# Create VPC Infrastructure (Task 1)

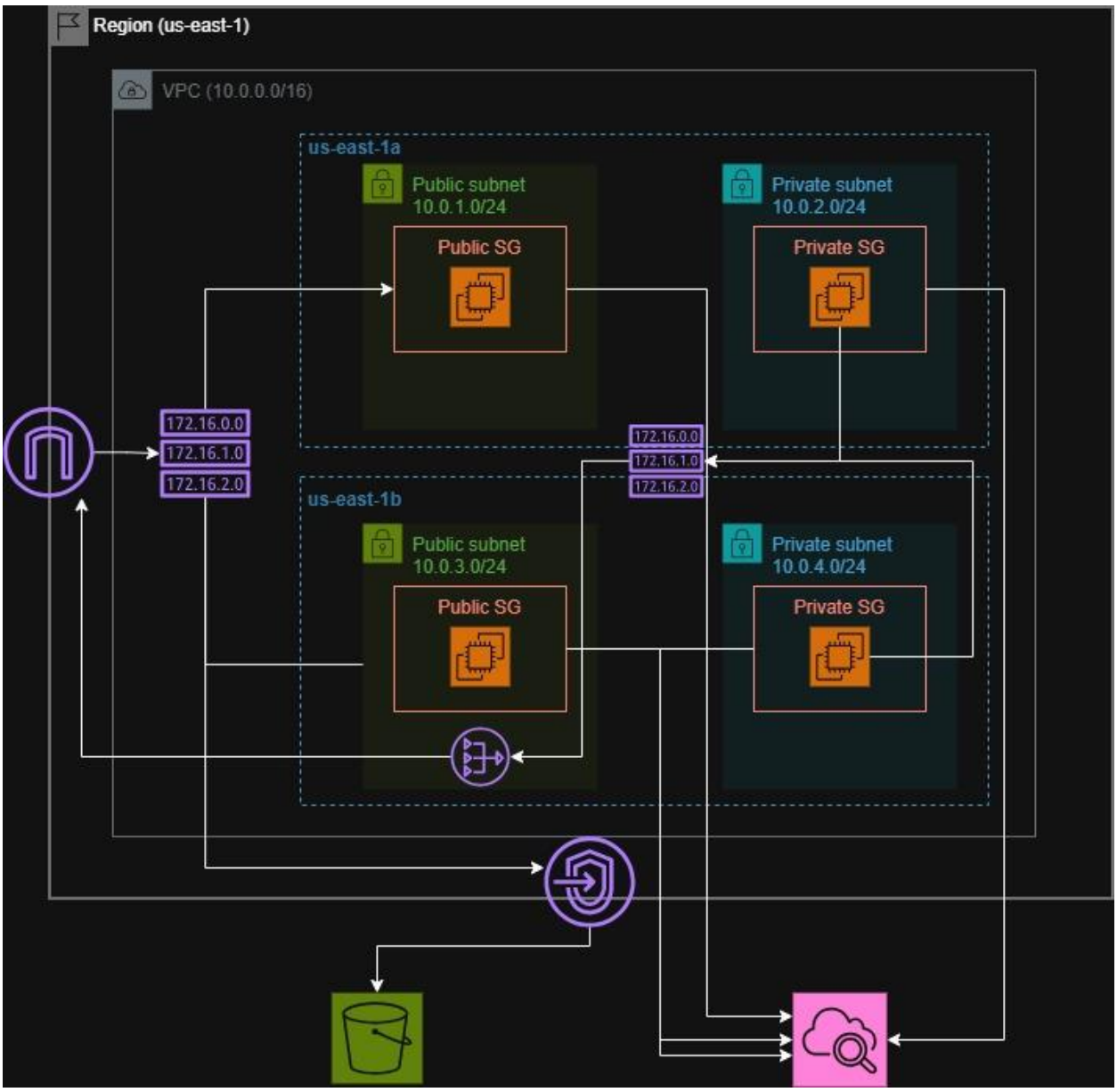

## Zaeem Attique Ashar

## Cloud Intern

### Task Description:

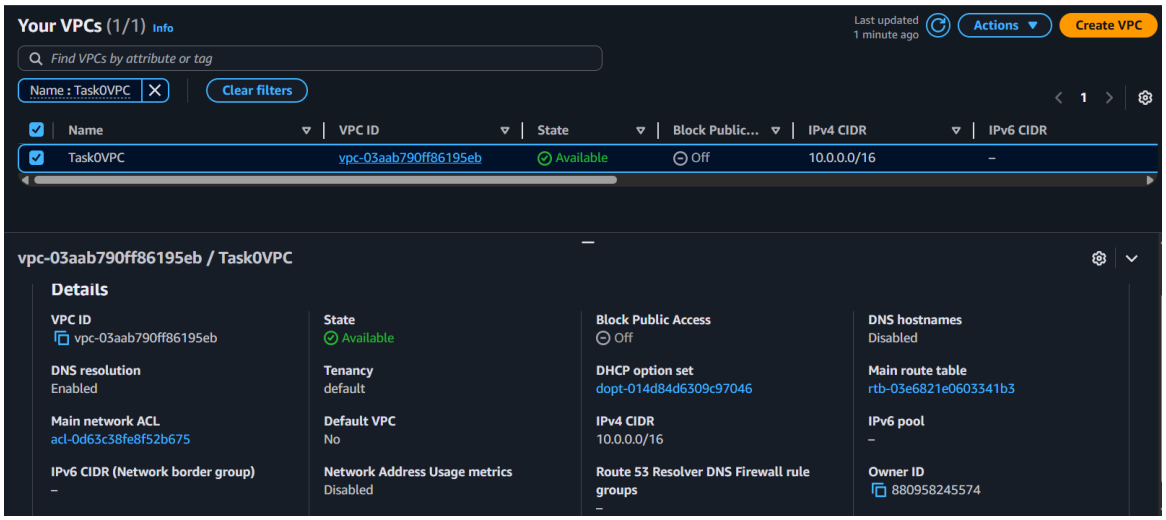
Create a custom Virtual Private Cloud (VPC) setup to host secure and scalable AWS resources across public and private subnets with proper routing, NAT, and security configurations. Also enable VPC Flow Logs for network traffic monitoring and store them in S3 Bucket as well by creating VPC Endpoint.

Architecture Diagram:



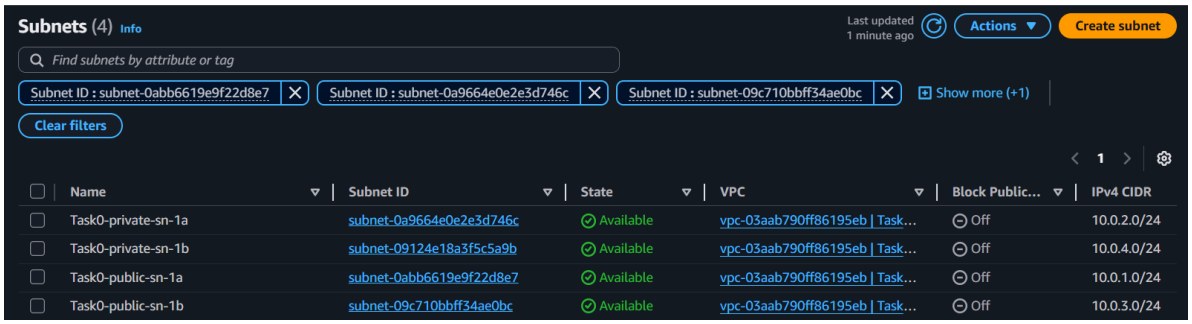
## Task 0.1: Create a custom VPC with a specified CIDR block

- Go to the VPC Dashboard and select the Create VPC button.
- Configuration:
    - Resources to create: VPC only
    - Name tag: Task0VPC
    - IPv4 CIDR block: IPv4 CIDR manual input
    - IPv4 CIDR: 10.0.0.0/16
    - IPv6 CIDR block: No IPv6 CIDR block
    - Tenancy: Default
- Click on create VPC.



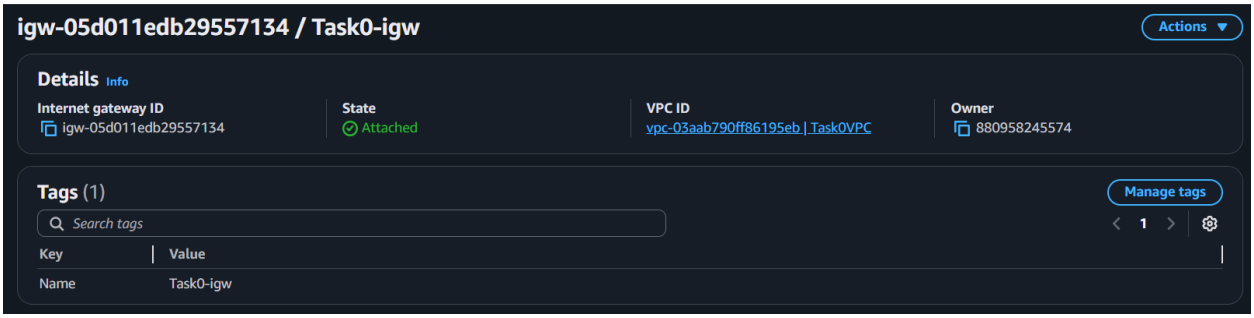

## Task 0.2: Define public and private subnets across multiple Availability Zones

- Go to the subnets tab in the VPC dashboard.
- Click on create subnets button and select the VPC created in the last step.
- Create subnet with the following config:
    - Name: Task0-pub-1a, Zone: us-east-1a, CIDR Block: 10.0.1.0/24
    - Name: Task0-priv-1a, Zone: us-east-1a, CIDR Block: 10.0.2.0/24
    - Name: Task0-pub-1b, Zone: us-east-1b, CIDR Block: 10.0.3.0/24
    - Name: Task0-priv-1b, Zone: us-east-1b, CIDR Block: 10.0.4.0/24



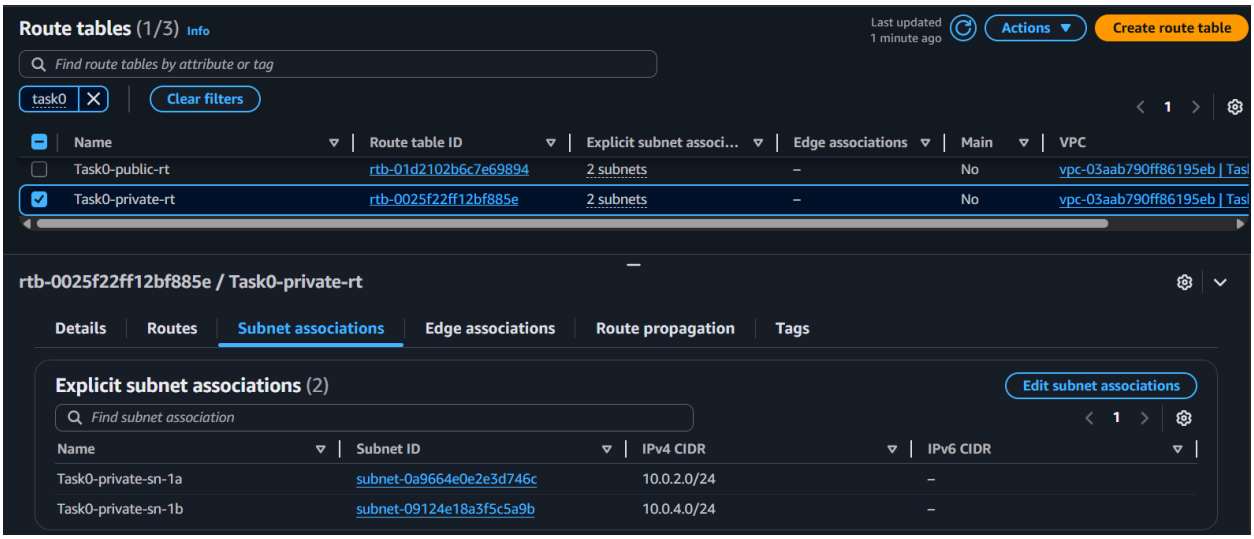

## Task 0.3: Set up an Internet Gateway and attach it to the VPC.

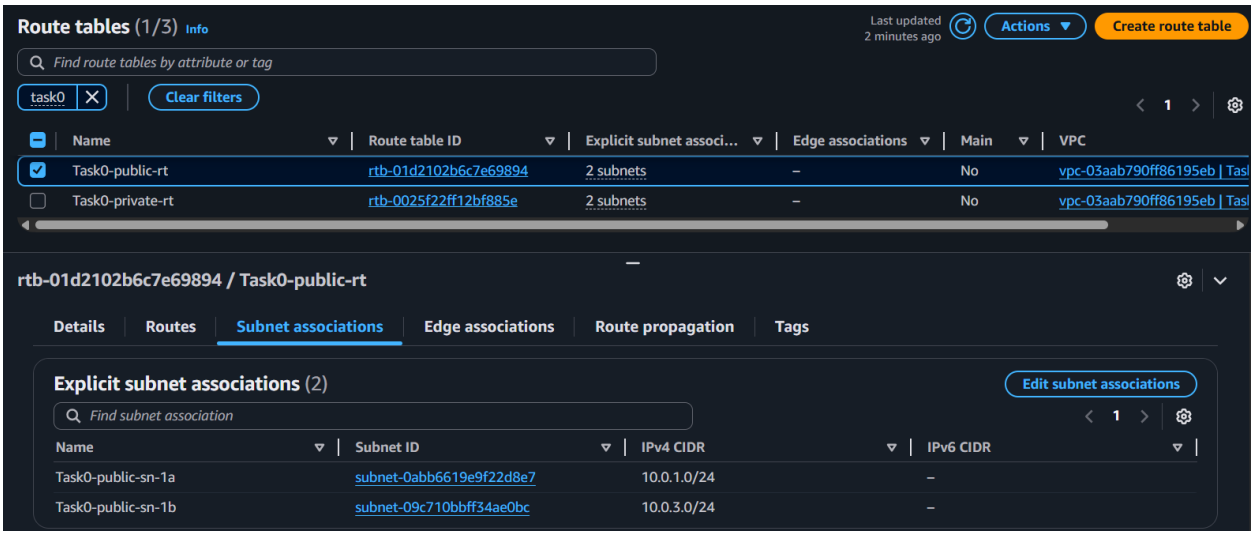
- Go to the Internet Gateways tab in the VPC Dashboard.
- Use the Create VPC button.
- Name your VPC and click Create.
- Now click on the Actions button and choose to attach VPC.
- Select the VPC previously created for the task and click on the attach IGW button.



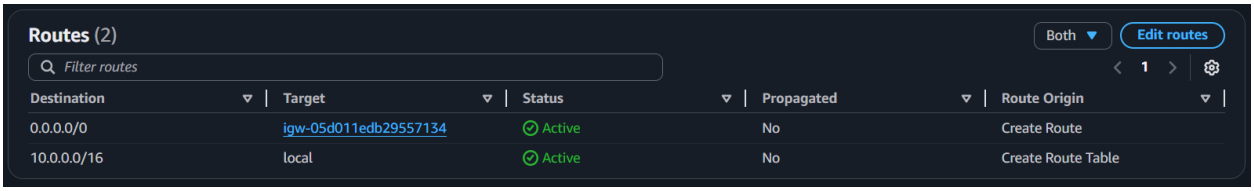

## Task 0.4: Create route tables and associate them with appropriate subnets.

- Go to route tables tab on the VPC Dashboard.
- Click on create route table and name the public route table.
- Select the VPC created for this task.
- Hit create route table.
- Repeat above steps to create a private route table.
- Now associate the public RT with the public subnet and private RT with private subnets.



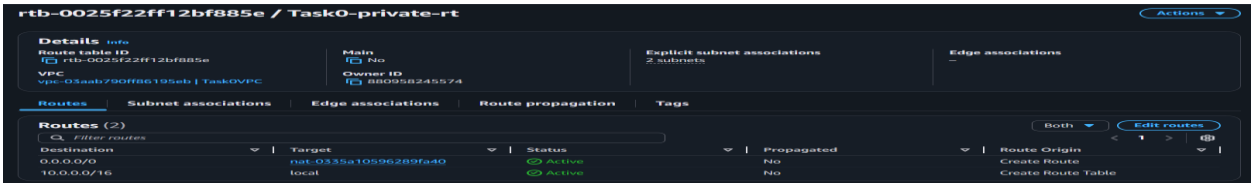




- Go to the public route table and create the following route(s)
    - Instance to IGW:
        - Destination: 0.0.0.0/0
        - Target: igw



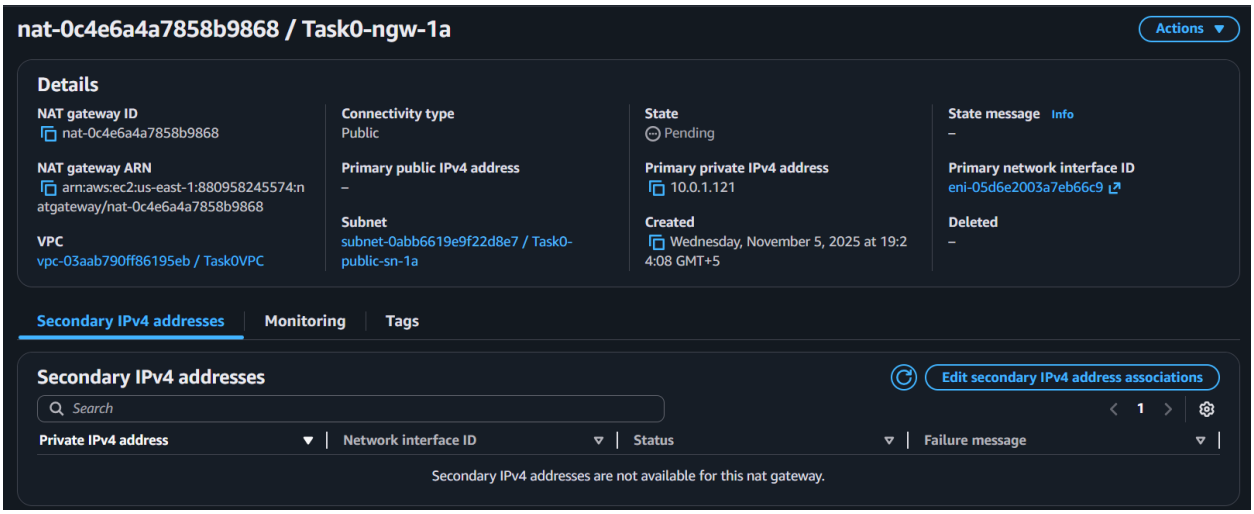

- Go to the private route table and create the following route
    - Instance to NGW:
        - Destination: 0.0.0.0/0

- Target: NGW




## Task 0.5: Configure NAT Gateway in a public subnet for private subnet internet access.

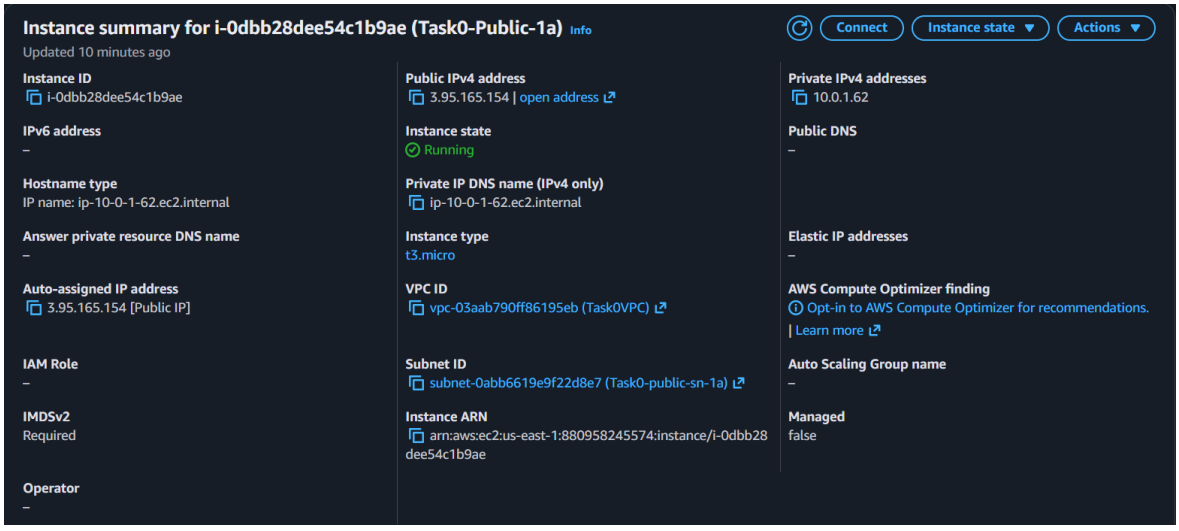
- Go to the NAT Gateway tab in the VPC Dashboard.
- Click on create NAT Gateway, set a name for the NAT Gateway.
- Select the public subnet Task0-public-sn-1a.
- Click on the allocate elastic IP button and then create the NAT Gateway.



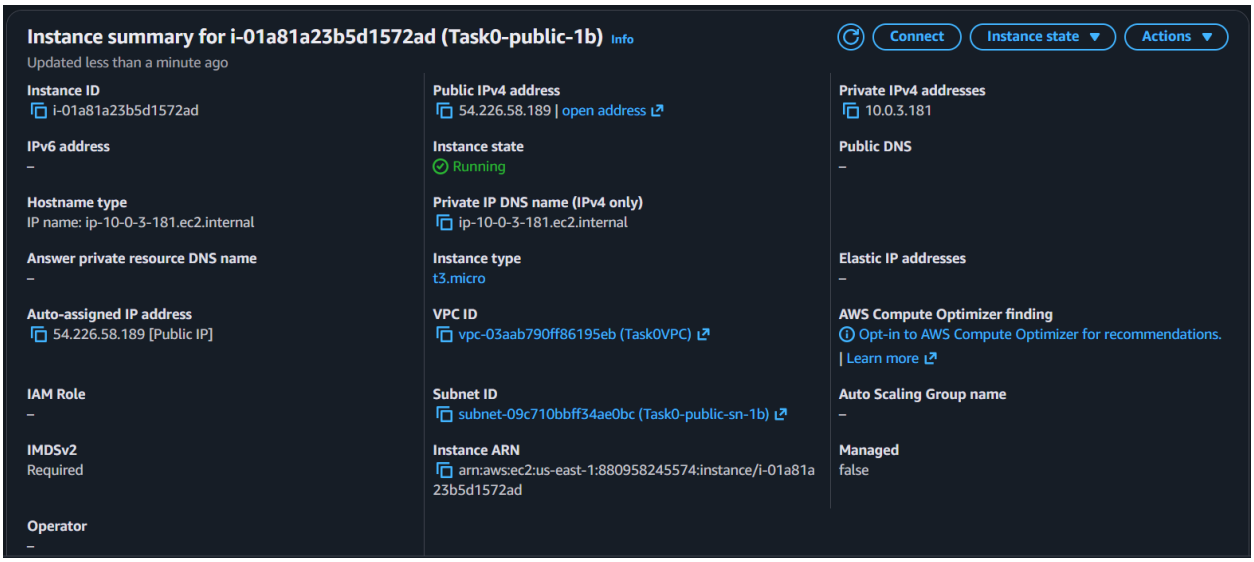

- Follow the same steps to create another NAT Gateway in the Task0-private-sn-1b.


## Task 0.6: Launch EC2 instances in private and public subnets as needed

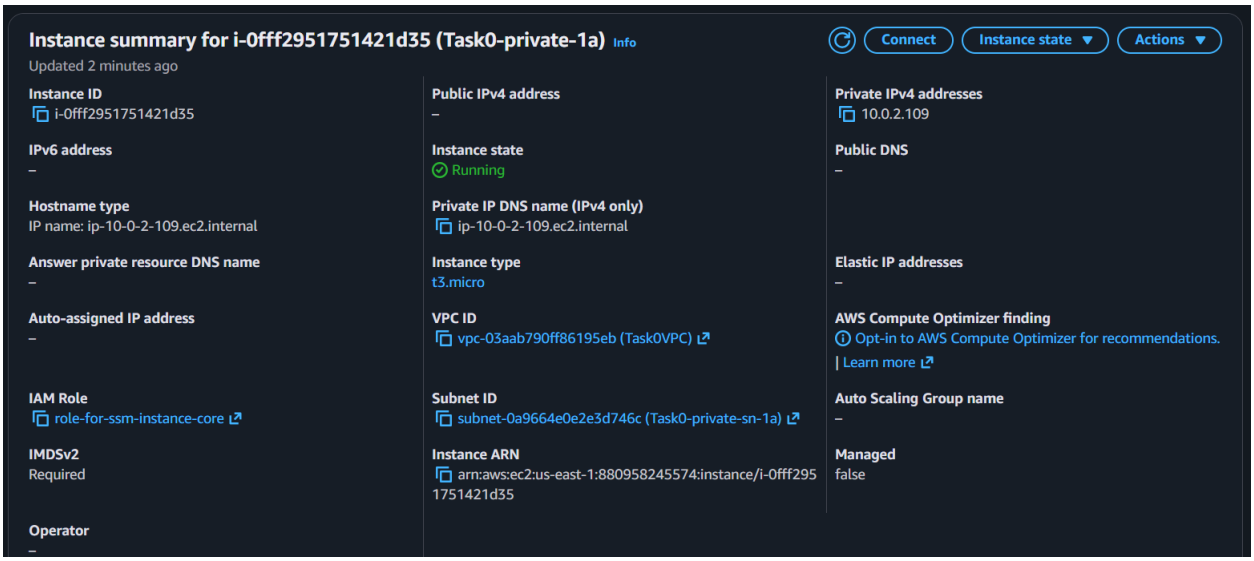
- Go to the EC2 Dashboard and click on the launch instance button.
- Configure the Public instance 1 as follows:
  - Name: Task0-public-1a
  - AMI: Amazon Linux 2023 (x86)
  - Instance type: t3.micro
  - Keypair: Proceed without key pair
  - VPC: Task0VPC
  - Subnet: Task0-private-sn-1a
  - Auto-assign public IP: Enabled
  - Create security group, rule: Type=HTTP, Source Type=Anywhere
  - Storage: 8GiB, gp3
  - Connect via SSH and run command *sudo dnf install nginx*



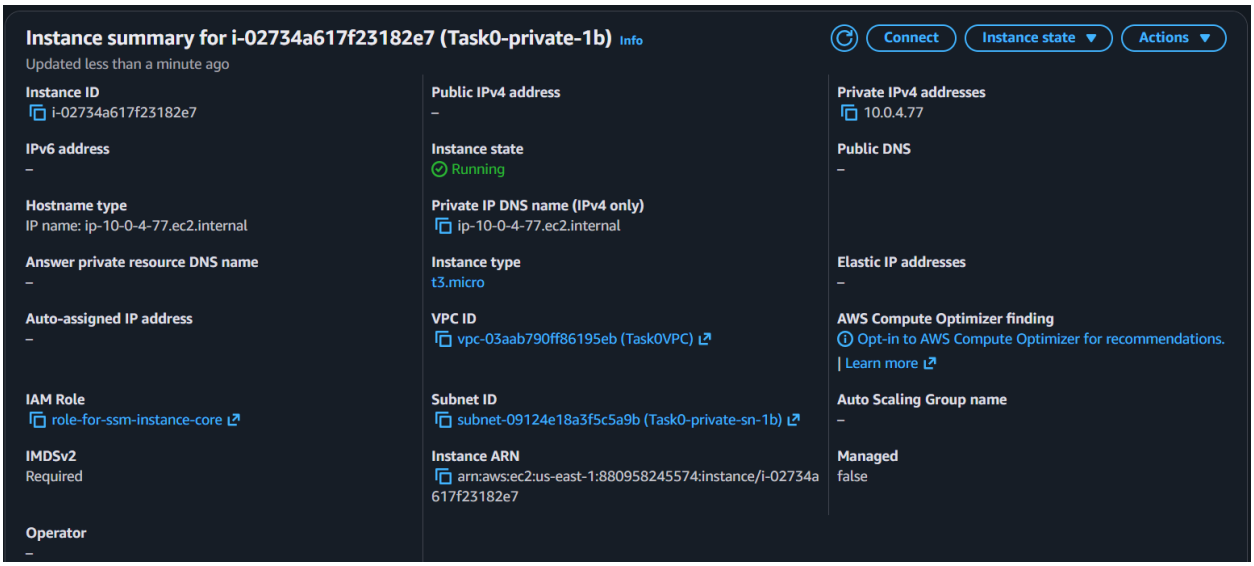
- Configure the Public instance 2 as follows:
    - Name: Task0-public-1b
    - AMI: Amazon Linux 2023 (x86)
    - Instance type: t3.micro
    - Keypair: Proceed without key pair
    - VPC: Task0VPC
    - Subnet: Task0-private-sn-1b
    - Auto-assign public IP: Enabled
    - Create security group, rule: Type=HTTP, Source Type=Anywhere
    - Storage: 8GiB, gp3
    - Connect via SSH and run command *sudo dnf install nginx*




- Configure the Private instance 1 as follows:
    - Name: Task0-private-1a
    - AMI: Amazon Linux 2023 (x86)
    - Instance type: t3.micro
    - Keypair: Proceed without key pair
    - VPC: Task0VPC
    - Subnet: Task0-private-sn-1a
    - Auto-assign public IP: Enabled
    - Create security group, rule: Type=HTTP, Source Type=Anywhere
    - Attach Role: ssmManagedInstanceCore
    - Storage: 8GiB, gp3




- Configure the Private instance 2 as follows:
  - Name: Task0-private-1b
  - AMI: Amazon Linux 2023 (x86)
  - Instance type: t3.micro
  - Keypair: Proceed without key pair
  - VPC: Task0VPC
  - Subnet: Task0-private-sn-1b
  - Auto-assign public IP: Enabled
  - Attach Role: ssmManagedInstanceCore
  - Create security group, rule: Type=HTTP, Source Type=Anywhere
  - Storage: 8GiB, gp3

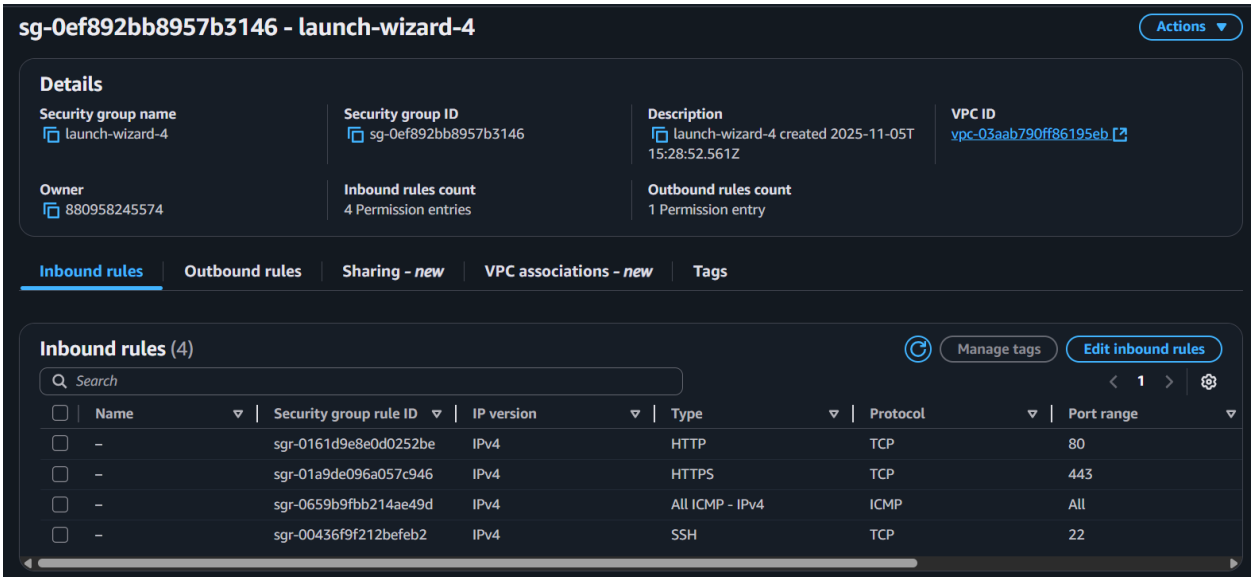


# Task 0.7: Implement security groups and network ACLs for traffic control

Security Group for public instance in SN 1a:

- In the EC2 Dashboard, go to the security groups tab.
- Click on the create security group button.
- Give the new SG a name, Select the VPC: Task0VPC
- Now create the following inbound rules.
    - Type: HTTP, Source: 0.0.0.0/0 (Hosting Web Server)
    - Type: HTTPS, Source: 0.0.0.0/0 (Secure connection to the webserver)
    - Type: ICMP, Source: 0.0.0.0/0 (Test traffic like ping)
    - Type: SSH, Source: 0.0.0.0/0 (Connecting to the machine via SSH)

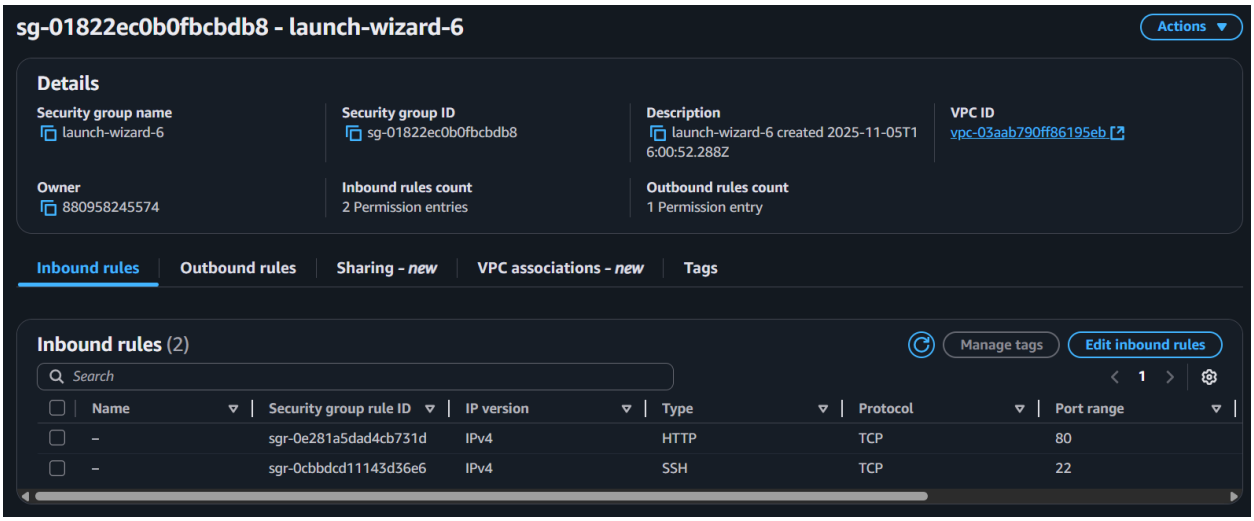
Create the same SG for the public instance in SN 1b.




Security Group for private instance in SN 1a:

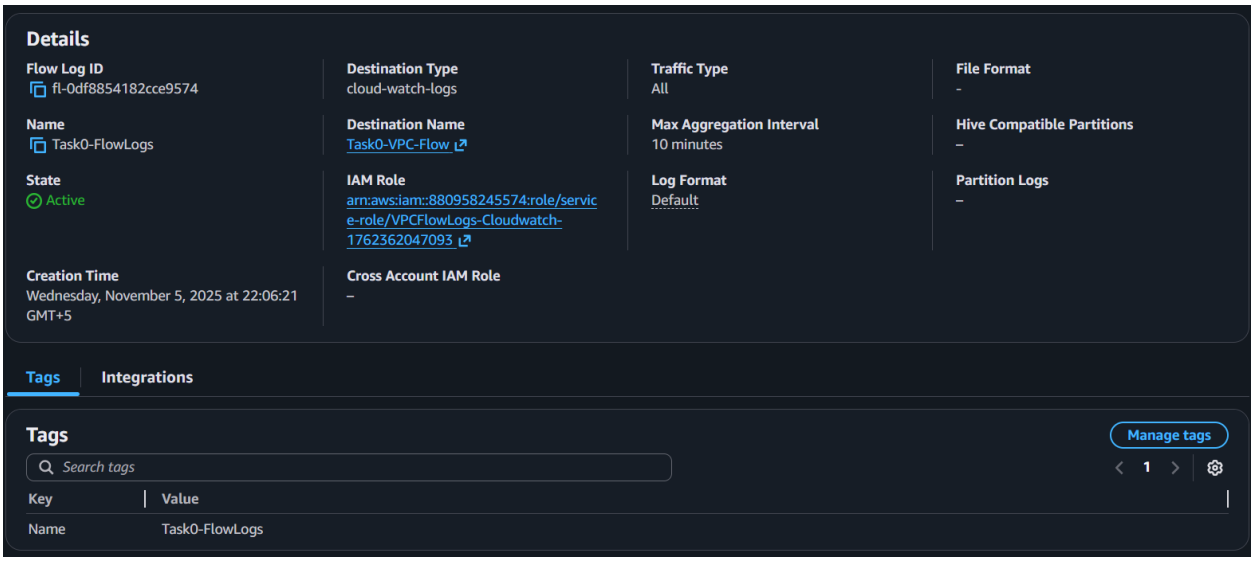
- In the EC2 Dashboard, go to the security groups tab.
- Click on the create security group button.
- Give the new SG a name, Select the VPC: Task0VPC
- Now create the following inbound rules.
    - Type: HTTP, Source: 0.0.0.0/0 (for connecting via SSM)
    - Type: All Traffic, Destination: 0.0.0.0/0 (sending traffic to NGW)

Create the same SG for the private instance in SN 1b



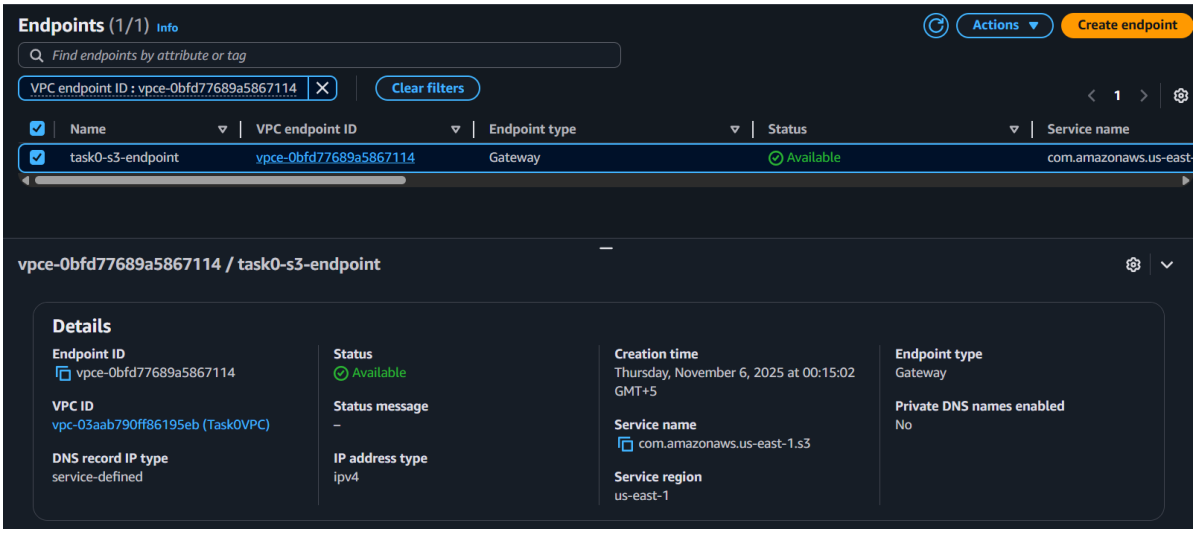

## Task 0.8: Enable VPC Flow Logs for network traffic monitoring

- Go to CloudWatch Dashboard
- Go to logs -> log groups tab.
- Create a new log group, give it the name Task0-VPC-Flow.
- Now go to the VPC Dashboard and open your VPC tab.
- Select VPC Task0VPC from the list.
- Now use the action dropdown and select create flow log.
- Using the following configurations:
    - Name: Task0-FlowLogs
    - Filter: all
    - Maximum aggregation interval: 10 min
    - Destination: Save to CW Logs
    - Destination log group: Task0-VPC-Flow
    - Service access: Create and use a new service role
    - Log record format: AWS default format
- Click on the create Flow Log button.



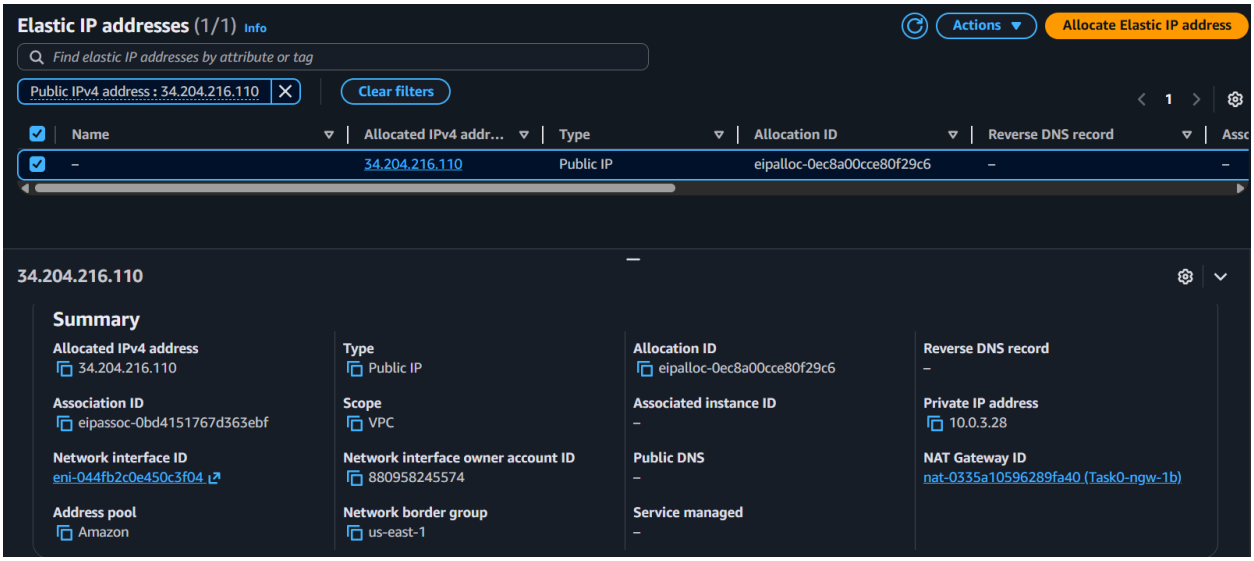

## Task 0.9: Create VPC Endpoints for secure private connectivity to AWS services

- Create an S3 bucket for demonstration purposes.
- Go to the VPC Dashboard and open the Endpoints tab.
- Following is the configuration:
    - Name: task0-s3-endpoint
    - Type: AWS Services
    - Service Name: com.amazonaws.us-east-1.s3 (Gateway)
    - VPC: Task0VPC
    - Route Table: Task0-rt-public
    - Policy: Full Access

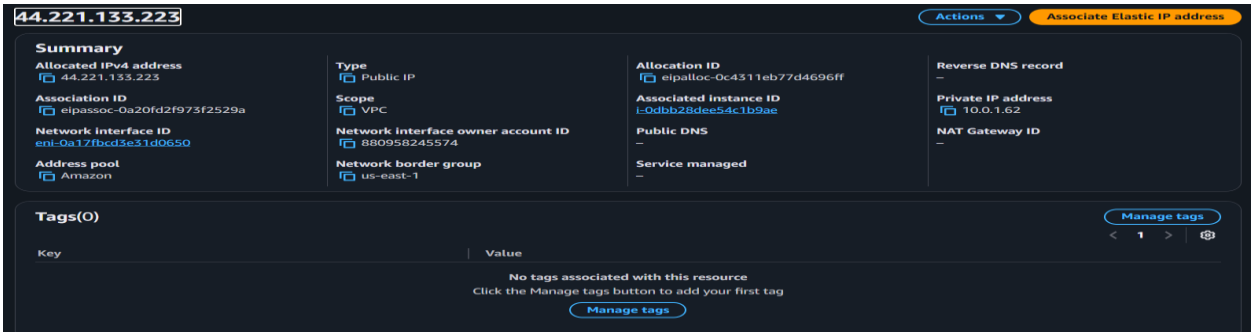


# Task 0.10: Use Elastic IPs for NAT Gateway or public-facing resources

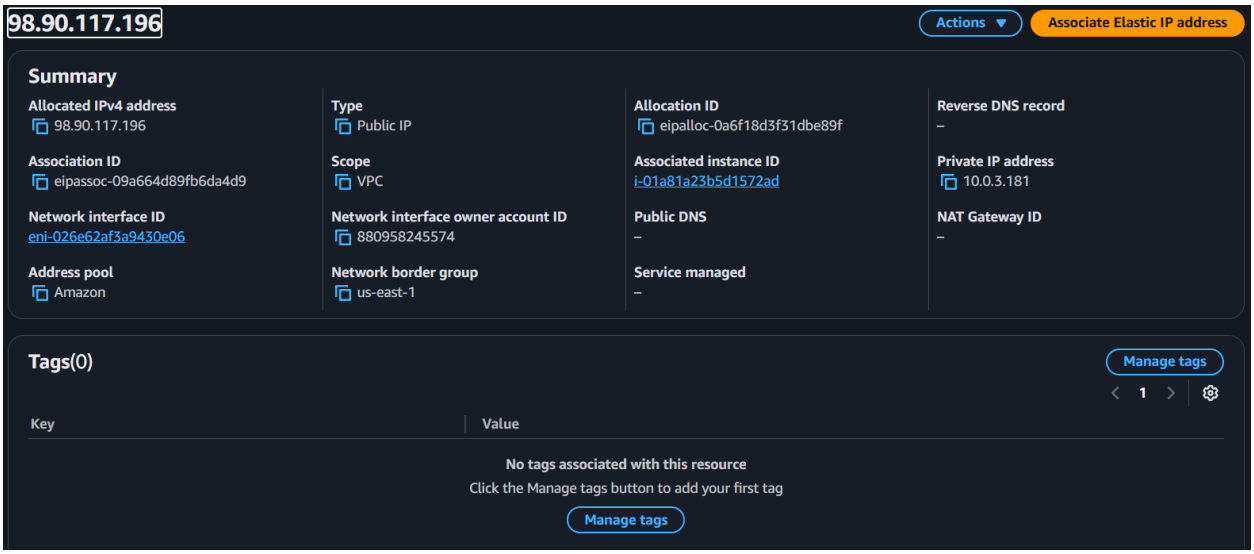
- While creating the NGW, use the allocate elastic IP button to auto-assign EIP.




- Now go to the VPC Dashboard and go to the Elastic IP tab.
- Select the allocate elastic IP button. Continue with the following configuration.
    - Amazon's pool of IPv4 addresses
    - Network Border Group: us-east-1.
- Then create the EIP.
- From the EIP dashboard, go to actions and attach the EIP to a public facing EC2 Instance ().



- Follow the same steps to allocate and attach the EIP to the other public facing EC2 instance.

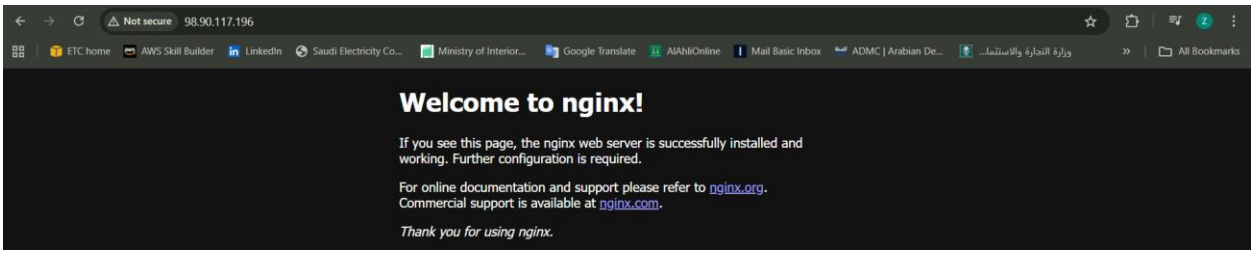



We can now access the NGINX webserver on the EIP.

## Task 0.11: Test connectivity between subnets and to the internet

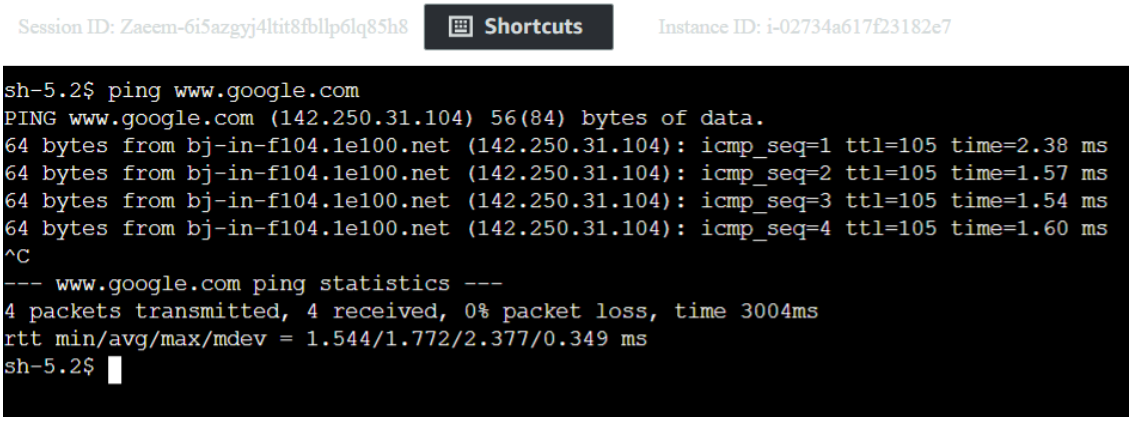
- Connect to public EC2 instance using SSH from UI.
- Ping www.google.com to check internet connectivity via IGW.

- Enter the EIP of public machine to test if HTTP traffic flows via IGW and is allowed by the SG:




The same can be done with the other public EC2 instance to check connectivity.

- Connect to the private EC2 instance using the Session Manager via NGW
- Use the ping command to check if the traffic is flowing through the NGW.




- To check the flow logs, head over to the CloudWatch Dashboard.
- Click on logs and then open the logs group tab.
- Select an interface to check the logs, and they will be displayed.