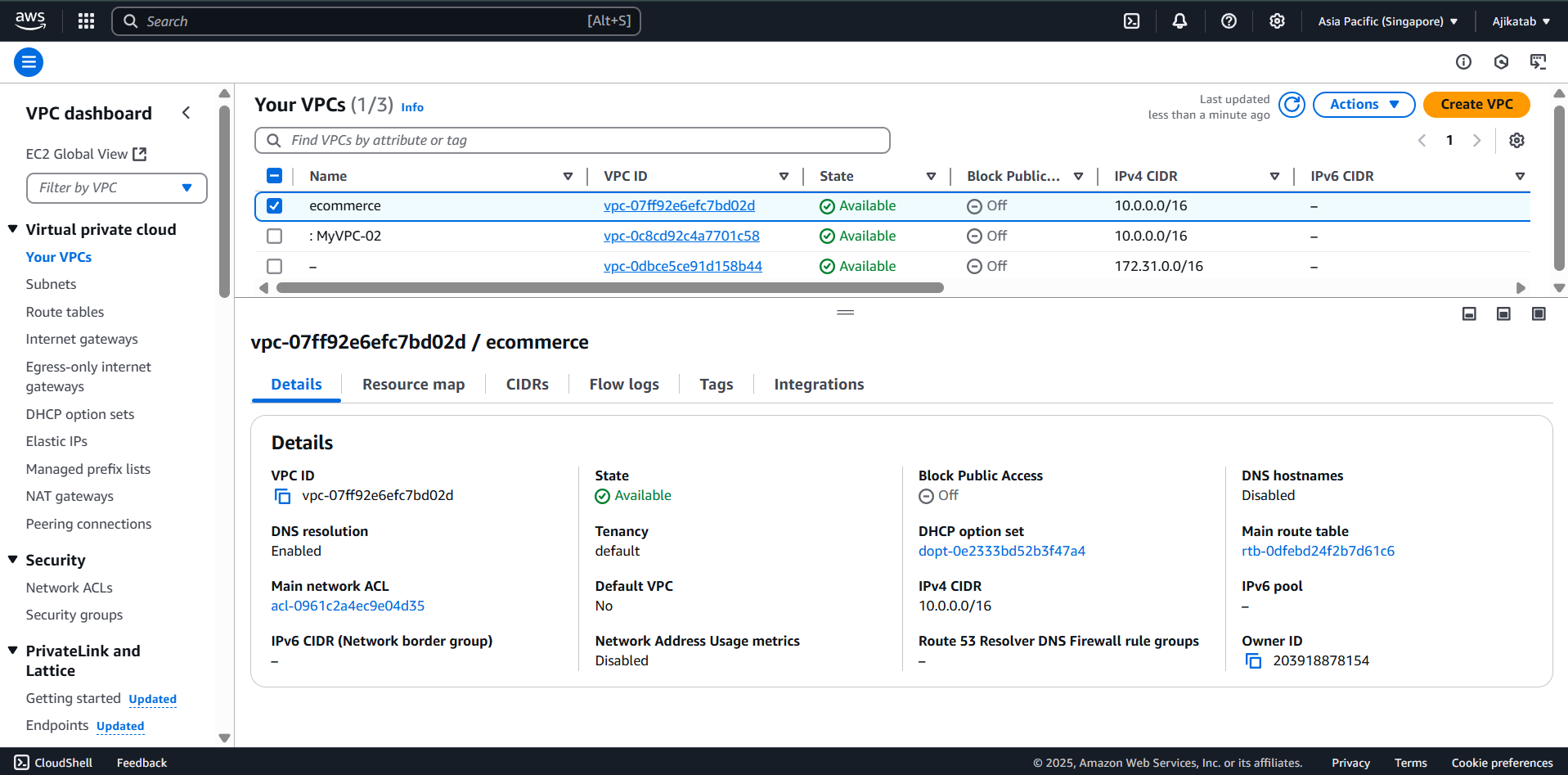
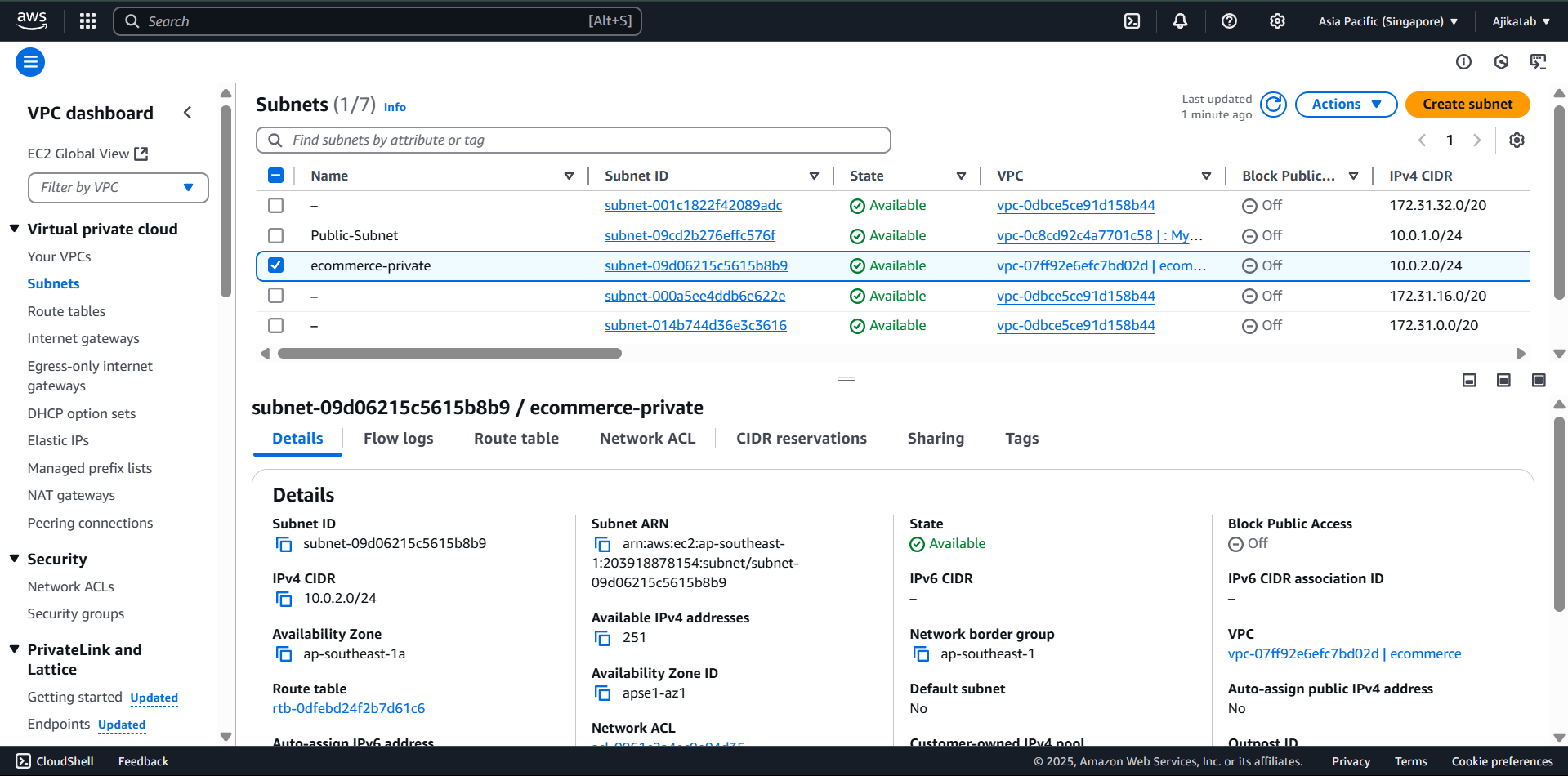
Nama: Faturohman Fahrizi Katab

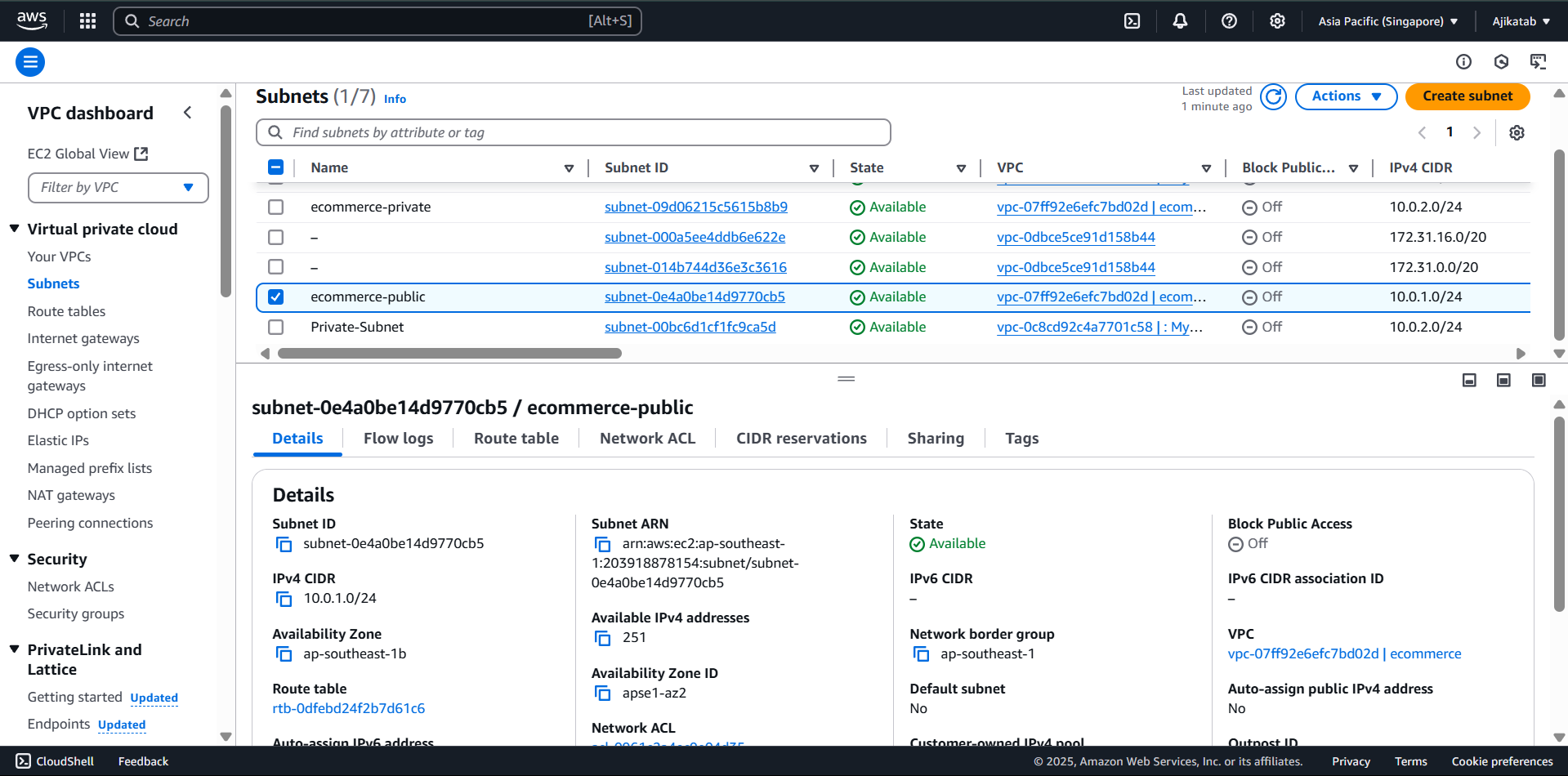
NRP: 152022072

1. Membuat VPC dengan nama Ecommerce

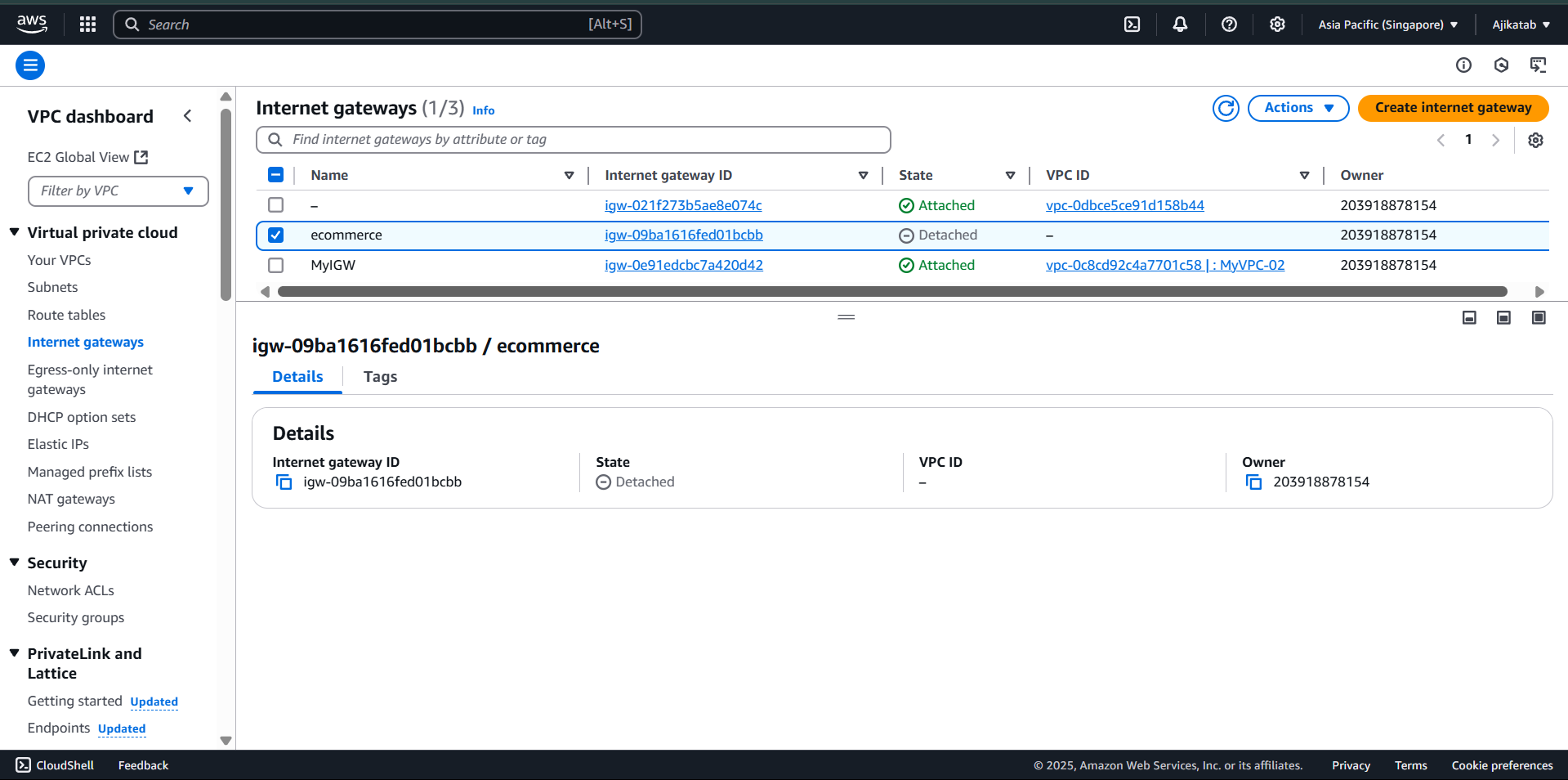


1. Membuat 2 subnet dengan nama ecommerce-private dan ecommerce-public

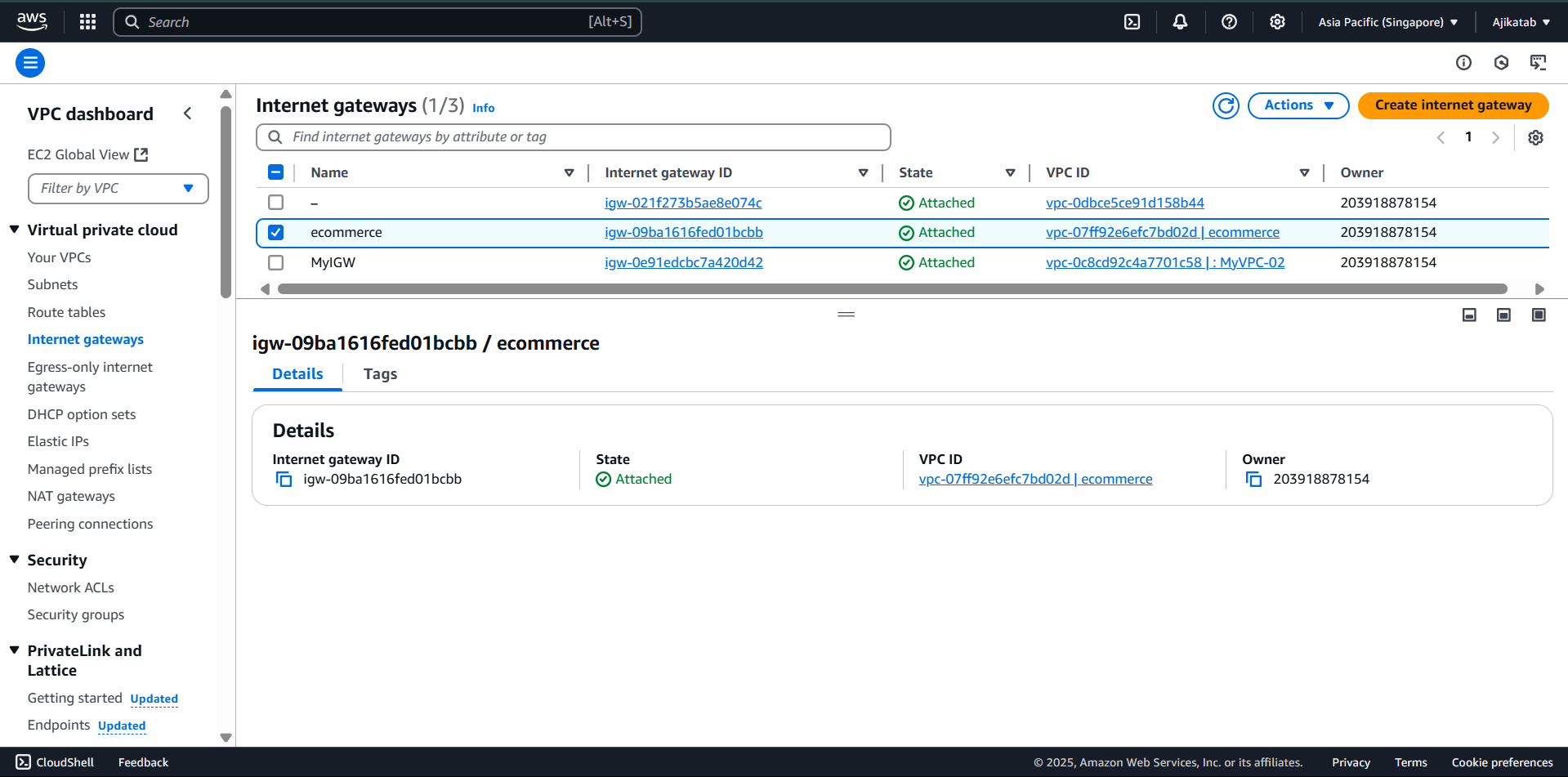




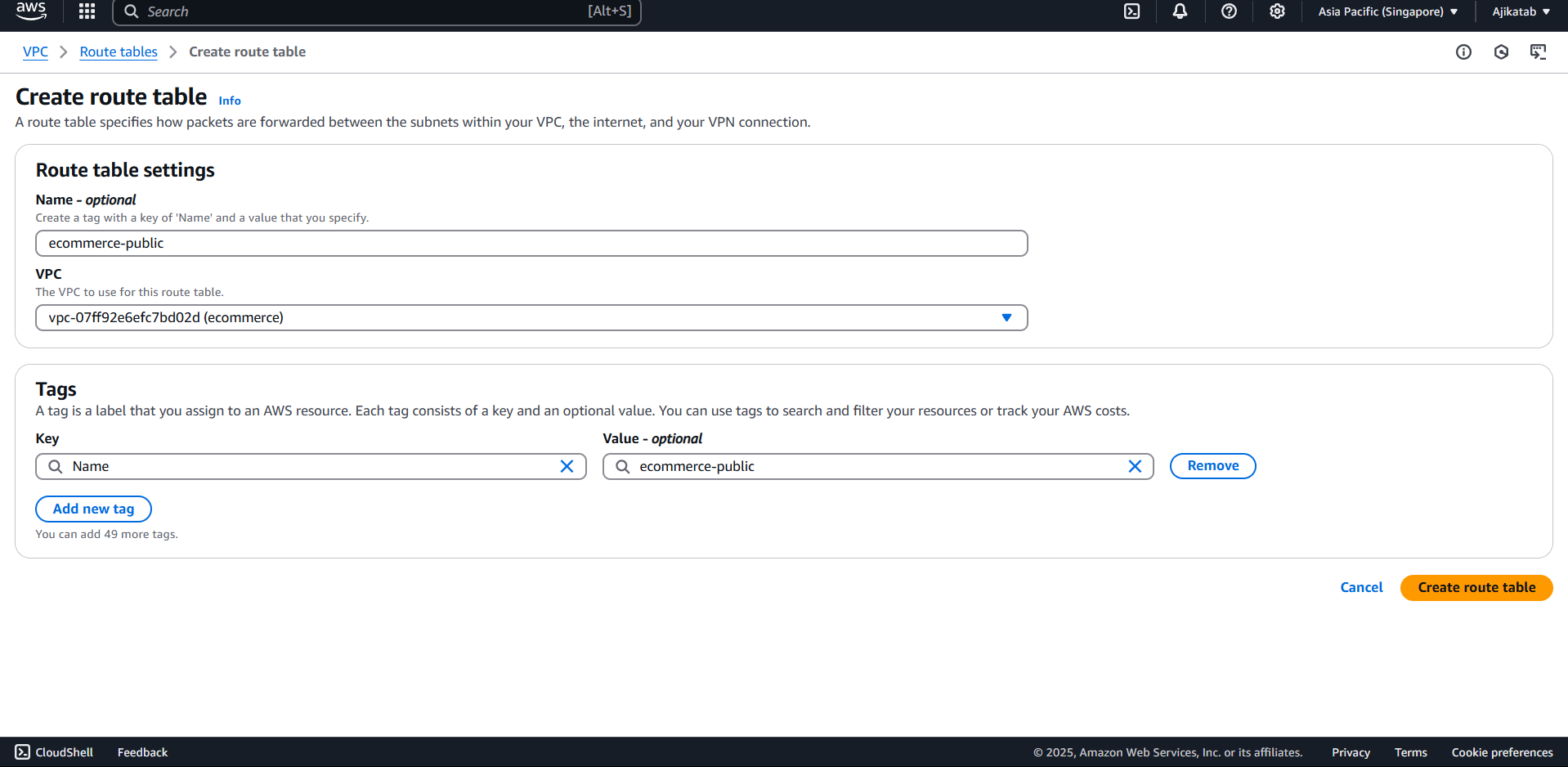
1. Lalu membuat internet Gateways dengan nama ecommerce



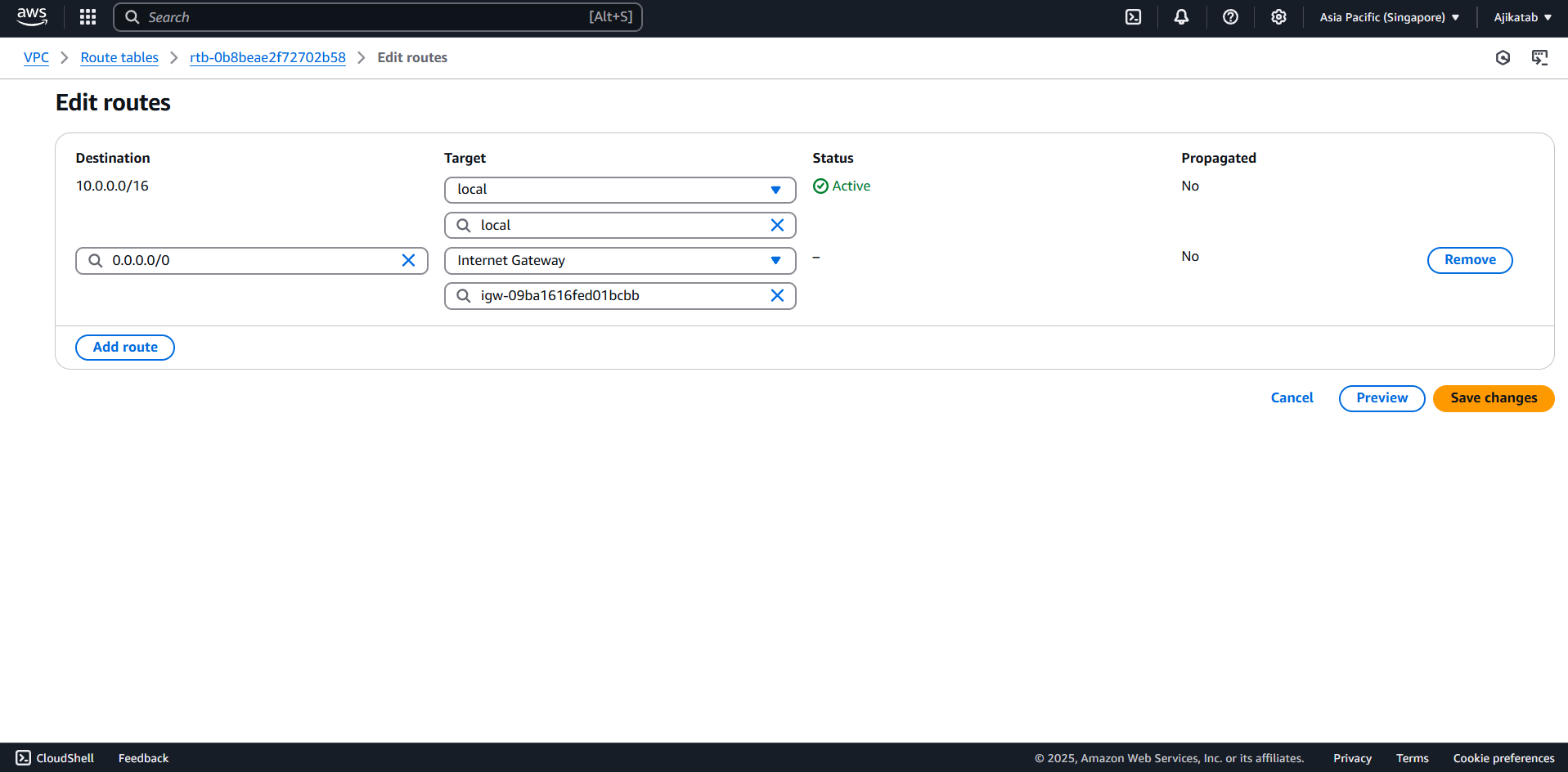
1. Lalu attack internet Gateways ke VPC yang tadi sudah di buat



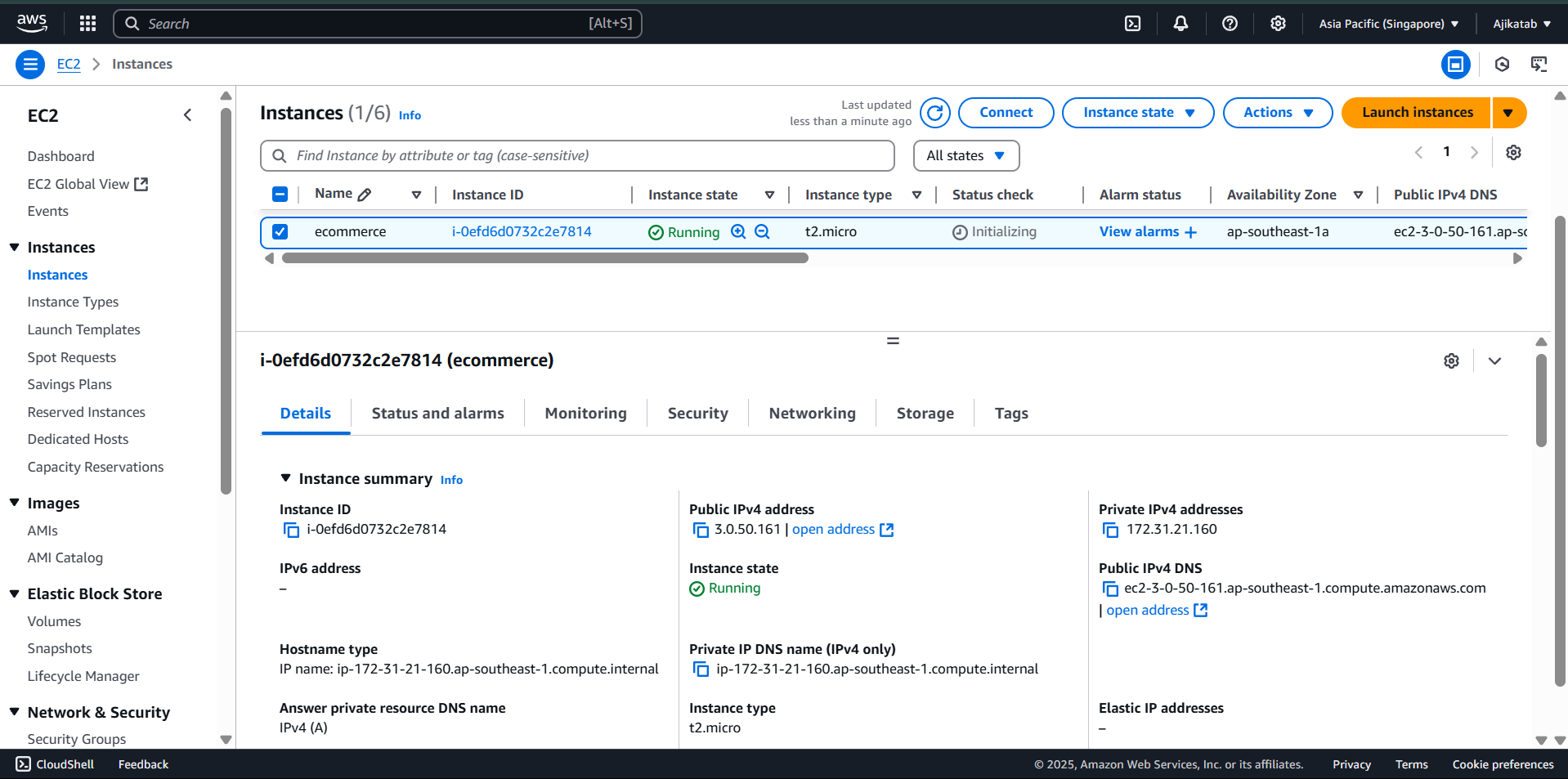
1. Lalu buat route table dengan nama ecommers-public



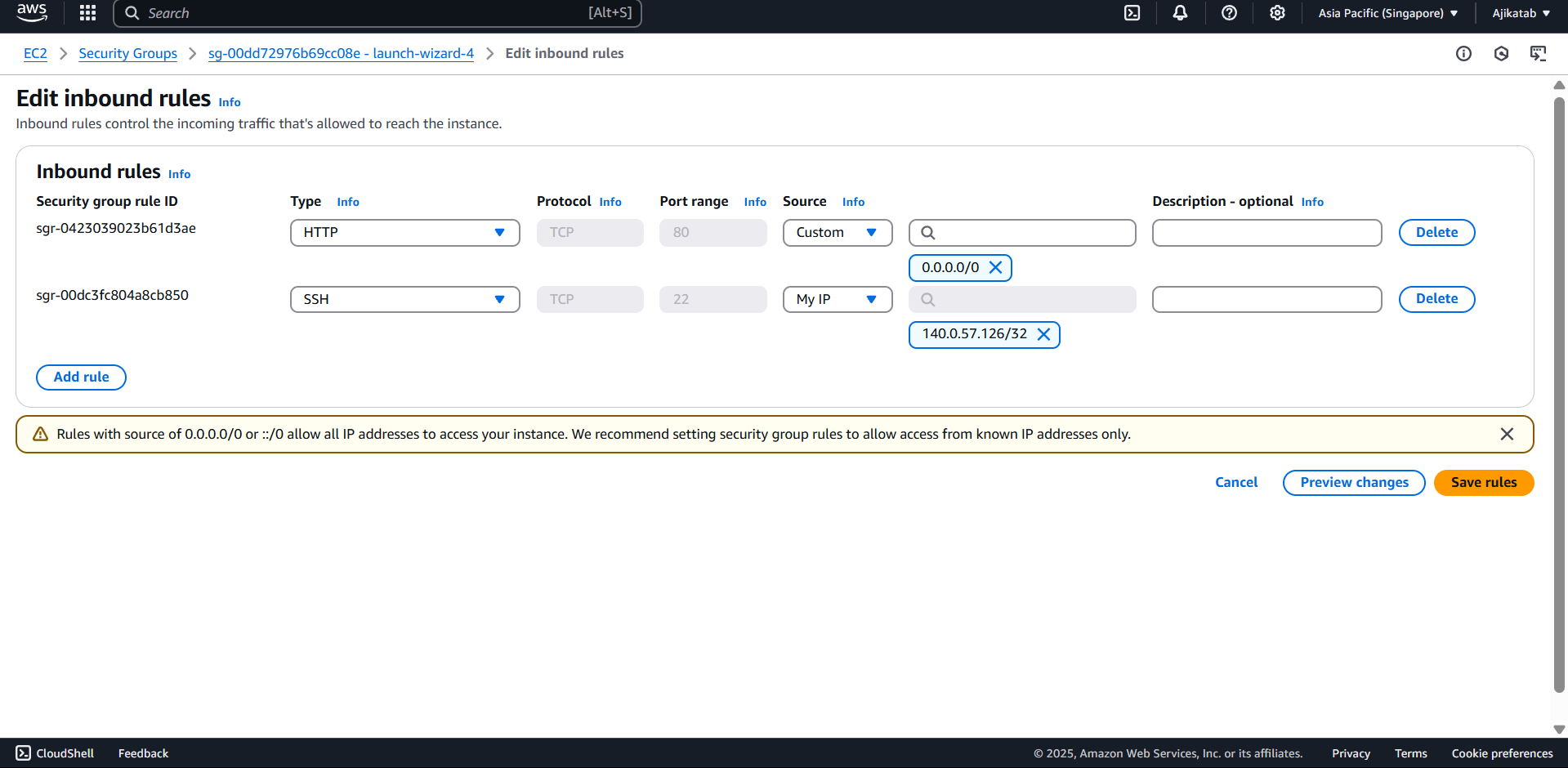
1. Lalu edit routenya dengan target internet gateways yang tadi sudah di buat



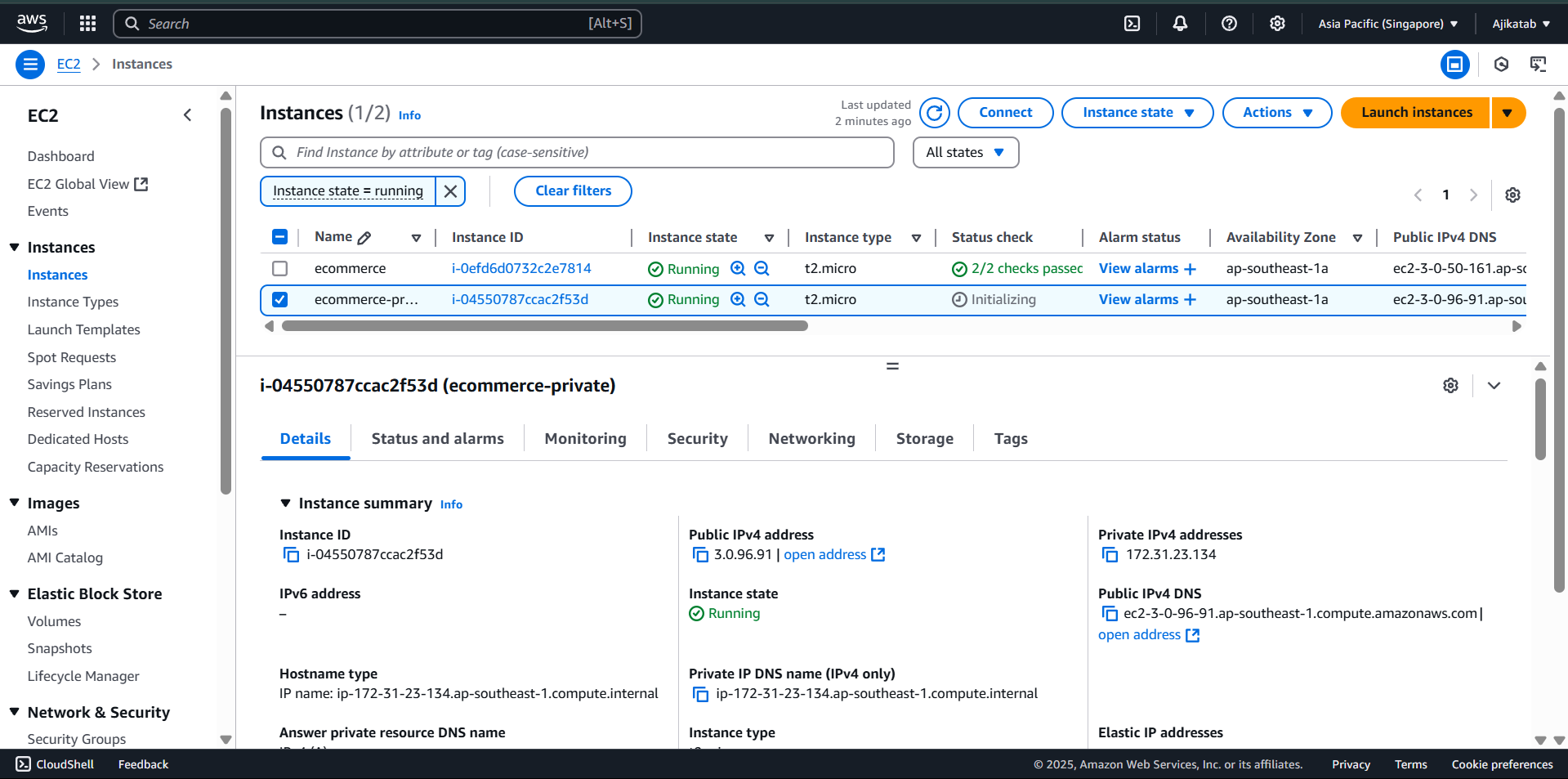
1. Buat EC2 dengan nama ecommerce



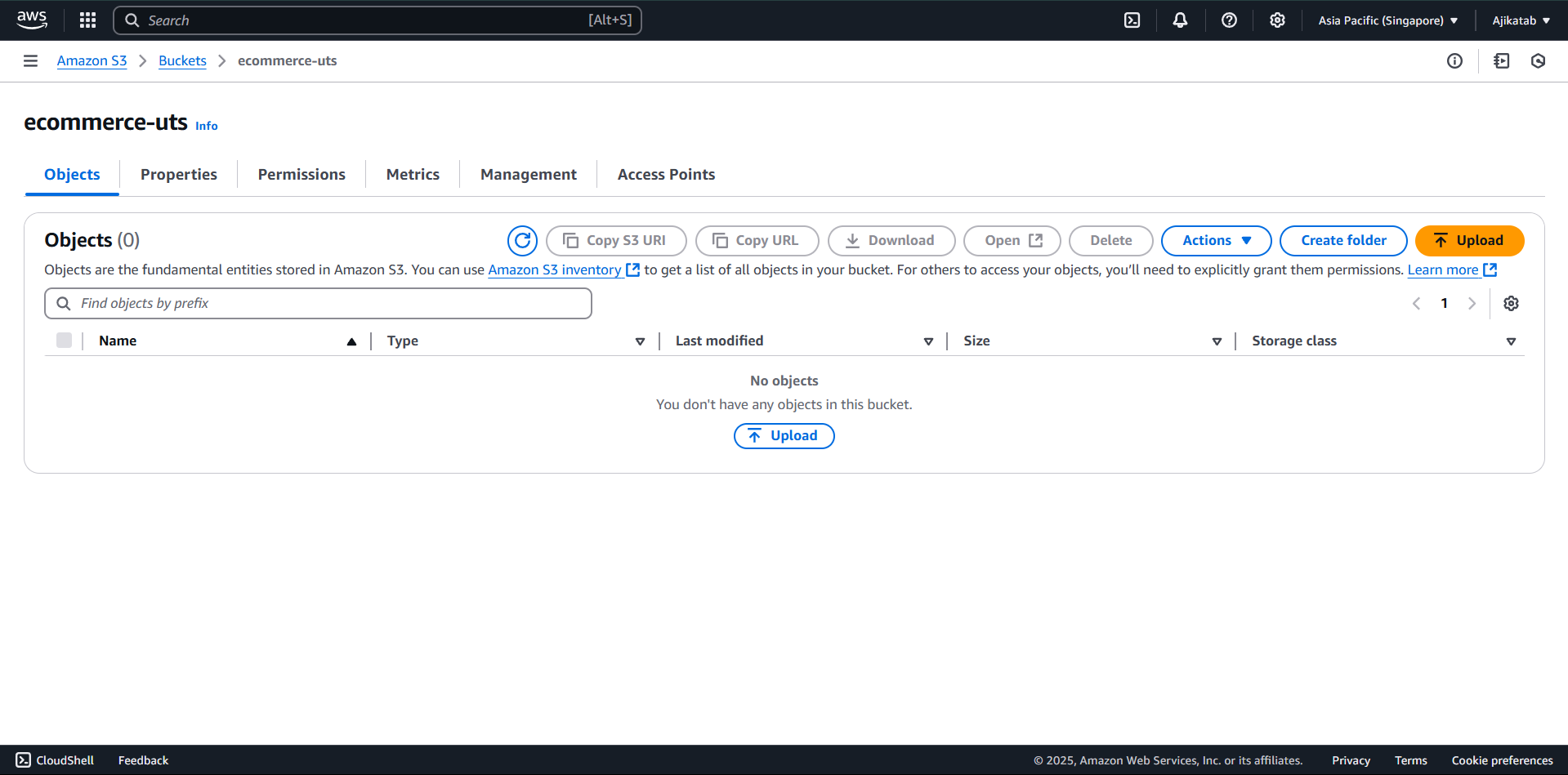
1. Lalu edit pada inbound rulesnya dengan mengganti source pada ssh menjadi “my ip”



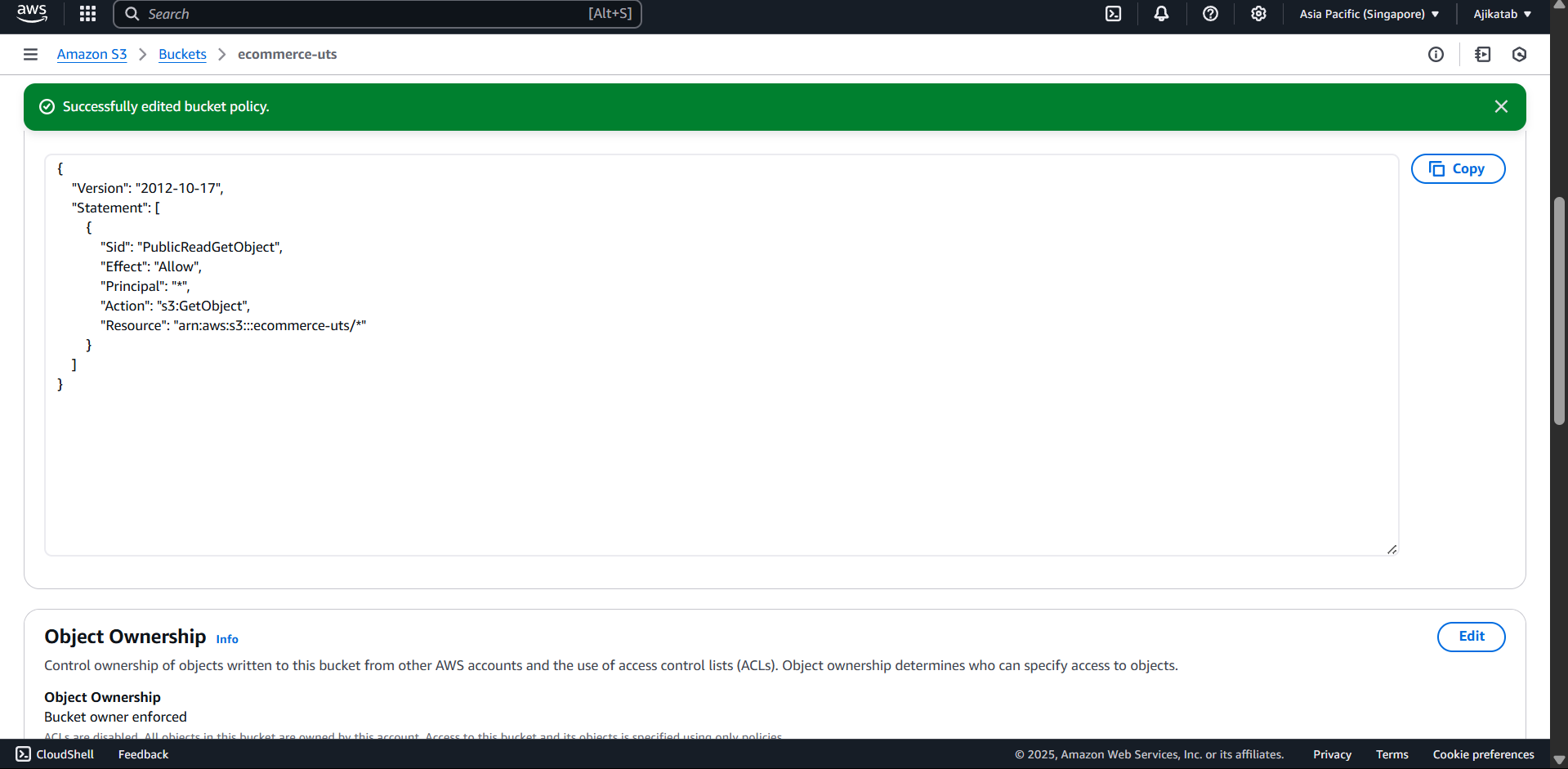
1. Membuat EC2 untuk yg private denagan nama ecommerce-private



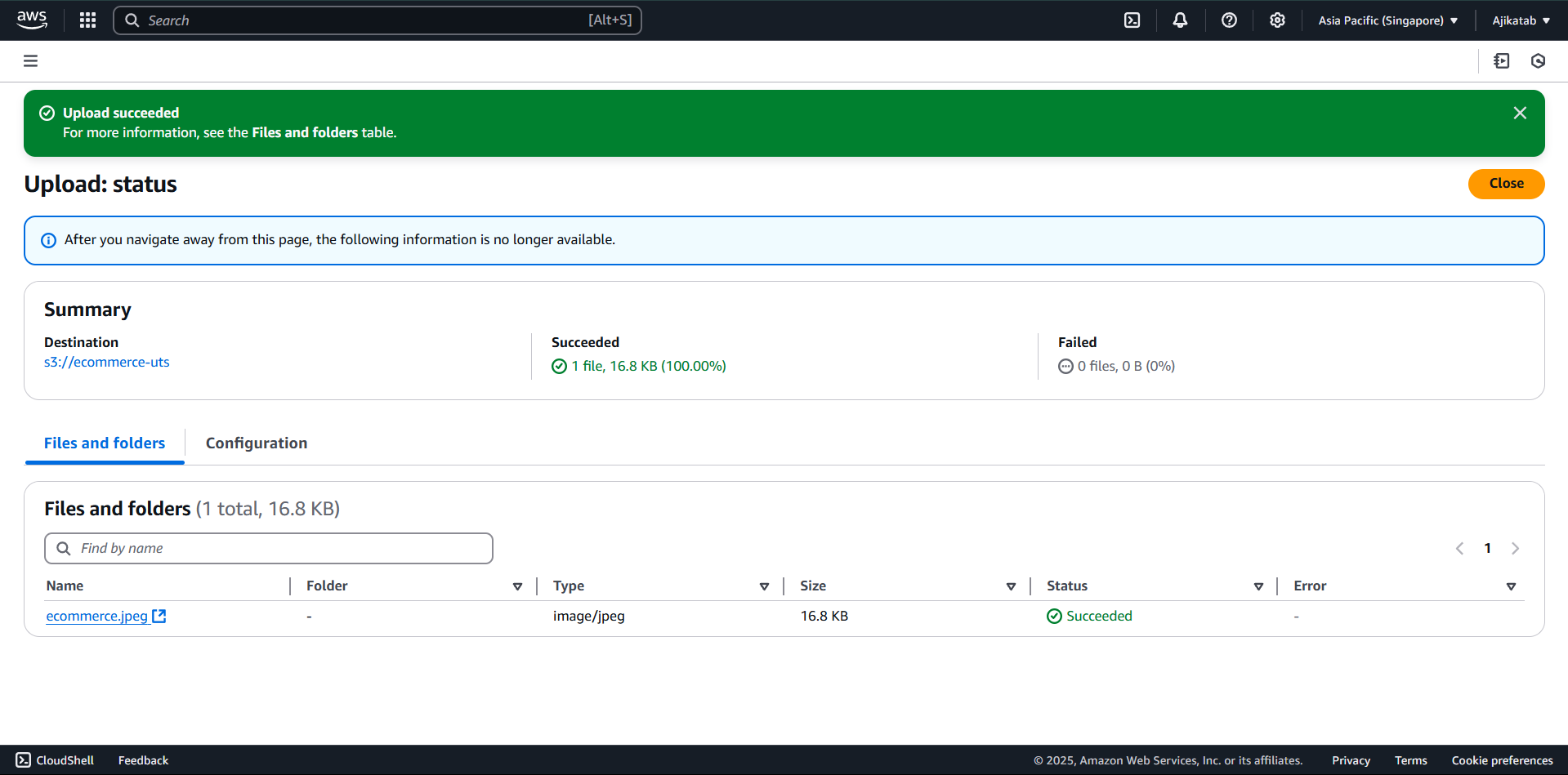
1. Selanjutnya buat bucket dengan mengan nama ecommerce-uts. uncheck block all public access agar gambar dapat di akses secara public



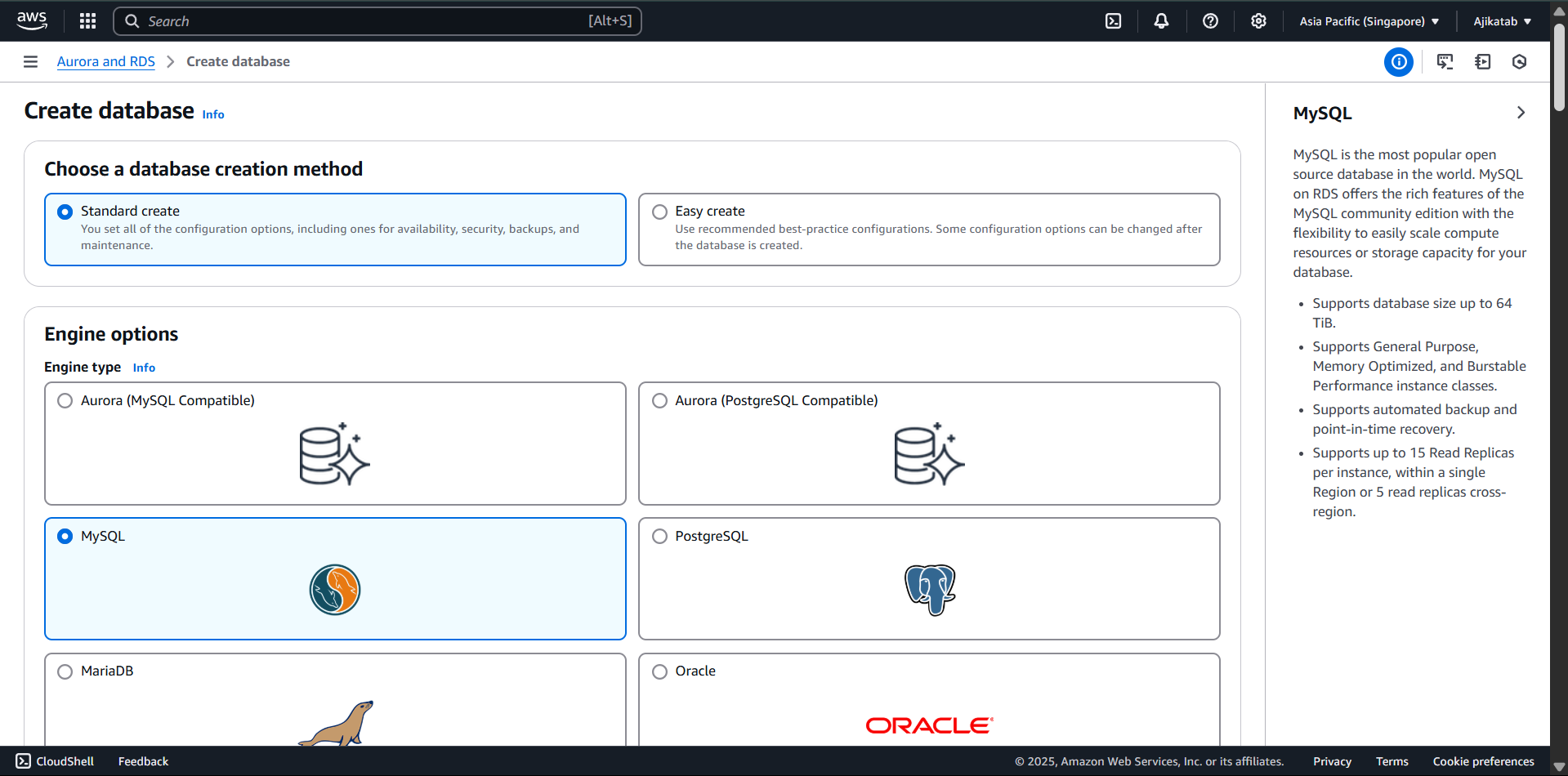
1. Selanjutnya edit bagian bucket policy pada bagian permissions

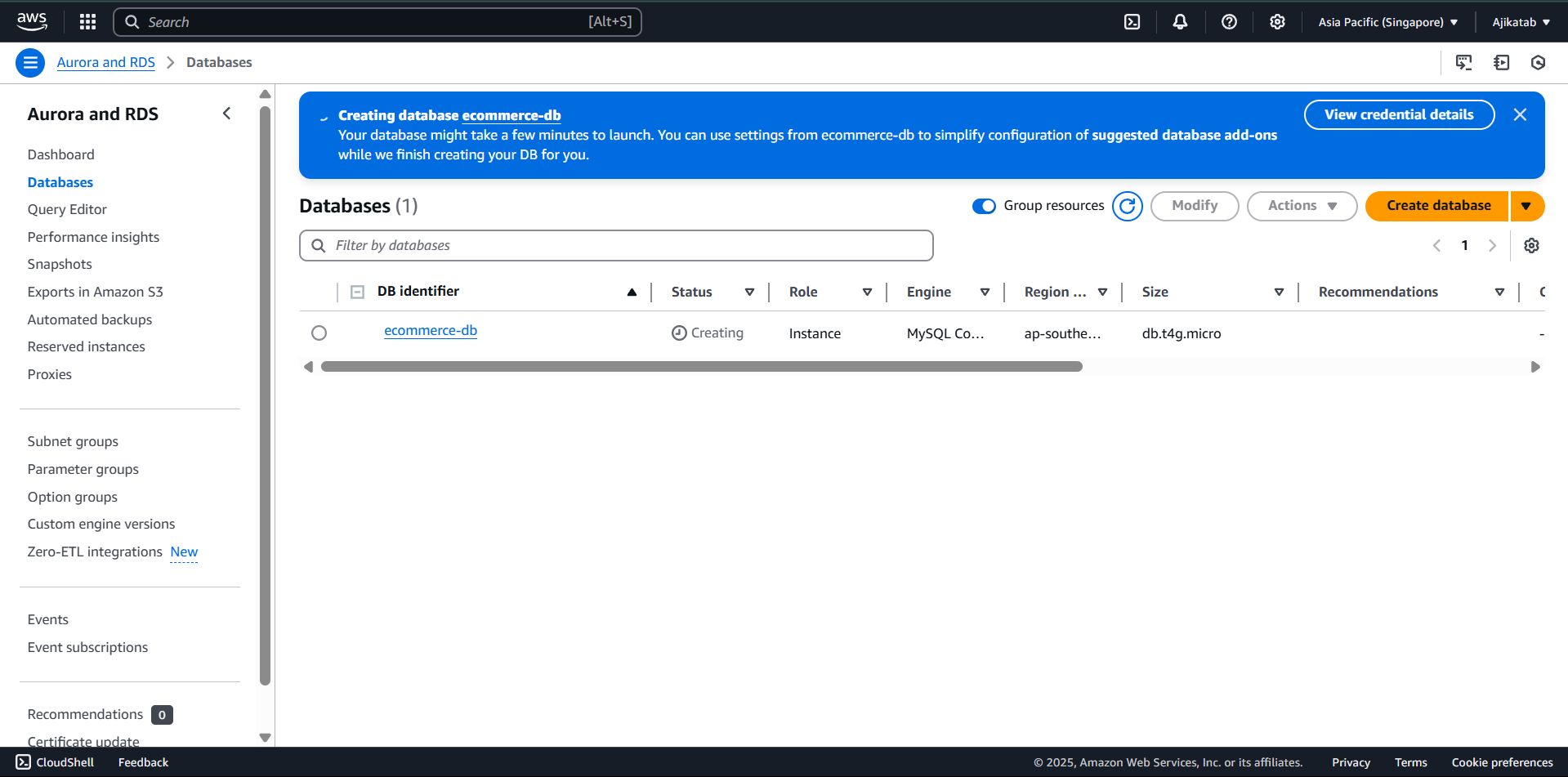


1. Upload gambar di bucket

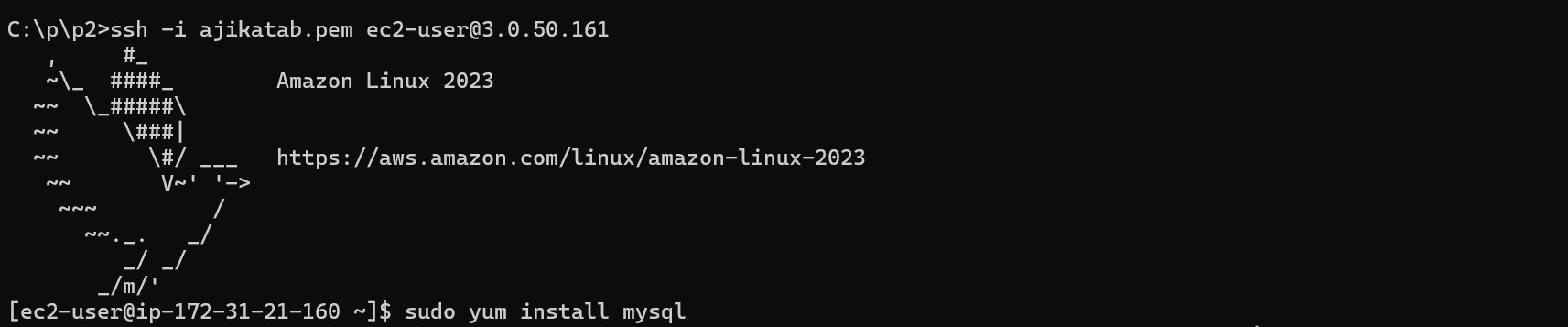


1. Buat Database di Aurora and RDS dengan menggunakan engine options mysql

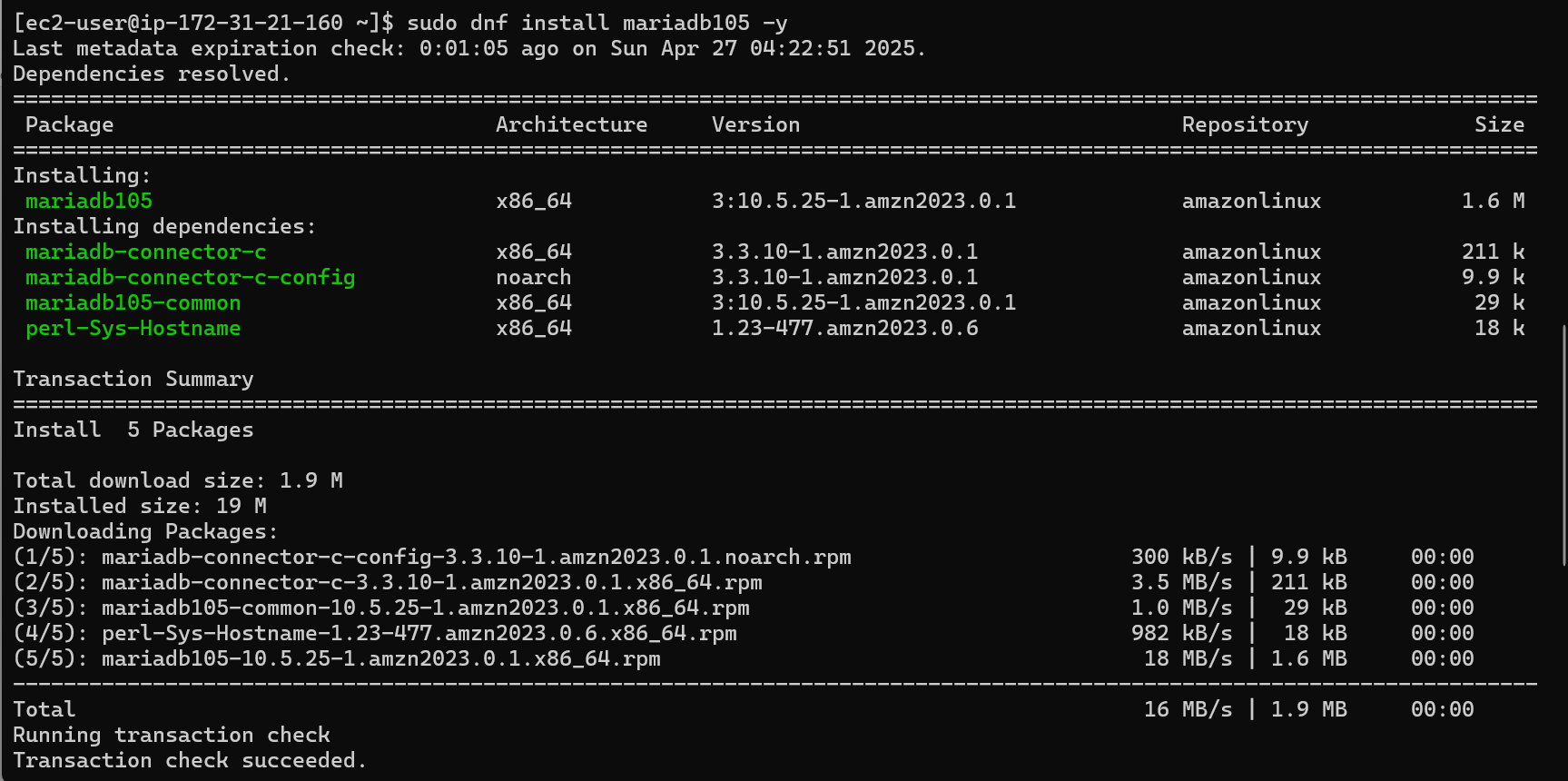




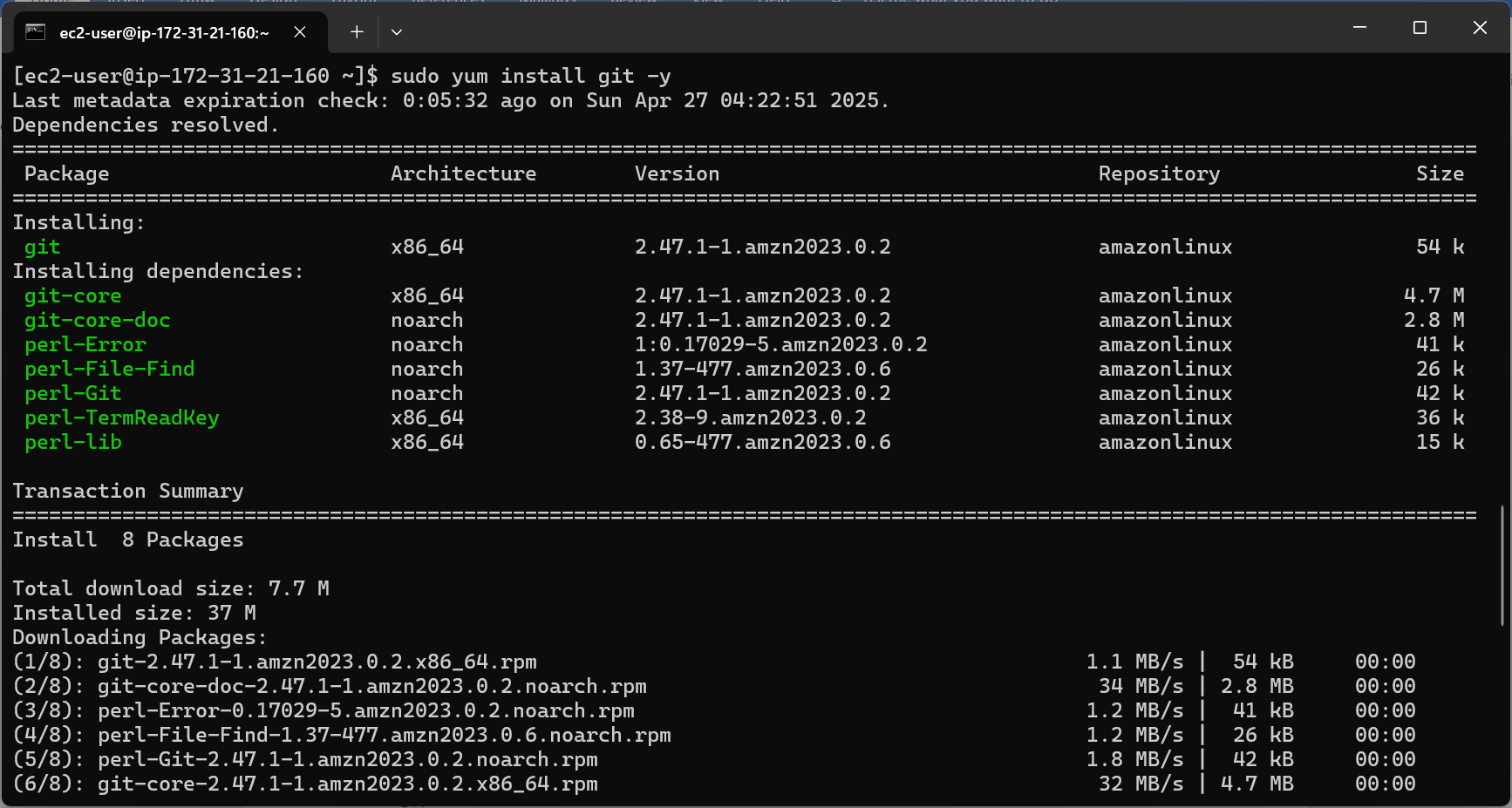
1. Login ssh “ssh -i ajikatab.pem ec2-user@3.0.50.161”



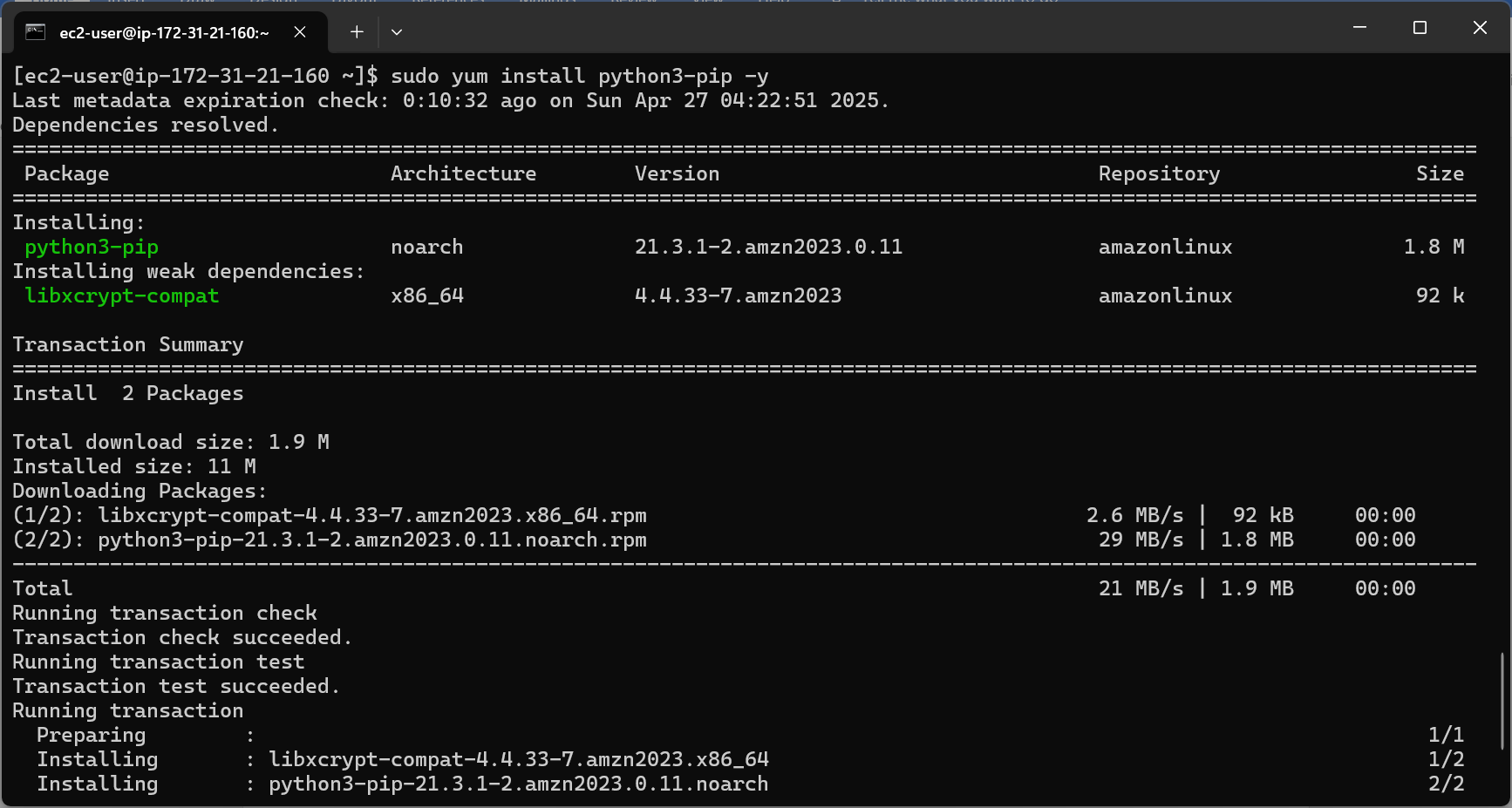
1. Install mysql-client “sudo dnf install mariadb105 -y”



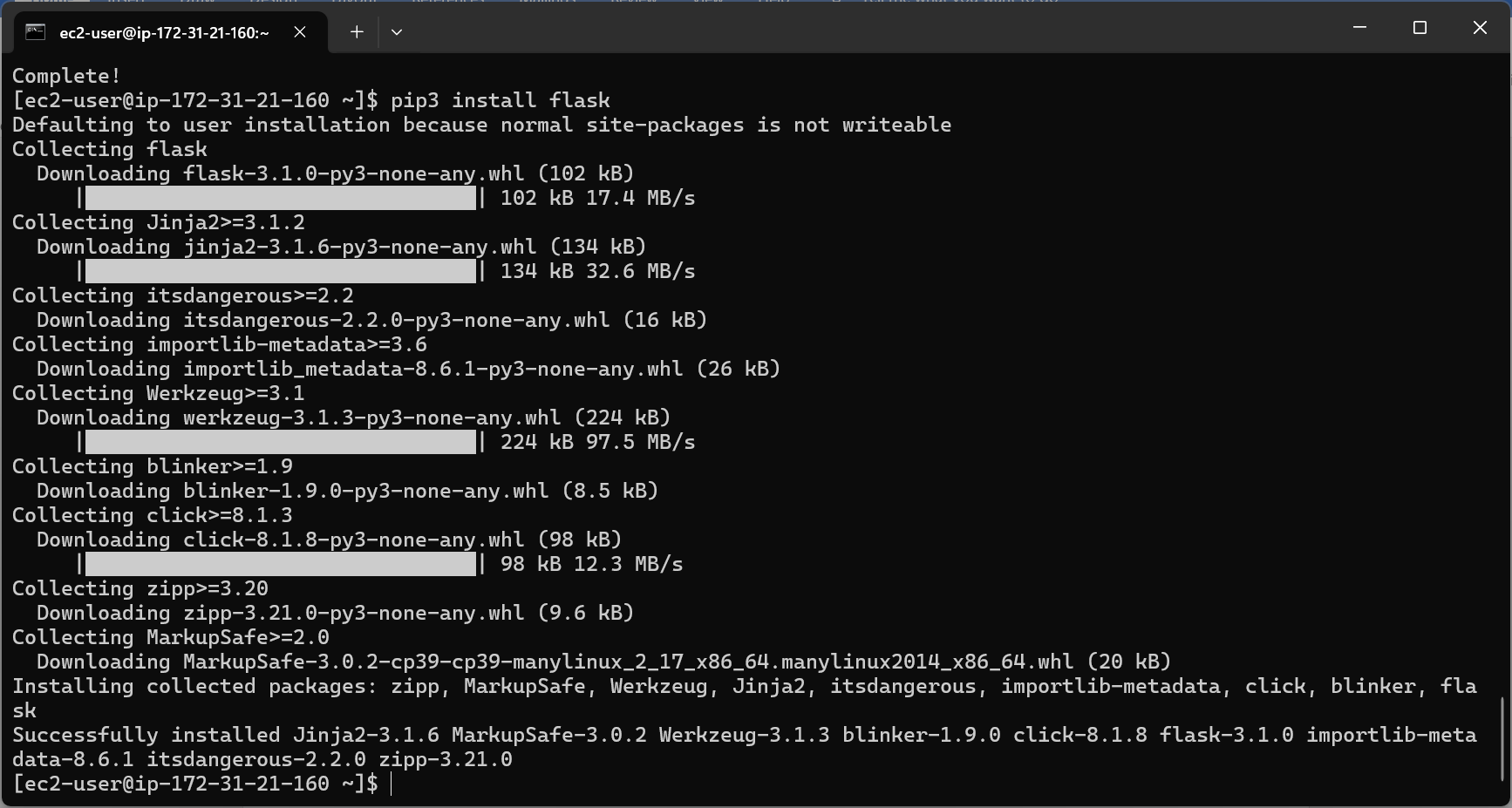
1. Install git “sudo yum install git -y”



1. Install python “sudo yum install python3-pip -y”



1. Install flask “pip3 install flask”



1. Membuat kodingan app.py dan html untuk tampilan webnya. Dan buatkan juga deploy.yml

|  |
| --- |
| from flask import Flask, render\_template\_string  import boto3  import pymysql  app = Flask(\_\_name\_\_)  # Inisialisasi boto3 untuk S3  s3 = boto3.client('s3')  # Fungsi untuk mendapatkan URL gambar produk dari S3  def get\_image\_url\_from\_s3(bucket\_name, file\_name):      url = s3.generate\_presigned\_url('get\_object', Params={'Bucket': bucket\_name, 'Key': file\_name})      return url  # Fungsi untuk mendapatkan koneksi ke RDS MySQL  def get\_rds\_connection():      rds = boto3.client('rds')      db\_instance = rds.describe\_db\_instances(DBInstanceIdentifier='ecommerce-db')      endpoint = db\_instance['DBInstances'][0]['Endpoint']['Address']      connection = pymysql.connect(          host=endpoint,          user='admin',  # Ganti dengan username RDS kamu          password='yourpassword',  # Ganti dengan password RDS kamu          db='ecommerce\_db'  # Ganti dengan nama database RDS kamu      )      return connection  # Fungsi untuk mendapatkan harga produk dari RDS  def get\_product\_price(product\_id):      connection = get\_rds\_connection()      with connection.cursor() as cursor:          cursor.execute("SELECT price FROM products WHERE product\_id = %s", (product\_id,))          result = cursor.fetchone()          return result[0] if result else None  # Endpoint untuk menampilkan harga produk dan URL gambar dari S3  @app.route('/')  def index():      # Contoh data produk yang diambil dari database dan S3      products = [          {              'name': 'Product 1',              'description': 'This is the description for product 1.',              'price': get\_product\_price(1),  # Ambil harga dari RDS              'image\_url': get\_image\_url\_from\_s3('ecommerce-bucket', 'product\_1.jpg')  # Gambar dari S3          },          {              'name': 'Product 2',              'description': 'This is the description for product 2.',              'price': get\_product\_price(2),  # Ambil harga dari RDS              'image\_url': get\_image\_url\_from\_s3('ecommerce-bucket', 'product\_2.jpg')  # Gambar dari S3          },          # Tambahkan produk lainnya sesuai kebutuhan      ]        # HTML untuk halaman      html\_code = """      <!DOCTYPE html>      <html lang="en">      <head>          <meta charset="UTF-8">          <meta name="viewport" content="width=device-width, initial-scale=1.0">          <title>Product List</title>          <style>              body {                  font-family: Arial, sans-serif;                  margin: 0;                  padding: 0;                  background-color: #f4f4f9;              }              .container {                  width: 80%;                  margin: 0 auto;                  padding: 20px;              }              h1 {                  text-align: center;                  color: #333;              }              .product-list {                  display: flex;                  flex-wrap: wrap;                  gap: 20px;                  justify-content: space-around;              }              .product-card {                  background-color: white;                  border-radius: 8px;                  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);                  width: 250px;                  padding: 15px;                  text-align: center;              }              .product-card img {                  width: 100%;                  height: 200px;                  object-fit: cover;                  border-radius: 8px;              }              .product-card h2 {                  font-size: 20px;                  margin: 10px 0;              }              .product-card p {                  color: #777;                  font-size: 16px;              }              .product-card .price {                  color: #4CAF50;                  font-weight: bold;              }          </style>      </head>      <body>          <div class="container">              <h1>Our Products</h1>                <div class="product-list">                  <!-- Looping Products -->                  {% for product in products %}                      <div class="product-card">                          <img src="{{ product.image\_url }}" alt="Product Image">                          <h2>{{ product.name }}</h2>                          <p>{{ product.description }}</p>                          <p class="price">${{ product.price }}</p>                      </div>                  {% endfor %}              </div>          </div>      </body>      </html>      """      return render\_template\_string(html\_code, products=products)  if \_\_name\_\_ == '\_\_main\_\_':      app.run(debug=True) |

|  |
| --- |
| name: Deploy to EC2  on:    push:      branches:        - main  jobs:    build-and-deploy:      runs-on: ubuntu-latest      steps:      - name: Checkout code        uses: actions/checkout@v3      - name: Setup SSH        uses: webfactory/ssh-agent@v0.5.4        with:          ssh-private-key: ${{ secrets.EC2\_SSH\_KEY }}      - name: Deploy to EC2        run: |          ssh -o StrictHostKeyChecking=no ec2-user@<Public\_IP\_EC2> << 'EOF'            cd /var/www/html            git pull origin main            systemctl restart apache2          EOF |

1. Masukan kedalam git dan setting secret keynya

