# Step 1: Launch an EC2 Instance

## Log in to AWS Management Console:

* + Go to the AWS Management Console at https://aws.amazon.com/console/
  + Sign in with your AWS credentials.

## Navigate to EC2 Dashboard:

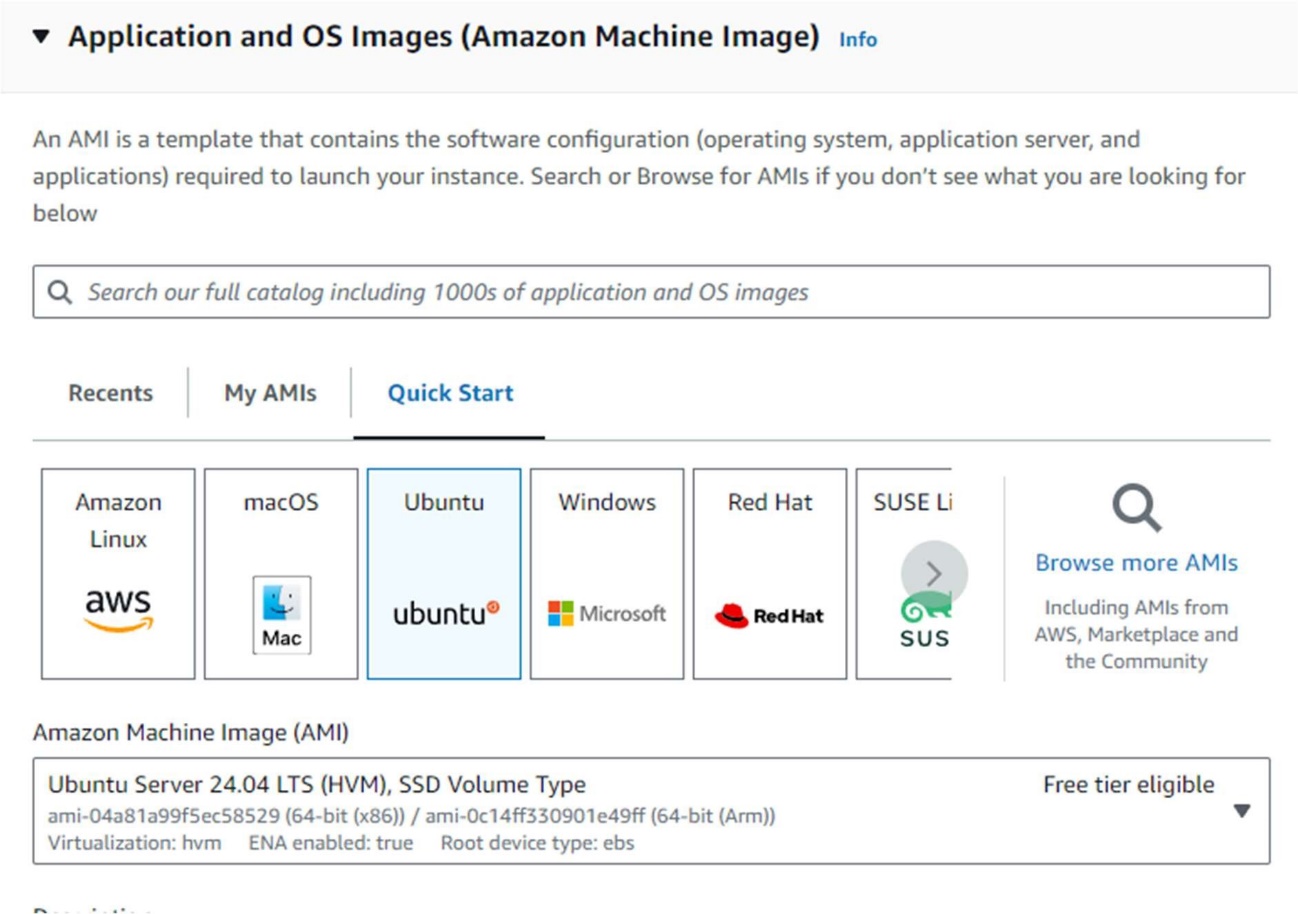
* + In the AWS Management Console, type "EC2" in the search bar and select EC2 to navigate to the EC2 Dashboard.

## Launch an Instance:

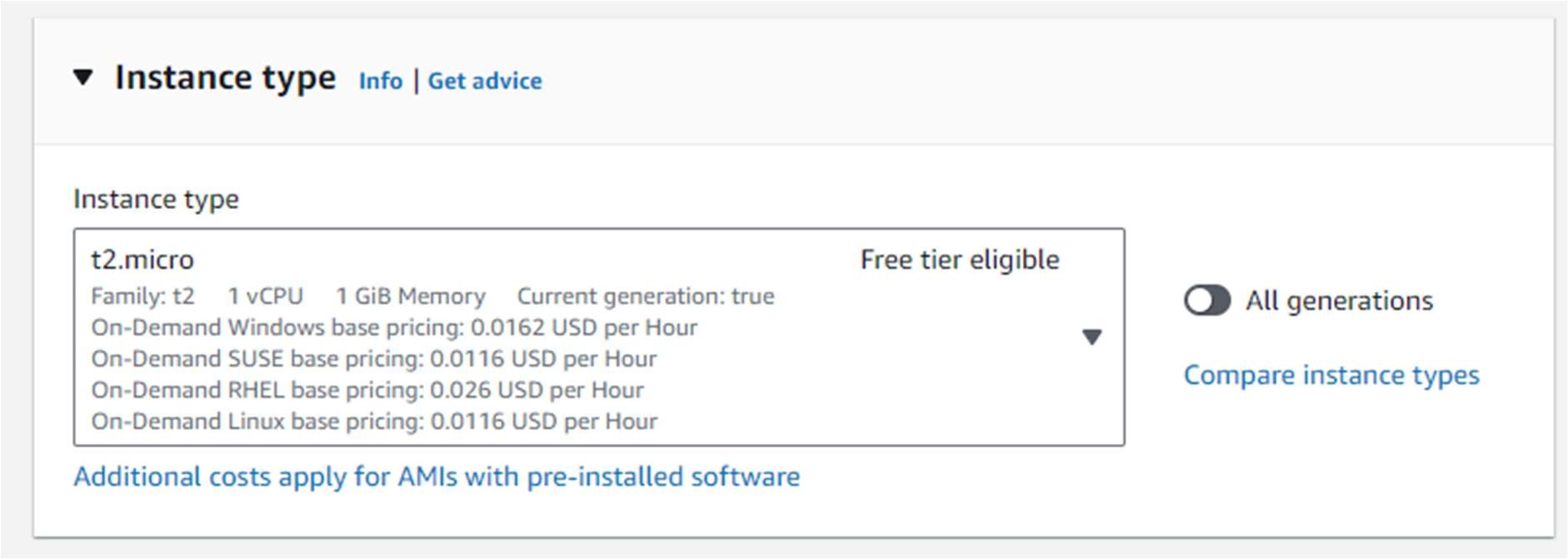
* + Click on the "Launch Instance" button.



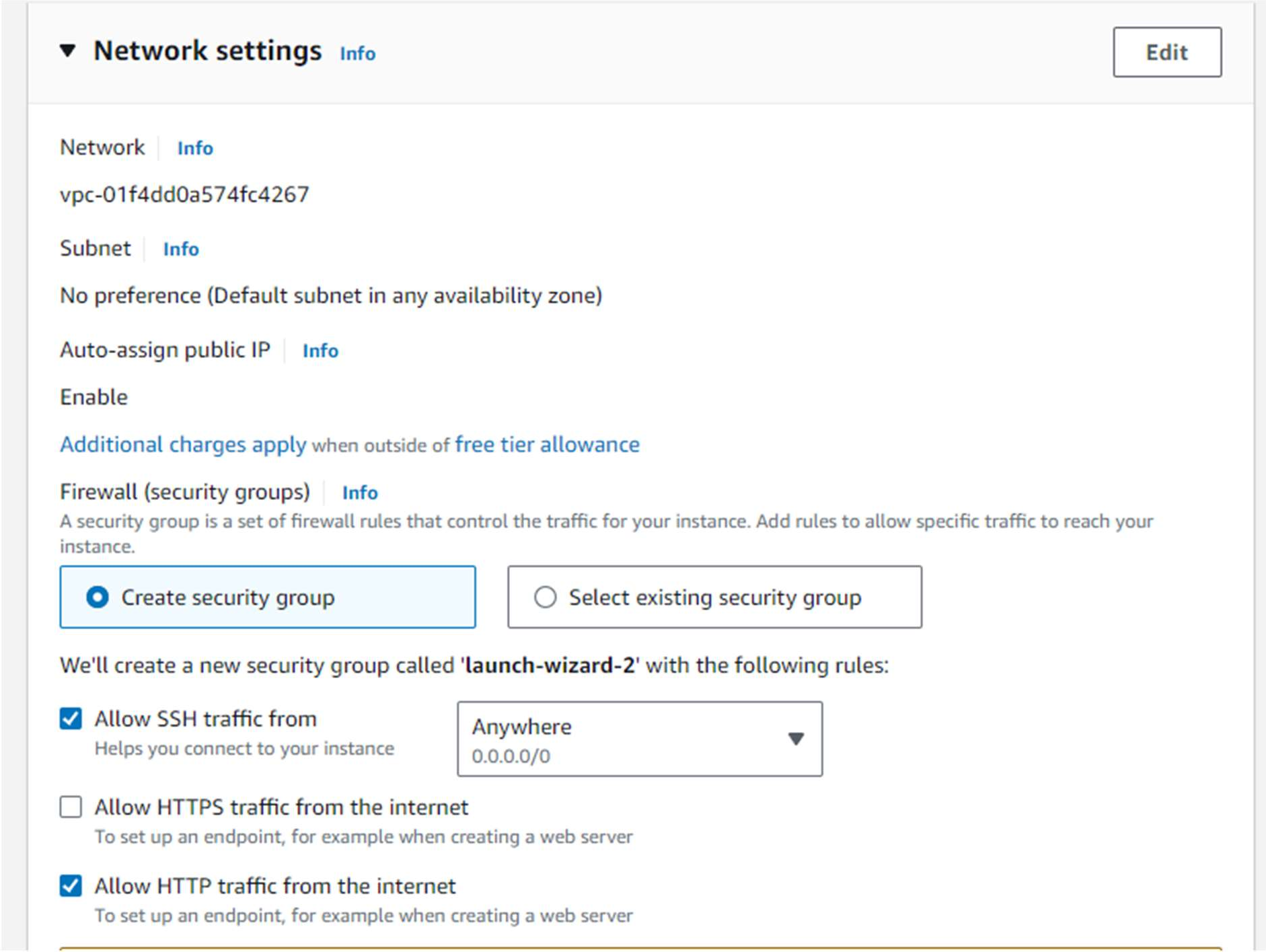
* + Choose an Amazon Machine Image (AMI): Select "Ubuntu Server 20.04 LTS (HVM), SSD Volume Type".



* + Choose an Instance Type: Select t2.micro (eligible for the free tier).



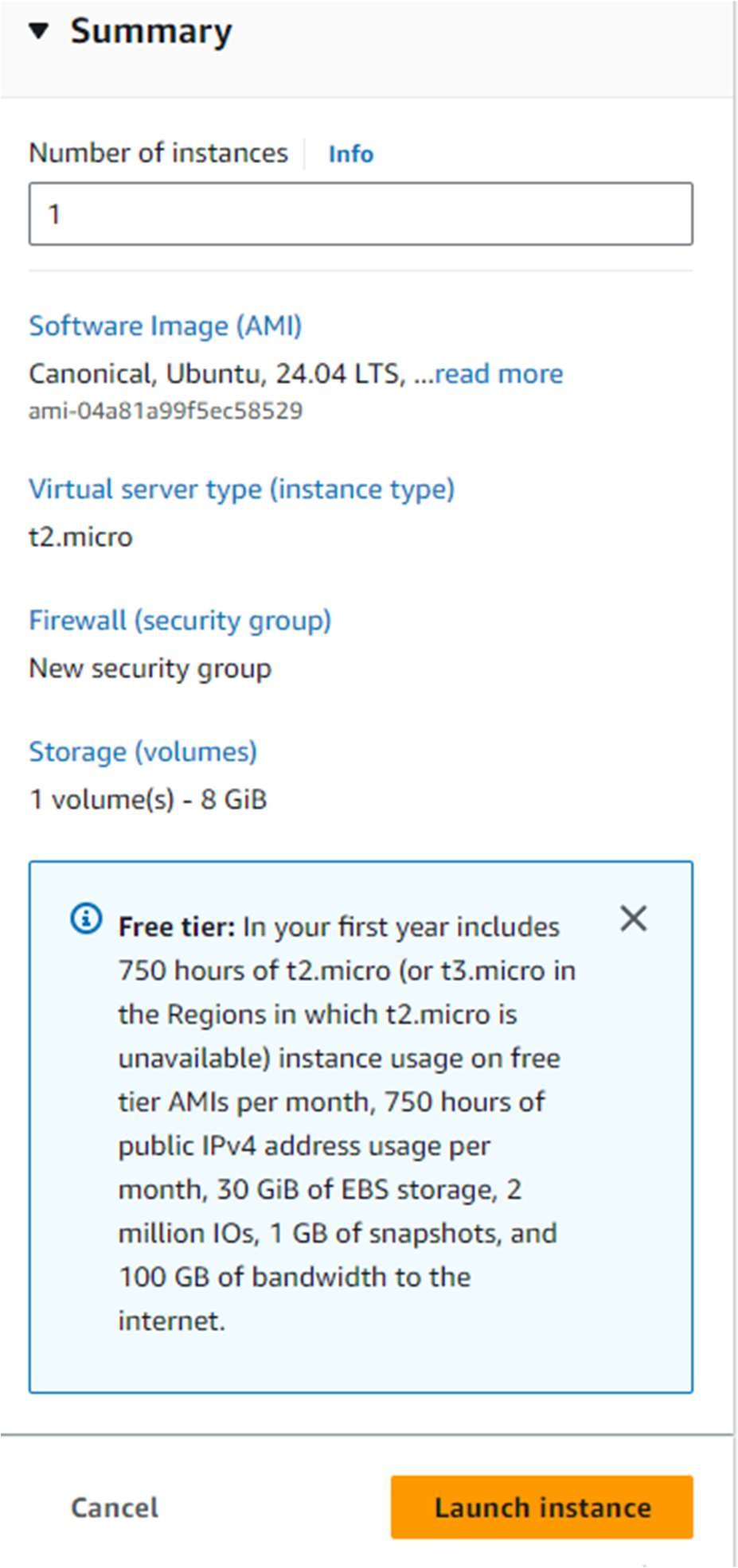
* + Configure Instance:
    - Select an existing key pair or create a new one.
    - Network: Choose the default VPC.
    - Subnet: Choose a subnet in the US-East-1 (N. Virginia) region.
    - Enable Auto-assign Public IP.



* + Add Storage: Keep the default settings.
  + Add Tags: Add a tag to identify your instance (e.g., Key: Name, Value: Nginx).

## Review and Launch:

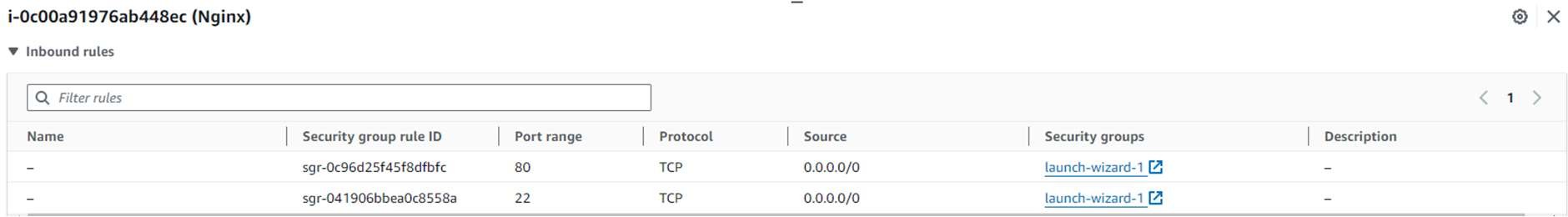
* + Review your instance settings and click "Launch".





## Configure Security Group:

* + Add a new security group with the following rules:
    - Type: HTTP, Protocol: TCP, Port Range: 80, Source: 0.0.0.0/0
    - Type: SSH, Protocol: TCP, Port Range: 22, Source: 0.0.0.0/0



# Step 2: Connect to Your Instance

## Connect to the EC2 Instance:

* + In the EC2 Dashboard, select your instance.
  + Click on "Connect" and follow the instructions to connect to your instance using SSH.

# Step 3: Install Apache and PHP

1. **Update the package index**:

sudo apt update -y

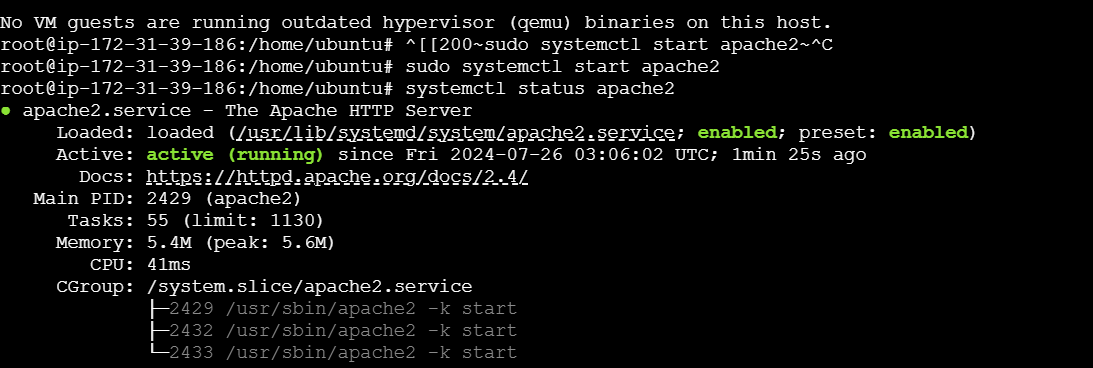
1. **Install Apache**:

sudo apt install apache 2 -y

1. **Start Apache**:

sudo systemctl start apache2

sudo systemctl enable apache2



1. **Install PHP**:

sudo apt install php php-mysql -y

1. **Restart Apache** to apply PHP installation:

sudo systemctl restart apache2

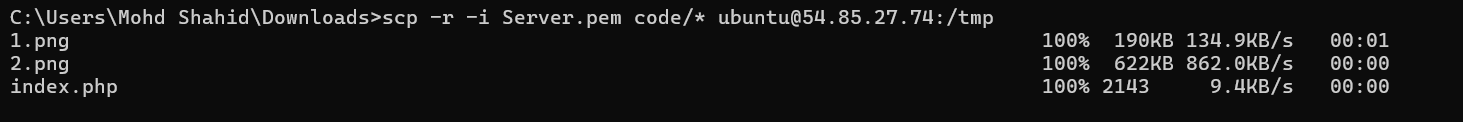
# Step 4: Create an RDS Instance

1. **Navigate to RDS Dashboard**:
   * Click on **Create Database**.
   * Choose **Standard Create**.
   * Select **MySQL**.
   * Choose a DB instance class (e.g., db.t3.micro).
   * Set storage and other configurations.
   * In the **Settings** section:
     + DB instance identifier: my-rds-instance.
     + Master username: intel.
     + Master password: intel123.
   * Configure additional settings (VPC, subnet, security groups).
2. **Create the RDS instance**.

# Step 5: Upload Website Files

1. **Upload your PHP website files** to the Apache document root:
   * Delete the default index file.
   * The default document root is /var/www/html/.
   * You can use SCP or any other method to transfer files. For example, using SCP:

scp -r -i your-key.pem path-to-your-local-files/\* ec2-user@your-ec2-public-ip:/tmp



* + Then move all the file into var/www/html

mv \* /var/www/html



# Step 6: Create Database & Table in RDS instance

1. **Connect to the RDS Instance**:
   * Obtain the endpoint from the RDS dashboard.
   * Connect MySQL:

mysql -h <RDS\_ENDPOINT> -u admin -p

1. **Create the Database and Table**:

CREATE DATABASE intel;

USE intel;

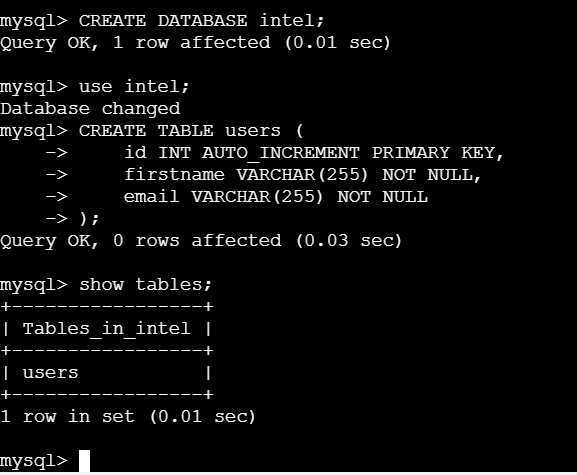
CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

firstname VARCHAR(255) NOT NULL,

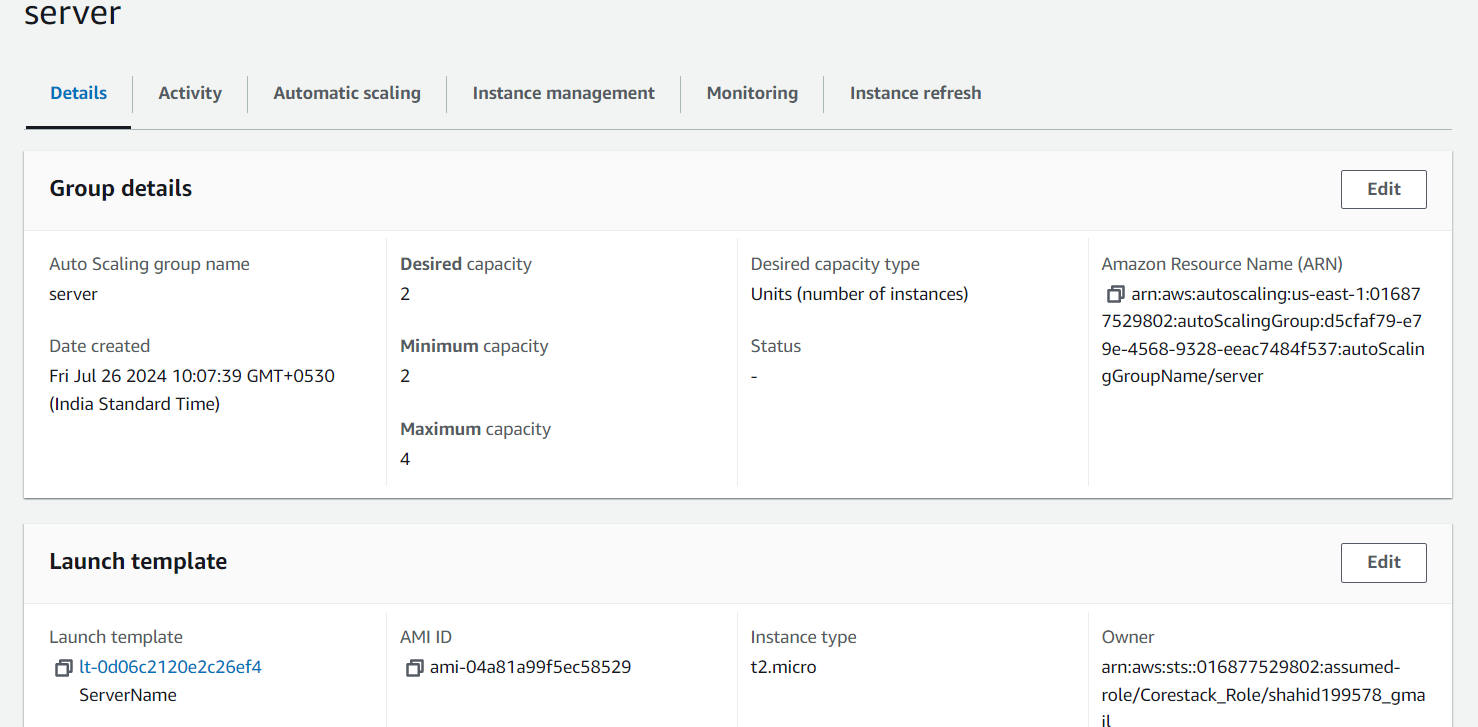
email VARCHAR(255) NOT NULL

);

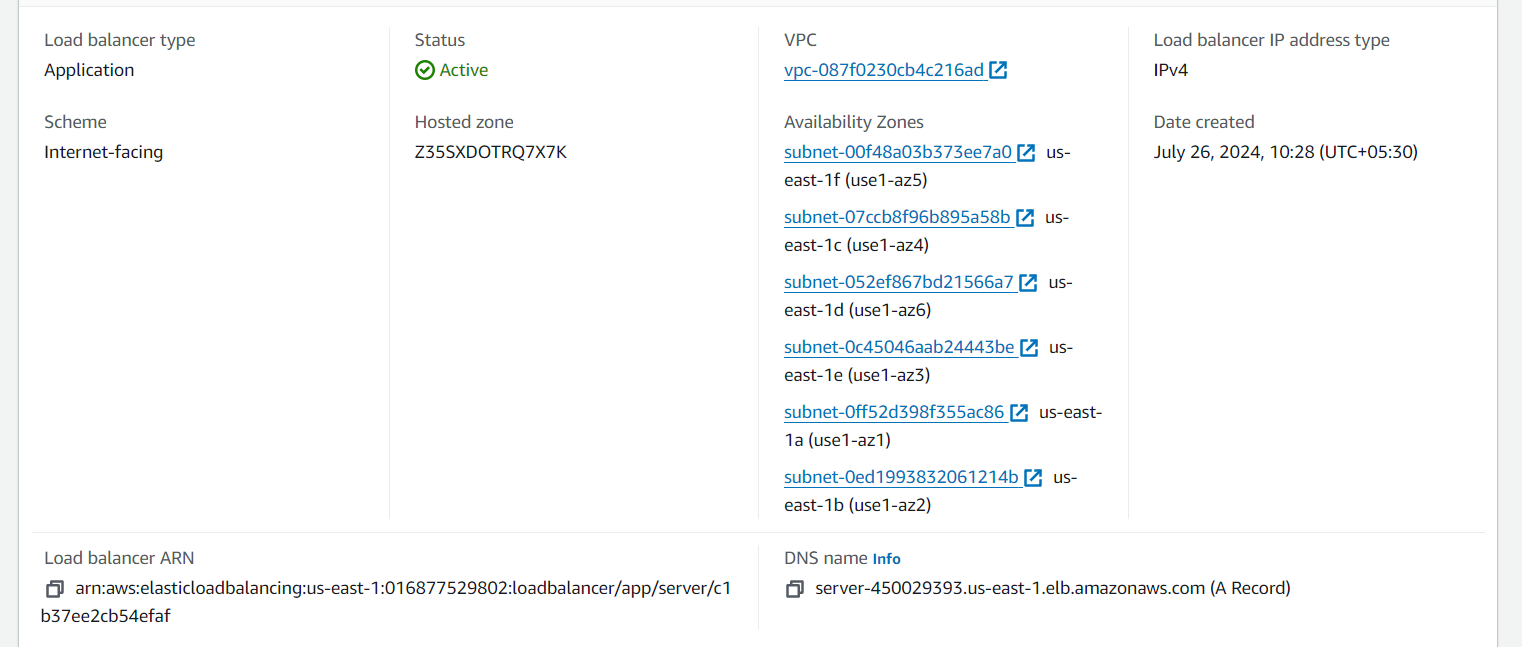


# Step 7: Enable Auto Scaling on These Instances (Minimum 2)

1. **Create a Launch Template**:
   * Navigate to **Launch Templates** in the EC2 dashboard.
   * Click on **Create launch template**.
   * Fill in template details and instance configuration.
   * Ensure to use the same AMI, instance type, and security group as your manually launched instance.
2. **Create an Auto Scaling Group**:
   * Navigate to **Auto Scaling Groups**.
   * Click on **Create Auto Scaling group**.
   * Choose your launch template.
   * Set the desired capacity to 2, minimum capacity to 2, and maximum capacity to 4.
   * Configure network and subnets.
   * Set up scaling policies (optional).

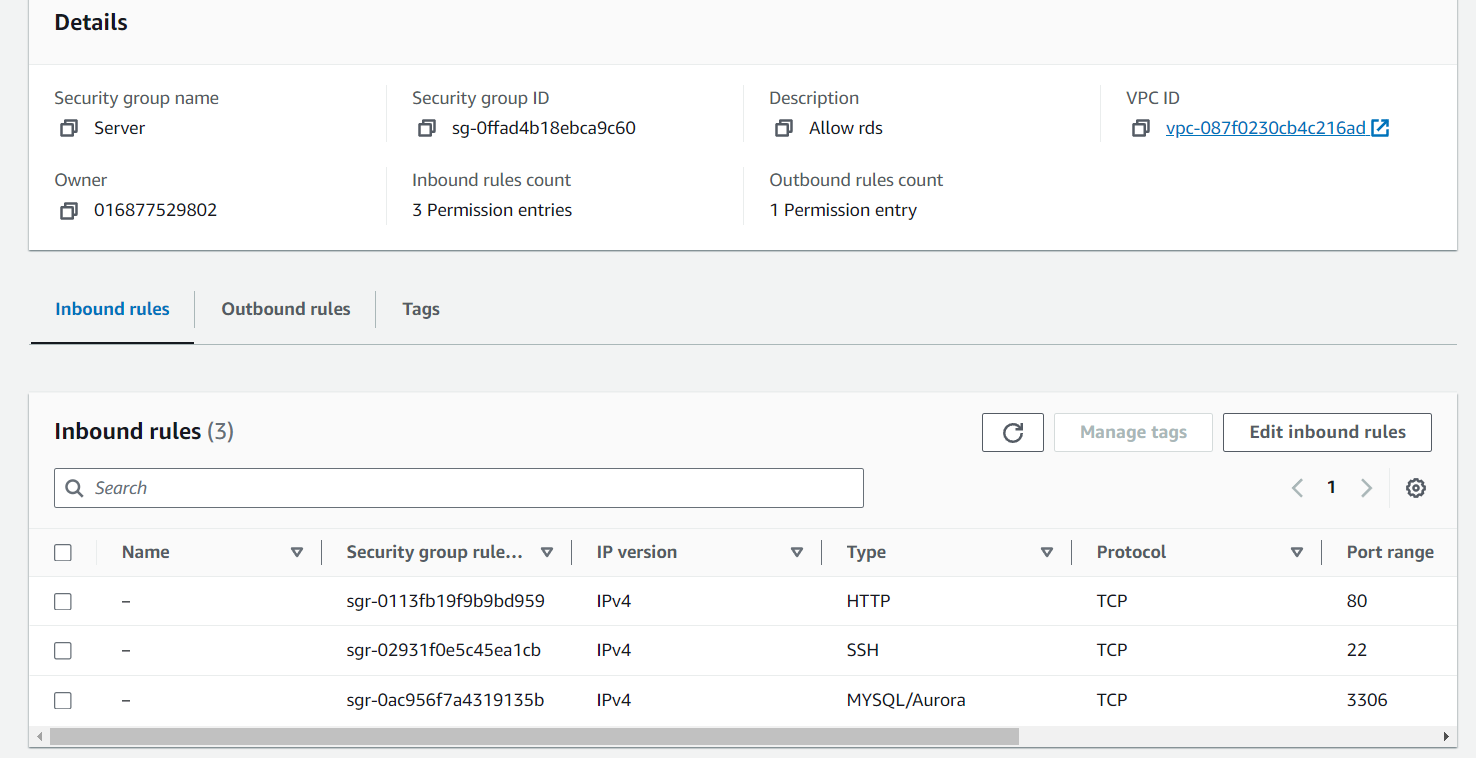


# Step 8: Create a Load Balancer

1. **Navigate to the EC2 Dashboard**:
   * Click on **Load Balancers** under the Load Balancing section.
   * Click on **Create Load Balancer**.
   * Choose **Application Load Balancer**.
   * Configure the load balancer:
     + Name: my-load-balancer.
     + Scheme: Internet-facing.
     + Listeners: HTTP (port 80).
     + Availability Zones: Select the VPC and subnets.
2. **Configure Security Groups** for the load balancer:
   * Ensure it allows HTTP traffic.
3. **Configure Routing**:
   * Create a target group:
     + Name: my-target-group.
     + Target type: Instances.
     + Protocol: HTTP.
     + Port: 80.
     + Health checks: HTTP.
   * Register your instances in the target group.
4. **Review and Create** the load balancer.

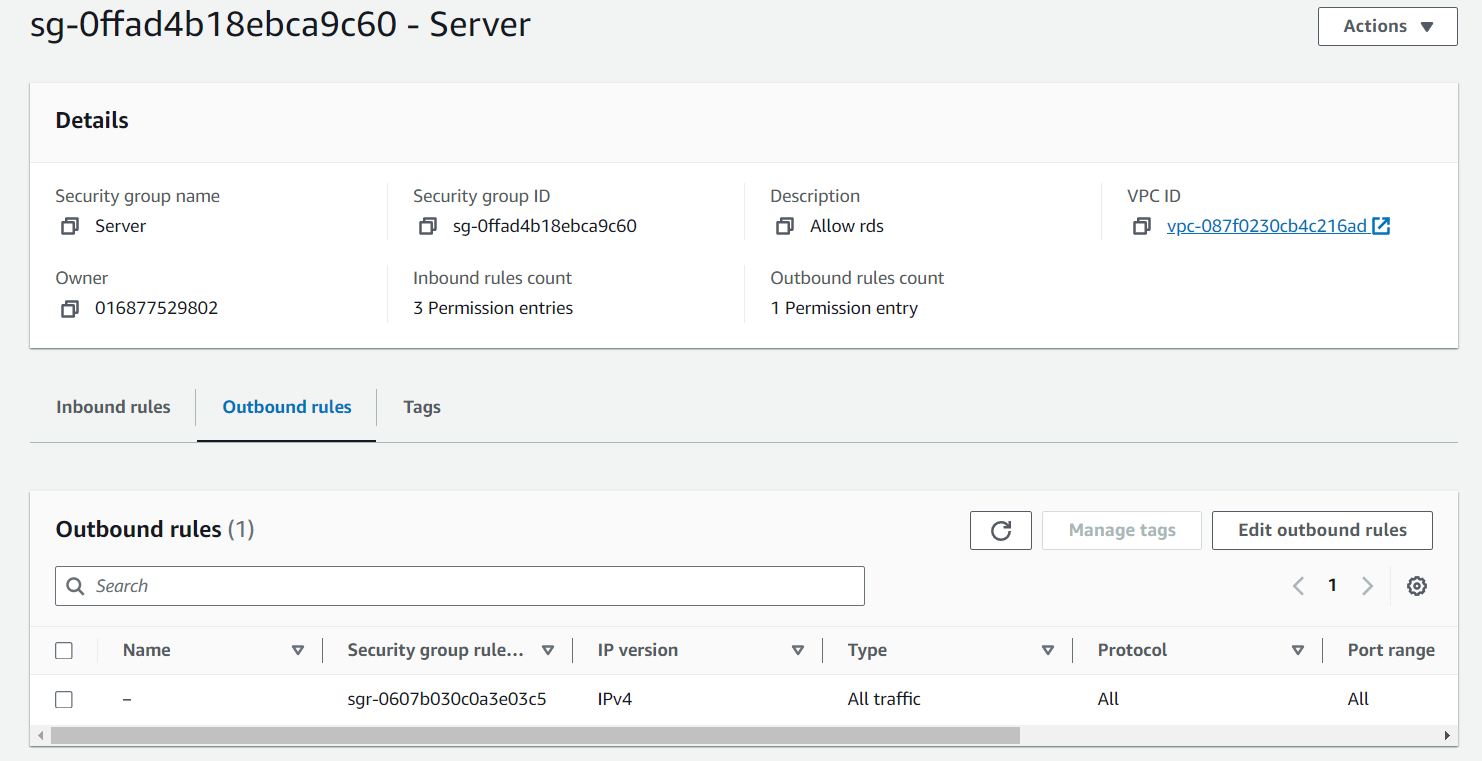
# Step 9: Allow Traffic from EC2 to RDS Instance

1. **Modify RDS Security Group**:
   * Go to the **RDS dashboard**, select your instance.
   * Click on **Modify** > **Security Groups**.
   * Add a rule to allow inbound MySQL/Aurora traffic (port 3306) from the EC2 instance's security group.



**2. Allow All Traffic to EC2 Instance**

1. **Modify EC2 Security Group**:
   * Go to the **EC2 dashboard**, select your instance.
   * Click on **Security Groups**.
   * Edit inbound rules to allow all traffic:
     + Custom TCP Rule, Source: 0.0.0.0/0 (All traffic).



# Step 11: Final Steps

1. **Test the Configuration**:
   * Ensure the website is accessible via the domain name.