoDB

S3

FTP

Kafka

RabbitMQ

Cassandra

More

Config

Main

SSH Tunnel

Socks Proxy

HTTP Proxy

Host

127.0.0.1

Port

3306

Username

root

Password

Password

Database

Database

Advanced

Socket Path

e.g. /var/run/mysqld/mysqld.sock

Use

Features

Event

Trigger

Use Connection String

SSL

Compatible Mode


Save

Connect

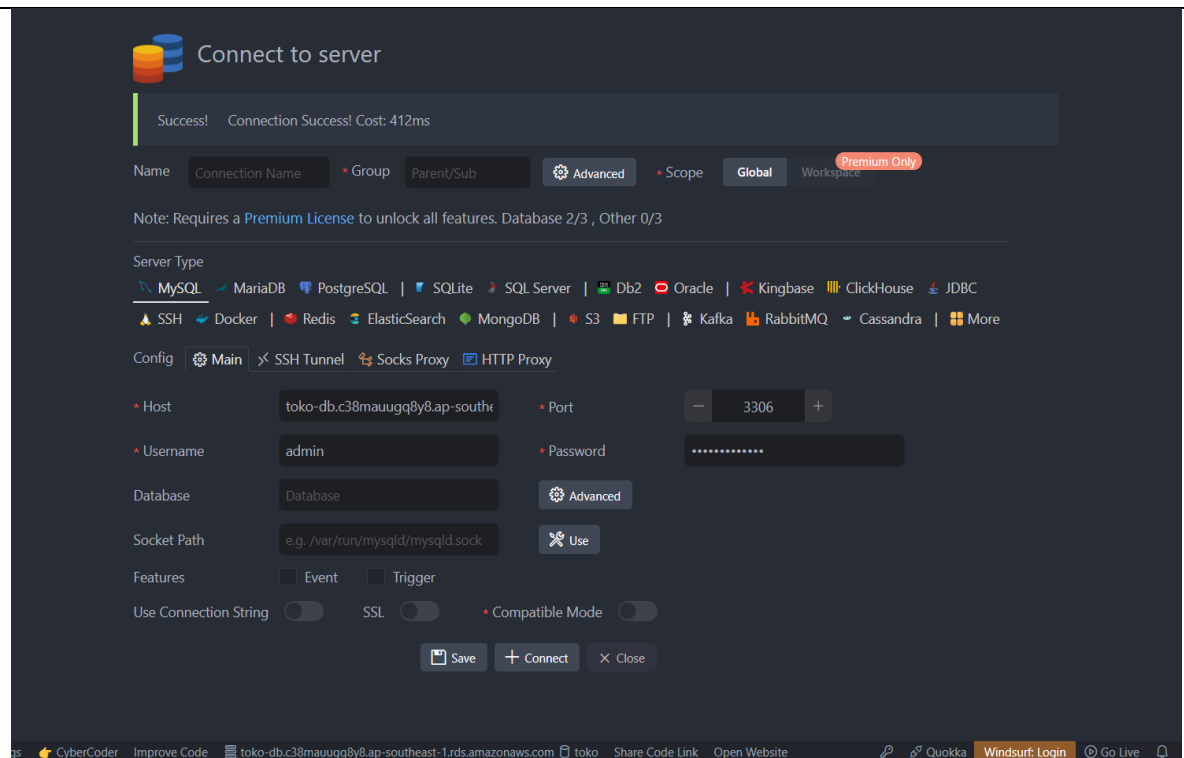
Close

Sesuaikan isian dengan yang sudah di buat sebelumnya, pada bagian host masukkan endpoint dari RDS yang sudah dibuat tadi

#### Endpoint

 toko-db.c38mauugq8y8.ap-southeast-1.rds.amazonaws.com

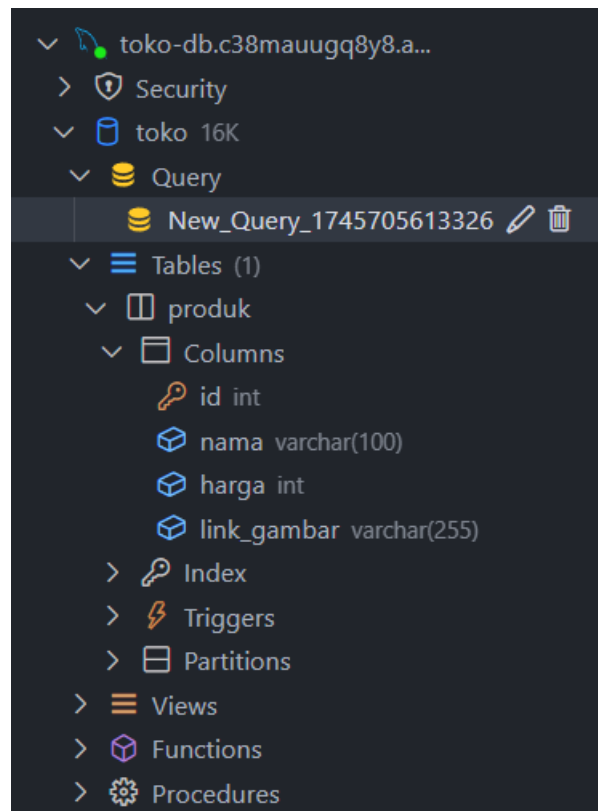
Apabila sudah tekan connect



Apabila berhasil maka akan muncul notifikasi seperti berikut

Selanjutnya buat database contohnya disini yaitu toko

Buat table dengan nama produk dengan isian nama, harga dan link\_gambar



Apabila sudah, tambahkan isiannya

Search Results					
		* id int	nama varchar(100)	harga int	link_gambar varchar(255)
<input type="checkbox"/>	>	1	Produk A	50000	https://cc-uts.s3.ap-southea
<input type="checkbox"/>	>	2	Produk B	100000	https://cc-uts.s3.ap-southea

Pada bagian link\_gambar masukan link\_gambar dari gambar yang ada di aws

#### Object URL

<https://cc-uts.s3.ap-southeast-1.amazonaws.com/Produk+A.jpg>

## Pengembangan aplikasi web

Setup EC2 Frontend nya

Jalankan perintah berikut untuk setup

**sudo yum install httpd -y**

**sudo systemctl enable httpd**

**sudo systemctl start httpd**

**sudo yum install git -y**

Siapkan repository untuk web yang sudah dibuat

Lakukan clone dari repository tersebut

**git clone https://github.com/username/repo.git**

```
[ec2-user@ip-10-0-1-36 ~]$ git clone https://github.com/MFadlansy/webiste-uts-cc.git
Cloning into 'webiste-uts-cc'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), done.
[ec2-user@ip-10-0-1-36 ~]$ ls
webiste-uts-cc
```

cd repo

```
[ec2-user@ip-10-0-1-36 webiste-uts-cc]$ ls
api index.html
```

Selanjutnya pindahkan file index.html ke dalam /var/www/html

```
[ec2-user@ip-10-0-1-36 webiste-uts-cc]$ sudo mv index.html /var/www/html/
[ec2-user@ip-10-0-1-36 webiste-uts-cc]$ ls /var/www/html
index.html
```

sudo chown ec2-user:ec2-user /var/www/html/index.html

Selanjutnya setup untuk Backend

```
[ec2-user@ip-10-0-1-36 ~]$ ssh -i "toko-be.pem" ec2-user@10.0.2.52

#_
~\  #####_      Amazon Linux 2023
~~ \#####\
~~  \###|
~~  \#/  --- https://aws.amazon.com/linux/amazon-linux-2023
~~   V~'  '--->
~~~
~~.  .
~~ /  \
~~ /m/ '

Last login: Sat Apr 26 15:29:53 2025 from 10.0.1.36
```

Akses ssh Backend lewat EC2 Frontend

Agar backend dapat terkoneksi dengan internet, buat NAT Gateway dengan subnet public

```
[ec2-user@ip-10-0-2-52 api]$ sudo yum install php php-cli php-fpm php-mysqlnd -y
Amazon Linux 2023 repository
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
53 MB/s | 36 MB 00:00
149 kB/s | 15 kB 00:00
```

Package	Architecture	Version	Repository	Size
<b>Installing:</b>				
php8.4	x86_64	8.4.5-1.amzn2023.0.1	amazonlinux	17 k
php8.4-fpm	x86_64	8.4.5-1.amzn2023.0.1	amazonlinux	2.0 M
php8.4-mysqlnd	x86_64	8.4.5-1.amzn2023.0.1	amazonlinux	157 k
<b>Installing dependencies:</b>				
apr	x86_64	1.7.5-1.amzn2023.0.4	amazonlinux	129 k
apr-util	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	98 k
generic-logos-httpd	noarch	18.0.0-12.amzn2023.0.3	amazonlinux	19 k
httpd-core	x86_64	2.4.62-1.amzn2023	amazonlinux	1.4 M
httpd-filesystem	noarch	2.4.62-1.amzn2023	amazonlinux	14 k
httpd-tools	x86_64	2.4.62-1.amzn2023	amazonlinux	81 k
libbrotli1	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	315 k
libsodium	x86_64	1.0.19-4.amzn2023	amazonlinux	176 k
libssl1	x86_64	1.1.43-1.amzn2023.0.1	amazonlinux	183 k
mailcap	noarch	2.1.49-3.amzn2023.0.3	amazonlinux	33 k
nginx-filesystem	noarch	1:1.26.3-1.amzn2023.0.1	amazonlinux	9.6 k
php8.4-cli	x86_64	8.4.5-1.amzn2023.0.1	amazonlinux	3.8 M
php8.4-common	x86_64	8.4.5-1.amzn2023.0.1	amazonlinux	797 k
php8.4-pdo	x86_64	8.4.5-1.amzn2023.0.1	amazonlinux	100 k
php8.4-process	x86_64	8.4.5-1.amzn2023.0.1	amazonlinux	53 k
php8.4-xml	x86_64	8.4.5-1.amzn2023.0.1	amazonlinux	972 k
<b>Installing weak dependencies:</b>				
apr-util-openssl	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	17 k
httpd	x86_64	2.4.62-1.amzn2023	amazonlinux	48 k
mod_http2	x86_64	2.0.27-1.amzn2023.0.3	amazonlinux	166 k
mod_lua	x86_64	2.4.62-1.amzn2023	amazonlinux	61 k
php8.4-mbstring	x86_64	8.4.5-1.amzn2023.0.1	amazonlinux	542 k

Karena disini Backend nya memakai php maka disini kita menginstall dependensi yang diperlukan terlebih dahulu

Tambahkan kode di bawah ini pada **sudo nano /etc/httpd/conf.d/php-fpm.conf**

```
GNU nano 8.3
<IfModule mod_proxy_fcgi.c>
    SetHandler proxy_fcgi://127.0.0.1:9000
</IfModule>
```

Ketik **sudo nano /etc/httpd/conf/httpd.conf** lalu

Tambahkan konfigurasi

<VirtualHost \*: 80>

DocumentRoot /var/www/html

ServerName 54.251.19.217

<Directory /var/www/html/api>

AllowOverride All

Require all granted

</Directory>

</VirtualHost>

```
GNU nano 8.3 /etc/httpd/conf/httpd.conf
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
#
# EnableMMAP and EnableSendfile: On systems that support it,
# memory-mapping or the sendfile syscall may be used to deliver
# files. This usually improves server performance, but must
# be turned off when serving from networked-mounted
# filesystems or if support for these functions is otherwise
# broken on your system.
# Defaults if commented: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
#EnableSendfile on
#
# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf

<VirtualHost *:80>
    DocumentRoot /var/www/html
    ServerName 54.251.19.217

    <Directory /var/www/html/api>
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>

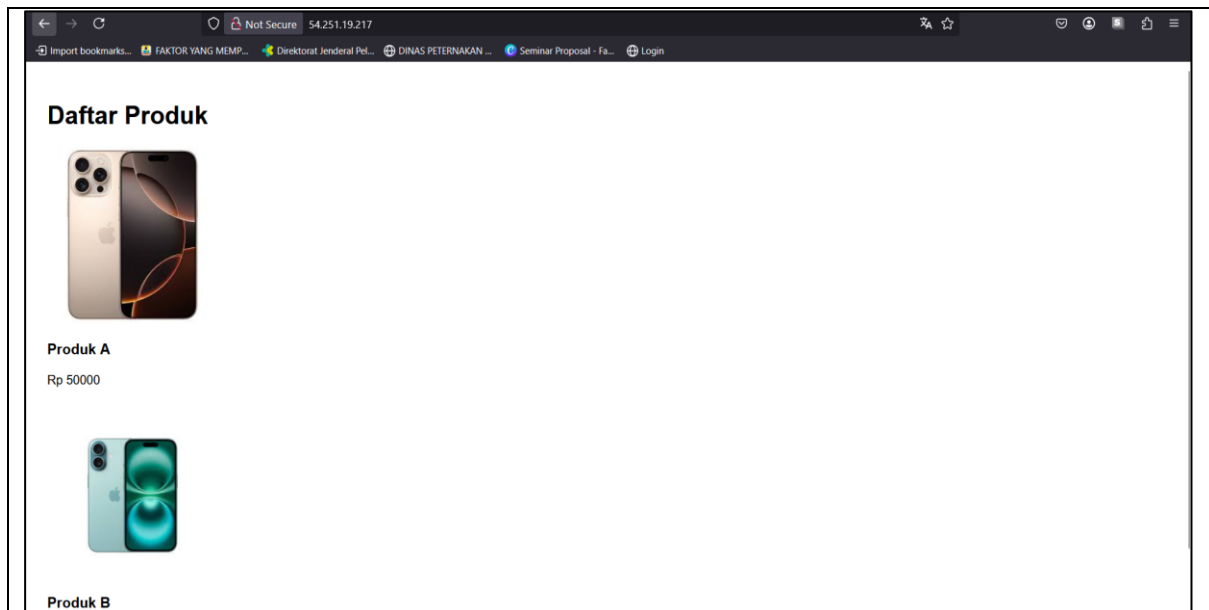
⌘ Help      ⌘ Write Out  ⌘ Where Is  ⌘ Cut       ⌘ Execute   ⌘ Location  ⌘ Undo     ⌘ Set Mark  ⌘ To Bracket
⌘ Exit      ⌘ Read File  ⌘ Replace   ⌘ Paste     ⌘ Justify   ⌘ Go To Line ⌘ Redo     ⌘ Copy      ⌘ Where Was
```

Pindahkan file index.php ke dalam **/var/www/html**

```
[ec2-user@ip-10-0-2-52 ~]$ sudo mv /var/www/html/api/index.php /var/www/html/
```

Setelah itu coba jalankan public ip dari EC2 Frontend

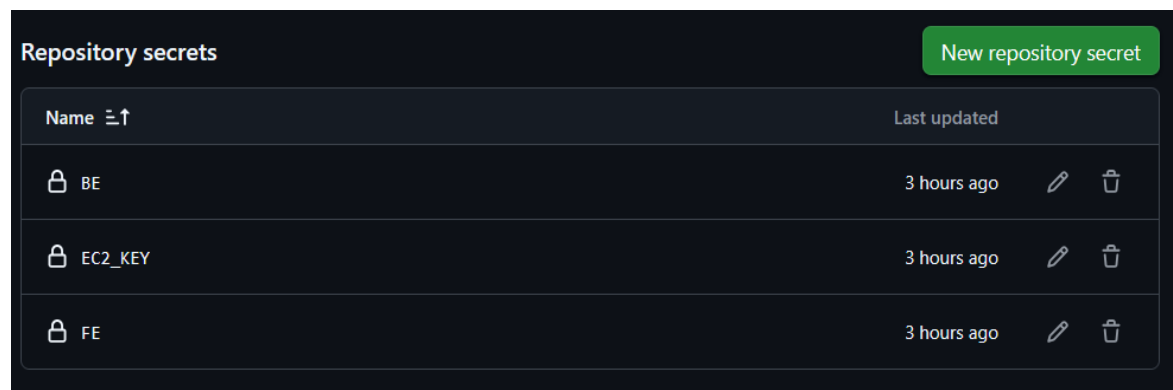




Apabila berhasil maka akan muncul seperti diatas, dimana data yang ditampilkan diambil dari RDS yang sudah dibuat sebelumnya

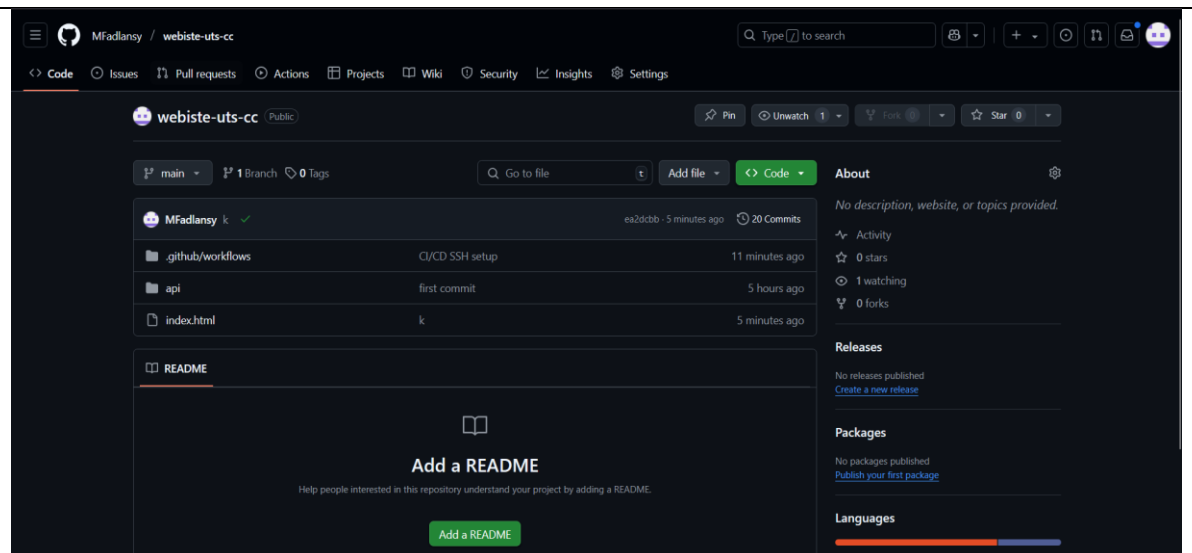
### Implementasi CI/CD dengan GitHub Actions

Menambahkan Secret pada Repository Github, yaitu **BE** yaitu IP private BE, **FE** yaitu IP public FE, dan **EC2\_KEY** yaitu key yang digunakan di instance.



Selanjutnya buat direktori `.github/workflows` pada repository lokal, yang di dalamnya buat file `ci-cd.yml` yang dimana nantinya dipakai untuk melakukan CI/CD setiap melakukan push ke github nya

Selanjutnya push file `ci-cd.yml` tersrbut ke github



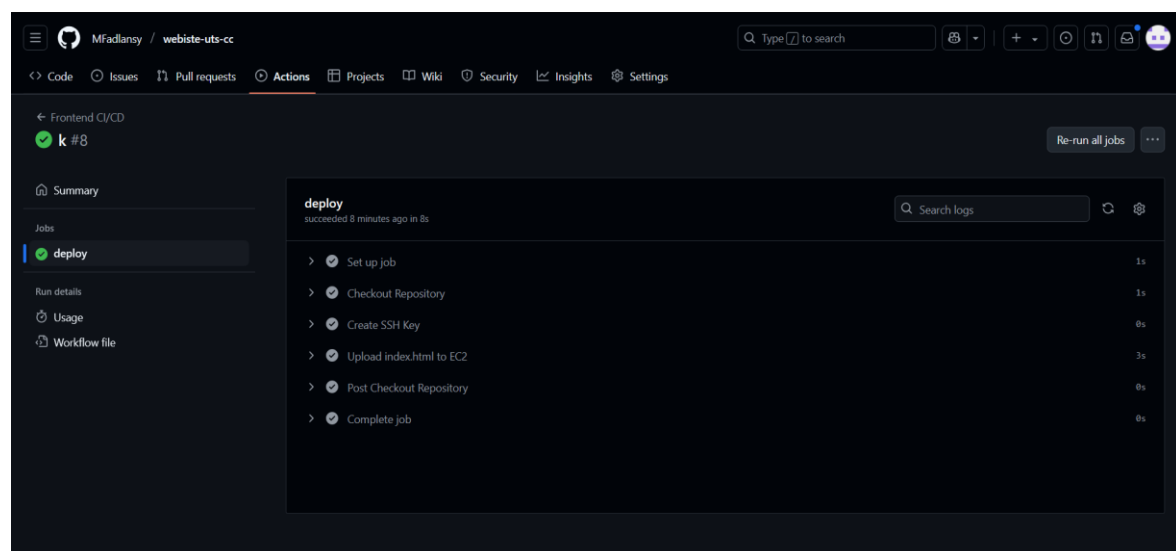
Setelah di push, kita coba melakukan perubahan pada website dan melihat apakah dapat berubah secara otomatis

```
<h1>Produk Di Jual</h1>
```

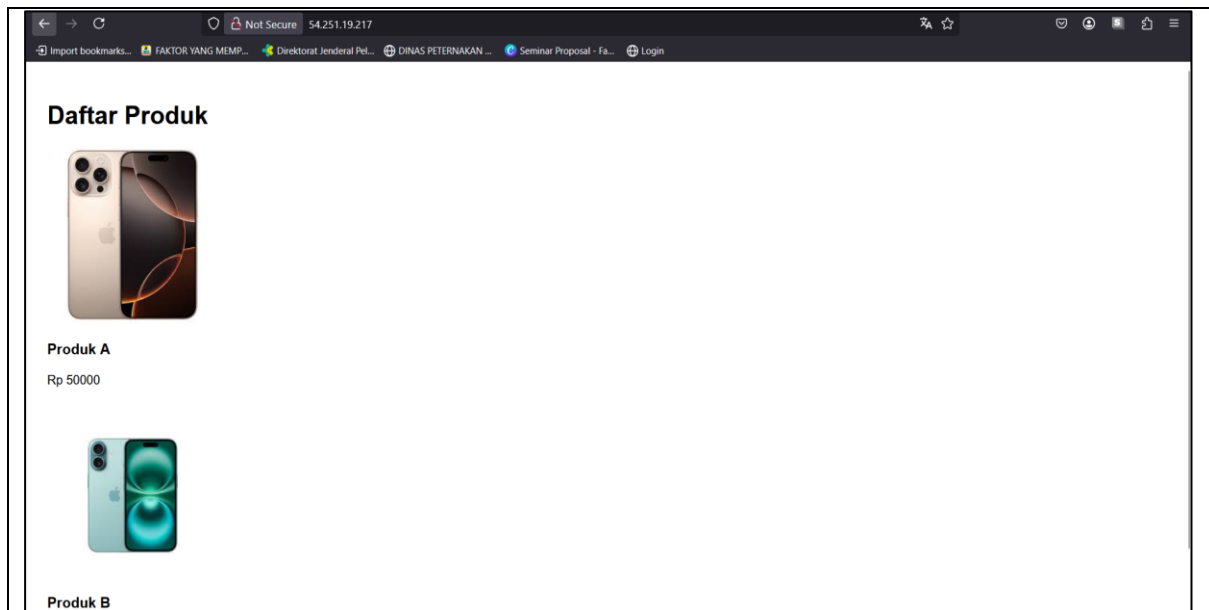
Kita coba ganti header pada websitenya

Lalu push ke github file index.html nya

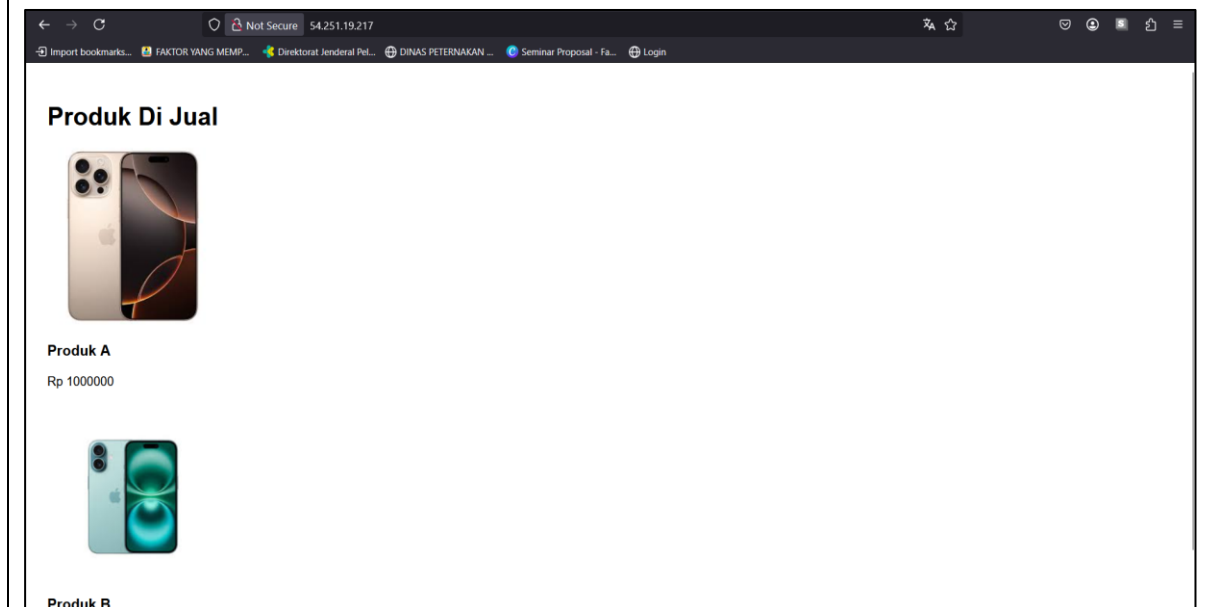
Cek pada tab actions untuk melihat proses deploy nya



Setelah itu kita coba lihat website nya apakah berubah atau tetap



Ini adalah tampilan sebelum di push ke github



Dan ini adalah tampilan setelah di push ke github

Terlihat bahwa header nya sudah berubah sesuai dengan yang diubah sebelumnya, yang awalnya “**Daftar Produk**” menjadi “**Produk Di Jual**” sesuai dengan yang diubah disini

```
<h1>Produk Di Jual</h1>
```