Nama : Ariq Bagus Sugiharto

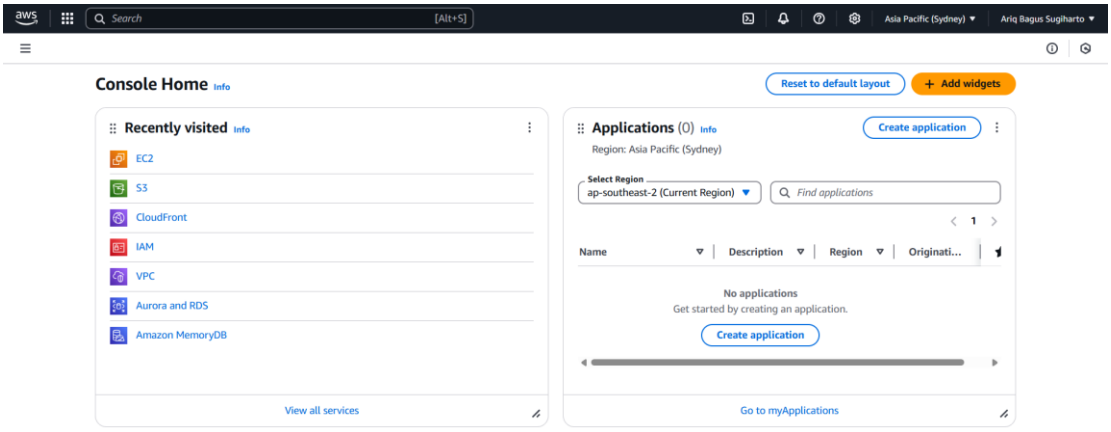
NRP : 15-2021-036

UTS IFB-452 Komputasi Awan

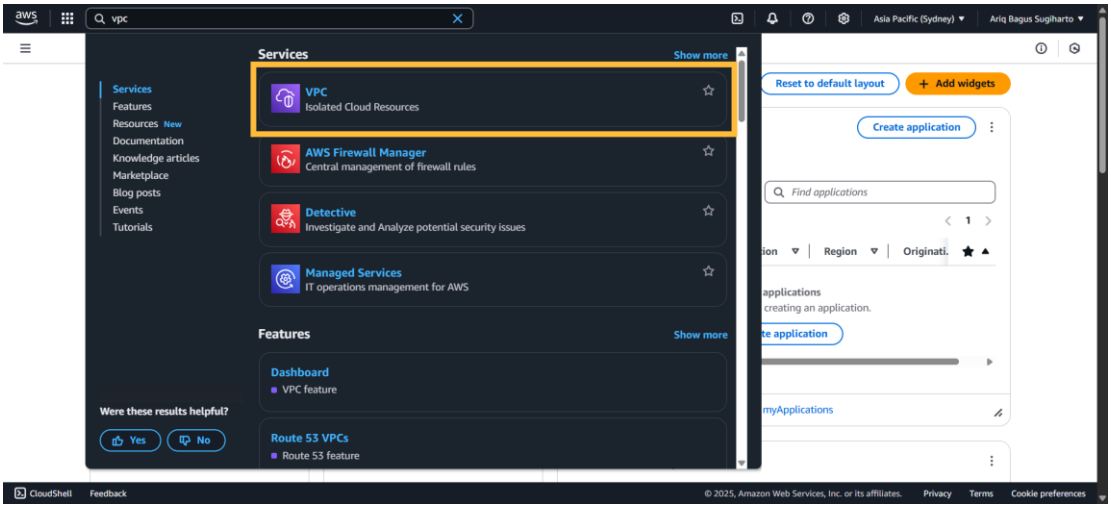
Link repository github: https://github.com/AbbysGud/ecommerce-cloud.git

Bagian 1 - Infrastruktur Dasar dengan Amazon EC2 & VPC

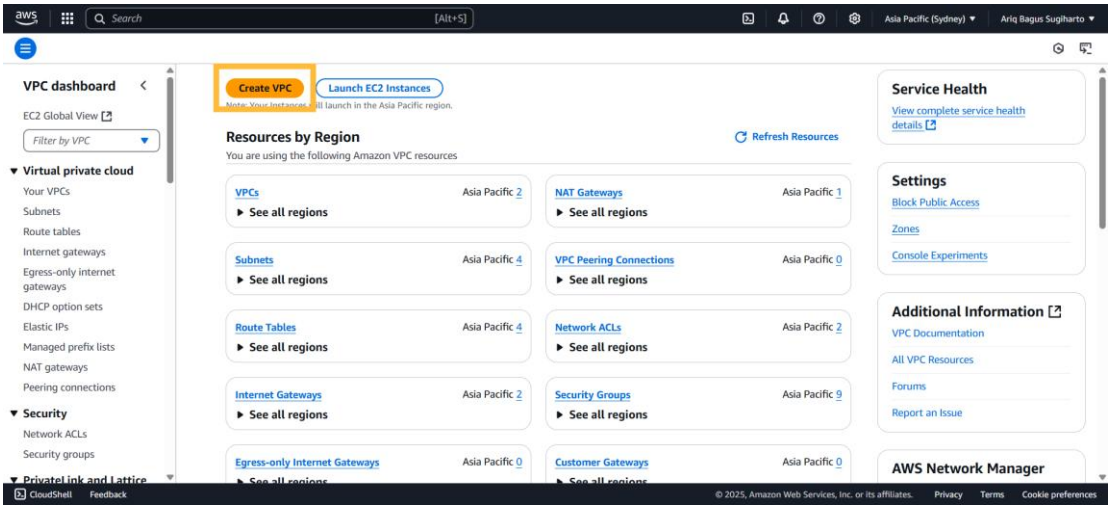
1. Membuat VPC dengan Subnet
   a. Masuk ke AWS Console.



   b. Cari VPC kemudian buka layanan VPC.



c. Pilih Create VPC.

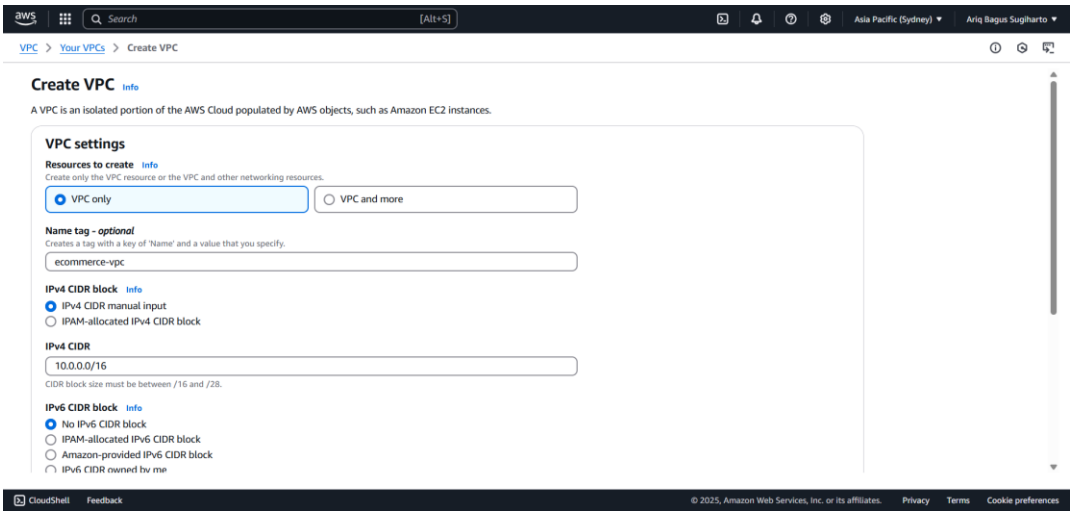


i. Masukkan nama untuk VPC.

ii. Tentukan blok CIDR IPv4 10.0.0.0/16.

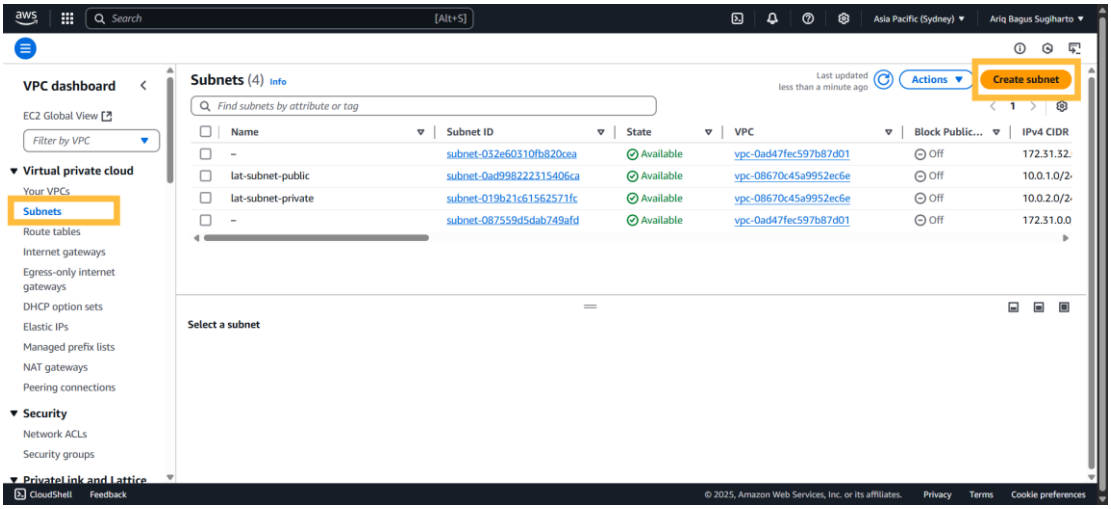
iii. Pilih opsi untuk IPv6 jika diperlukan.

iv. Klik “Create” untuk membuat VPC

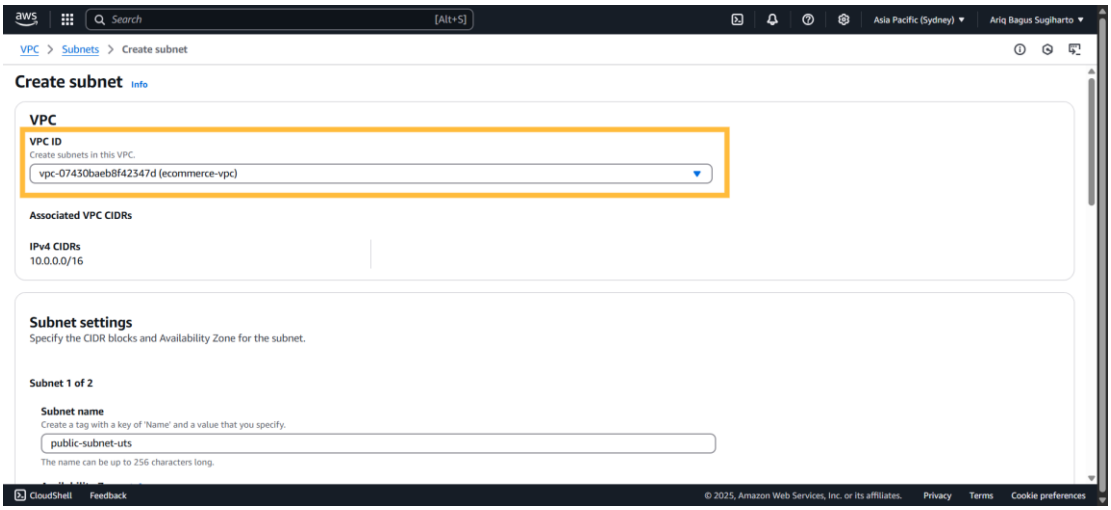


2. Buat 2 Subnet (1 public, 1 private)

a. Pilih Subnet, kemudian pilih Create Subnet.



b. Pilih VPC yang tadi dibuat.

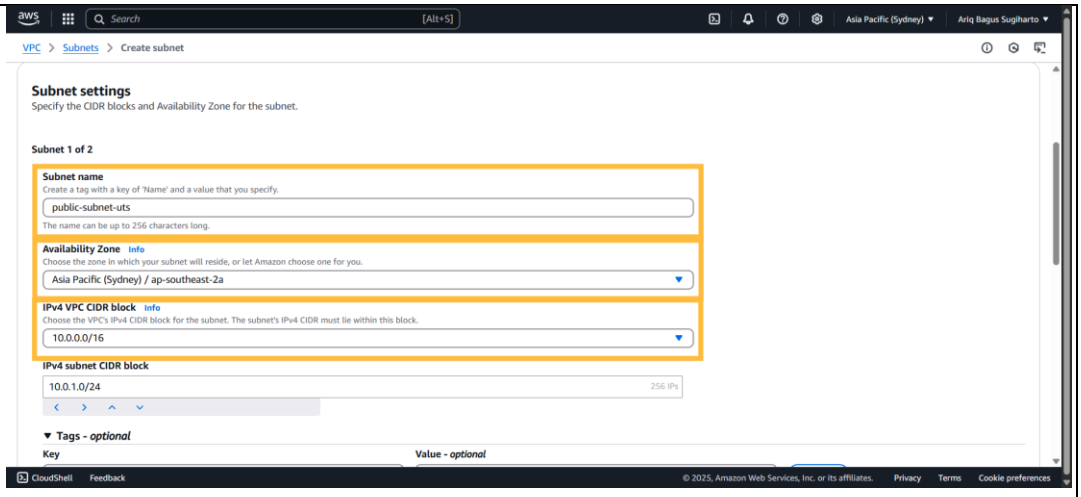


c. Public Subnet

i. Masukkan Nama untuk subnetnya, misal **public-subnet-uts**

ii. Pilih Availability Zone, misal **ap-southeast-2a**

iii. Masukkan CIDRnya, misal **10.0.1.0/24**

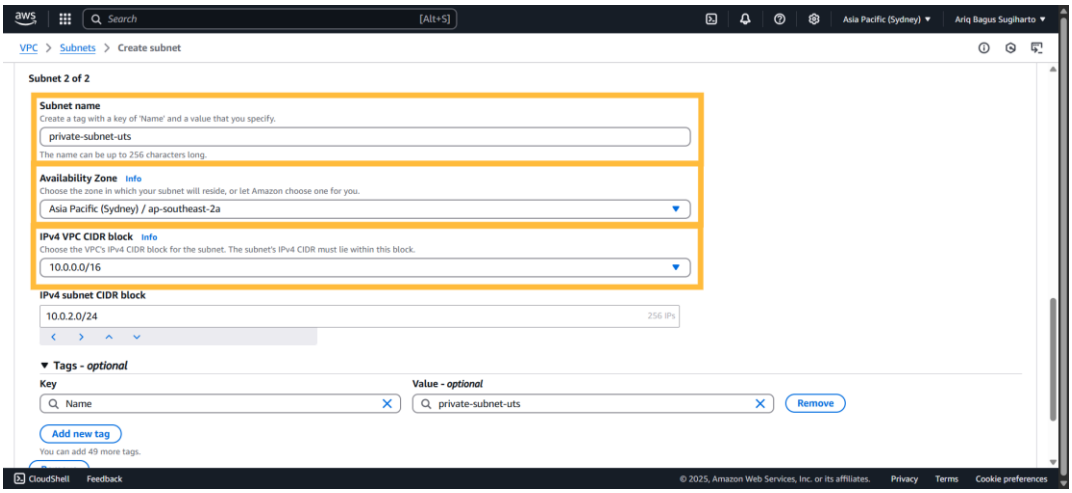


d. Private Subnet

i. Masukkan Nama untuk subnetnya, misal **private-subnet-uts**

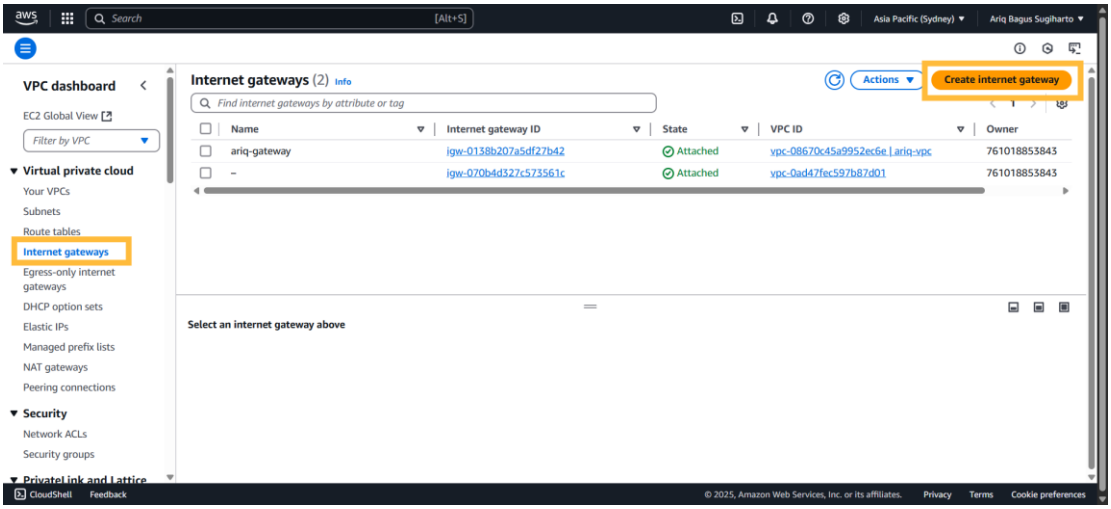
ii. Pilih Availability Zone, misal **ap-southeast-2a**

iii. Masukkan CIDRnya, misal **10.0.2.0/24**

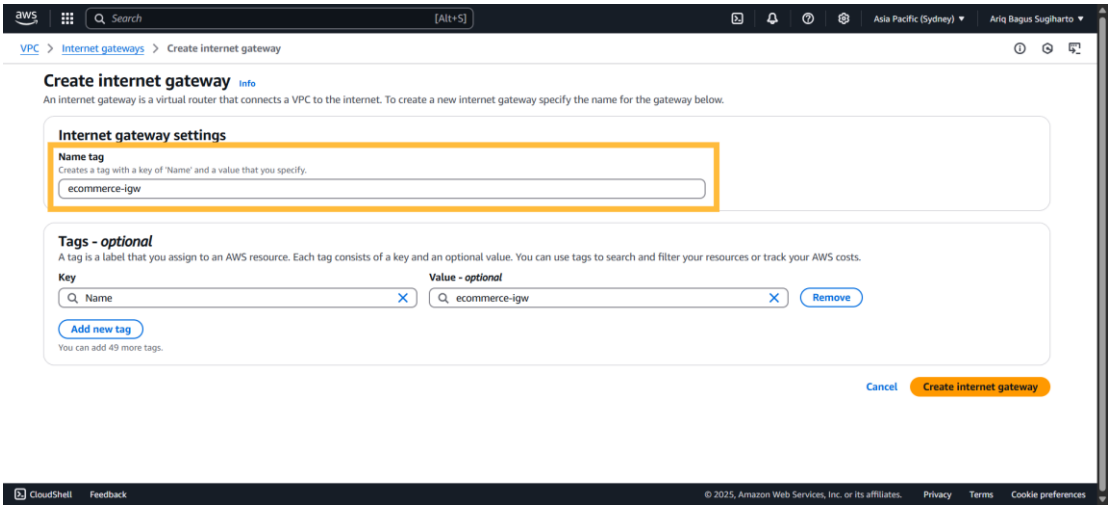


3. Buat Internet Gateway dan Hubungkan ke VPC

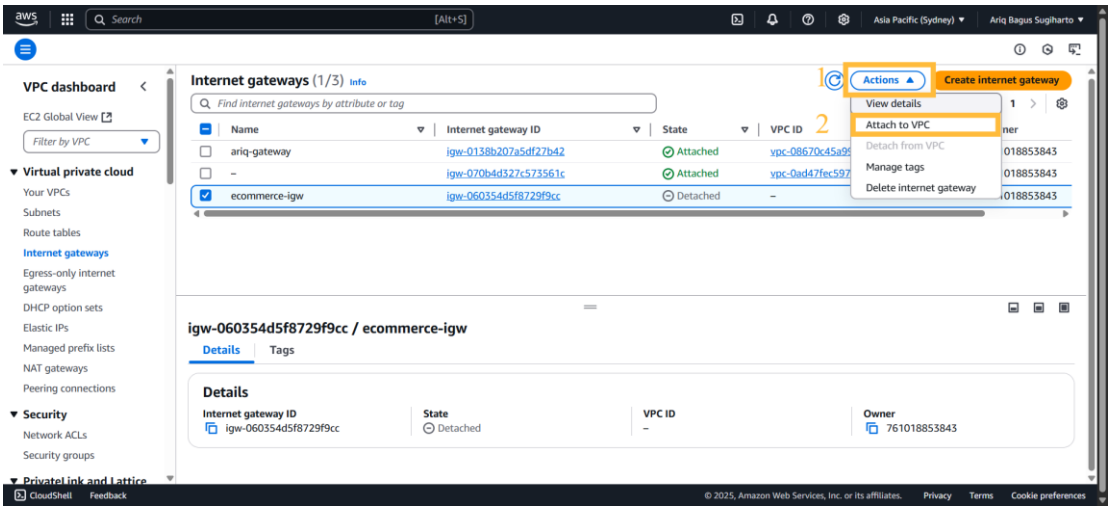
a. Buka menu Internet Gateway, lalu klik Create Internet Gateway

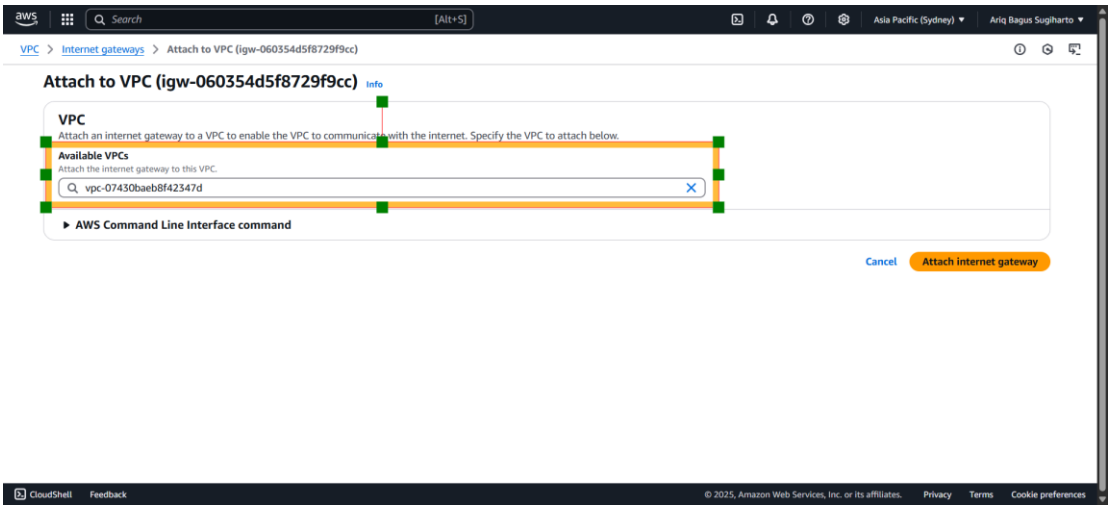


b. Berikan nama internet gatewaynya, misal **ecommerce-igw**



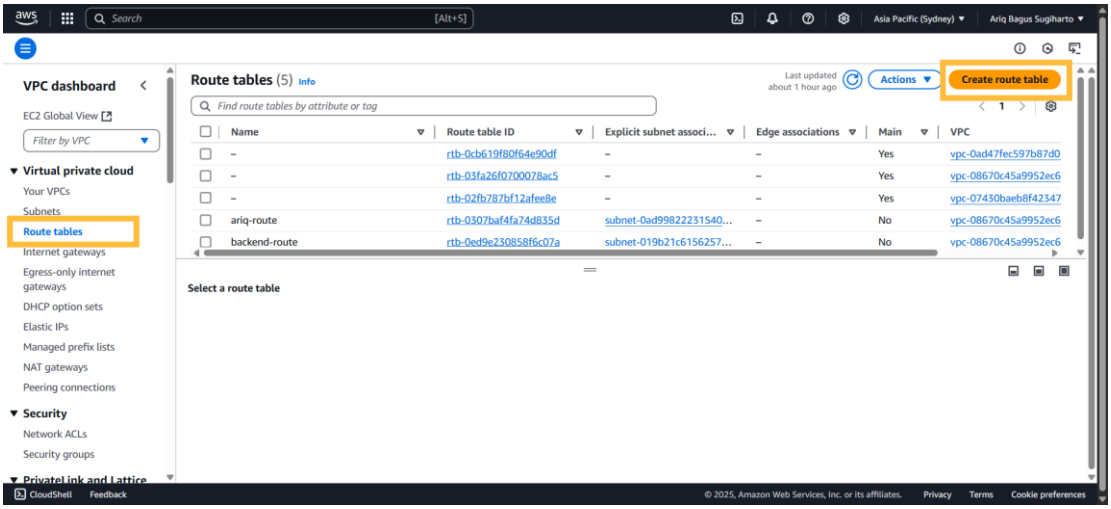
c. Attach ke vpc yang tadi dibuat



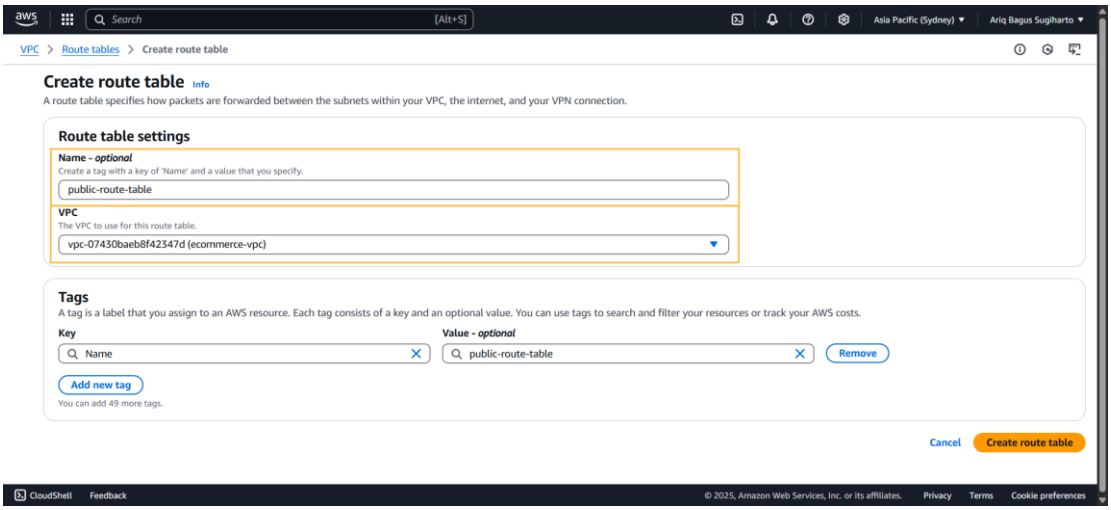


4. Buat Route Table dan Routing

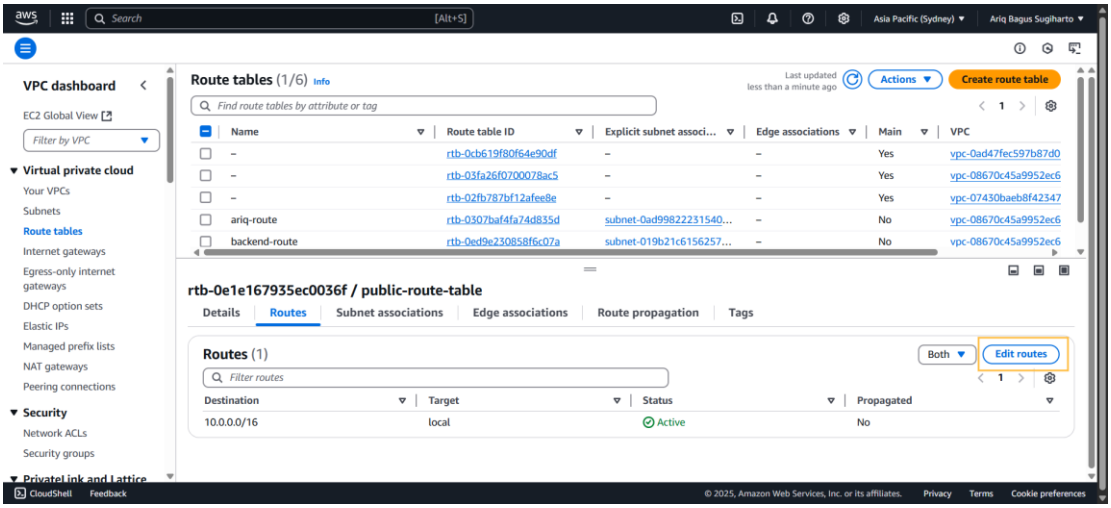
a. Pilih Route Tables di sidebar lalu klik Create Route Table



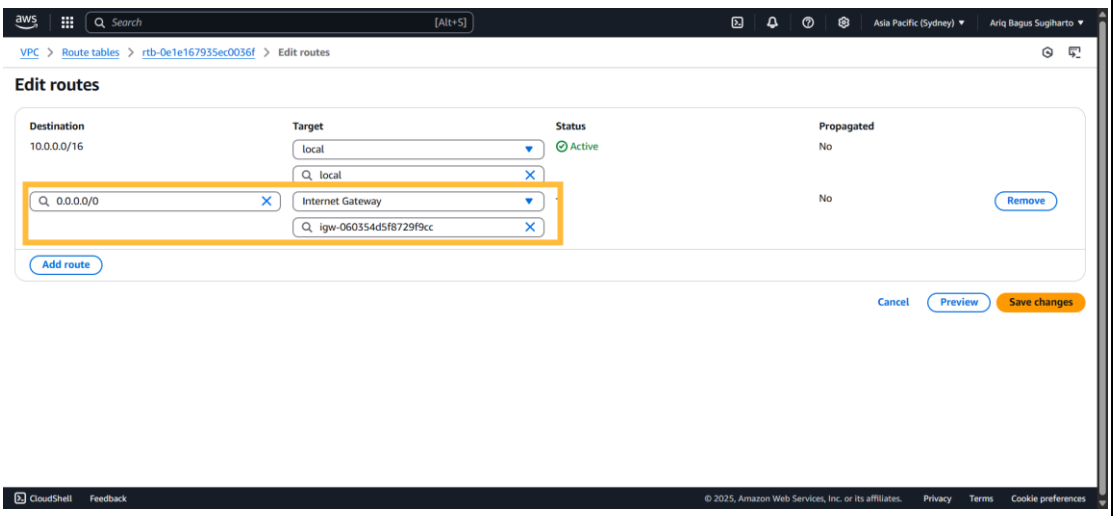
b. Masukkan Nama dan Pilih VPC yang telah dibuat



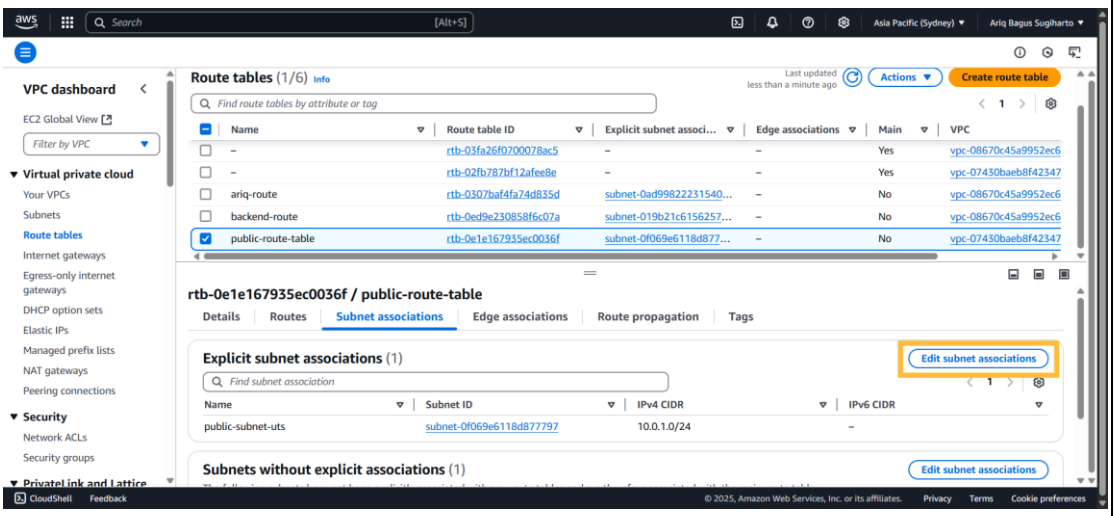
c. Selanjutnya pilih route table yang telah dibuat, klik routes, klik edit routes



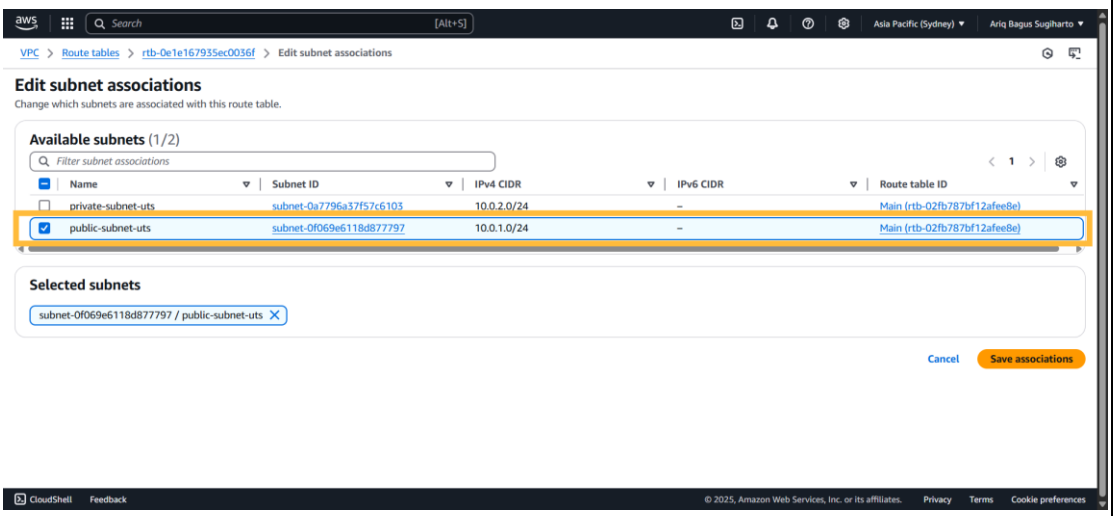
d. Tambahkan Route **0.0.0.0/0** ke **internet gateway** yang sebelumnya dibuat, lalu save



e. Selanjutnya pilih route table tadi, klik subnet associations, lalu klik edit subnet associations

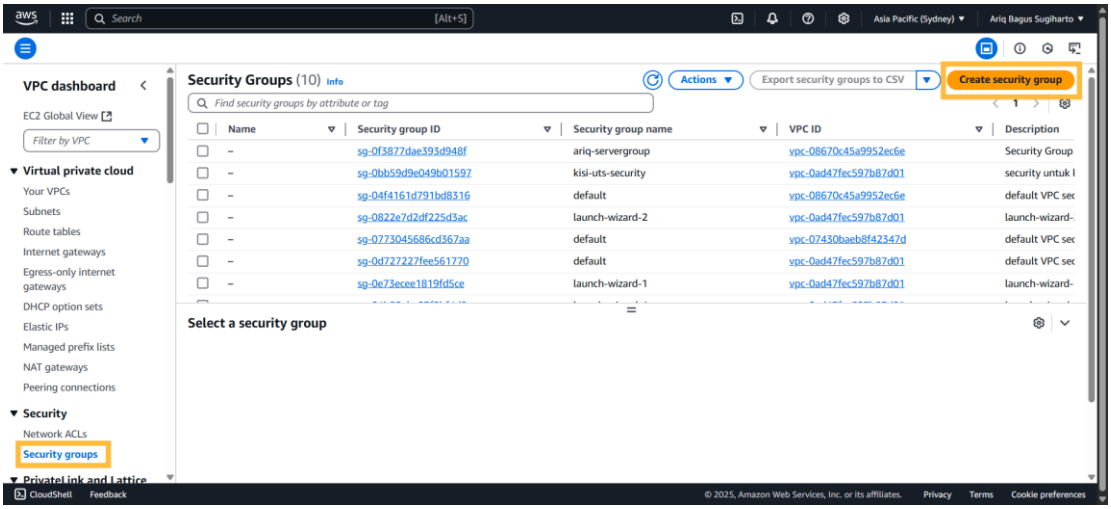


f. Tambahkan public subnet yang telah dibuat ke route table, lalu save



5. Buat Security Group

a. Pilih Security Groups di sidebar, lalu klik Create Security Group

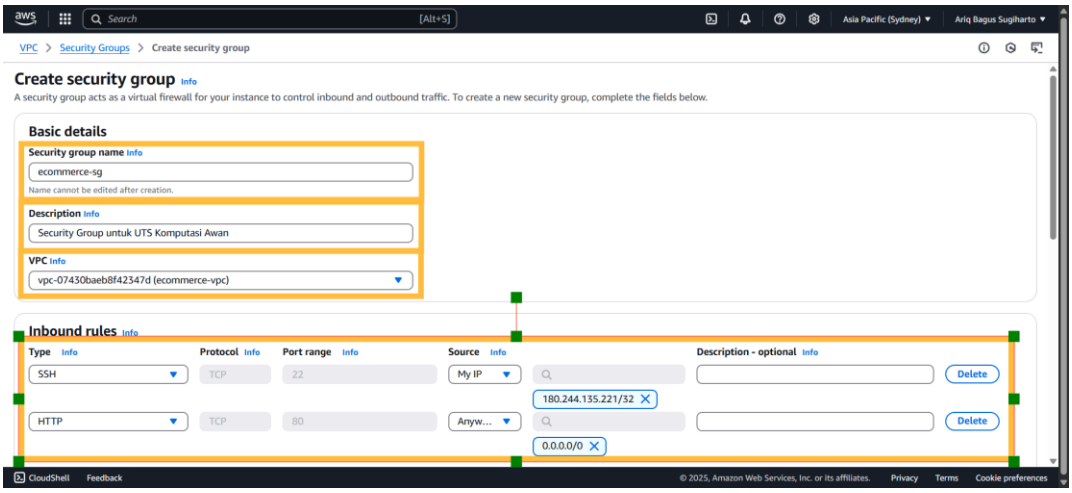


i. Masukkan Nama Security Group, misal **ecommerce-sg**

ii. Masukkan Deskripsi untuk Security Groupnya

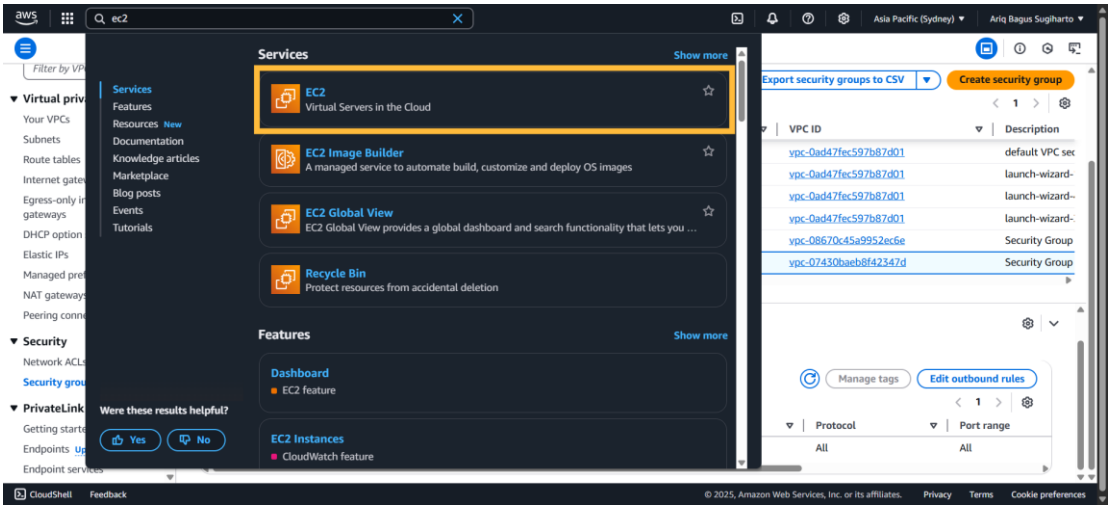
iii. Pilih VPC yang telah dibuat

iv. Tambahkan Inbound Rules yaitu **SSH (22)** dengan sourcenya **My IP** dan **HTTP (80)** dengan sourcenya **Anywhere** (untuk be ssh-nya harus dari ip security group fe, karena akses instancenya dari fe)

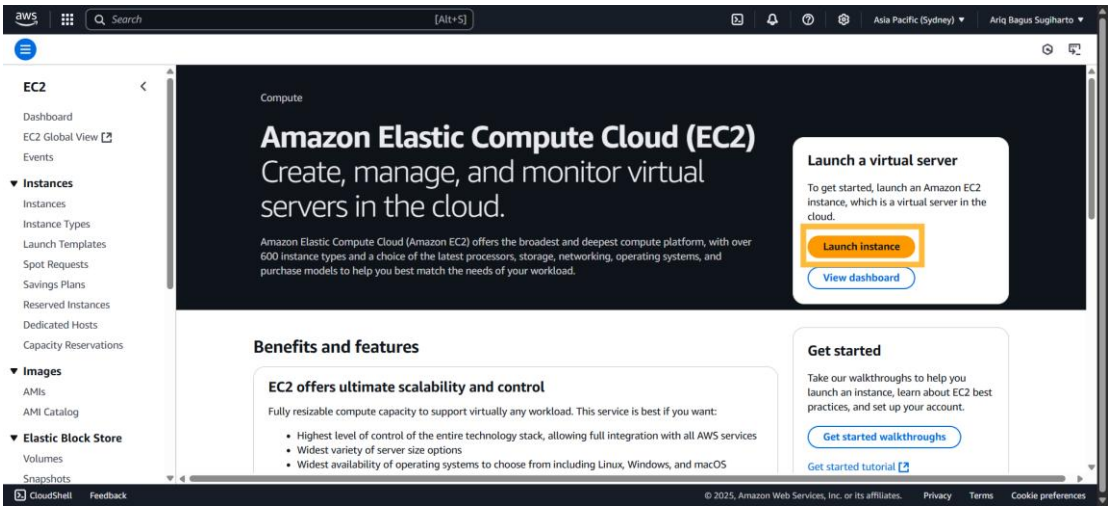


6. Launch EC2 Instances

a. Cari EC2 lalu buka EC2.



b. Pilih Launch Instances

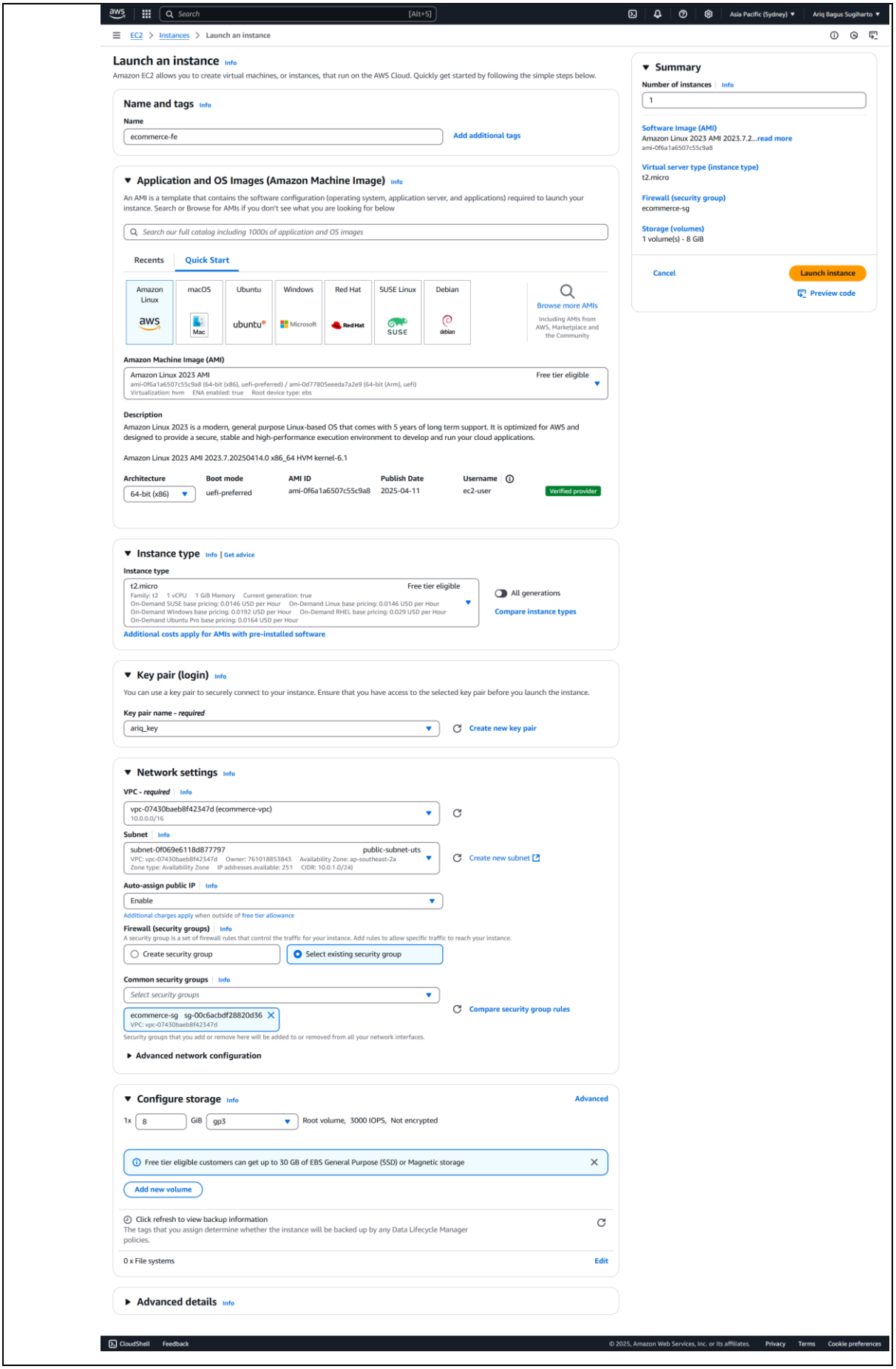


c. Buat dua buah instance sesuai ketentuan berikut:

i. Masukkan nama untuk tiap instance, misal **ecommerce-fe** dan **ecommerce-be**

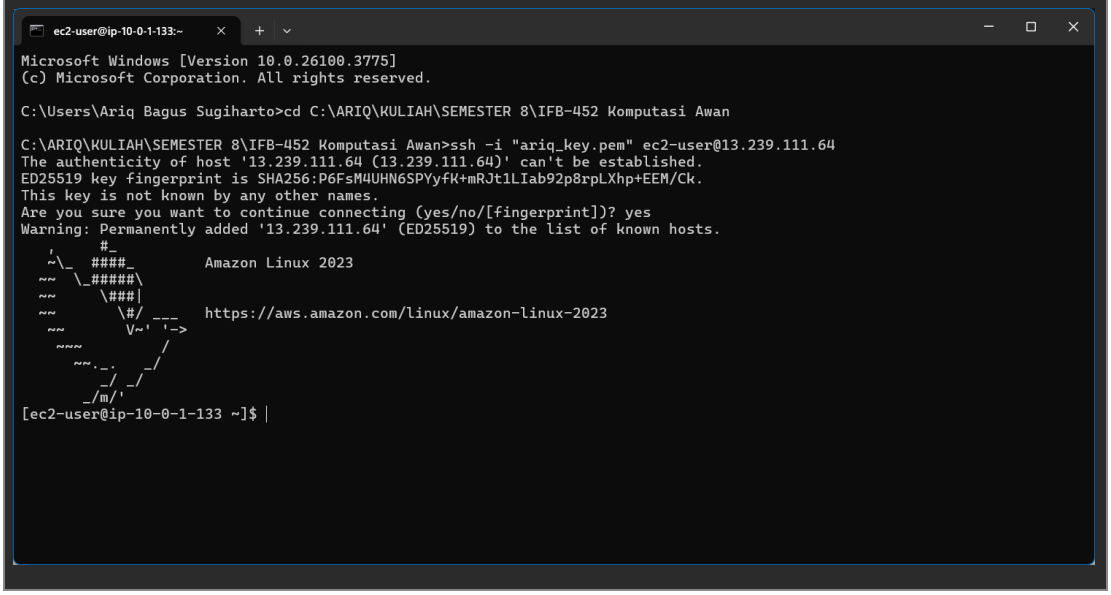
ii. Untuk key pair, bisa buat baru atau gunakan yang sudah ada

iii. Kemudian atur konfigurasi networknya menjadi vpc yang sudah dibuat, dengan subnetnya public untuk yang front-end, dan security groupnya juga ganti menjadi security group yang telah dibuat sebelumnya


aws
Search
[Alt+S]
Asia Pacific (Sydney)
Ariq Bagus Sugiharto
EC2 > Instances > Launch an instance
Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
ecommerce-fe
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Debian
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Amazon Linux 2023 AMI
ami-0f6a1a6507c55c9a8 (64-bit (x86), uefi-preferred) / ami-0d77805eeeda7a2e9 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs
Free tier eligible
Description
Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.
Amazon Linux 2023 AMI 2023.7.20250414.0 x86_64 HVM kernel-6.1
Architecture
64-bit (x86)
Boot mode
uefi-preferred
AMI ID
ami-0f6a1a6507c55c9a8
Publish Date
2025-04-11
Username
ec2-user
Verified provider
Instance type Info | Get advice
Instance type
t2.micro
Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand SUSE base pricing: 0.0146 USD per Hour On-Demand Linux base pricing: 0.0146 USD per Hour
On-Demand Windows base pricing: 0.0192 USD per Hour On-Demand RHEL base pricing: 0.029 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0164 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
ariq_key
Create new key pair
Network settings Info
VPC - required Info
vpc-07430baeb8f42347d (ecommerce-vpc)
10.0.0.0/16
Subnet Info
subnet-0f069e6118d877797
public-subnet-uts
VPC: vpc-07430baeb8f42347d Owner: 761018853843 Availability Zone: ap-southeast-2a
Zone type: Availability Zone IP addresses available: 251 CIDR: 10.0.1.0/24)
Create new subnet
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
Common security groups Info
Select security groups
ecommerce-sg sg-00c6acbdf28820d36
VPC: vpc-07430baeb8f42347d
Compare security group rules
Security groups that you add or remove here will be added to or removed from all your network interfaces.
Advanced network configuration
Configure storage Info
Advanced
1x
8
GiB
gp3
Root volume, 3000 IOPS, Not encrypted
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage
Add new volume
Click refresh to view backup information
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.
0 x File systems
Edit
Advanced details Info
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0f6a1a6507c55c9a8
Virtual server type (instance type)
t2.micro
Firewall (security group)
ecommerce-sg
Storage (volumes)
1 volume(s) - 8 GiB
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

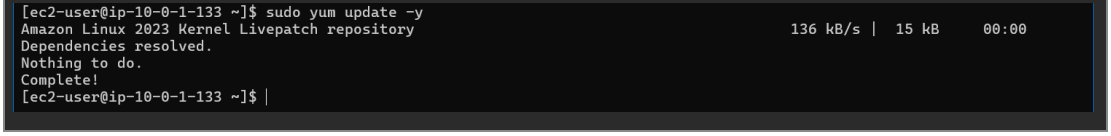
d. Connect ke ssh front-end dengan command berikut:

```
ssh -i “/path_key.pem” ec2-user@ipv4_public
```



e. Cek apakah terhubung ke internet menggunakan command berikut:

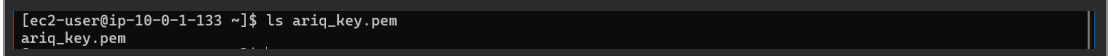
```
sudo yum update -y
```



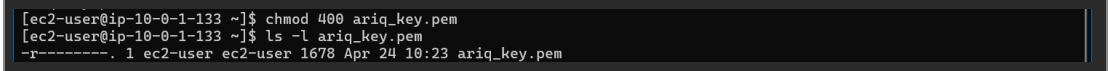
f. Connect ke back-end dengan command berikut:

```
scp -i "key.pem" key.pem ec2-user@ ipv4_public:/home/ec2-user/
```

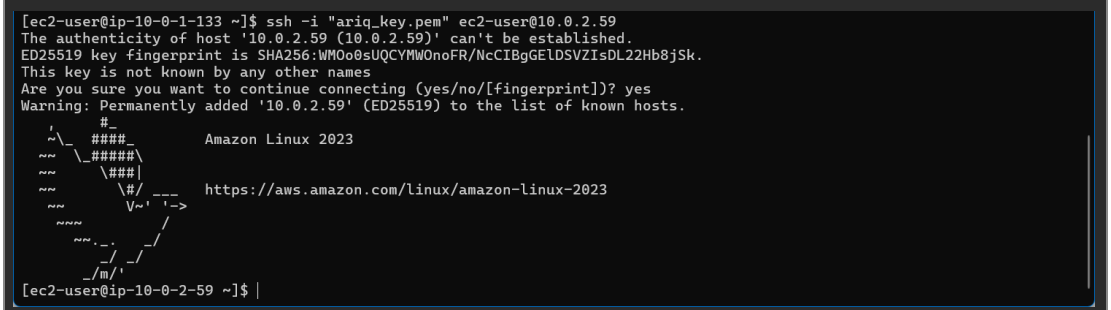

```
ls key.pem
```




```
chmod 400 key.pem

ls -l key.pem
```

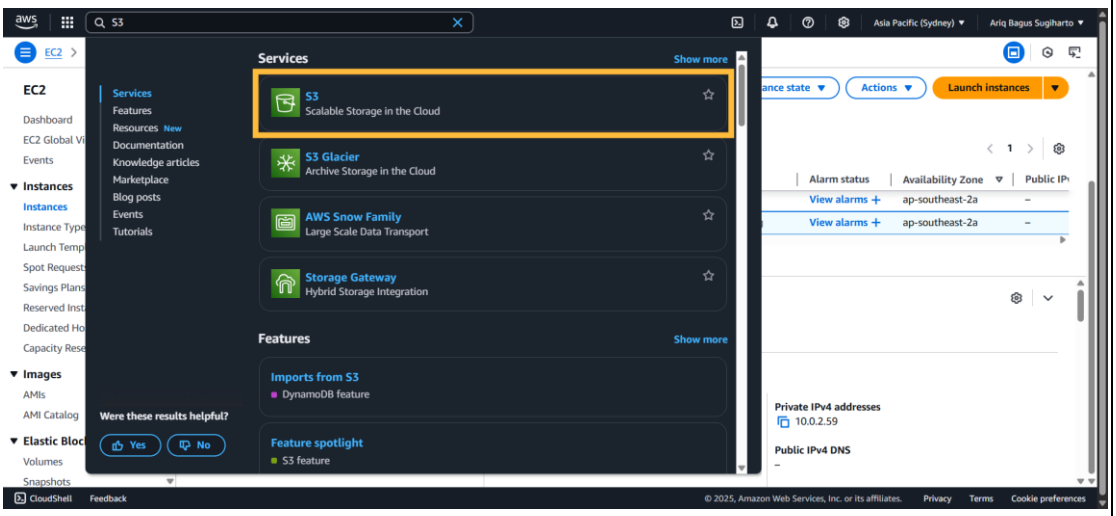



```
ssh -i “key.pem” ec2-user@private_ip_address
```

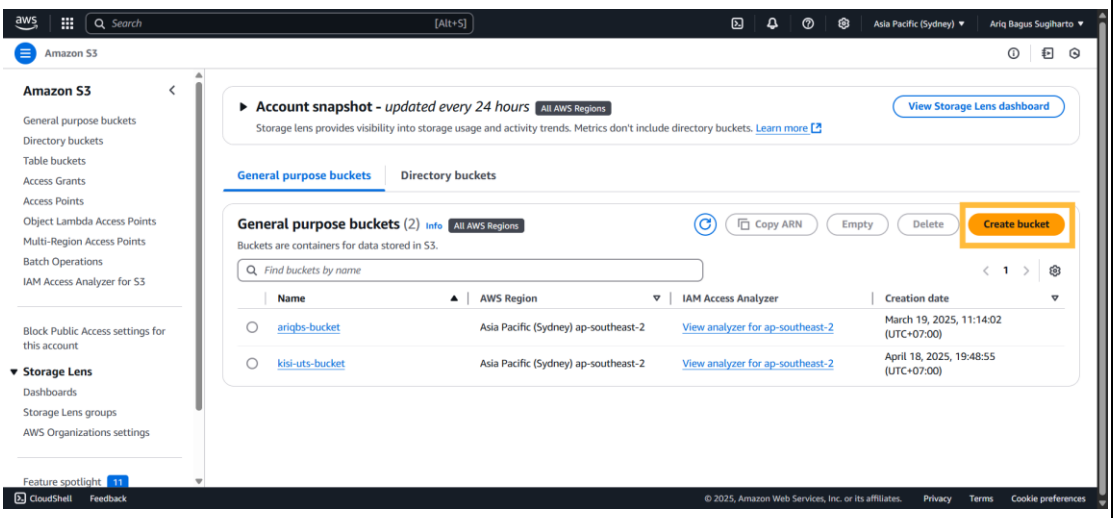


Bagian 2 - Penyimpanan Data dengan (Amazon S3 & Database)

1. Buat Bucket S3
   a. Cari S3 lalu Pilih S3

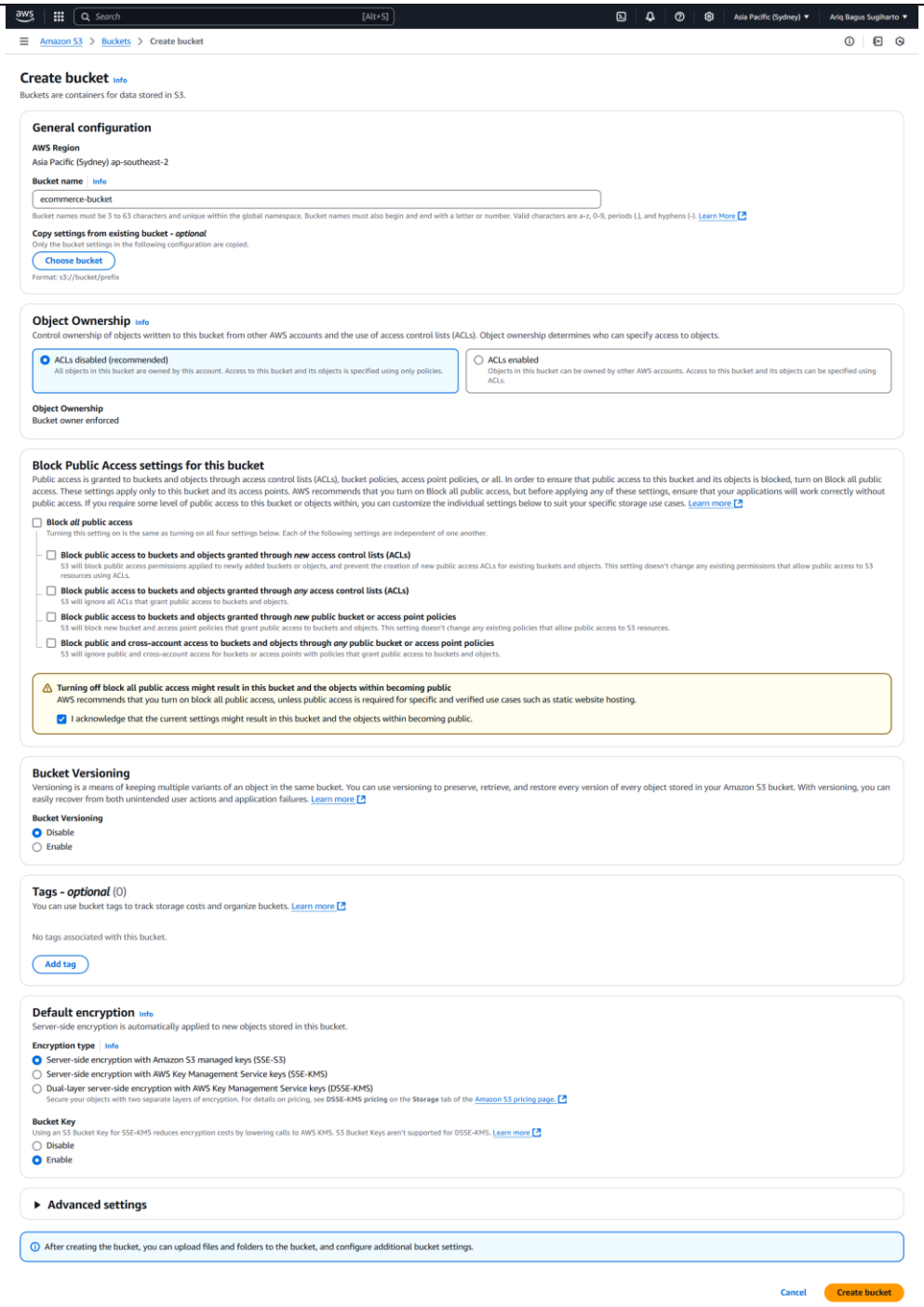


   b. Klik Create Bucket

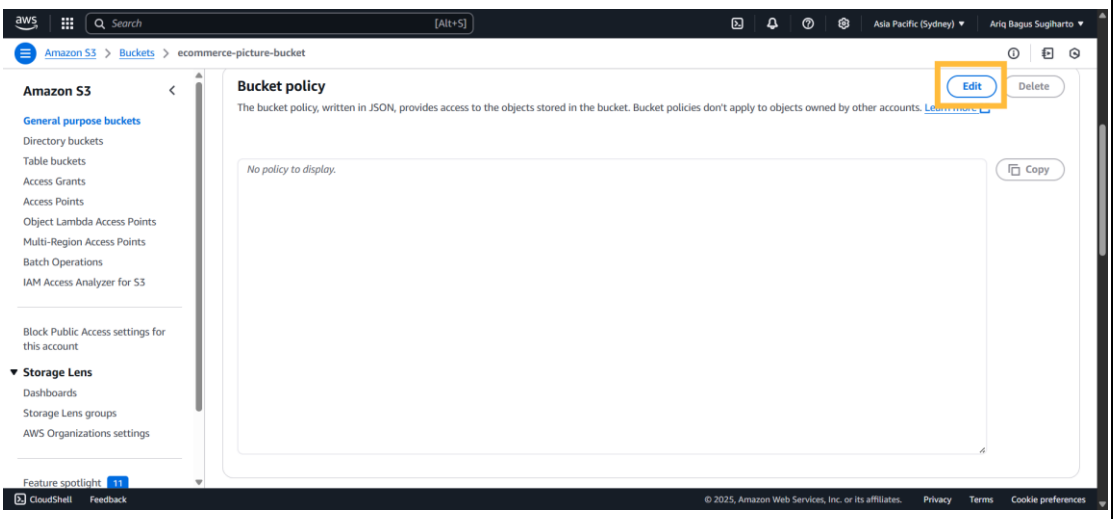


      i. Masukkan Nama Bucket, misal **ecommerce-picture-bucket**

      ii. Uncheck **block public access** agar bisa diakses public isinya

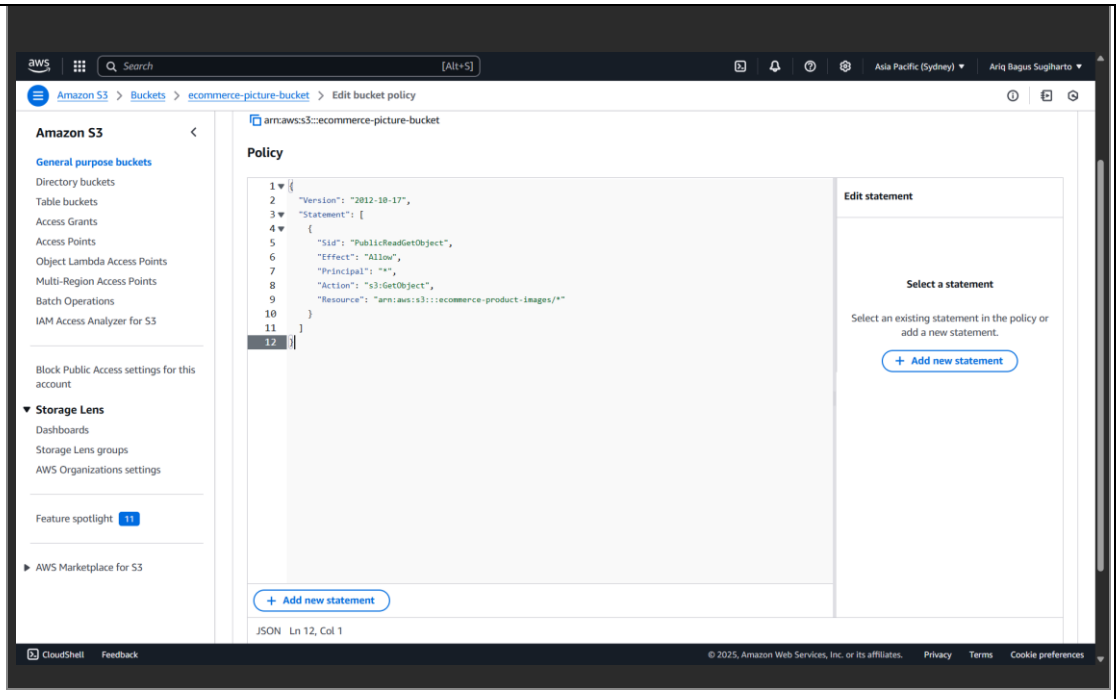

aws
Search
[Alt+S]
Asia Pacific (Sydney)
Ariq Bagus Sugiharto
Amazon S3 > Buckets > Create bucket
Create bucket Info
Buckets are containers for data stored in S3.
General configuration
AWS Region
Asia Pacific (Sydney) ap-southeast-2
Bucket name Info
ecommerce-bucket
Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). Learn More
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket
Format: s3://bucket/prefix
Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.
ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.
Object Ownership
Bucket owner enforced
Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.
Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.
I acknowledge that the current settings might result in this bucket and the objects within becoming public.
Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more
Bucket Versioning
Disable
Enable
Tags - optional (0)
You can use bucket tags to track storage costs and organize buckets. Learn more
No tags associated with this bucket.
Add tag
Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.
Encryption type Info
Server-side encryption with Amazon S3 managed keys (SSE-S3)
Server-side encryption with AWS Key Management Service keys (SSE-KMS)
Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the Amazon S3 pricing page.
Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more
Disable
Enable
Advanced settings
After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.
Cancel
Create bucket

c. Selanjutnya buka bucket yang telah dibuat, pilih permission, lalu bucket policy dan pilih edit

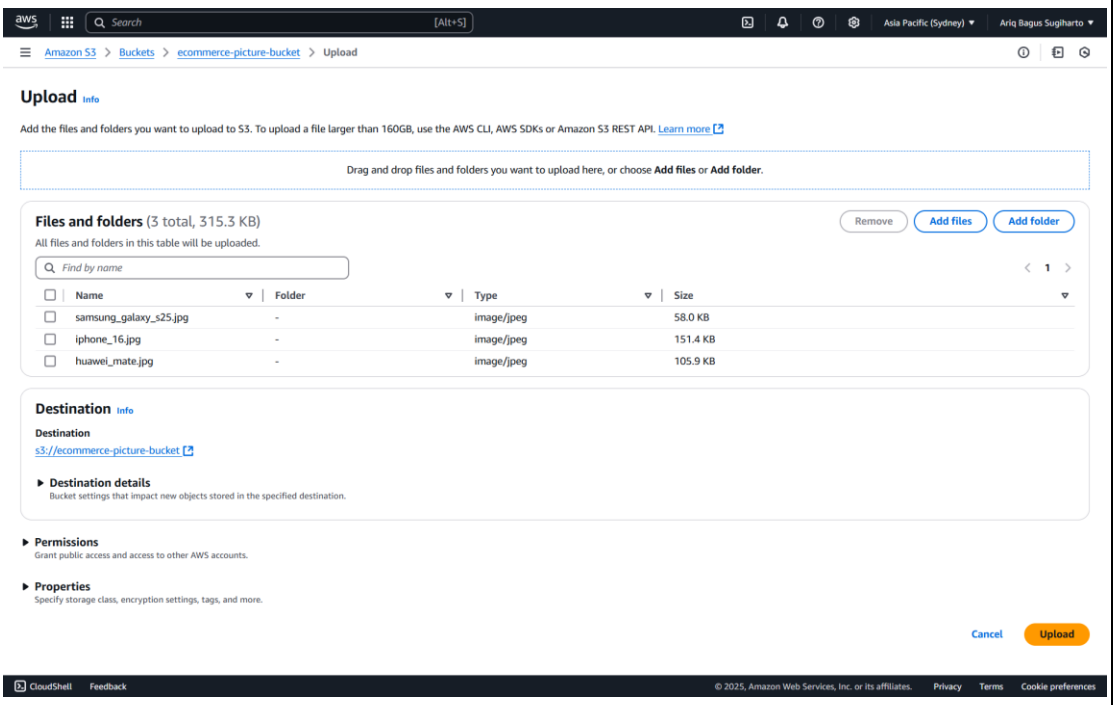


d. Tambahkan policy berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::ecommerce-picture-bucket/*"
    }
  ]
}
```

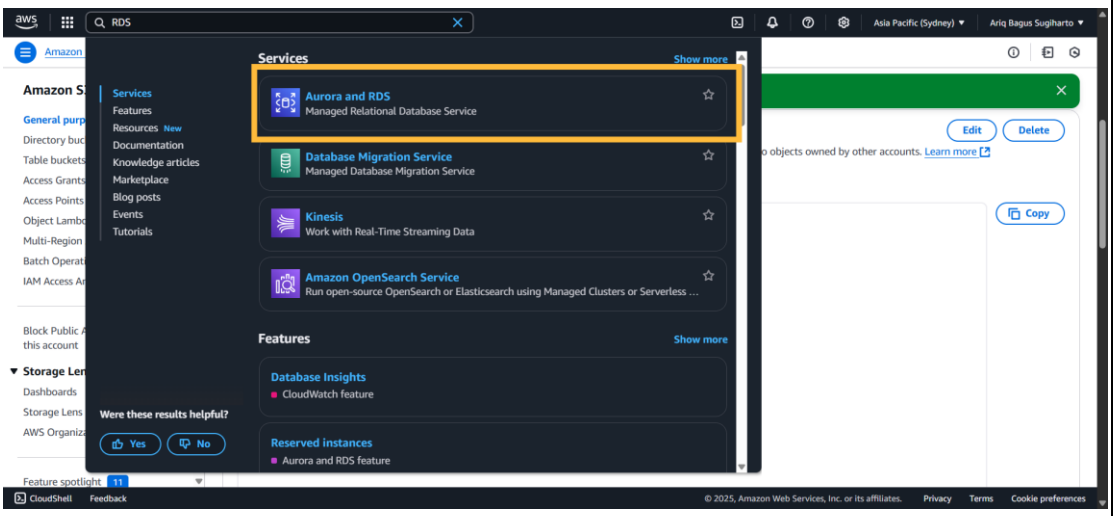


e. Tambahkan gambar ke dalam bucketnya, dengan cara klik upload di bucket.

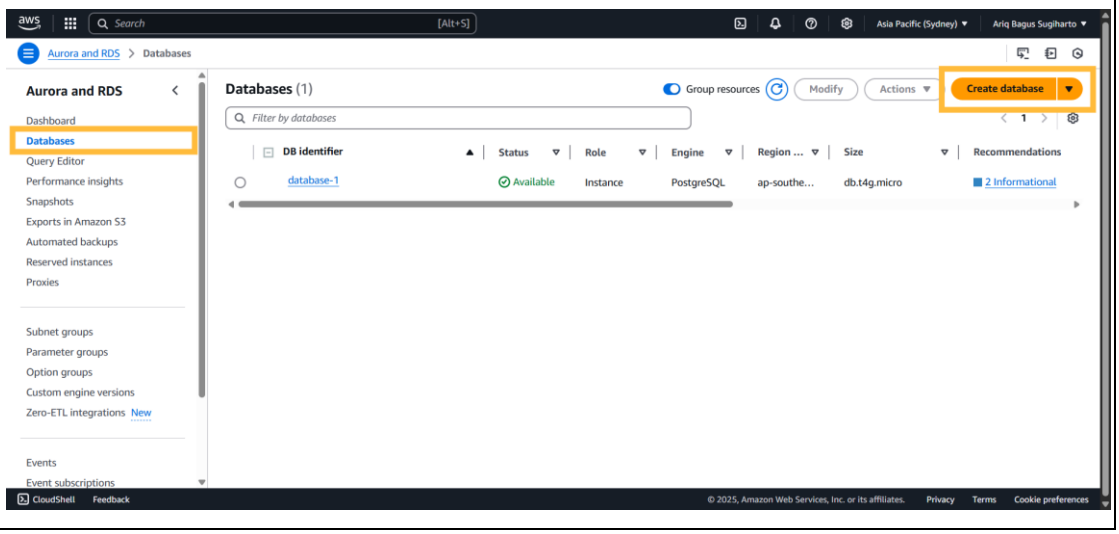


2. Buat RDS MySQL

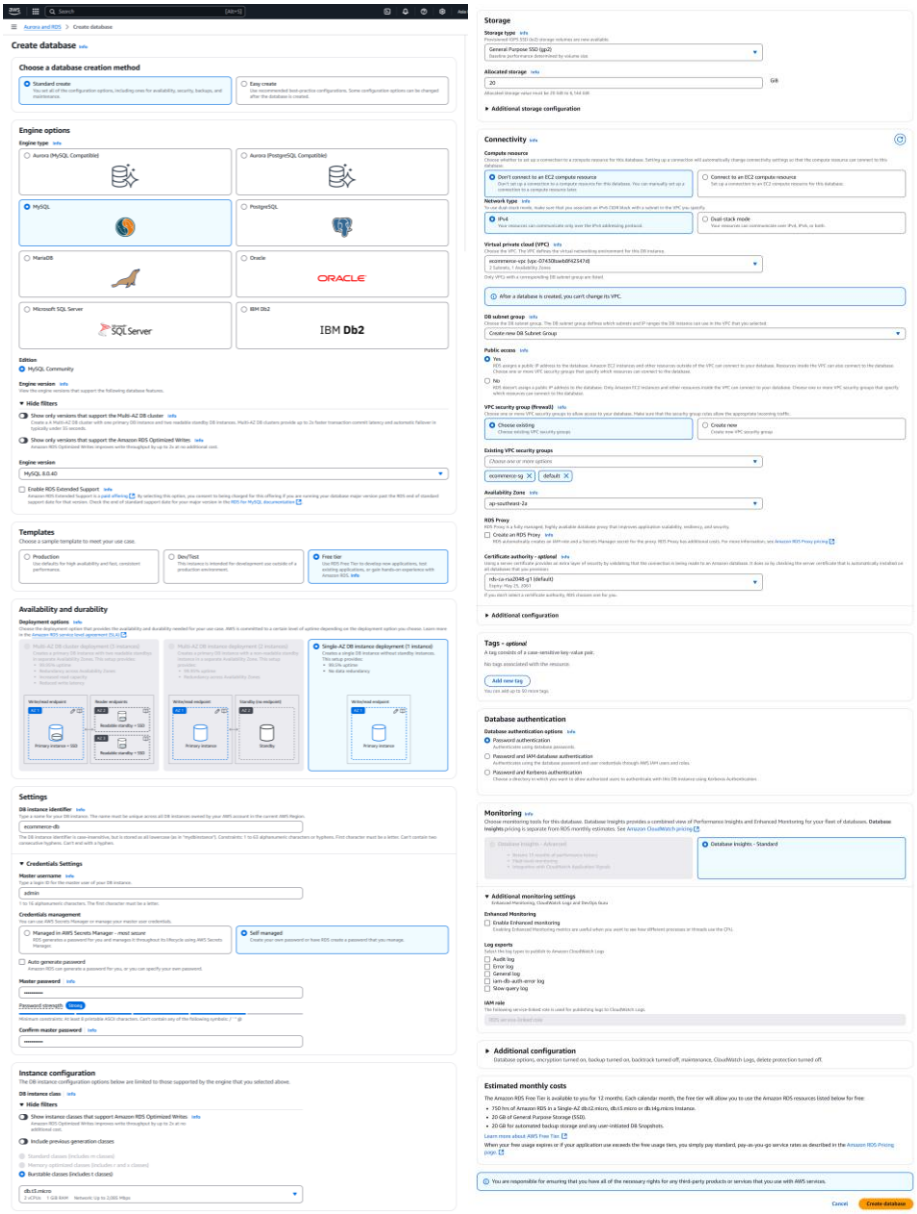
a. Cari RDS lalu pilih RDS



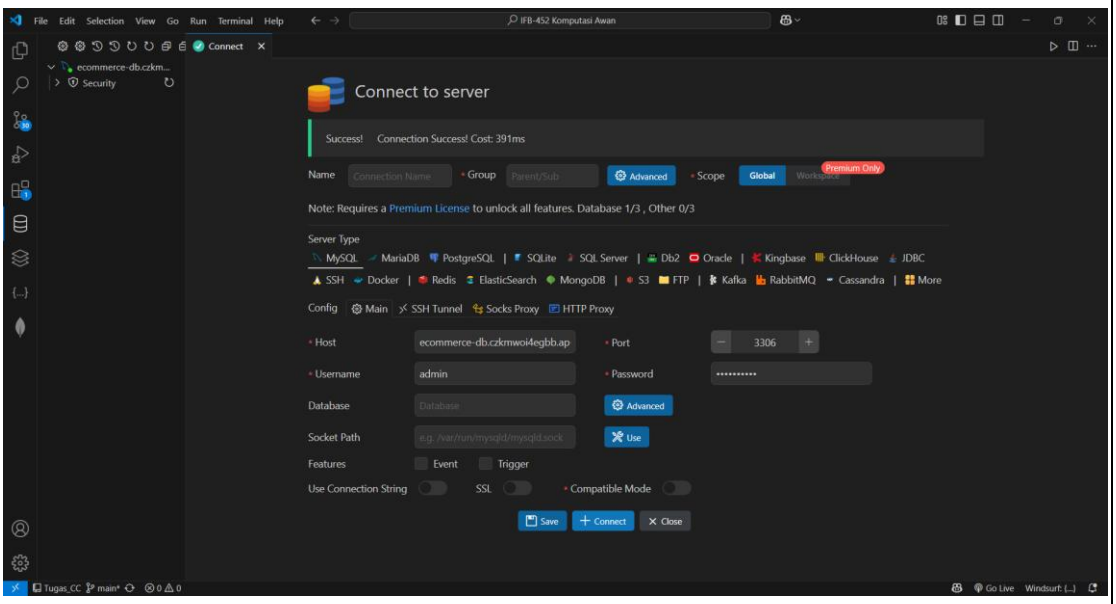
b. Pilih Databases di sidebar, lalu klik Create database



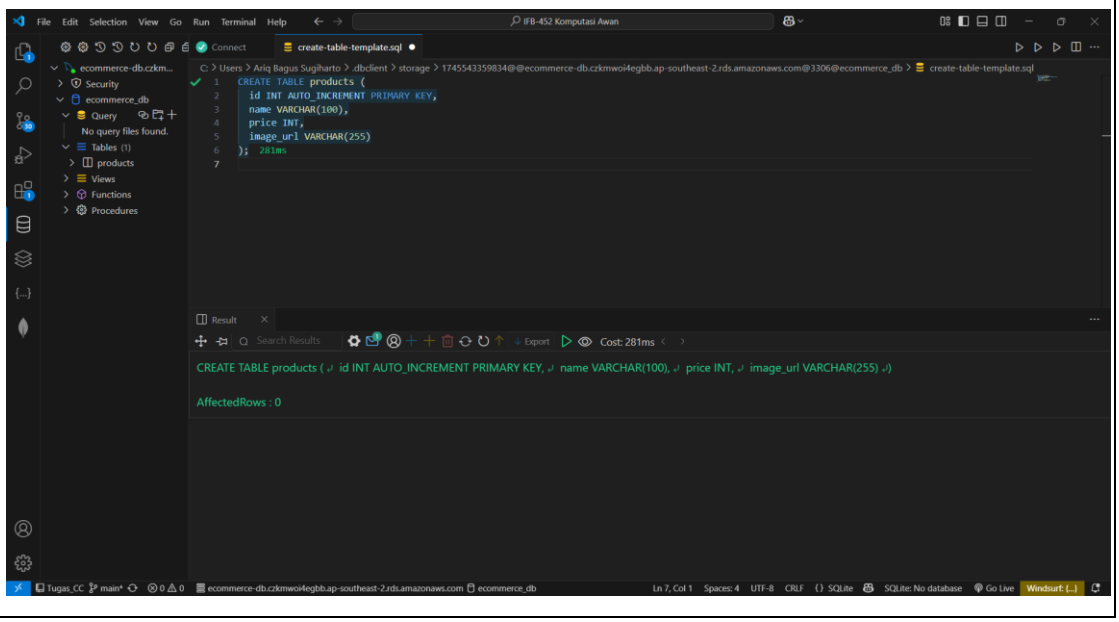
i. Input konfigurasi sesuai berikut:



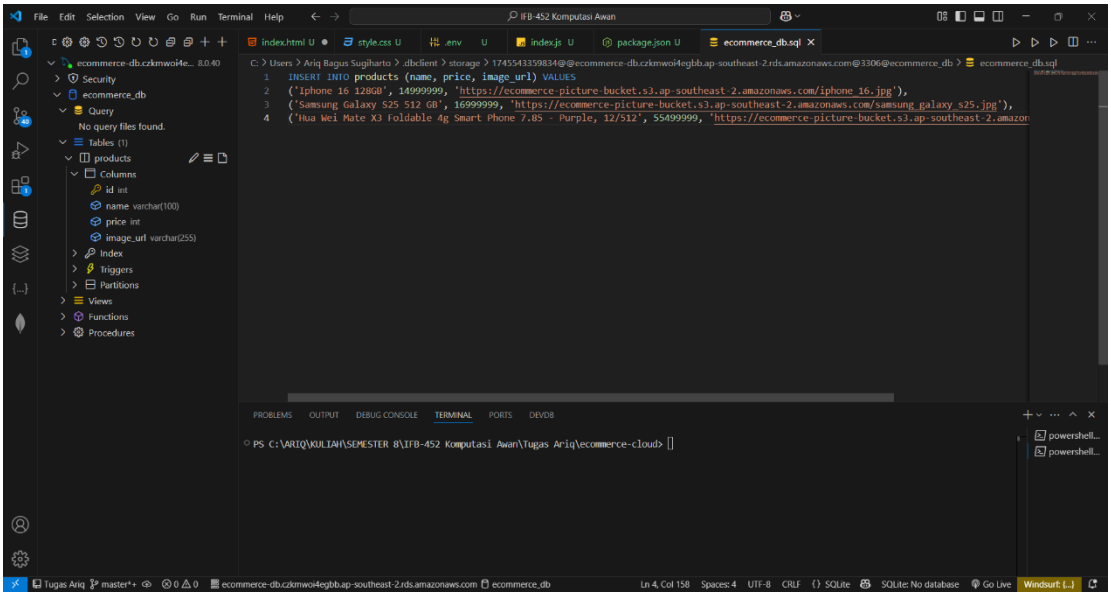
c. Buka database di Visual Studio Code



d. Buat Database, misal **ecommerce-db**, lalu buat tabel, misal **products**



e. Tambahkan data ke dalam tabel **products**.



Bagian 3 - Pengembangan Aplikasi Web

1. Setup Frontend EC2

   a. Lakukan command berikut untuk installasi kebutuhan di frontend:

```
sudo yum install httpd -y
```

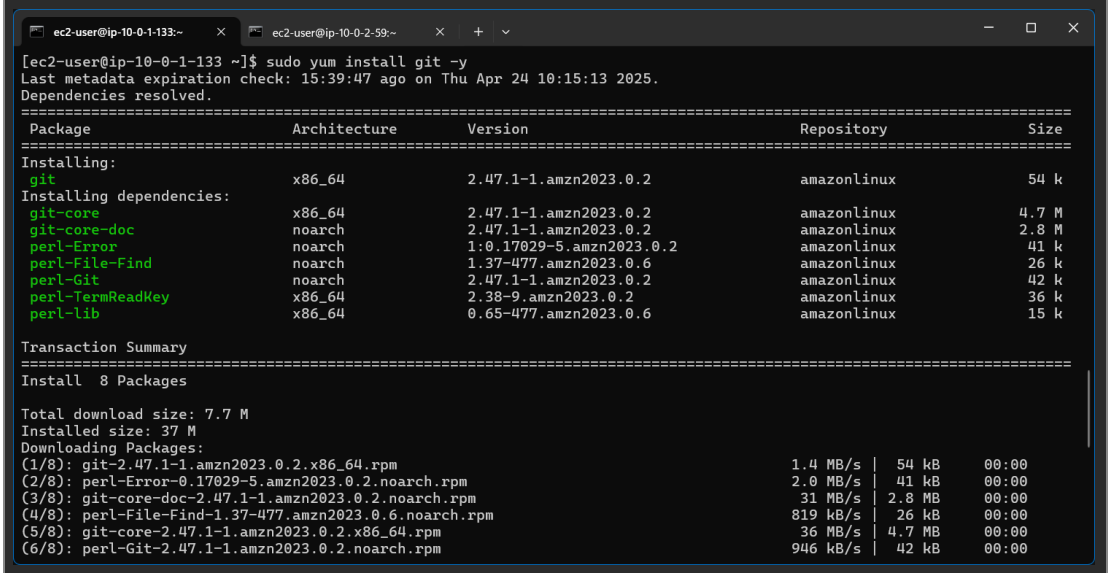
```
[ec2-user@ip-10-0-1-133 ~]$ sudo yum install httpd -y
Last metadata expiration check: 15:26:43 ago on Thu Apr 24 10:15:13 2025.
Dependencies resolved.
================================================================================
 Package              Architecture        Version                  Repository     Size
================================================================================
Installing:
 httpd                x86_64              2.4.62-1.amzn2023        amazonlinux    48 k
Installing dependencies:
 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
 libbrotli            x86_64              1.0.9-4.amzn2023.0.2     amazonlinux   315 k
 mailcap              noarch              2.1.49-3.amzn2023.0.3    amazonlinux    33 k
Installing weak dependencies:
 apr-util-openssl     x86_64              1.6.3-1.amzn2023.0.1     amazonlinux    17 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

Transaction Summary
================================================================================
Install  12 Packages

Total download size: 2.3 M
Installed size: 6.9 M
Downloading Packages:
(1/12): apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64.rpm          485 kB/s |  17 kB     00:00
```

```
sudo systemctl enable httpd
```

```
sudo systemctl start httpd
```

```
[ec2-user@ip-10-0-1-133 ~]$ sudo systemctl start httpd
[ec2-user@ip-10-0-1-133 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
```

sudo yum install git -y

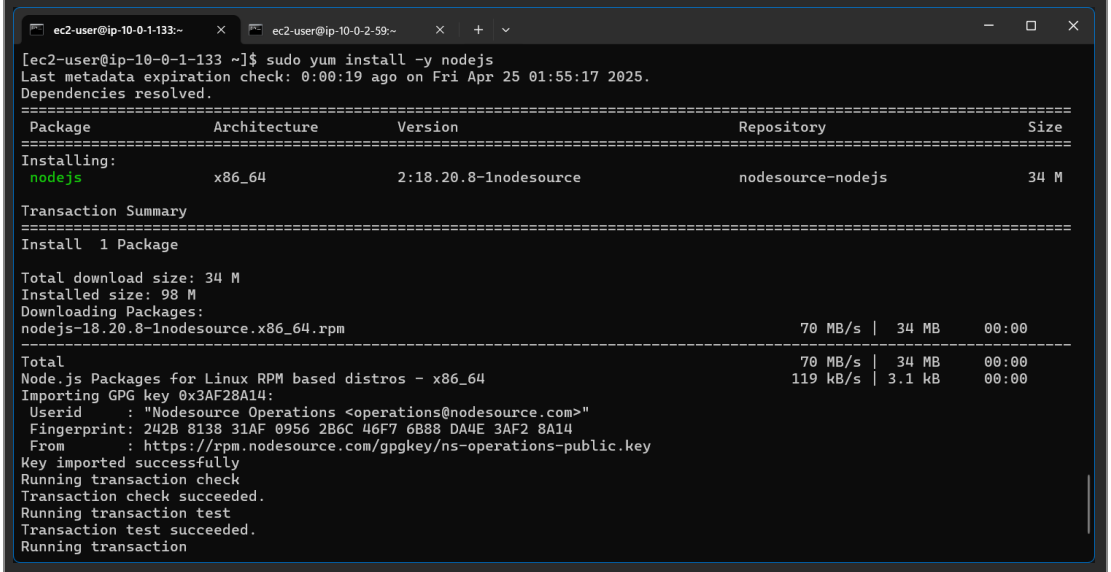


```
[ec2-user@ip-10-0-1-133 ~]$ sudo yum install git -y
Last metadata expiration check: 15:39:47 ago on Thu Apr 24 10:15:13 2025.
Dependencies resolved.
================================================================================
 Package                Architecture        Version                    Repository                Size
================================================================================
Installing:
 git                    x86_64              2.47.1-1.amzn2023.0.2      amazonlinux               54 k
Installing dependencies:
 git-core               x86_64              2.47.1-1.amzn2023.0.2      amazonlinux              4.7 M
 git-core-doc           noarch              2.47.1-1.amzn2023.0.2      amazonlinux              2.8 M
 perl-Error             noarch              1:0.17029-5.amzn2023.0.2   amazonlinux               41 k
 perl-File-Find         noarch              1.37-477.amzn2023.0.6      amazonlinux               26 k
 perl-Git               noarch              2.47.1-1.amzn2023.0.2      amazonlinux               42 k
 perl-TermReadKey       x86_64              2.38-9.amzn2023.0.2        amazonlinux               36 k
 perl-lib               x86_64              0.65-477.amzn2023.0.6      amazonlinux               15 k

Transaction Summary
================================================================================
Install  8 Packages

Total download size: 7.7 M
Installed size: 37 M
Downloading Packages:
(1/8): git-2.47.1-1.amzn2023.0.2.x86_64.rpm                 1.4 MB/s |  54 kB     00:00
(2/8): perl-Error-0.17029-5.amzn2023.0.2.noarch.rpm         2.0 MB/s |  41 kB     00:00
(3/8): git-core-doc-2.47.1-1.amzn2023.0.2.noarch.rpm         31 MB/s | 2.8 MB     00:00
(4/8): perl-File-Find-1.37-477.amzn2023.0.6.noarch.rpm      819 kB/s |  26 kB     00:00
(5/8): git-core-2.47.1-1.amzn2023.0.2.x86_64.rpm             36 MB/s | 4.7 MB     00:00
(6/8): perl-Git-2.47.1-1.amzn2023.0.2.noarch.rpm            946 kB/s |  42 kB     00:00
```

curl -sL https://rpm.nodesource.com/setup_18.x | sudo bash -

```
[ec2-user@ip-10-0-1-133 ~]$ curl -sL https://rpm.nodesource.com/setup_18.x | sudo bash -
2025-04-25 01:55:16 - Cleaning up old repositories...
2025-04-25 01:55:16 - Old repositories removed
2025-04-25 01:55:16 - Supported architecture: x86_64
2025-04-25 01:55:16 - Added N|Solid repository for LTS version: 18.x
2025-04-25 01:55:16 - dnf available, updating...
Node.js Packages for Linux RPM based distros - x86_64          13 MB/s | 1.0 MB     00:00
Metadata cache created.
N|Solid Packages for Linux RPM based distros - x86_64         8.2 MB/s | 637 kB     00:00
Metadata cache created.
2025-04-25 01:55:17 - Repository is configured and updated.
2025-04-25 01:55:17 - You can use N|solid Runtime as a node.js alternative
2025-04-25 01:55:17 - To install N|solid Runtime, run: dnf install nsolid -y
2025-04-25 01:55:17 - Run 'dnf install nodejs -y' to complete the installation.
```

sudo yum install -y nodejs



```
[ec2-user@ip-10-0-1-133 ~]$ sudo yum install -y nodejs
Last metadata expiration check: 0:00:19 ago on Fri Apr 25 01:55:17 2025.
Dependencies resolved.
================================================================================
 Package             Architecture        Version                   Repository                Size
================================================================================
Installing:
 nodejs              x86_64              2:18.20.8-1nodesource     nodesource-nodejs         34 M

Transaction Summary
================================================================================
Install  1 Package

Total download size: 34 M
Installed size: 98 M
Downloading Packages:
nodejs-18.20.8-1nodesource.x86_64.rpm                         70 MB/s |  34 MB     00:00
--------------------------------------------------------------------------------
Total                                                         70 MB/s |  34 MB     00:00
Node.js Packages for Linux RPM based distros - x86_64        119 kB/s | 3.1 kB     00:00
Importing GPG key 0x3AF28A14:
 Userid     : "Nodesource Operations <operations@nodesource.com>"
 Fingerprint: 242B 8138 31AF 0956 2B6C 46F7 6B88 DA4E 3AF2 8A14
 From       : https://rpm.nodesource.com/gpgkey/ns-operations-public.key
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
```

sudo npm install -g pm2

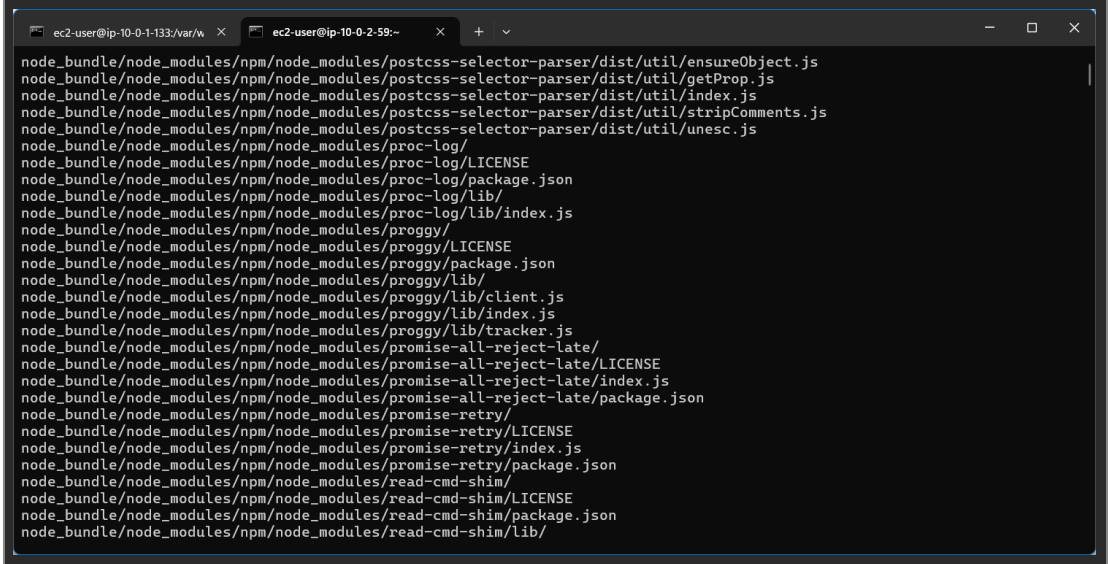
```
[ec2-user@ip-10-0-1-133 ~]$ sudo npm install -g pm2

changed 134 packages in 12s

13 packages are looking for funding
  run `npm fund` for details
```

mkdir ~/node_bundle

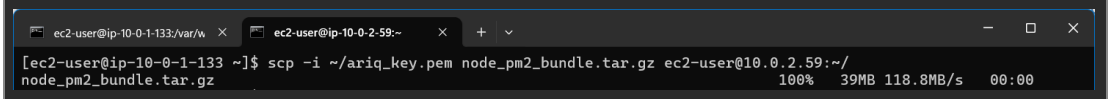
cp -r /usr/bin/node /usr/bin/npm /usr/bin/pm2 ~/node_bundle/

cp -r /usr/lib/node_modules ~/node_bundle/

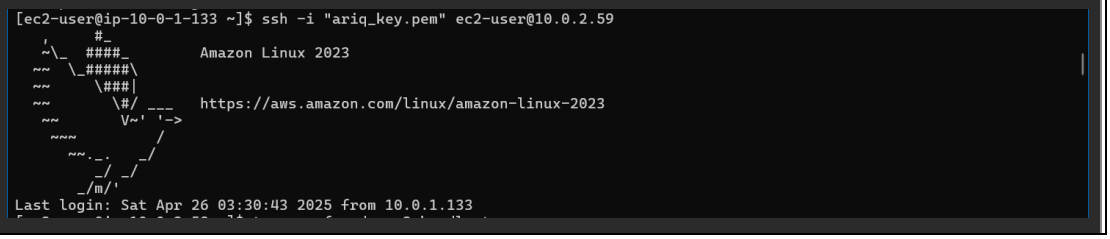
tar -czvf node_pm2_bundle.tar.gz -C ~/ node_bundle



scp -i key.pem node_pm2_bundle.tar.gz ec2-user@ipv4_private:~/



ssh -i key.pem ec2-user@ipv4_private



```
tar -xzvf node_pm2_bundle.tar.gz
```

```
[ec2-user@ip-10-0-2-59 ~]$ tar -xzvf node_pm2_bundle.tar.gz
node_bundle/
node_bundle/node
node_bundle/npm
node_bundle/pm2
node_bundle/node_modules/
node_bundle/node_modules/pm2/
node_bundle/node_modules/pm2/.gitattributes
node_bundle/node_modules/pm2/LICENSE
node_bundle/node_modules/pm2/pm2
node_bundle/node_modules/pm2/.mocharc.js
node_bundle/node_modules/pm2/constants.js
node_bundle/node_modules/pm2/index.js
node_bundle/node_modules/pm2/paths.js
node_bundle/node_modules/pm2/preinstall.js
node_bundle/node_modules/pm2/package.json
node_bundle/node_modules/pm2/bun.lock
node_bundle/node_modules/pm2/CHANGELOG.md
node_bundle/node_modules/pm2/CONTRIBUTING.md
node_bundle/node_modules/pm2/README.md
node_bundle/node_modules/pm2/run.sh
node_bundle/node_modules/pm2/GNU-AGPL-3.0.txt
node_bundle/node_modules/pm2/node_modules/
node_bundle/node_modules/pm2/node_modules/ansi-colors/
node_bundle/node_modules/pm2/node_modules/ansi-colors/LICENSE
node_bundle/node_modules/pm2/node_modules/ansi-colors/index.js
node_bundle/node_modules/pm2/node_modules/ansi-colors/symbols.js
node_bundle/node_modules/pm2/node_modules/ansi-colors/package.json
node_bundle/node_modules/pm2/node_modules/ansi-colors/README.md
```

```
echo 'export PATH=$HOME/node_bundle:$PATH' >> ~/.bashrc

source ~/.bashrc

(EXIT INSTANCE BE, KEMBALI KE FE)

cd /usr/lib/node_modules/pm2

tar czf ~/pm2_bundle.tar.gz .

scp -i ~/ariq_key.pem ~/pm2_bundle.tar.gz ec2-user@10.0.2.59:~/
```

```
[ec2-user@ip-10-0-2-59 ~]$
logout
Connection to 10.0.2.59 closed.
[ec2-user@ip-10-0-1-133 ~]$ cd /usr/lib/node_modules/pm2
[ec2-user@ip-10-0-1-133 pm2]$ tar czf ~/pm2_bundle.tar.gz .
[ec2-user@ip-10-0-1-133 pm2]$ scp -i ~/ariq_key.pem ~/pm2_bundle.tar.gz ec2-user@10.0.2.59:~/
pm2_bundle.tar.gz                                             100% 4919KB 116.4MB/s   00:00
```

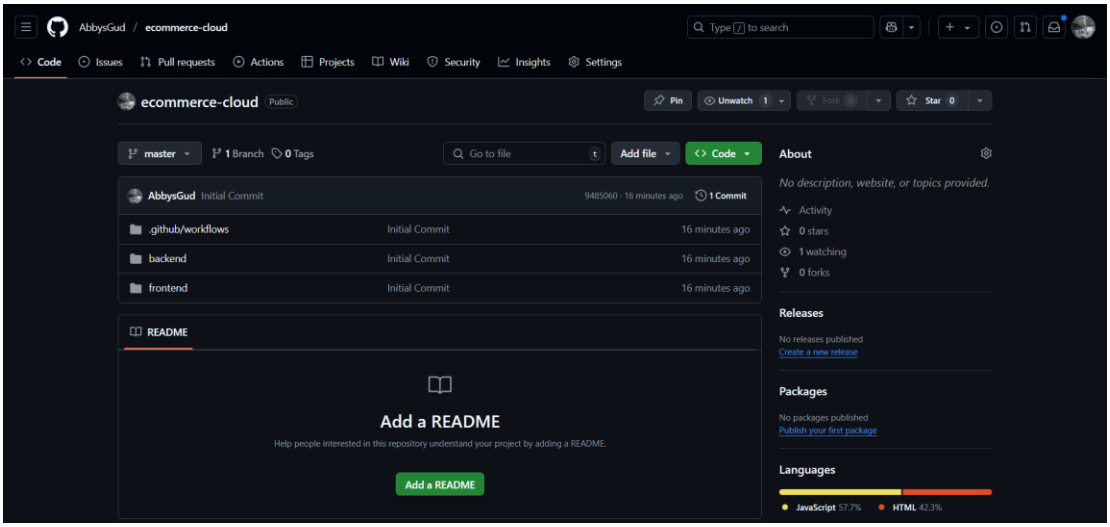
```
rm -rf ~/node_bundle/pm2

mkdir -p ~/node_bundle/pm2

tar xzf ~/pm2_bundle.tar.gz -C ~/node_bundle/pm2

node ~/node_bundle/pm2/bin/pm2 -v
```

```
[ec2-user@ip-10-0-2-59 ~]$ rm -rf ~/node_bundle/pm2
[ec2-user@ip-10-0-2-59 ~]$ mkdir -p ~/node_bundle/pm2
[ec2-user@ip-10-0-2-59 ~]$ tar xzf ~/pm2_bundle.tar.gz -C ~/node_bundle/pm2
[ec2-user@ip-10-0-2-59 ~]$ node ~/node_bundle/pm2/bin/pm2 -v

                      -------------

__/\\\\\\\\\\\\\____/\\\\____________/\\\\____/\\\\\\\\\_____
 _\/\\\/////////\\\_\/\\\\\\________/\\\\\\__/\\\///////\\\___
  _\/\\\_______\/\\\_\/\\\//\\\____/\\\//\\\_\///______\//\\\__
   _\/\\\\\\\\\\\\\/__\/\\\\///\\\/\\\/_\/\\\___________/\\\/___
    _\/\\\/////////____\/\\\__\///\\\/___\/\\\________/\\\//_____
     _\/\\\_____________\/\\\____\///_____\/\\\_____/\\\//________
      _\/\\\_____________\/\\\_____________\/\\\___/\\\/___________
       _\/\\\_____________\/\\\_____________\/\\\__/\\\\\\\\\\\\\\\_
        _\///______________\///______________\///__\///////////////__


                      Runtime Edition

       PM2 is a Production Process Manager for Node.js applications
                    with a built-in Load Balancer.

              Start and Daemonize any application:
              $ pm2 start app.js

              Load Balance 4 instances of api.js:
              $ pm2 start api.js -i 4
```

```
echo "alias pm2='node ~/node_bundle/pm2/bin/pm2'" >> ~/.bashrc
source ~/.bashrc
```

```
[ec2-user@ip-10-0-2-59 ~]$ echo "alias pm2='node ~/node_bundle/pm2/bin/pm2'" >> ~/.bashrc
[ec2-user@ip-10-0-2-59 ~]$ source ~/.bashrc
```

b. Selanjutnya siapkan repository yang sudah berisi program web untuk frontend dan backendnya. Disini digunakan html native untuk frontend, dan node.js untuk backend



c. Selanjutnya clone repository ke EC2 Frontend.

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

```
[ec2-user@ip-10-0-1-133 ~]$ git clone https://github.com/AbbysGud/ecommerce-cloud.git
Cloning into 'ecommerce-cloud'...
remote: Enumerating objects: 963, done.
remote: Counting objects: 100% (963/963), done.
remote: Compressing objects: 100% (733/733), done.
remote: Total 963 (delta 157), reused 963 (delta 157), pack-reused 0 (from 0)
Receiving objects: 100% (963/963), 1008.00 KiB | 17.68 MiB/s, done.
Resolving deltas: 100% (157/157), done.
```

d. Kirim folder frontend ke /var/www/html

```
[ec2-user@ip-10-0-1-133 ~]$ sudo cp -r ~/ecommerce-cloud/* /var/www/html/
[ec2-user@ip-10-0-1-133 ~]$ ls
ariq_key.pem  ecommerce-cloud
[ec2-user@ip-10-0-1-133 ~]$ cd /var/www/html
[ec2-user@ip-10-0-1-133 html]$ ls
backend  frontend
```

e. Tambahkan kofigurasi di httpd.conf

```
LoadModule proxy_module modules/mod_proxy. so
LoadModule proxy_http_module modules/mod_proxy_http.so

ProxyRequests Off

<Proxy *>
    Require all granted
</Proxy>

ProxyPass /api http://ipv4_private:3000
ProxyPassReverse /api http://ipv4_private:3000
```

```
ec2-user@ip-10-0-1-133:~   ec2-user@ip-10-0-2-59:~/back
 GNU nano 8.3                          /etc/httpd/conf/httpd.conf                          Modified
# 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

LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so

# Proxy to backend API
ProxyRequests Off
<Proxy *>
    Require all granted
</Proxy>

ProxyPass /api http://10.0.2.59:3000
ProxyPassReverse /api http://10.0.2.59:3000

File Name to Write: /etc/httpd/conf/httpd.conf
^G Help          M-D DOS Format      M-A Append          M-B Backup File
^C Cancel        M-M Mac Format      M-P Prepend         ^T Browse
```

f. Kirim backend ke EC2 Backend menggunakan scp.

```
cd repo

scp -r -i key.pem backend ec2-user@ipv4_private:~/
```

```
ec2-user@ip-10-0-1-133:~
[ec2-user@ip-10-0-1-133 ~]$ cd ecommerce-cloud
[ec2-user@ip-10-0-1-133 ecommerce-cloud]$ scp -r -i ~/ariq_key.pem backend ec2-user@10.0.2.59:~/
.env                                                                                    100%  120   172.1KB/s   00:00
index.js                                                                                100% 1091     1.1MB/s   00:00
mime                                                                                    100%  371   339.6KB/s   00:00
mime.cmd                                                                                100%  299   293.5KB/s   00:00
mime.ps1                                                                                100%  769     1.7MB/s   00:00
.package-lock.json                                                                      100%   38KB  21.9MB/s   00:00
HISTORY.md                                                                              100% 5096    12.3MB/s   00:00
LICENSE                                                                                 100% 1167     3.1MB/s   00:00
README.md                                                                               100% 4123    11.3MB/s   00:00
index.js                                                                                100% 5252     7.2MB/s   00:00
package.json                                                                            100% 1157     2.3MB/s   00:00
LICENSE                                                                                 100% 1103     1.7MB/s   00:00
README.md                                                                               100% 1245     1.6MB/s   00:00
array-flatten.js                                                                        100% 1195     1.1MB/s   00:00
package.json                                                                            100%  879     2.3MB/s   00:00
HISTORY.md                                                                              100%   16KB  25.3MB/s   00:00
LICENSE                                                                                 100% 1172     1.2MB/s   00:00
README.md                                                                               100%   19KB  25.9MB/s   00:00
SECURITY.md                                                                             100% 1193     2.5MB/s   00:00
index.js                                                                                100% 2681     7.1MB/s   00:00
read.js                                                                                 100% 4325    10.2MB/s   00:00
json.js                                                                                 100% 5299     5.4MB/s   00:00
raw.js                                                                                  100% 1884     2.7MB/s   00:00
text.js                                                                                 100% 2285     5.6MB/s   00:00
urlencoded.js                                                                           100% 6404    15.6MB/s   00:00
package.json                                                                            100% 1472     2.3MB/s   00:00
History.md                                                                              100% 1775     4.3MB/s   00:00
LICENSE                                                                                 100% 1153     2.7MB/s   00:00
```

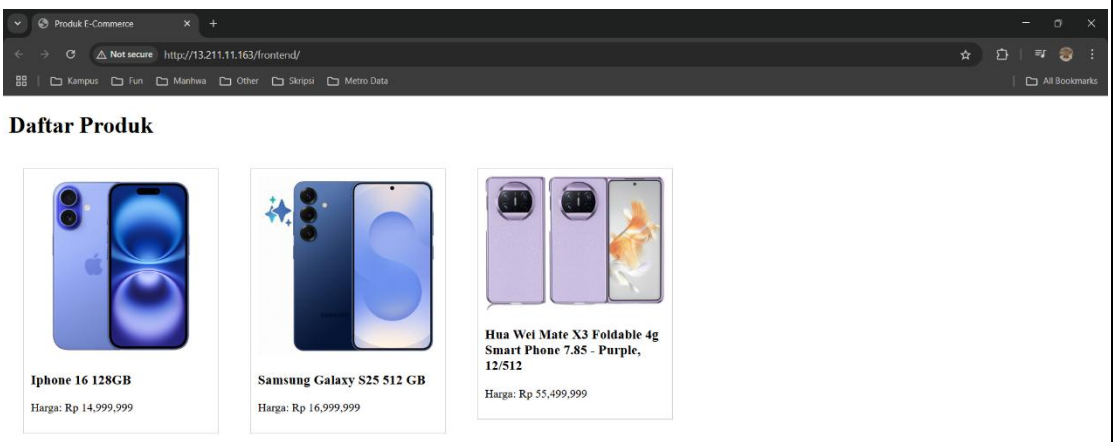
g. Selanjutnya di instance EC2 Backend mulai node.js.

```
cd backend

pm2 start index.js
```

```
[ec2-user@ip-10-0-2-59 ~]$ cd backend
[ec2-user@ip-10-0-2-59 backend]$ pm2 start index.js
[PM2] Starting /home/ec2-user/backend/index.js in fork_mode (1 instance)
[PM2] Done.
```

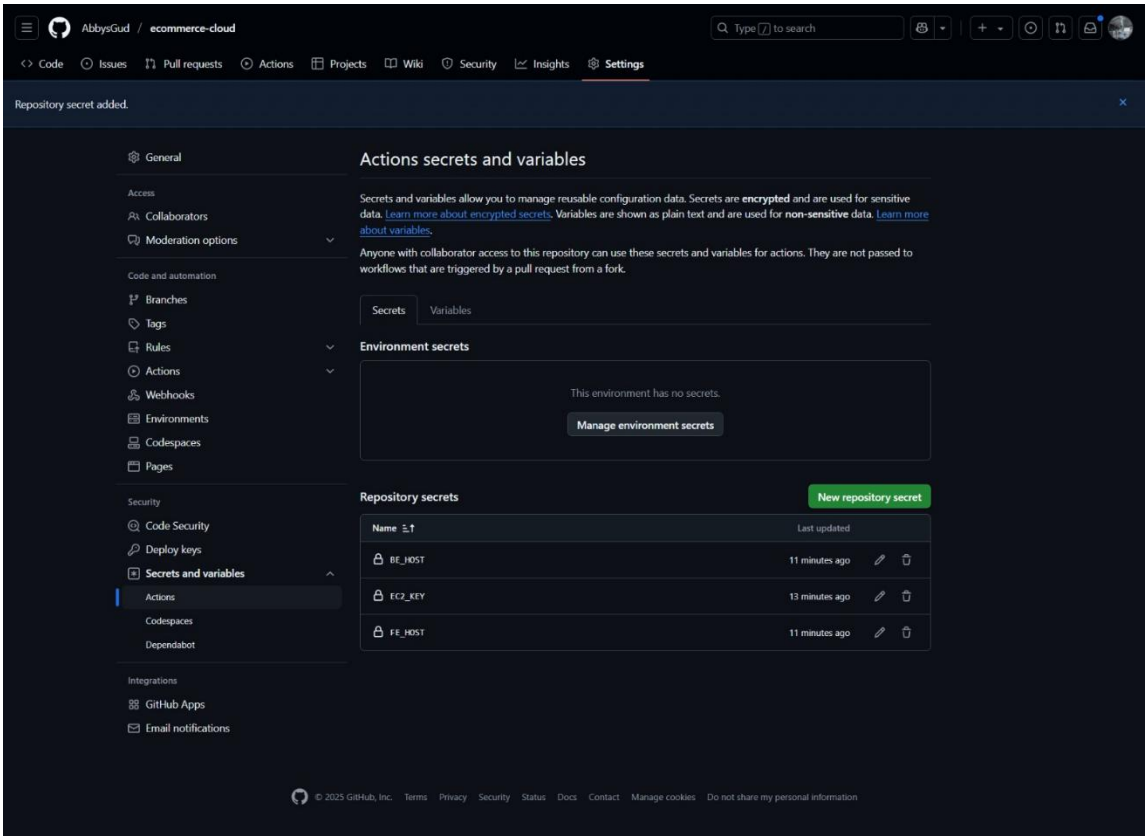
id	name	mode	↺	status	cpu	memory
0	index	fork	0	online	0%	22.4mb

h. Periksa apakah berhasil atau tidak, jika berhasil maka harusnya ada gambar yang diambil dari Bucket S3, dan ada data produk yang diambil dari RDS Database.



Bagian 4 - Implementasi CI/CD dengan GitHub Actions

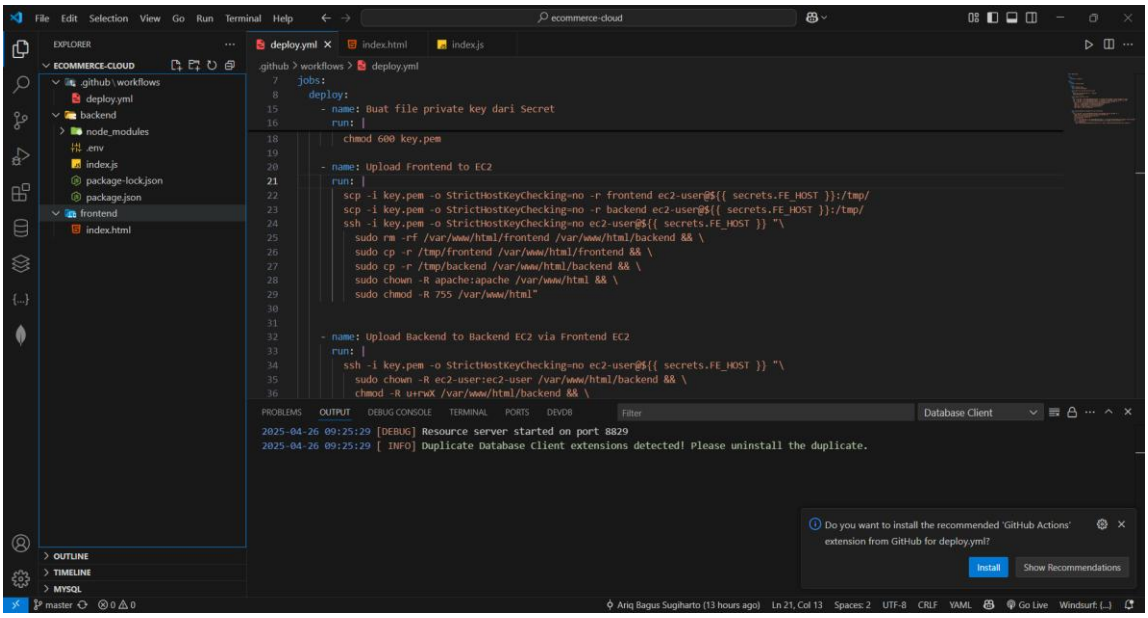
1. Tambahkan Secret pada Repository Github, disini ditambahkan **BE_HOST** yaitu IP private BE, **FE_HOST** yaitu IP public FE, dan **EC2_KEY** yaitu key yang digunakan di instance.



2. Di repository lokal, tambahkan direktori .github/workflows/, didalamnya tambahkan file deploy.yml untuk melakukan CICD setiap push perubahan di lokal ke dua instance EC2.

```
name: Deploy App

on:
  push:
    branches: [ master ]

jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
    - name: Checkout repo
      uses: actions/checkout@v3

    - name: Buat file private key dari Secret
      run: |
        echo "${{ secrets.EC2_KEY }}" > key.pem
```

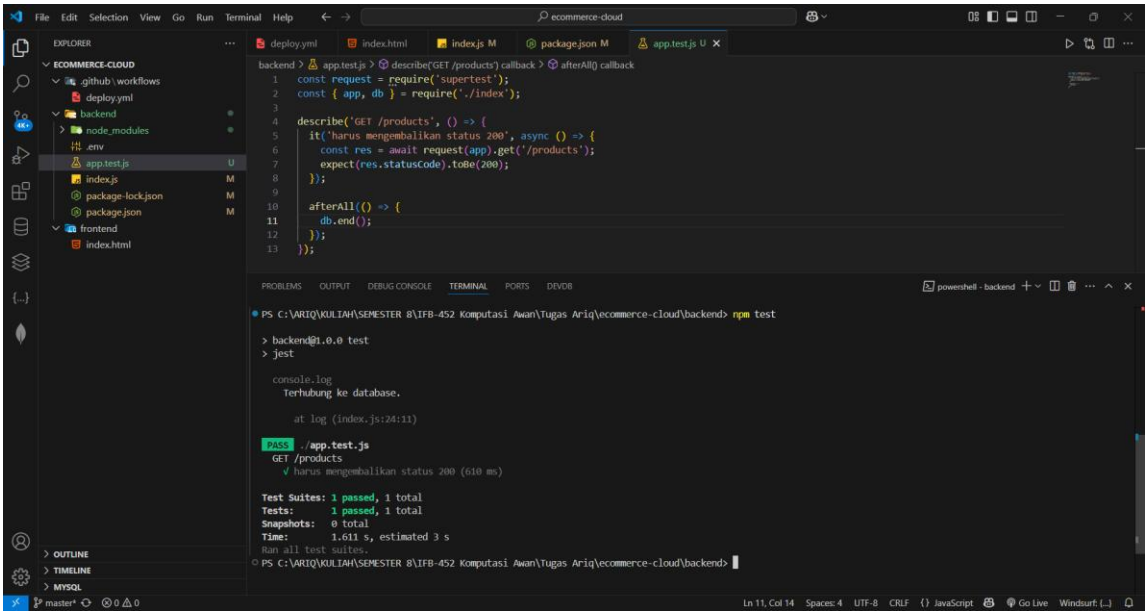
```
        chmod 600 key.pem

    - name: Upload Frontend to EC2
      run: |
        scp -i key.pem -o StrictHostKeyChecking=no -r frontend ec2-user@${{
secrets.FE_HOST }}:/tmp/
        scp -i key.pem -o StrictHostKeyChecking=no -r backend ec2-user@${{
secrets.FE_HOST }}:/tmp/
        ssh -i key.pem -o StrictHostKeyChecking=no ec2-user@${{
secrets.FE_HOST }} "\
          sudo rm -rf /var/www/html/frontend /var/www/html/backend && \
          sudo cp -r /tmp/frontend /var/www/html/frontend && \
          sudo cp -r /tmp/backend /var/www/html/backend && \
          sudo chown -R apache:apache /var/www/html && \
          sudo chmod -R 755 /var/www/html"


    - name: Upload Backend to Backend EC2 via Frontend EC2
      run: |
        ssh -i key.pem -o StrictHostKeyChecking=no ec2-user@${{
secrets.FE_HOST }} "\
          sudo chown -R ec2-user:ec2-user /var/www/html/backend && \
          chmod -R u+rwX /var/www/html/backend && \
          cd /var/www/html/backend && \
          npm install && \
          scp -i ~/ariq_key.pem -o StrictHostKeyChecking=no -r
/var/www/html/backend ec2-user@${{ secrets.BE_HOST }}:~/ && \
          ssh -i ~/ariq_key.pem -o StrictHostKeyChecking=no ec2-user@${{
secrets.BE_HOST }} '\
            cd ~/backend && \
            node ~/node_bundle/pm2/bin/pm2 restart all || node
~/node_bundle/pm2/bin/pm2 start index.js'"
```



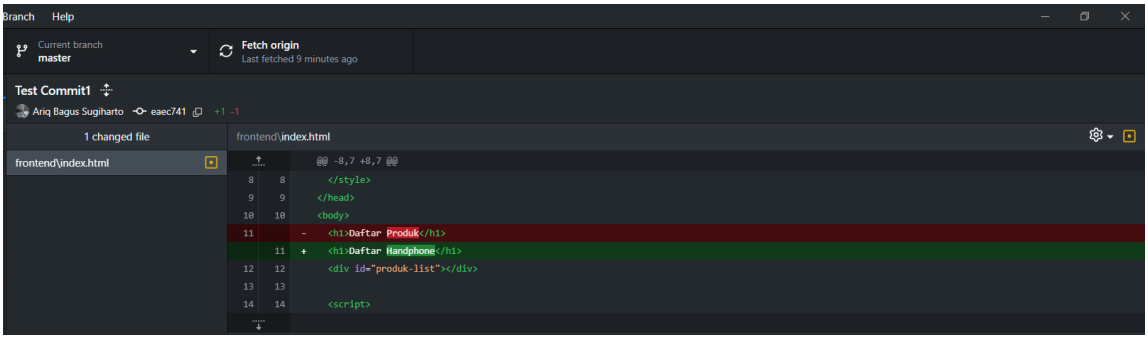
3. Selanjutnya tambahkan app.test.js di backend anda untuk testing terlebih dahulu apakah ada error atau tidak sebelum di CICD, selain itu juga tambahkan script didalam action.

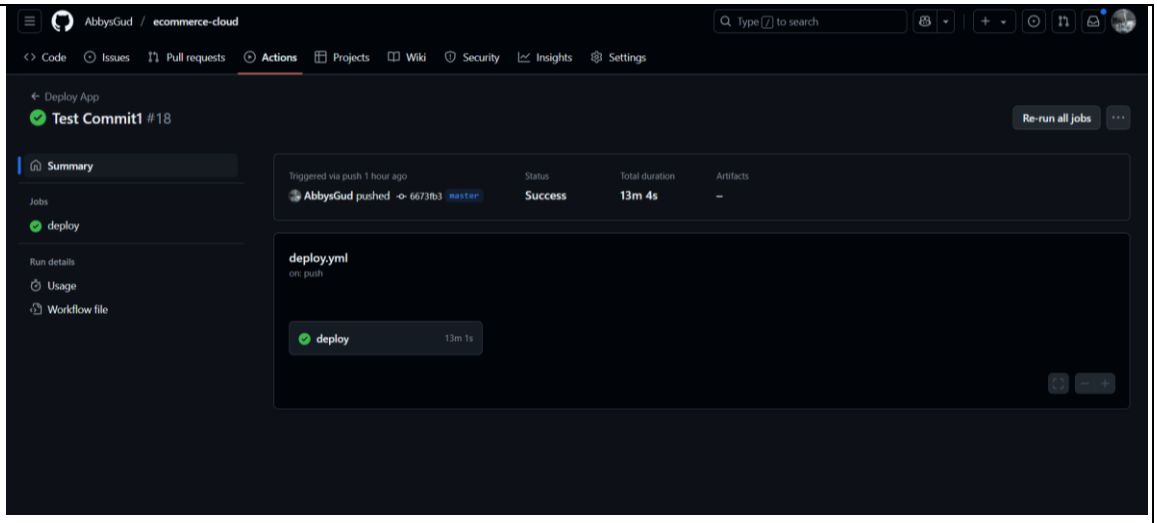
```
    - name: Install dependencies
      run: |
        cd backend
        npm ci

    - name: Run tests
      run: |
        cd backend
        npm run test
```

Contoh test berhasil:

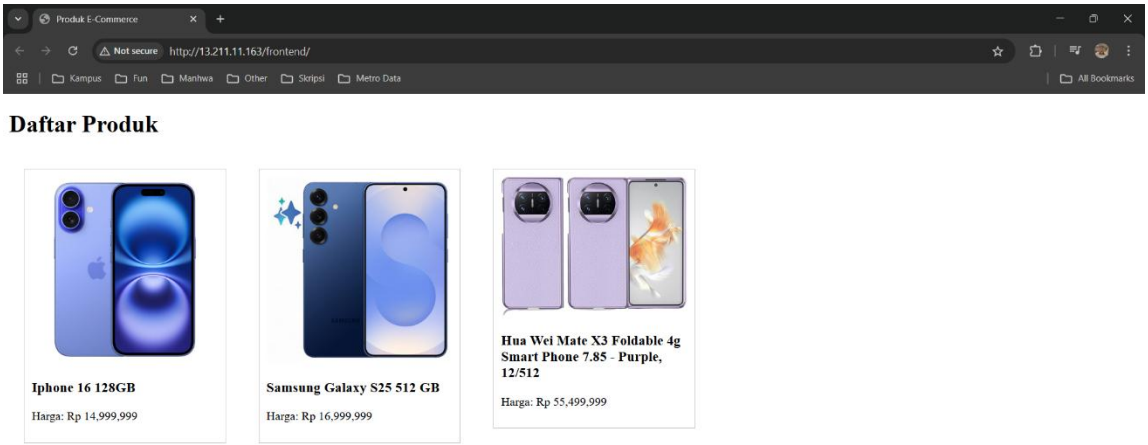


4. Lakukan perubahan di lokal, misalnya mengganti Daftar Produk menjadi Daftar Handphone, Commit dan Push perubahan tersebut dan periksa di repository pada Action untuk melihat hasil CICD apakah berhasil atau tidak.

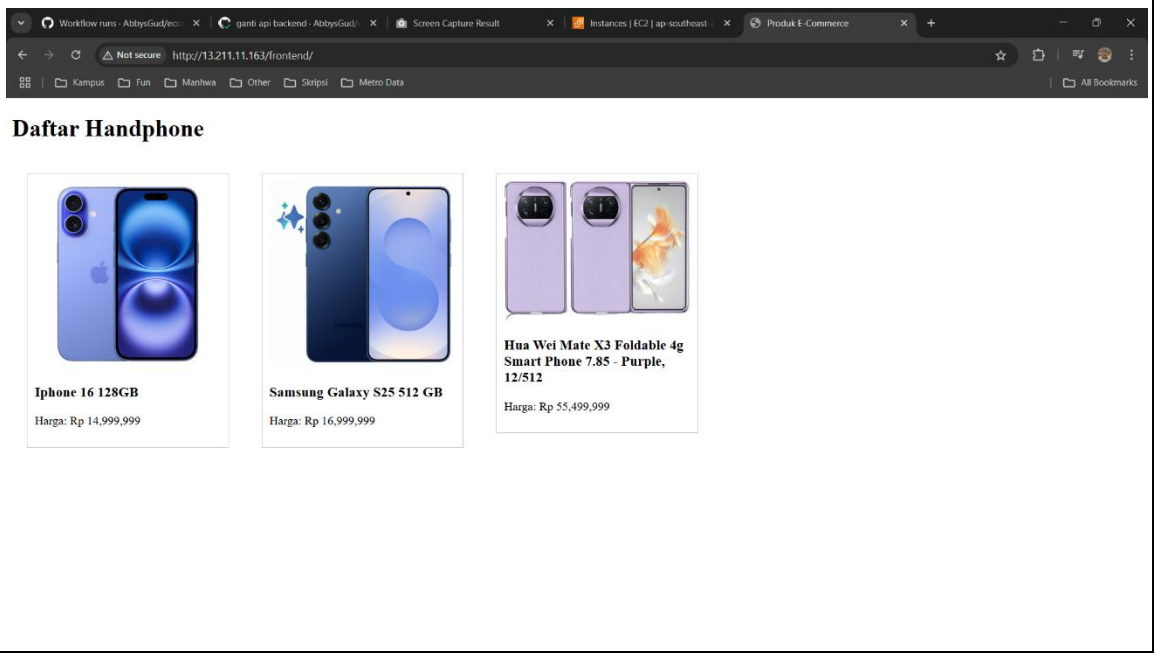



AbbysGud / ecommerce-cloud
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Deploy App
Test Commit1 #18
Re-run all jobs
Summary
Jobs
deploy
Run details
Usage
Workflow file
Triggered via push 1 hour ago
AbbysGud pushed 6673fb3 master
Status
Success
Total duration
13m 4s
Artifacts
–
deploy.yml
on: push
deploy
13m 1s

Sebelum:


Produk E-Commerce
Not secure
http://13.211.11.163/frontend/
Kampus
Fun
Manhwa
Other
Skripsi
Metro Data
All Bookmarks
Daftar Produk
Iphone 16 128GB
Harga: Rp 14,999,999
Samsung Galaxy S25 512 GB
Harga: Rp 16,999,999
Hua Wei Mate X3 Foldable 4g Smart Phone 7.85 - Purple, 12/512
Harga: Rp 55,499,999

Sesudah:


Workflow runs - AbbysGud/ecc
ganti api backend - AbbysGud/
Screen Capture Result
Instances | EC2 | ap-southeast
Produk E-Commerce
Not secure
http://13.211.11.163/frontend/
Kampus
Fun
Manhwa
Other
Skripsi
Metro Data
All Bookmarks
Daftar Handphone
Iphone 16 128GB
Harga: Rp 14,999,999
Samsung Galaxy S25 512 GB
Harga: Rp 16,999,999
Hua Wei Mate X3 Foldable 4g Smart Phone 7.85 - Purple, 12/512
Harga: Rp 55,499,999