

Nama : Ariq Bagus Sugiharto

NRP : 15-2021-036

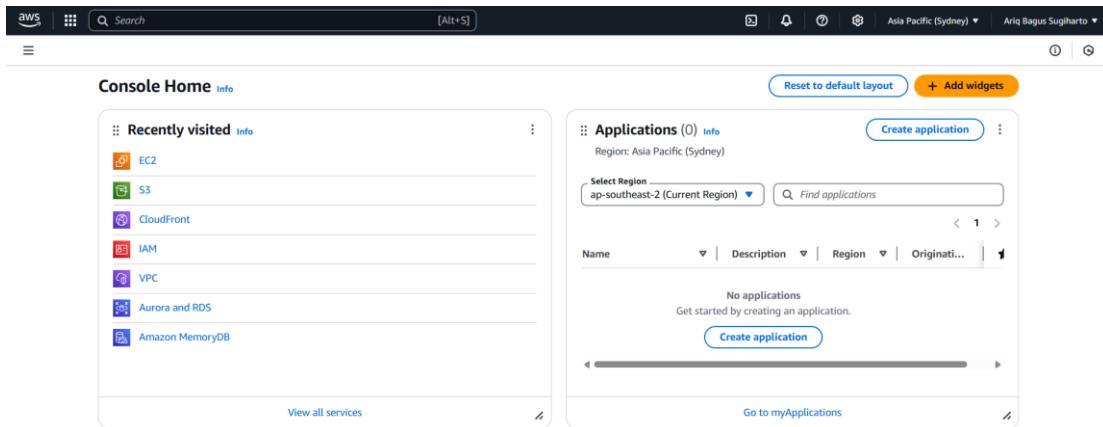
UTS IFB-452 Komputasi Awan

Link repository github: <https://github.com/AbbyGud/ecommerce-cloud.git>

Bagian 1 - Infrastruktur Dasar dengan Amazon EC2 & VPC

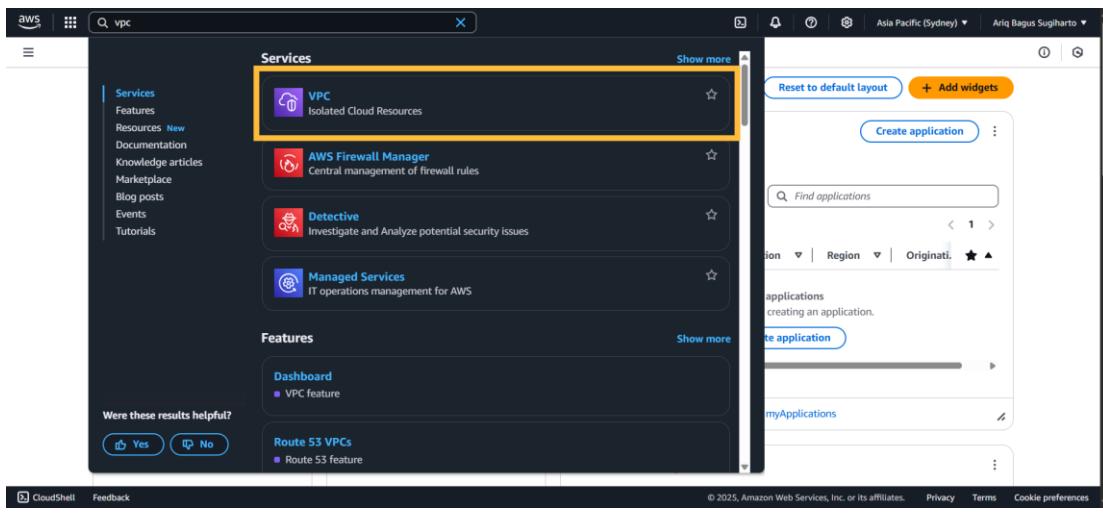
1. Membuat VPC dengan Subnet

- Masuk ke AWS Console.



The screenshot shows the AWS Console Home page. On the left, there's a sidebar with 'Recently visited' services: EC2, S3, CloudFront, IAM, VPC, Aurora and RDS, and Amazon MemoryDB. On the right, the 'Applications' section is displayed, showing a list of applications with a 'Create application' button. The region is set to Asia Pacific (Sydney).

- Cari VPC kemudian buka layanan VPC.



The screenshot shows the AWS search results for 'vpc'. The 'VPC' service is highlighted with a yellow box. Other services listed include AWS Firewall Manager, Detective, and Managed Services. The search bar at the top has 'vpc' typed into it. The region is set to Asia Pacific (Sydney).

c. Pilih Create VPC.

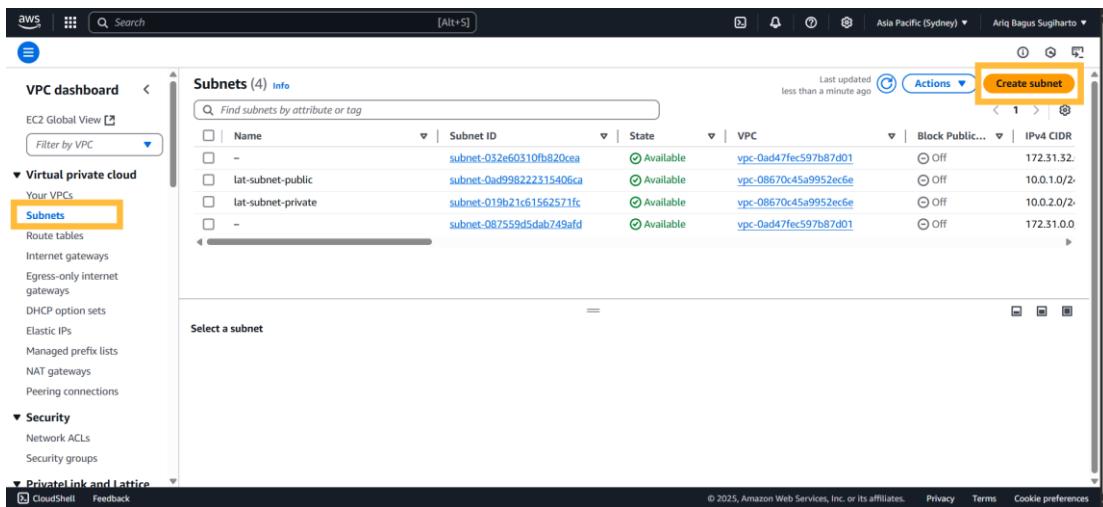
The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with options like 'Virtual private cloud', 'Security', and 'PrivateLink and Lattice'. In the center, under 'Resources by Region', there are several boxes for different VPC components: VPCs, Subnets, Route Tables, Internet Gateways, Egress-only Internet Gateways, NAT Gateways, Security Groups, and Customer Gateways. Each box has a 'See all regions' link. At the top right, there's a 'Create VPC' button, which is highlighted with a yellow box. Below it is another button labeled 'Launch EC2 Instances'. The top right corner shows the user's name 'Ariq Bagus Sugiharto' and the region 'Asia Pacific (Sydney)'. A 'Service Health' and 'Settings' box are also visible on the right side.

- Masukkan nama untuk VPC.
- Tentukan blok CIDR IPv4 10.0.0.0/16.
- Pilih opsi untuk IPv6 jika diperlukan.
- Klik "Create" untuk membuat VPC

The screenshot shows the 'Create VPC' configuration page. At the top, it says 'VPC > Your VPCs > Create VPC'. The main section is titled 'Create VPC' with a 'Info' link. It says 'A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.' Below this is a 'VPC settings' section. Under 'Resources to create', the 'VPC only' option is selected. There's a 'Name tag - optional' field containing 'ecommerce-vpc'. Under 'IPv4 CIDR block', the 'IPV4 CIDR manual input' option is selected, and the CIDR block '10.0.0.0/16' is entered. Under 'IPv6 CIDR block', the 'No IPv6 CIDR block' option is selected. At the bottom, there are 'CloudShell' and 'Feedback' buttons, and a copyright notice for 2025 Amazon Web Services, Inc. or its affiliates.

- Buat 2 Subnet (1 public, 1 private)

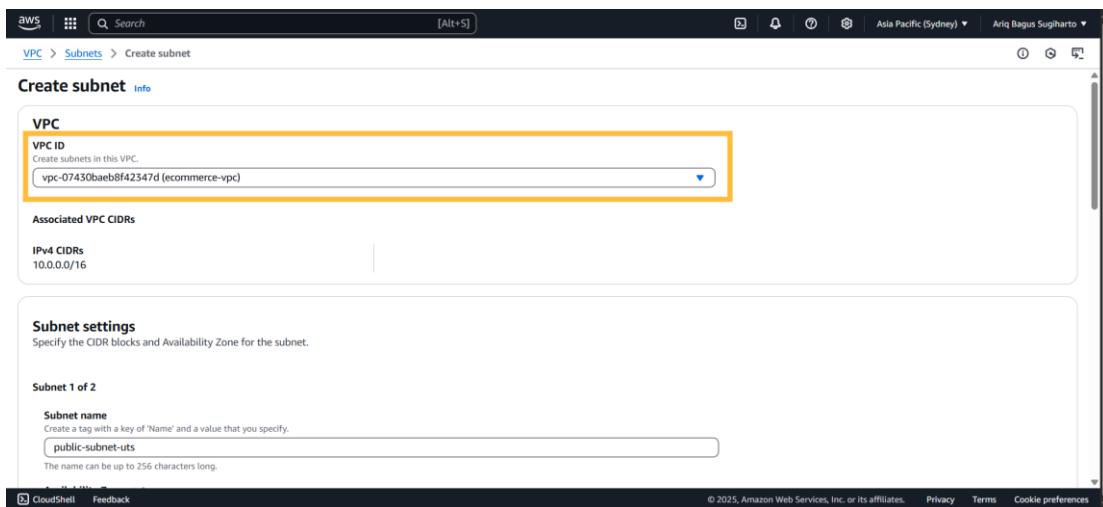
a. Pilih Subnet, kemudian pilih Create Subnet.



The screenshot shows the AWS VPC dashboard with the 'Subnets' section selected. The table lists four subnets:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
-	subnet-032e60310fb820cea	Available	vpc-0ad47fec597b87d01	Off	172.31.32.
lat-subnet-public	subnet-0ad9982221515406ca	Available	vpc-08670c45a9952ec6e	Off	10.0.1.0/24
lat-subnet-private	subnet-019b21c61562571fc	Available	vpc-08670c45a9952ec6e	Off	10.0.2.0/24
-	subnet-087559d5dab749af	Available	vpc-0ad47fec597b87d01	Off	172.31.0.0

b. Pilih VPC yang tadi dibuat.



The screenshot shows the 'Create subnet' wizard. The 'VPC ID' field is highlighted, displaying 'vpc-07450baeb8f42347d (ecommerce-vpc)'. The 'Associated VPC CIDRs' section shows 'IPV4 CIDRs 10.0.0.0/16'. The 'Subnet settings' section shows 'Subnet 1 of 2' with a 'Subnet name' field containing 'public-subnet-uts'.

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**

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

Availability Zone
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
10.0.1.0/24
 256 IPs

Tags - optional
Key Value - optional

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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**

Subnet 2 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

Availability Zone
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
10.0.2.0/24
 256 IPs

Tags - optional
Key Value - optional

You can add 49 more tags.

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

3. Buat Internet Gateway dan Hubungkan ke VPC

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

VPC dashboard

Internet gateways (2) Info

Name	Internet gateway ID	State	VPC ID	Owner
ariq-gateway	igw-0138b207a5df27b42	Attached	vpc-08670c45a9952ec6e ariq-vpc	761018853843
-	igw-070b4d27c573561c	Attached	vpc-0ad47feef597b87d01	761018853843

Select an internet gateway above

Create internet gateway

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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

Create internet gateway [Info](#)
An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="ecommerce-igw"/> Remove

[Add new tag](#)
You can add 49 more tags.

[Cancel](#) [Create internet gateway](#)

- c. Attach ke vpc yang tadi dibuat

VPC dashboard < [Actions](#) [Create internet gateway](#)

Internet gateways (1/3) [Info](#)

Name	Internet gateway ID	State	VPC ID
ariq-gateway	igw-0158b207a5df27b42	Attached	vpc-08670x45a9
-	igw-070b4d527c573561c	Attached	vpc-0ad47fec597
<input checked="" type="checkbox"/> ecommerce-igw	igw-060354d5f8729f9cc	Detached	-

[View details](#) [Detach from VPC](#) [Manage tags](#) [Delete internet gateway](#)

igw-060354d5f8729f9cc / ecommerce-igw

[Details](#) [Tags](#)

Details

Internet gateway ID	State	VPC ID	Owner
igw-060354d5f8729f9cc	Detached	-	761018853843

[CloudShell](#) [Feedback](#)

Attach to VPC (igw-060354d5f8729f9cc) [Info](#)

VPC
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

[AWS Command Line Interface command](#)

[Cancel](#) [Attach internet gateway](#)

[CloudShell](#) [Feedback](#)

4. Buat Route Table dan Routing

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

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected in the sidebar. The main area displays a table of existing route tables, each with columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, and VPC. A prominent orange box highlights the 'Create route table' button at the top right of the table.

b. Masukkan Nama dan Pilih VPC yang telah dibuat

The screenshot shows the 'Create route table' wizard. In the 'Route table settings' step, the 'Name' field is set to 'public-route-table' and the 'VPC' dropdown is set to 'vpc-07430baeb8f42347d (ecommerce-vpc)'. The 'Tags' section is also visible, showing a single tag 'public-route-table' attached to the route table.

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

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. It displays a table of route tables, with one specific route table ('rtb-0e1e167935ec0036f / public-route-table') selected and its details shown in the main pane. The 'Routes' tab is active, showing a table of routes with columns for Destination, Target, Status, and Propagated. An orange box highlights the 'Edit routes' button at the top right of the routes table.

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

The screenshot shows the 'Edit routes' page for a specific route table. A new route entry is being added:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway		No
	igw-060354d5f8729f9cc		

Buttons at the bottom include 'Add route', 'Cancel', 'Preview', and a highlighted 'Save changes' button.

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

The screenshot shows the 'Route tables (1/6) info' page. The 'public-route-table' is selected. Below it, the 'rtb-0e1e167935ec0036f / public-route-table' details are shown, including the 'Subnet associations' tab which is currently active. An 'Edit subnet associations' button is highlighted.

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

The screenshot shows the 'Edit subnet associations' page for the selected route table. In the 'Available subnets' section, the 'public-subnet-uts' subnet is selected and highlighted. In the 'Selected subnets' section, the same subnet is listed with a 'Save associations' button highlighted.

5. Buat Security Group

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

The screenshot shows the AWS VPC dashboard with the 'Security groups' section selected. A table lists 10 security groups, each with a Name, Security group ID, Security group name, VPC ID, and Description. The 'Create security group' button at the top right is highlighted with a yellow box.

Name	Security group ID	Security group name	VPC ID	Description
-	sg-0f3877dae393d948f	ariq-servergroup	vpc-08670c45a9952ec6e	Security Group
-	sg-0bb59d9e049b01597	kisi-uts-security	vpc-0ad47fec597b87d01	security untuk I
-	sg-04f41614791bd8316	default	vpc-08670c45a9952ec6e	default VPC sec
-	sg-0822e7d2df225d5ac	launch-wizard-2	vpc-0ad47fec597b87d01	launch-wizard-
-	sg-0773045686cd367aa	default	vpc-07430baeb8f42347d	default VPC sec
-	sg-0d727227fee561770	default	vpc-0ad47fec597b87d01	default VPC sec
-	sg-0e73ecee1819fd5ce	launch-wizard-1	vpc-0ad47fec597b87d01	launch-wizard-

- Masukkan Nama Security Group, misal **ecommerce-sg**
- Masukkan Deskripsi untuk Security Groupnya
- Pilih VPC yang telah dibuat
- Tambahkan Inbound Rules yaitu **SSH (22)** dengan sourceny **My IP** dan **HTTP (80)** dengan sourceny **Anywhere** (untuk be ssh-nya harus dari ip security group fe, karena akses instancenya dari fe)

The screenshot shows the 'Create security group' wizard. The 'Basic details' section includes fields for 'Security group name' (set to 'ecommerce-sg'), 'Description' (set to 'Security Group untuk UTS Komputasi Awan'), and 'VPC' (set to 'vpc-07430baeb8f42347d (ecommerce-vpc)'). The 'Inbound rules' section contains two rules: one for SSH (port 22) from 'My IP' and one for HTTP (port 80) from 'Anywhere'.

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	My IP	180.244.135.221/32
HTTP	TCP	80	Anywhere	0.0.0.0/0

6. Launch EC2 Instances

a. Cari EC2 lalu buka EC2.

The screenshot shows the AWS search results for 'ec2'. The 'EC2' service card is highlighted with a yellow box. The service card includes the title 'EC2 Virtual Servers in the Cloud' and a brief description: 'A managed service to automate build, customize and deploy OS images'. Below the service cards, there's a 'Features' section with a 'Dashboard' card. At the bottom, there are 'Were these results helpful?' buttons ('Yes' and 'No') and a feedback link.

b. Pilih Launch Instances

The screenshot shows the AWS EC2 landing page. The main heading is 'Amazon Elastic Compute Cloud (EC2) Create, manage, and monitor virtual servers in the cloud.' A call-to-action button 'Launch instance' is highlighted with a yellow box. To the right, there's a 'Get started' section with links to 'Get started walkthroughs' and 'Get started tutorial'.

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]

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

Recent | **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-0d77805eeda7a2e9 (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	Boot mode	AMI ID	Publish Date	Username	Verified provider
64-bit (x86)	uefi-preferred	ami-0f6a1a6507c55c9a8	2025-04-11	ec2-user	Verified provider

Instance type Info | Get advice

t2.micro 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: 7610188535845 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](#) Enabled 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-00x6acbdff28820d36 VPC: vpc-07430baeb8f42347d [Compare security group rules](#)

Advanced network configuration

Configure storage Info

Advanced 1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted [Edit](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage [X](#)

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

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

[ec2-user@ip-10-0-1-133 ~] + | Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ariq Bagus Sugiharto>cd C:\ARIQ\KULIAH\SEMESTER 8\IFB-452 Komputasi Awan

C:\ARIQ\KULIAH\SEMESTER 8\IFB-452 Komputasi Awan>ssh -i "ariq_key.pem" ec2-user@13.239.111.64
The authenticity of host '13.239.111.64 (13.239.111.64)' can't be established.
ED25519 key fingerprint is SHA256:P6FsMUHN6SPYyfK+mRJt1LIab92p8rpLXhp+EEM/Ck.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.239.111.64' (ED25519) to the list of known hosts.

[ec2-user@ip-10-0-1-133 ~]$ |
```

e. Cek apakah terhubung ke internet menggunakan command berikut:

```
sudo yum update -y
```

```
[ec2-user@ip-10-0-1-133 ~]$ sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-1-133 ~]$ |
```

f. Connect ke back-end dengan command berikut:

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

```
C:\ARIQ\KULIAH\SEMESTER 8\IFB-452 Komputasi Awan>scp -i "ariq_key.pem" ariq_key.pem ec2-user@13.239.111.64:/home/ec2-user/
[ec2-user@ip-10-0-1-133 ~]$ ariq_key.pem | 100% 1678 13.8KB/s 00:00
```

```
ls key.pem
```

```
[ec2-user@ip-10-0-1-133 ~]$ ls ariq_key.pem
```

```
chmod 400 key.pem
```

```
ls -l key.pem
```

```
[ec2-user@ip-10-0-1-133 ~]$ chmod 400 ariq_key.pem
[ec2-user@ip-10-0-1-133 ~]$ ls -l ariq_key.pem
-r-----. 1 ec2-user ec2-user 1678 Apr 24 10:23 ariq_key.pem
```

```
ssh -i "key.pem" ec2-user@private_ip_address
```

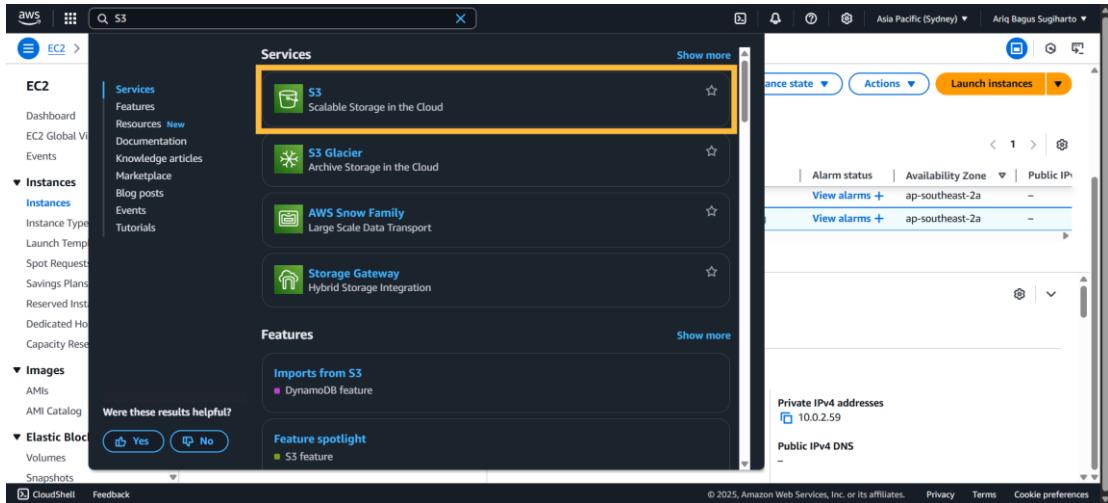
```
[ec2-user@ip-10-0-1-133 ~]$ ssh -i "ariq_key.pem" ec2-user@10.0.2.59
The authenticity of host '10.0.2.59 (10.0.2.59)' can't be established.
ED25519 key fingerprint is SHA256:WM0oOsUQCYMWOnoFR/NcCIBgGEldSVZIsDL22Hb8jSk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.59' (ED25519) to the list of known hosts.

[ec2-user@ip-10-0-2-59 ~]$ |
```

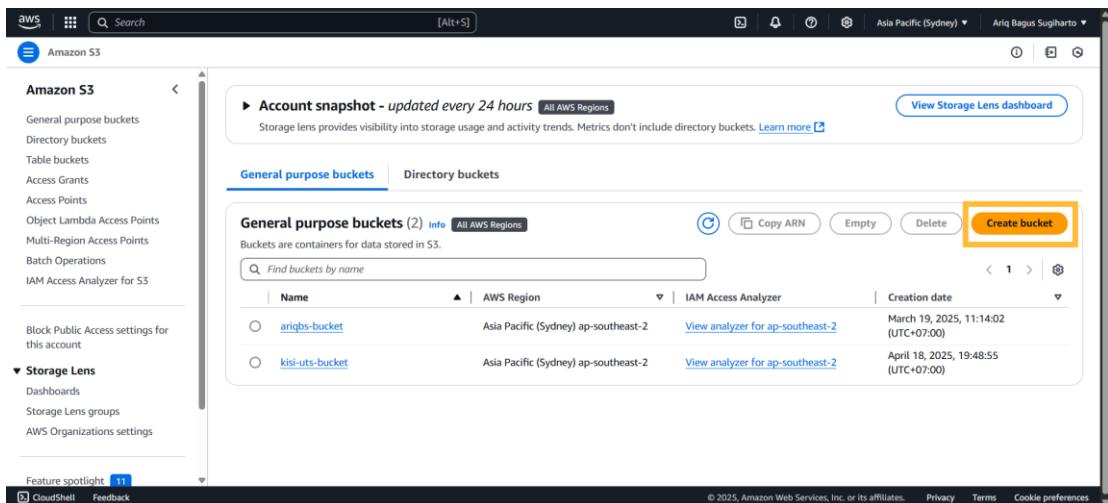
Bagian 2 - Penyimpanan Data dengan (Amazon S3 & Database)

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]

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](#)

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](#)

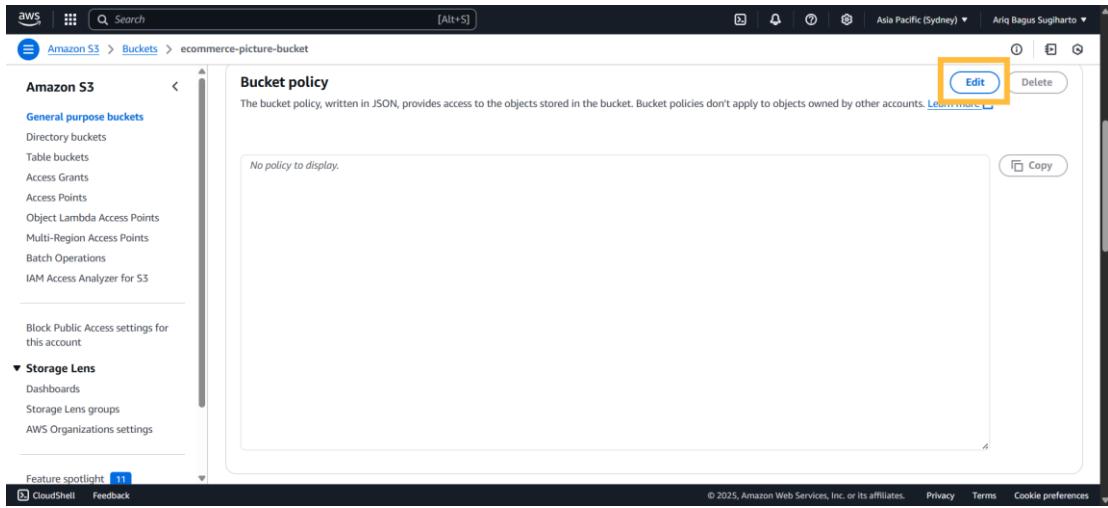
Disable
 Enable

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/*"
    }
  ]
}
```

The screenshot shows the AWS S3 Bucket Policy editor for the bucket 'ecommerce-picture-bucket'. The policy document is displayed in JSON format:

```

1▼ [
2  "Version": "2012-10-17",
3  "Statement": [
4    {
5      "Sid": "PublicReadGetObject",
6      "Effect": "Allow",
7      "Principal": "*",
8      "Action": "s3:GetObject",
9      "Resource": "arn:aws:s3:::ecommerce-product-images/*"
10    }
11  ]
12 ]

```

On the right side, there is a panel titled 'Edit statement' with a sub-section 'Select a statement' containing the message 'Select an existing statement in the policy or add a new statement.' Below it is a button '+ Add new statement'.

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

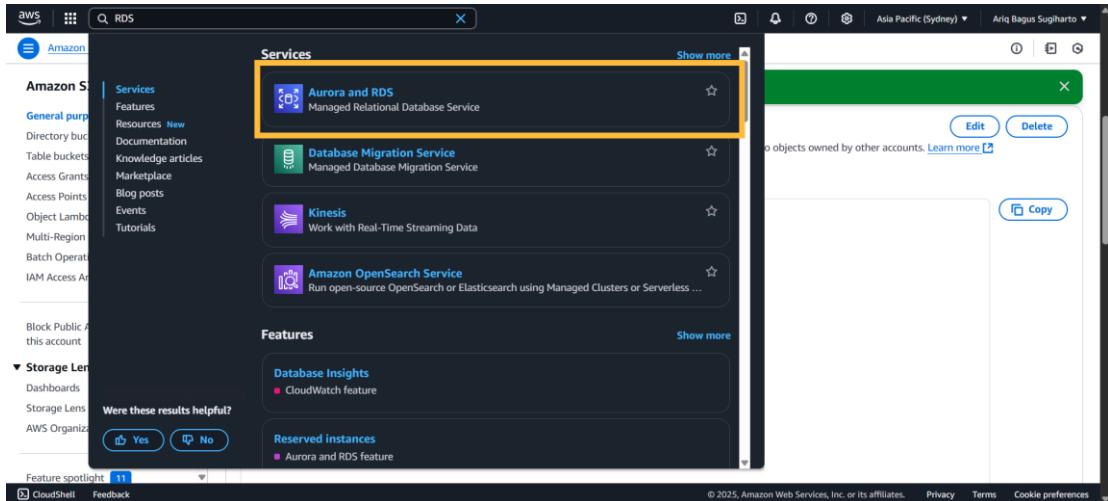
The screenshot shows the AWS S3 Upload interface for the bucket 'ecommerce-picture-bucket'. The 'Files and folders' section lists three files:

Name	Type	Size
samsung_galaxy_s25.jpg	image/jpeg	58.0 KB
iphone_16.jpg	image/jpeg	151.4 KB
huawei_mate.jpg	image/jpeg	105.9 KB

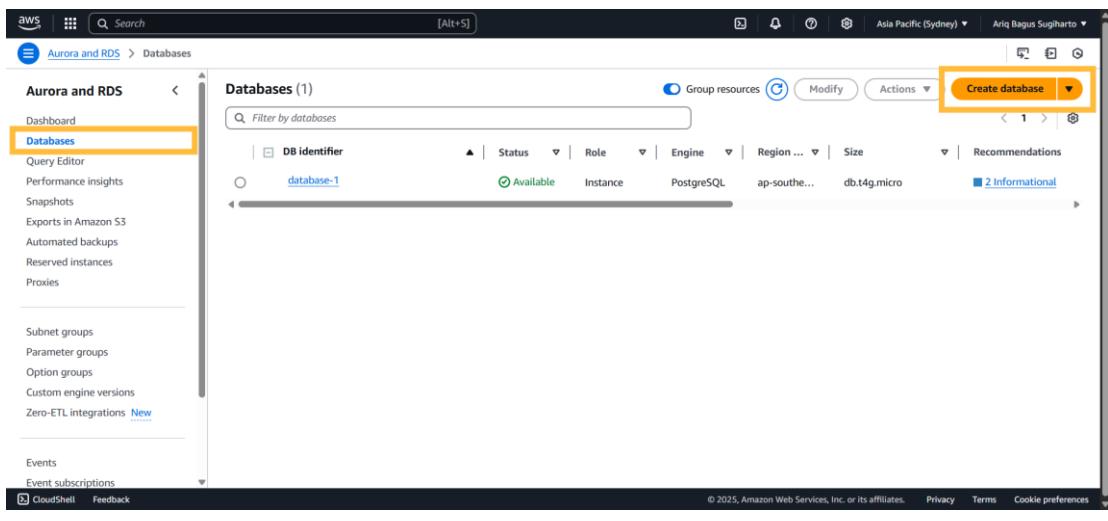
The 'Destination' section shows the destination as 's3://ecommerce-picture-bucket'. The 'Permissions' section indicates public access is granted. The 'Properties' section specifies storage class, encryption settings, and tags. At the bottom right are 'Cancel' and 'Upload' buttons.

2. Buat RDS MySQL

a. Cari RDS lalu pilih RDS



b. Pilih Databases di sidebar, lalu klik Create database



i. Input konfigurasi sesuai berikut:

Create database

Choose a database creation method

- Standard create: Use standard configuration options, including ones for availability, security, backups, and maintenance.
- Easy create: Use simplified configuration options. Some configuration options can be changed after the database is created.

Engine options

Engine type: [MySQL](#) ([MySQL, Compatible](#))

- Aurora (MySQL, Compatible)
- Aurora (PostgreSQL, Compatible)
- MySQL
- PostgreSQL
- MariaDB
- Oracle
- Microsoft SQL Server
- SQL Server

Storage

Storage type: [Standard storage](#) (Amazon EBS volumes are available)

General storage: [100 GB \(100\)](#)

Allocated storage: [20](#)

Allocated storage values range from 20 GB up to 1,000 GB.

Additional storage configuration

Connectivity

Compute resource: [Create a new VPC endpoint for this database. Setting up a connection will automatically change network settings so that the compute resources can connect to this database.](#)

- Don't create an EC2 compute resource
- Create an EC2 compute resource

Network type: [AWS Direct Connect](#)

Virtual private cloud (VPC): [Create a new VPC endpoint for this database. Setting up a connection will automatically change network settings so that the compute resources can connect to this database.](#)

DB subnet group: [Create a new DB subnet group](#)

Public access: [No](#)

DB proxy: [Amazon RDS Optimized Writer](#)

DB security group: [Create a new DB security group](#)

EC2 security group: [Create a new EC2 security group](#)

CloudWatch Metrics: [Create CloudWatch Metrics](#)

CloudWatch Metrics Insights: [Create CloudWatch Metrics Insights](#)

CloudWatch Metrics Insights Insights: [Create CloudWatch Metrics Insights Insights](#)

Availability and durability

Deployment option: [Multi-AZ instance deployment \(3 instances\)](#)

Choose the deployment option that provides the availability and durability needed for your use case. AWS RDS is optimized to a certain level of durability depending on the deployment option you choose. Learn more about deployment options.

- Multi-AZ instance deployment (3 instances): Provides the highest level of durability and availability. Three Amazon RDS instances are used to provide redundancy. This setup provides:
 - Read replicas
 - Write replicas
 - Read-only endpoints
 - Multi-AZ endpoints
- Single-AZ instance deployment (1 instance): Provides a single Amazon RDS instance. This setup provides:
 - Read replicas
 - Read-only endpoints

Settings

DB instance identifier: [aws_rds_staging_db](#)

The DB instance identifier is case-sensitive, but it is stored as lowercase (as in `MyXXXXXXXXXX`). Constraints: 1 to 24 alphanumeric characters or hyphens. First character must be a letter. Can't contain punctuation marks.

Credentials Settings

Master username: [admin](#)

Type a login for the administrator user of your RDS instance.

Administrative privileges: The first character must be a letter.

Credentials management:

- AWS IAM User: [Create a new IAM user for your RDS instance.](#)
- Self-managed: [Create your own password on RDS; create a password that you manage.](#)

Master password: [admin123456](#)

Master password: [admin123456](#)

Confirm master password: [admin123456](#)

Instance configuration

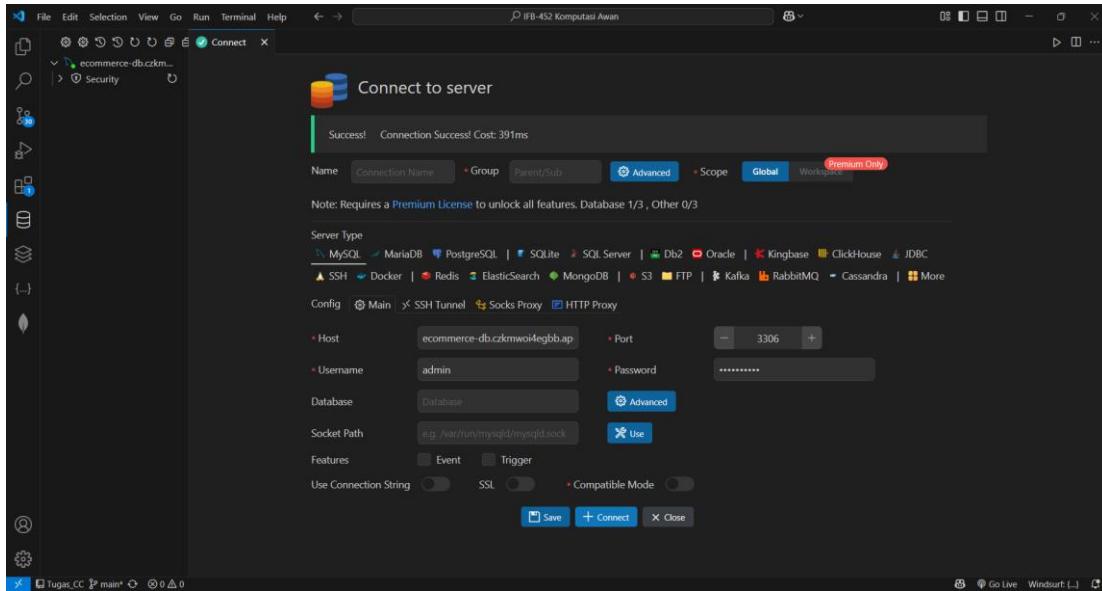
The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class: [db.t3.2xlarge](#)

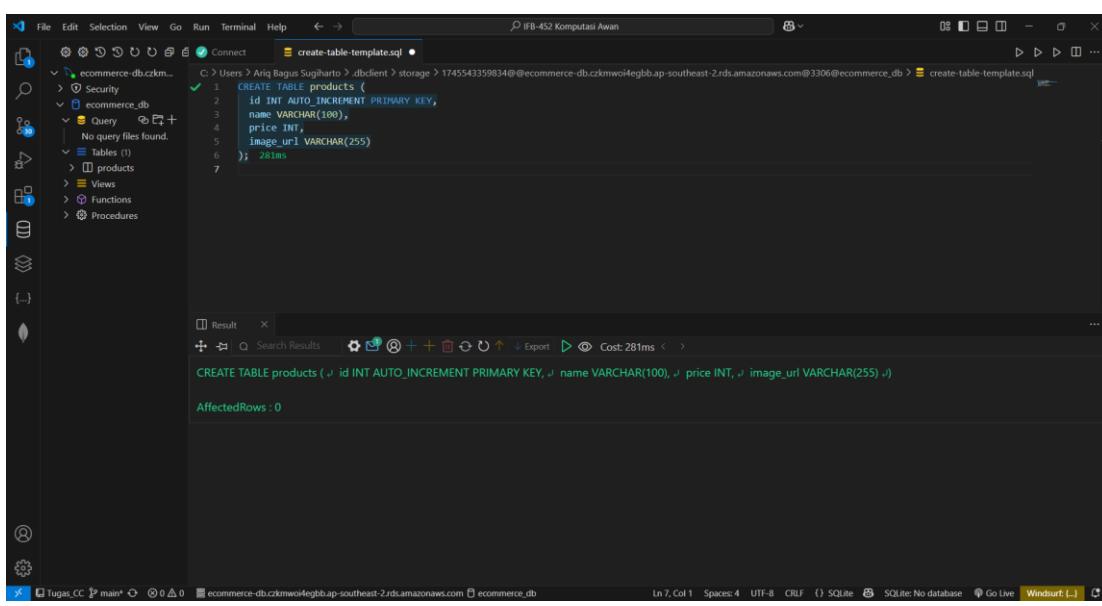
Additional configuration

You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

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.

The screenshot shows a SQLite database client interface with the following details:

- Database:** ecommerce_db
- Table:** products
- Columns:** id (int), name (varchar(100)), price (int), image_url (varchar(255))
- SQL Query:** A query is being run to insert data into the products table. The query is:INSERT INTO products (name, price, image_url) VALUES ('Iphone 16 128GB', 14999999, 'https://ecommerce-picture-bucket.s3.ap-southeast-2.amazonaws.com/iphone_16.jpg'), ('Samsung Galaxy S25 512 GB', 16999999, 'https://ecommerce-picture-bucket.s3.ap-southeast-2.amazonaws.com/samsung_galaxy_s25.jpg'), ('Huawei Mate X3 Foldable 4G Smart Phone 7.85 Purple, 12/512', 55499999, 'https://ecommerce-picture-bucket.s3.ap-southeast-2.amazonaws.com/huawei_mate_x3_foldable_4g_smart_phone_7.85_purple_12_512.jpg')

Bagian 3 - Pengembangan Aplikasi Web

1. Setup Frontend EC2

- 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:
 app-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
```

```
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
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
```

```
node_bundle/node_modules/npm/node_modules/postcss-selector-parser/dist/util/ensureObject.js
node_bundle/node_modules/npm/node_modules/postcss-selector-parser/dist/util/getProp.js
node_bundle/node_modules/npm/node_modules/postcss-selector-parser/dist/util/index.js
node_bundle/node_modules/npm/node_modules/postcss-selector-parser/dist/util/stripComments.js
node_bundle/node_modules/npm/node_modules/postcss-selector-parser/dist/util/unesc.js
node_bundle/node_modules/npm/node_modules/proc-log/
node_bundle/node_modules/npm/node_modules/proc-log/LICENSE
node_bundle/node_modules/npm/node_modules/proc-log/package.json
node_bundle/node_modules/npm/node_modules/proc-log/lib/
node_bundle/node_modules/npm/node_modules/proc-log/lib/index.js
node_bundle/node_modules/npm/node_modules/proggy/
node_bundle/node_modules/npm/node_modules/proggy/LICENSE
node_bundle/node_modules/npm/node_modules/proggy/package.json
node_bundle/node_modules/npm/node_modules/proggy/lib/
node_bundle/node_modules/npm/node_modules/proggy/lib/client.js
node_bundle/node_modules/npm/node_modules/proggy/lib/index.js
node_bundle/node_modules/npm/node_modules/proggy/lib/tracker.js
node_bundle/node_modules/npm/node_modules/promise-all-reject-late/
node_bundle/node_modules/npm/node_modules/promise-all-reject-late/LICENSE
node_bundle/node_modules/npm/node_modules/promise-all-reject-late/index.js
node_bundle/node_modules/npm/node_modules/promise-all-reject-late/package.json
node_bundle/node_modules/npm/node_modules/promise-retry/
node_bundle/node_modules/npm/node_modules/promise-retry/LICENSE
node_bundle/node_modules/npm/node_modules/promise-retry/index.js
node_bundle/node_modules/npm/node_modules/promise-retry/package.json
node_bundle/node_modules/npm/node_modules/read-cmd-shim/
node_bundle/node_modules/npm/node_modules/read-cmd-shim/LICENSE
node_bundle/node_modules/npm/node_modules/read-cmd-shim/package.json
node_bundle/node_modules/npm/node_modules/read-cmd-shim/lib/
```

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

```
[ec2-user@ip-10-0-1-133 ~]$ scp -i ~/ariq_key.pem node_pm2_bundle.tar.gz ec2-user@10.0.2.59:~/
node_pm2_bundle.tar.gz
          100%   39MB 118.8MB/s  00:00
```

```
ssh -i key.pem ec2-user@ipv4_private
```

```
[ec2-user@ip-10-0-1-133 ~]$ ssh -i "ariq_key.pem" ec2-user@10.0.2.59
#_#
~~\####_      Amazon Linux 2023
~~\#####\
~~\###|
~~  '#/ _-- https://aws.amazon.com/linux/amazon-linux-2023
~~  V~' '--->
~~  /
~~. _/ _/
~~/_m/`_
Last login: Sat Apr 26 03:30:43 2025 from 10.0.1.133
```

```
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
```

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

The screenshot shows a GitHub repository page for 'ecommerce-cloud'. The repository is public and has one branch ('master') and no tags. The last commit was made by 'AbbyGud' 16 minutes ago. The repository contains three main folders: '.github/workflows', 'backend', and 'frontend', each with an 'Initial Commit' message. A 'README' file is present but empty. The 'About' section indicates there is no description, website, or topics provided. The 'Releases' section shows no releases published, with a link to 'Create a new release'. The 'Packages' section shows no packages published, with a link to 'Publish your first package'. The 'Languages' section shows a chart where JavaScript accounts for 57.7% and HTML for 42.3% of the codebase.

- 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
```

```
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 ^D DOS Format ^A Append ^B Backup File
^C Cancel ^M Mac Format ^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 ~]$ 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
index.js
mime
mime.cmd
mime.ps1
.package-lock.json
HISTORY.md
LICENSE
README.md
index.json
package.json
LICENSE
README
README.md
array-flatten.js
package.json
HISTORY.md
LICENSE
README.md
SECURITY.md
index.js
read.js
json.js
raw.js
text.js
urlencoded.js
package.json
History.md
LICENSE
```

	100%	120	172.1KB/s	00:00
.env	100%	1091	1.1MB/s	00:00
index.js	100%	371	339.6KB/s	00:00
mime	100%	299	293.5KB/s	00:00
mime.cmd	100%	769	1.7MB/s	00:00
mime.ps1	100%	38KB	21.9MB/s	00:00
.package-lock.json	100%	5096	12.3MB/s	00:00
HISTORY.md	100%	1167	3.1MB/s	00:00
LICENSE	100%	4123	11.3MB/s	00:00
README.md	100%	5252	7.2MB/s	00:00
index.json	100%	1157	2.3MB/s	00:00
package.json	100%	1103	1.7MB/s	00:00
LICENSE	100%	1245	1.6MB/s	00:00
README	100%	1195	1.1MB/s	00:00
README.md	100%	879	2.3MB/s	00:00
array-flatten.js	100%	16KB	25.3MB/s	00:00
SECURITY.md	100%	1172	1.2MB/s	00:00
index.js	100%	19KB	25.9MB/s	00:00
read.js	100%	1193	2.5MB/s	00:00
json.js	100%	2681	7.1MB/s	00:00
raw.js	100%	4325	10.2MB/s	00:00
text.js	100%	5299	5.4MB/s	00:00
urlencoded.js	100%	1884	2.7MB/s	00:00
package.json	100%	2285	5.6MB/s	00:00
History.md	100%	6404	15.6MB/s	00:00
LICENSE	100%	1472	2.3MB/s	00:00
	100%	1775	4.3MB/s	00:00
	100%	1153	2.7MB/s	00:00

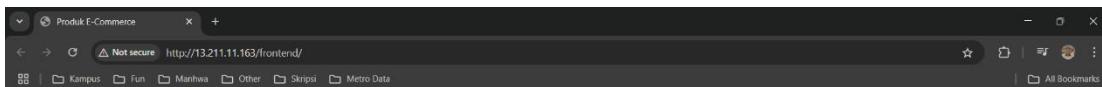
- g. Selanjutnya di instance EC2 Backend mulai node.js.

```
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	o	status	cpu	memory
0	index	fork	0	online	0%	22.4mb

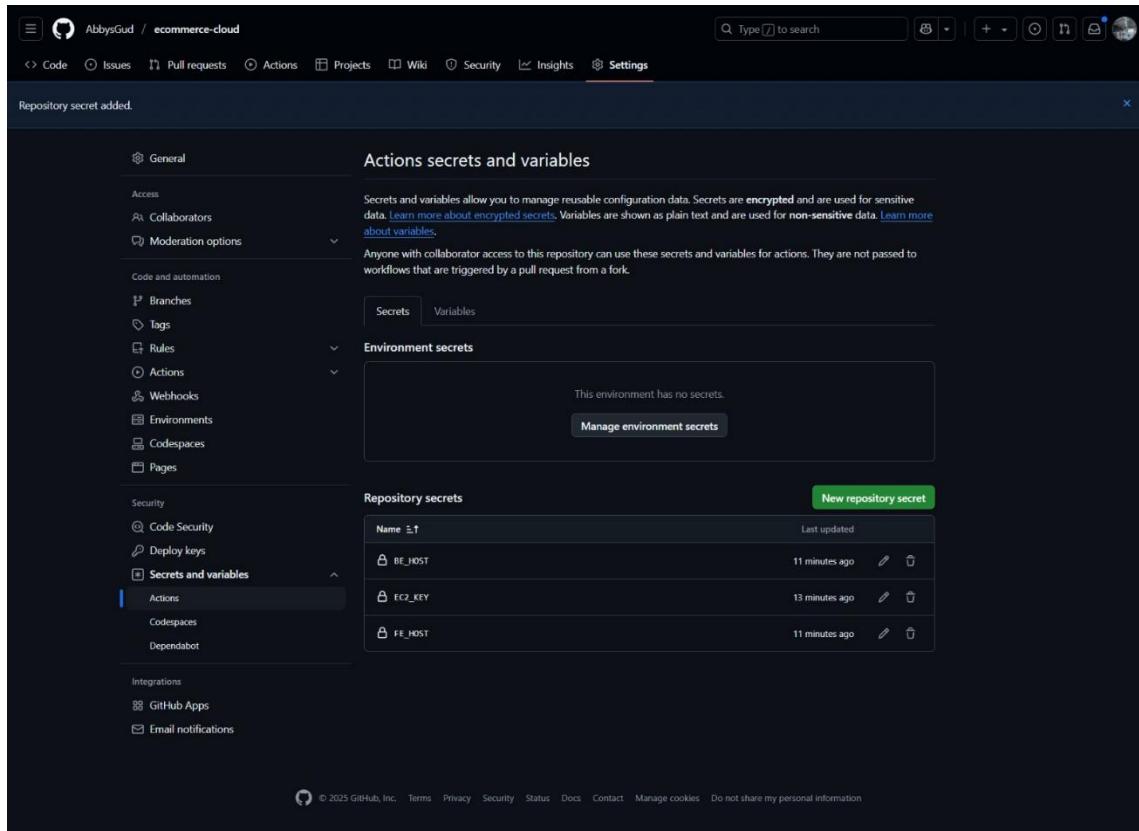
- h. Periksa apakah berhasil atau tidak, jika berhasil maka harusnya ada gambar yang diambil dari Bucket S3, dan ada data produk yang diambil dari RDS Database.



	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

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
```

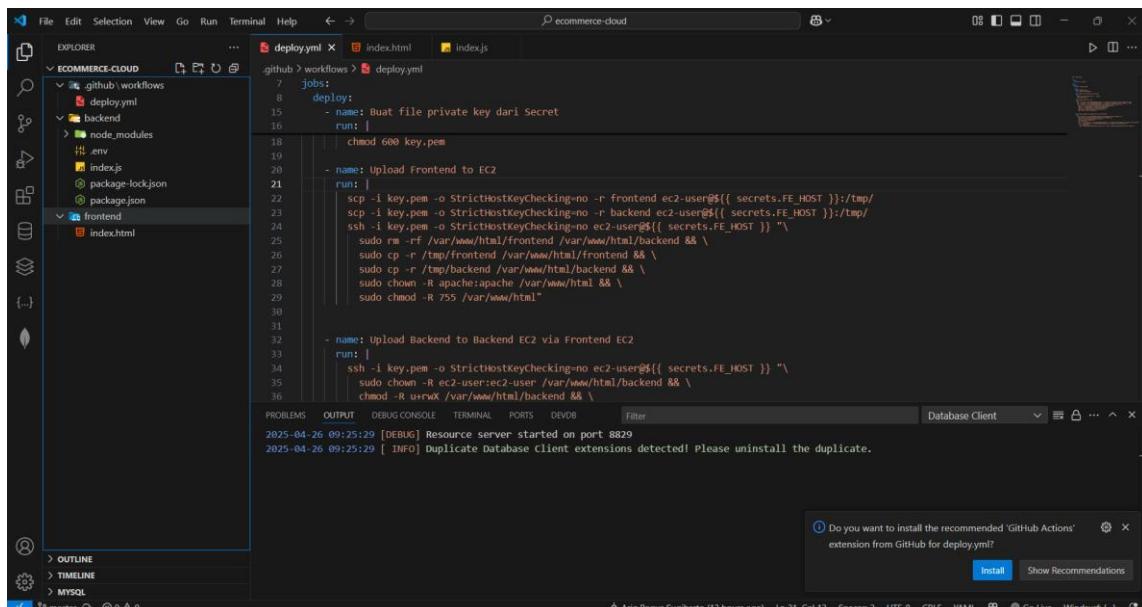
```

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:

```

backend > app.test.js > describe(GET /products) callback > afterAll() callback
1  const request = require('supertest');
2  const { app, db } = require('../index');
3
4  describe('GET /products', () => {
5    it('harus mengembalikan status 200', async () => {
6      const res = await request(app).get('/products');
7      expect(res.statusCode).toBe(200);
8    });
9
10   afterAll(() => {
11     db.end();
12   });
13 });

PS C:\VARIQ\KULIAH\SEMESTER 8\IFB-452 Komputasi Awan\Tugas Ariq\ecommerce-cloud\backend> npm test
> backend@1.0.0 test
> jest

console.log
  Terhubung ke database.

  at log (index.js:24:11)

PASS ./app.test.js
  GET /products
    ✓ harus mengembalikan status 200 (610 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.611 s, estimated 3 s
Ran all test suites.

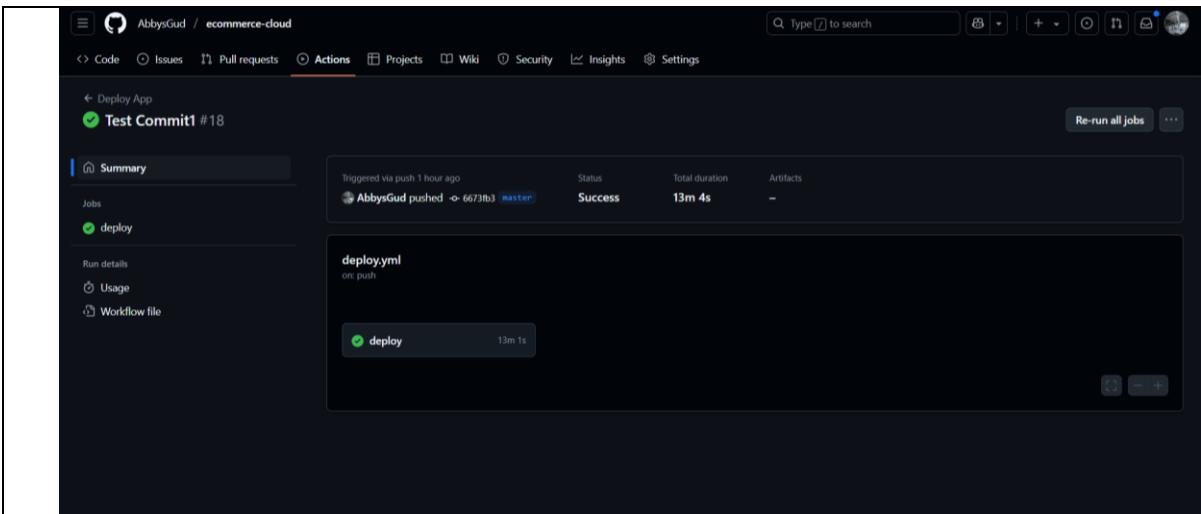
```

- 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.

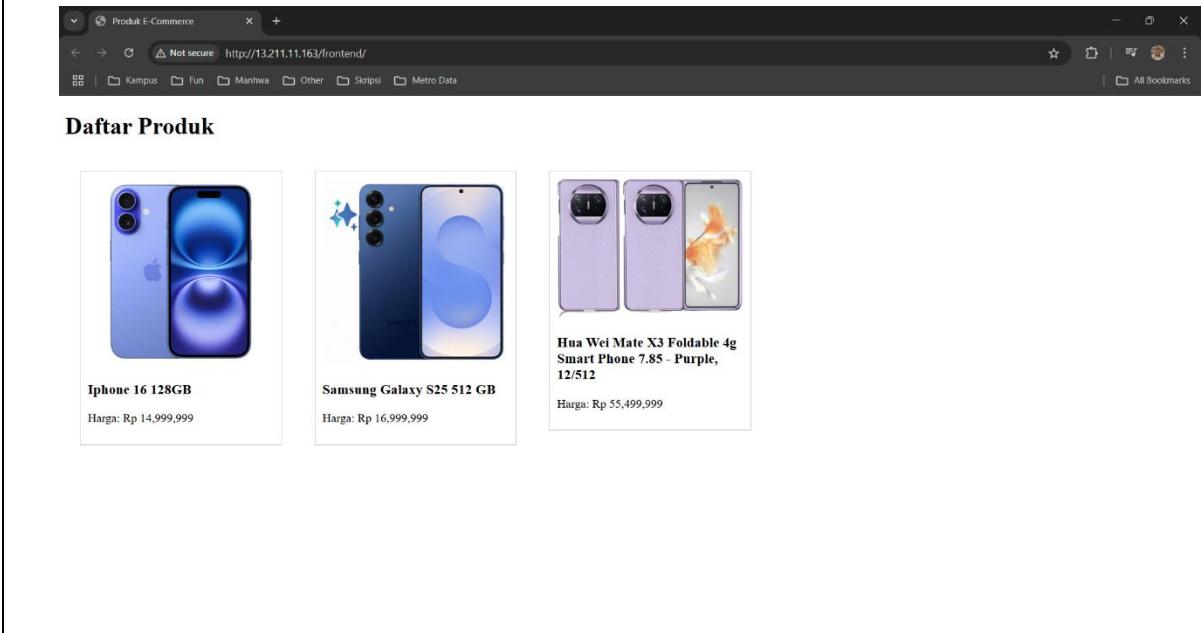
```

Test Commit1
Aniq Bagus Sugiharto -O- eaec741 +1 -1
1 changed file: frontend/index.html
frontend/index.html
  8 8      </style>
  9 9      </head>
10 10     <body>
11 11     -<h1>Daftar Produk</h1>
12 12     +<h1>Daftar Handphone</h1>
13 13     <div id="produk-list"></div>
14 14     <script>

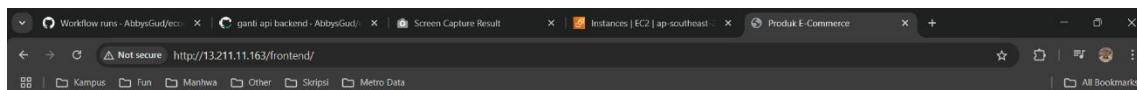
```



Sebelum:



Sesudah:



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