**AWS Employee Profile Management System**

**Application**: Employee profile of XYZ company – New employees input their information and upload photos. Existing employees can get their information.

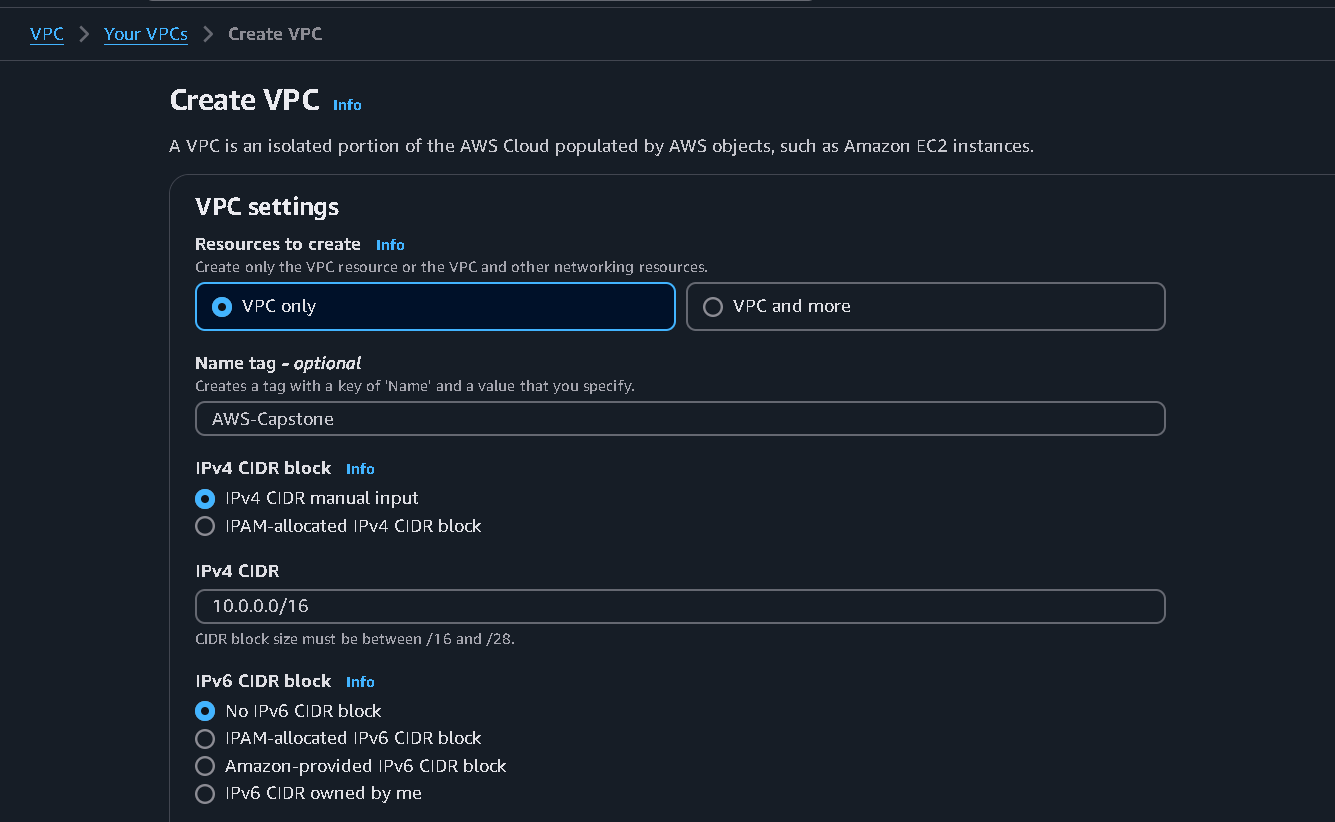
Solution:

Step 1:

Setup VPC for Load Balancer, Application EC2 instance and RDS Database - one public and two private subnets.

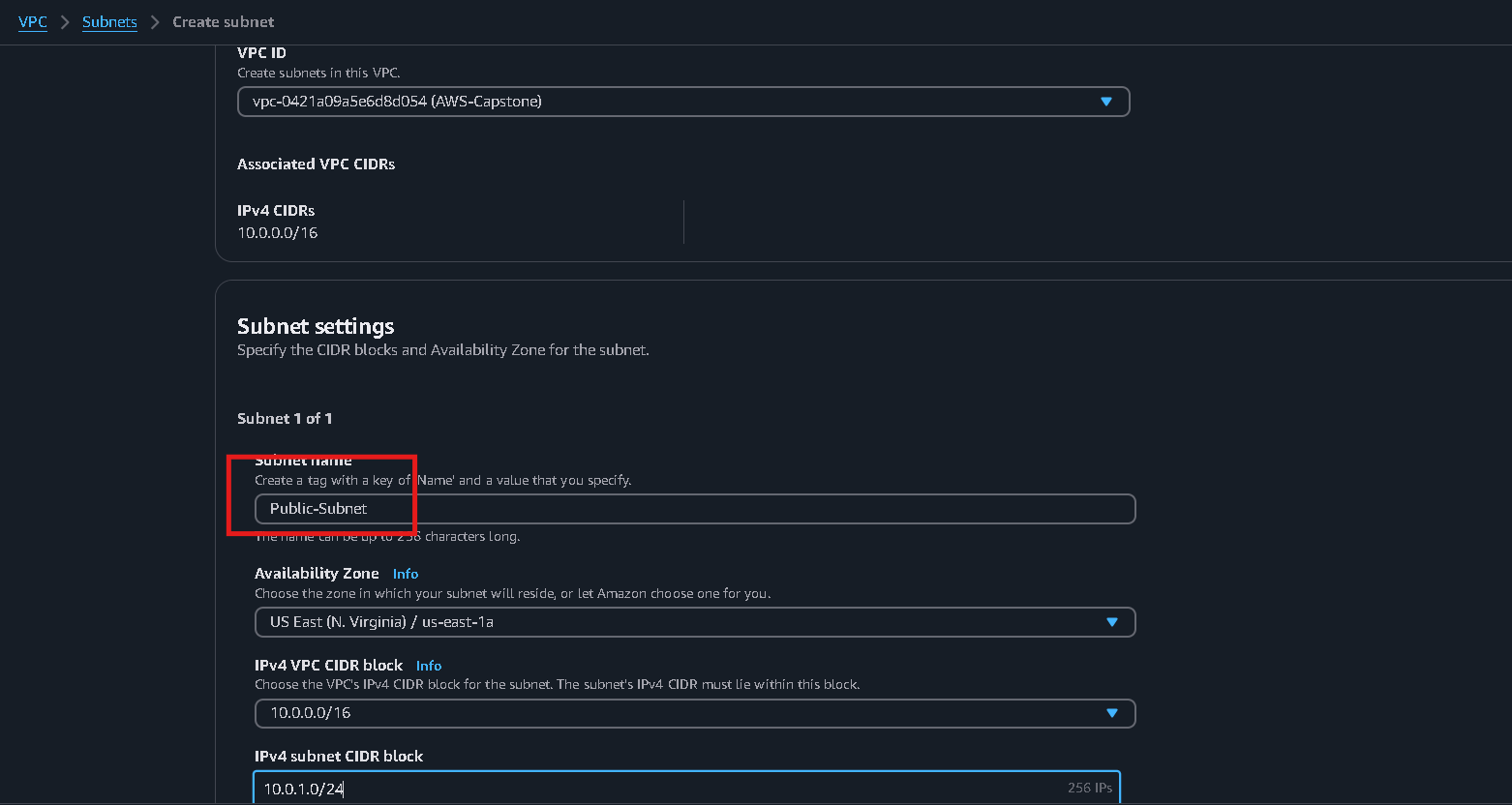
**\*Create a VPC**

* Go to the **VPC Dashboard** in the AWS Management Console.
* Click **Create VPC**.
* Choose the default IPv4 CIDR block (e.g., 10.0.0.0/16).
* Leave the other settings as default (you can customize them later if needed).

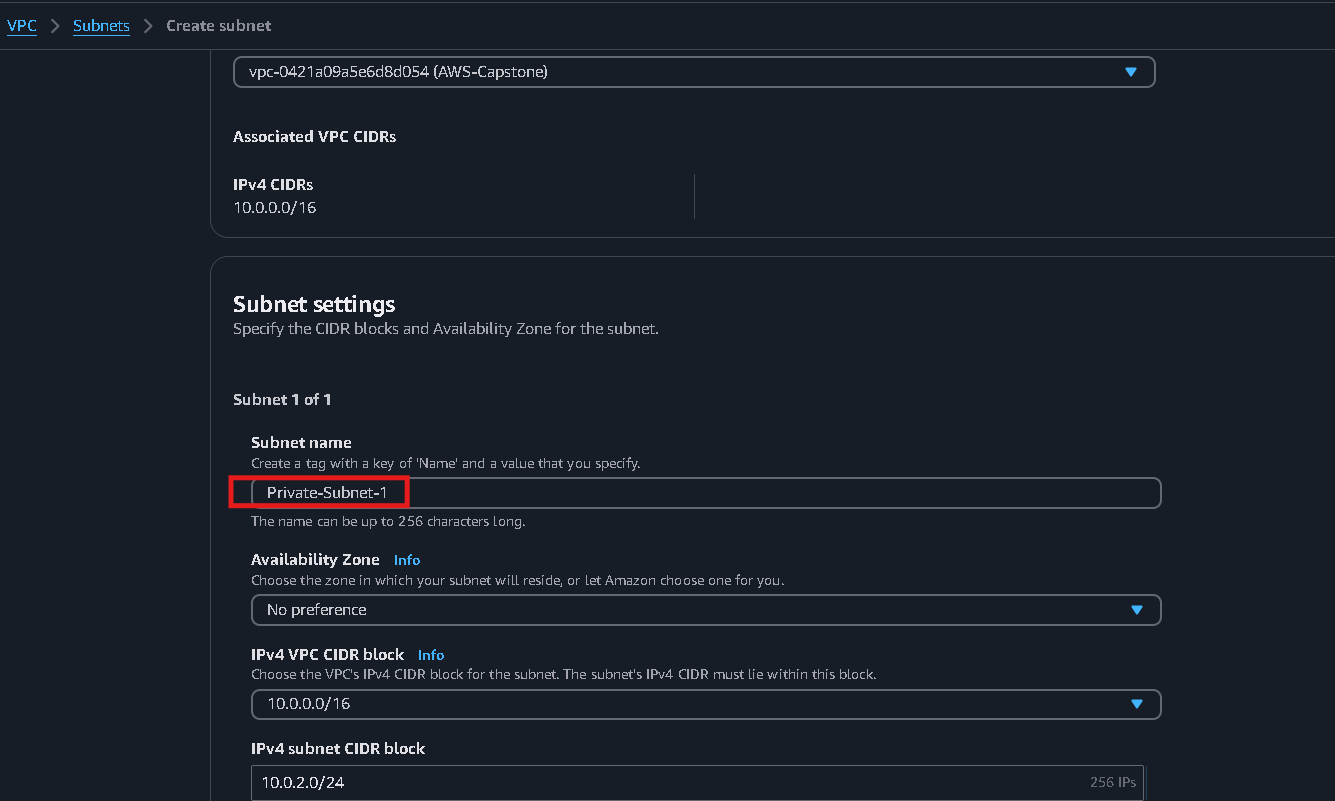


**\*Create Subnets**

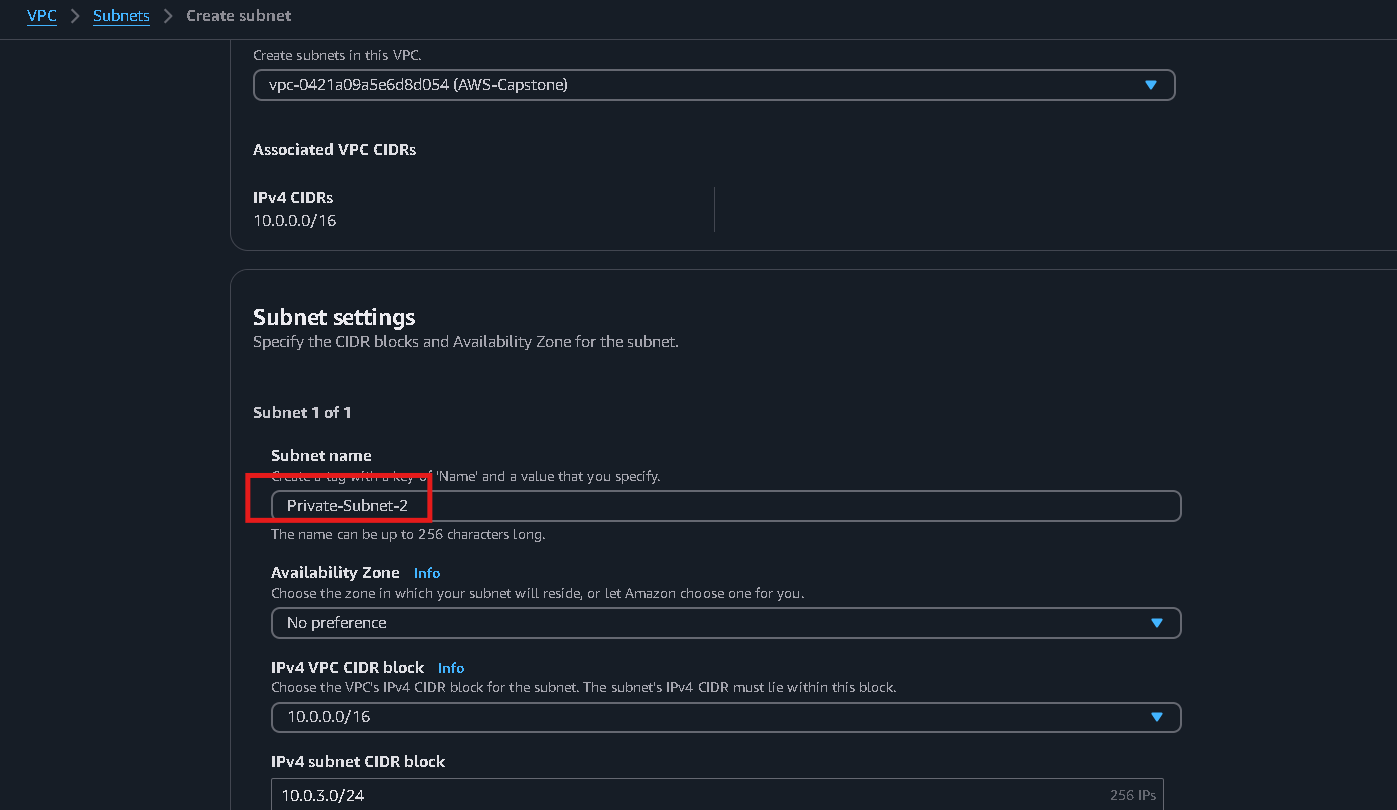
* Go to the **Subnets** section in the VPC Dashboard.
* Click **Create subnet**.
* **Public Subnet**:
  + Name: Public-Subnet
  + Availability Zone: Choose any AZ.
  + CIDR block: 10.0.1.0/24 (or any suitable range).



* **Private Subnet 1**:
  + Name: Private-Subnet-EC2
  + Availability Zone: Choose a different AZ (if desired).
  + CIDR block: 10.0.2.0/24.

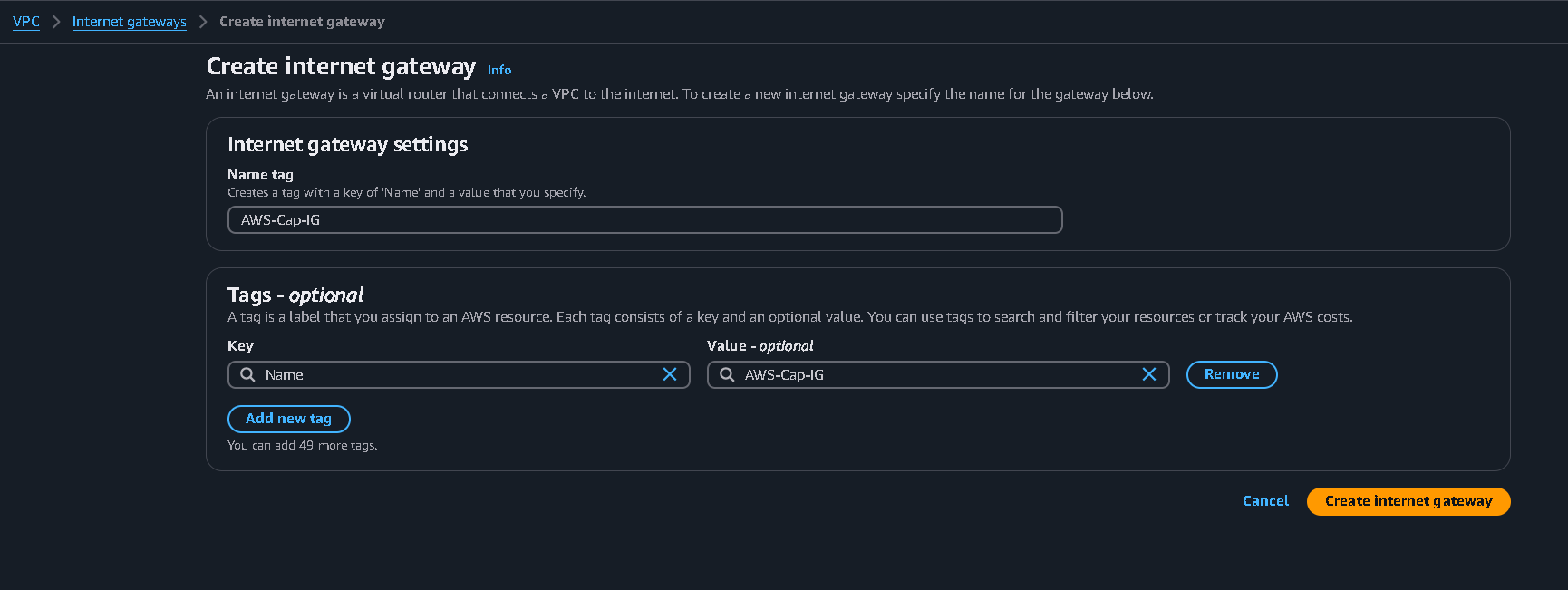


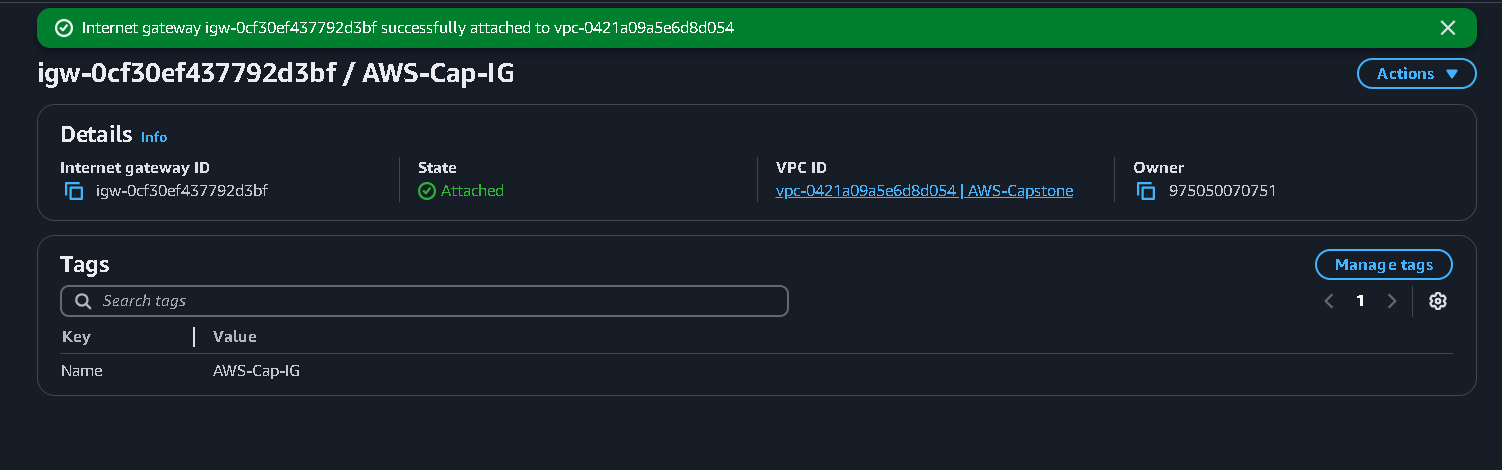
* **Private Subnet 2**:
  + Name: Private-Subnet-RDS
  + Availability Zone: Same or different AZ.
  + CIDR block: 10.0.3.0/24.



**\*Set Up Internet Gateway**

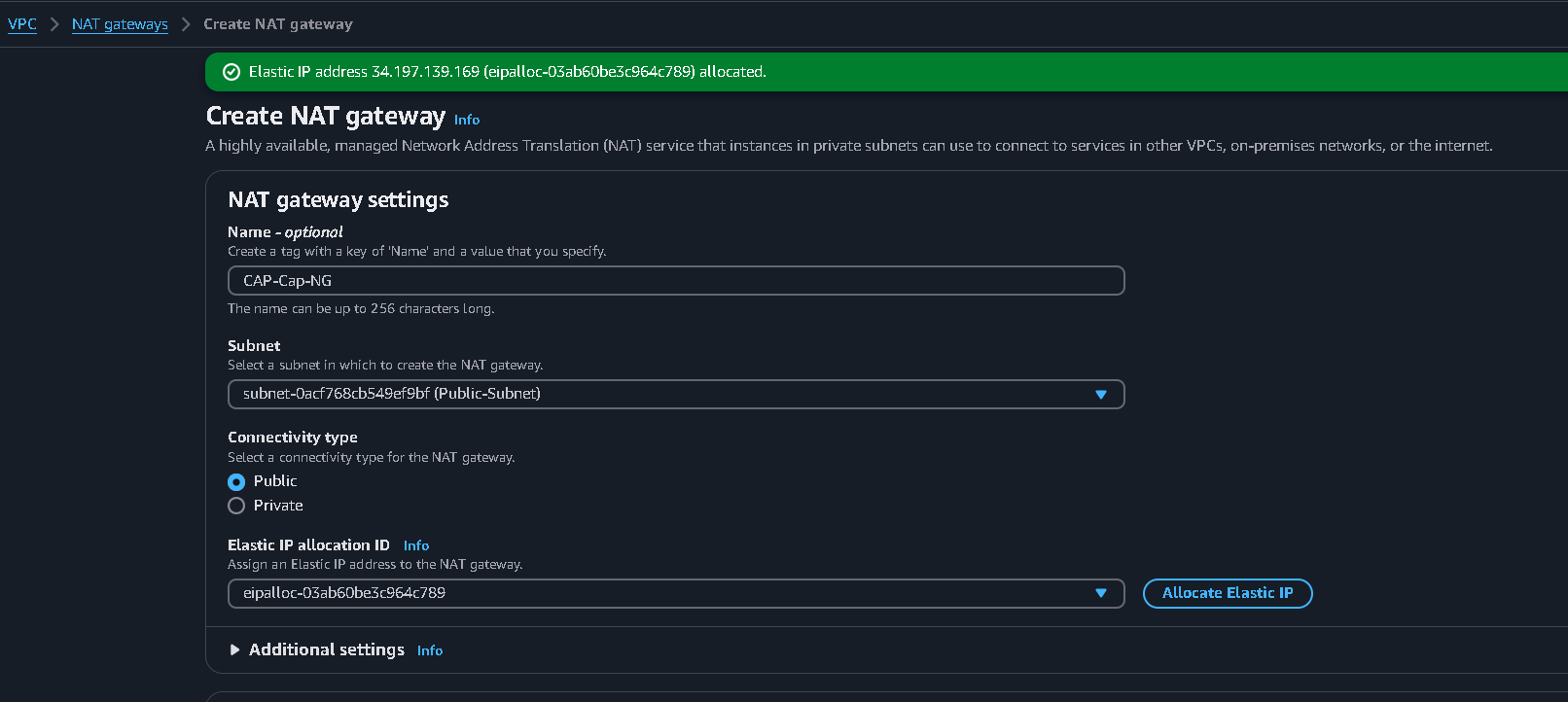
* In the VPC Dashboard, go to **Internet Gateways**.
* Click **Create internet gateway**.
* Name it Internet-Gateway.
* Attach it to the VPC you created earlier.





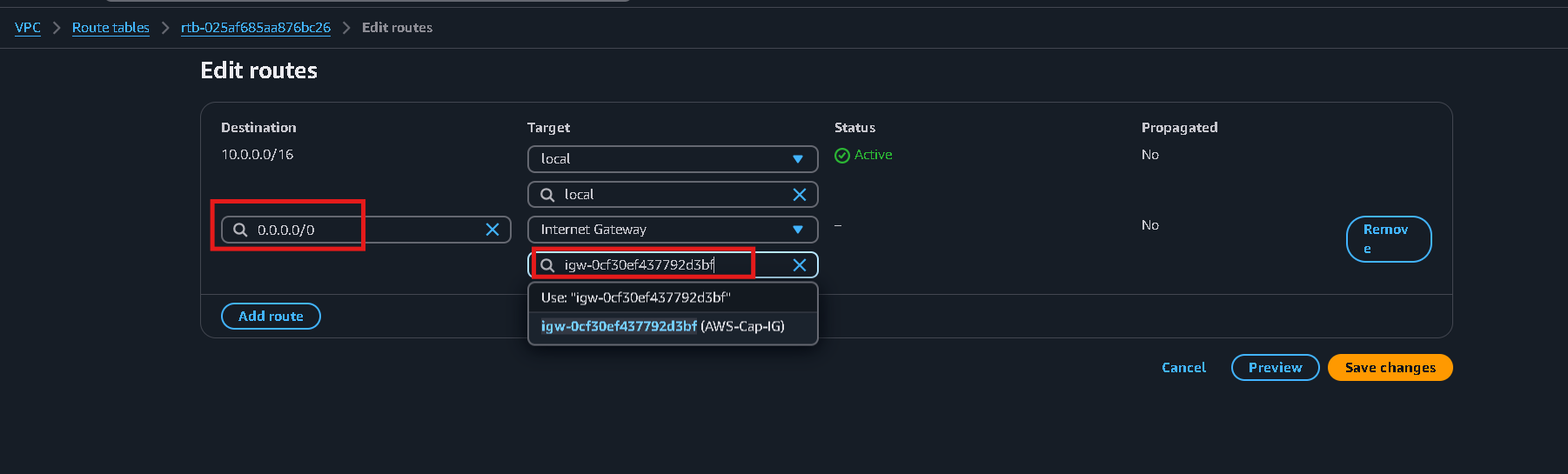
\***Set Up NAT Gateway (Optional)**

* Go to the **NAT Gateways** section in the VPC Dashboard.
* Create a NAT Gateway in the **Public Subnet**.
* Allocate an Elastic IP for the NAT Gateway.
* Update the private subnet route tables to route internet-bound traffic through the NAT Gateway.



\***Configure Route Tables**

* Go to the **Route Tables** section in the VPC Dashboard.
* By default, a main route table is created.need to modify it:
  + For the **Public Subnet**, modify the route table to have a route:
    - Destination: 0.0.0.0/0
    - Target: **Internet Gateway**



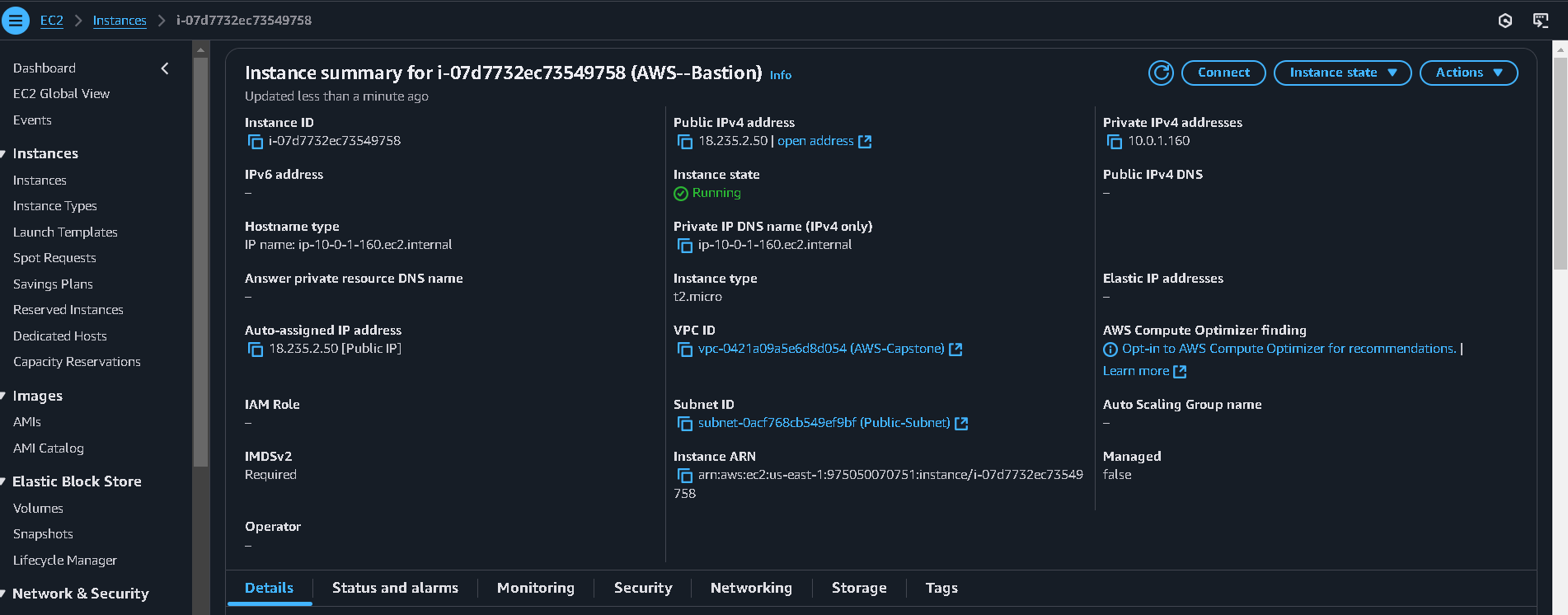
* For the **Private Subnets**:
  + Create a new route table for private subnets.
  + Add routes for outbound traffic through a **NAT Gateway**

**Step 2:**

Create Bastion Host Launch an EC2 instance in the Public Subnet

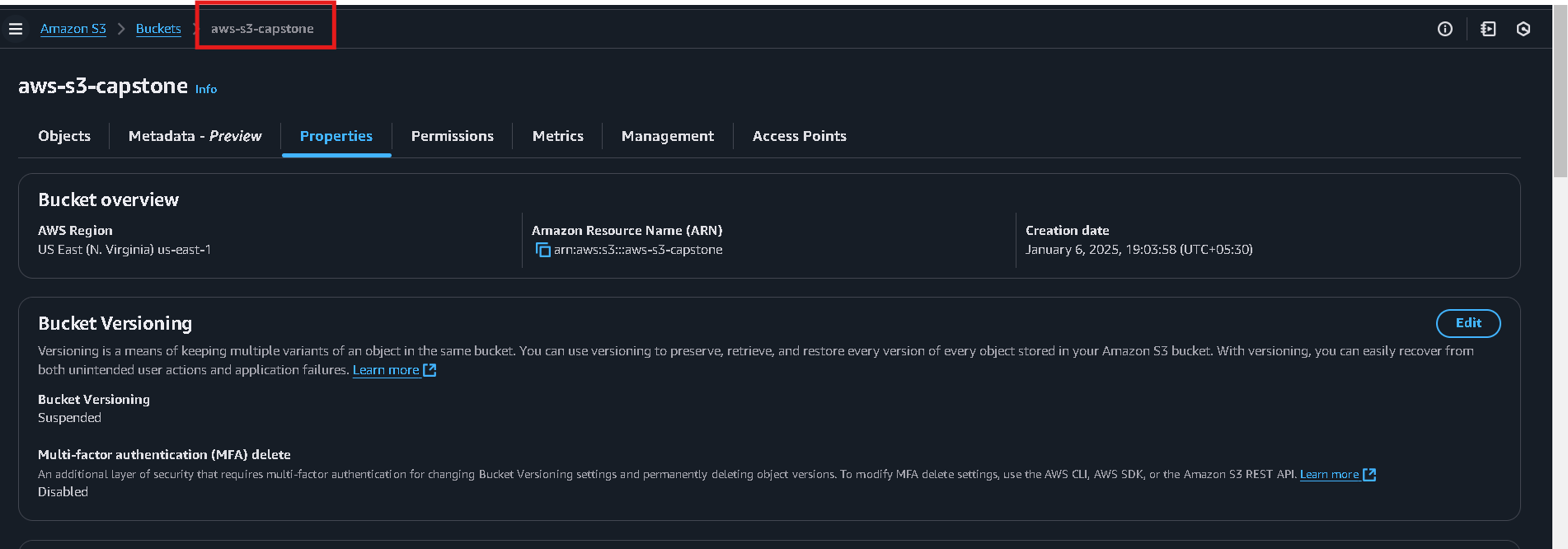
**Launch the Bastion Host in the Public Subnet**

1. **Go to EC2 Dashboard**: In the AWS Management Console, navigate to the **EC2 Dashboard**.
2. **Launch a New EC2 Instance**:
   * Click **Launch Instance** to create a new EC2 instance.
   * Choose an AMI, such as **Ubuntu**.
   * Choose an instance type (e.g., t2.micro for testing purposes).
3. **Network Settings**:
   * In the **Network** section, select the **VPC** already created.
   * Select the **Public Subnet** for this instance.
   * Assign a **Public IP** to the Bastion Host (this is important because the Bastion Host needs to be reachable from the internet for SSH access).
4. **Configure Security Group**:
   * Create a **new Security Group** for the Bastion Host, or use an existing one. Add the following inbound rules:
     + **Type**: SSH
     + **Protocol**: TCP
     + **Port**: 22
     + **Source**: your\_ip/32 (for security, restrict access to your own IP address).
     + This will allow SSH access to the Bastion Host only from your IP address.
   * Add the following outbound rules (default is usually fine):
     + **Type**: All traffic
     + **Protocol**: All traffic
     + **Port range**: All
     + **Destination**: 0.0.0.0/0 (allowing traffic to any destination).
5. **Review and Launch**:
   * Review your settings and click **Launch**.
   * Make sure to use a **new key pair** or an existing key pair to access the instance (you will need this for SSH access).



Step 4:

Create S3 Bucket To store images For Static Website Hosting



Step 5:

Create RDS Instance Connect to it and create “employees” table

**Create an RDS Instance:**

Create an RDS MySQL Database Instance

1. Navigate to RDS Dashboard:

o Go to RDS from the AWS Management Console.

2. Create a Database:

o Click Create Database.

o Choose MySQL and Standard Create.

o Configuration:

\*B Instance Identifier: Employe-db.

\*Master Username: admin.

\* Password.

\* Instance Size: Choose an appropriate instance (e.g., db.t2.micro for testing).

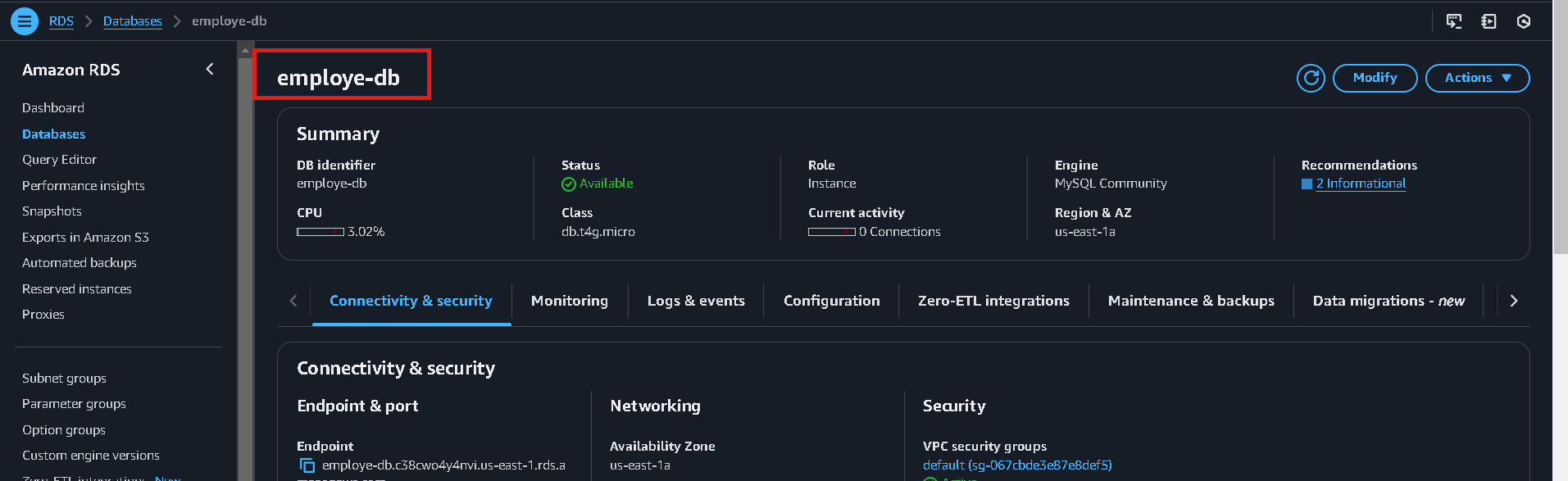
\* Connectivity:

\* Choose the VPC that includes your EC2 instances.

\* Set Publicly Accessible to No.

\* Security Group: Allow inbound MySQL traffic (port 3306).

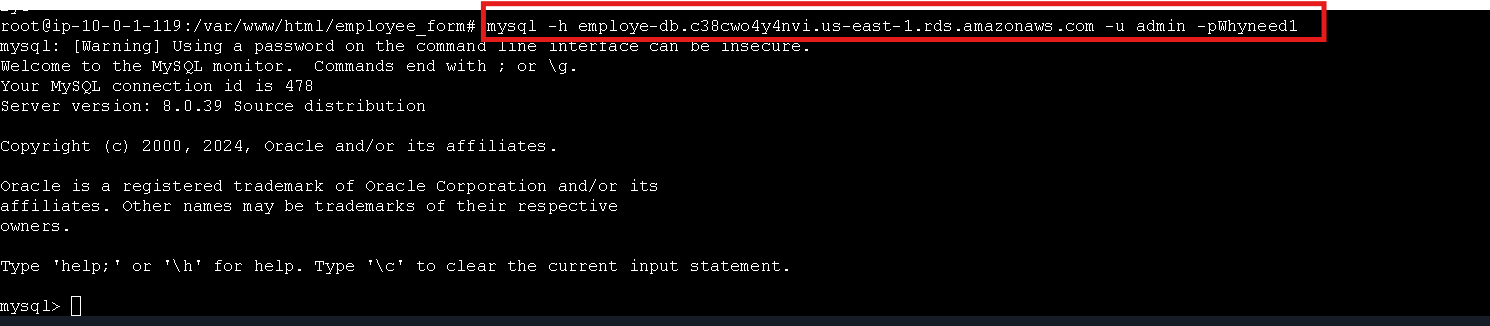
\* Review and create the RDS instance



**Now install MySQL in ubuntu instance.**

Once RDS instance comes in available state and connect database instance to the EC2

instance by copying RDS Endpoint following credentials.



Create Database &amp; Table in RDS instance:

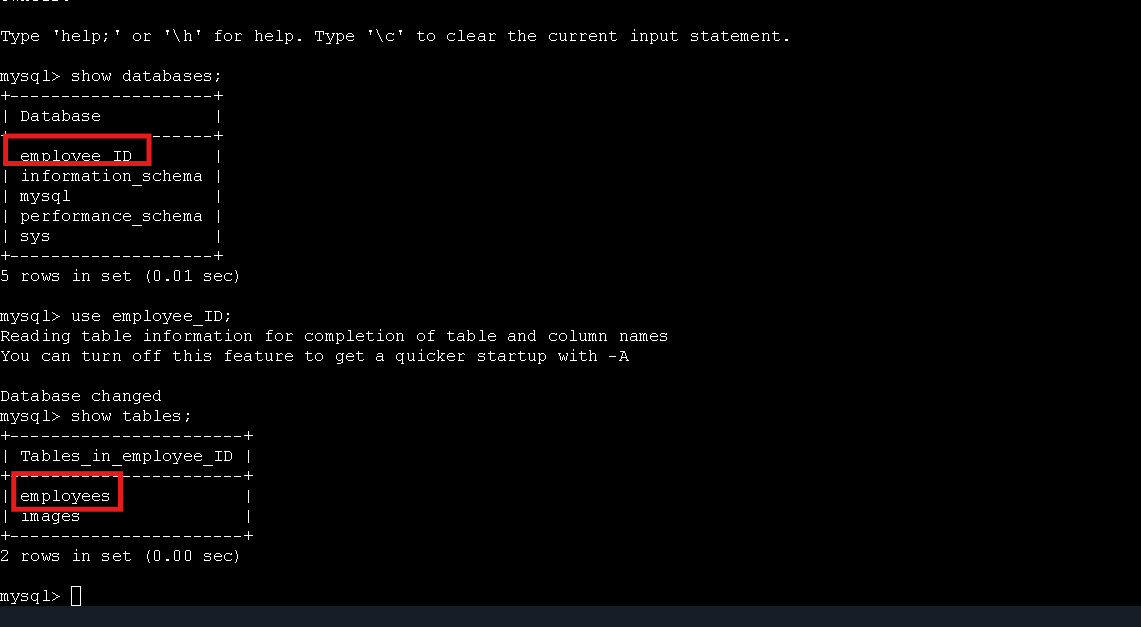
a. Database name: employee\_ID

b. Table name: employees

c. Database password: Whyneed1

Verify the “intel” database is present or not.

Create a “data” table.



Now installs PHP 5.6 along with the php5.6-mysqli extension. PHP is a scripting

language commonly used for web development, and the mysqli extension allows PHP to

interact with MySQL databases, enabling it to connect to and execute SQL queries on a

MySQL database.

**Step 6:**

Domain Name Setup Get a Domain Name Setup Route 53

**Create a Hosted Zone in Route 53**

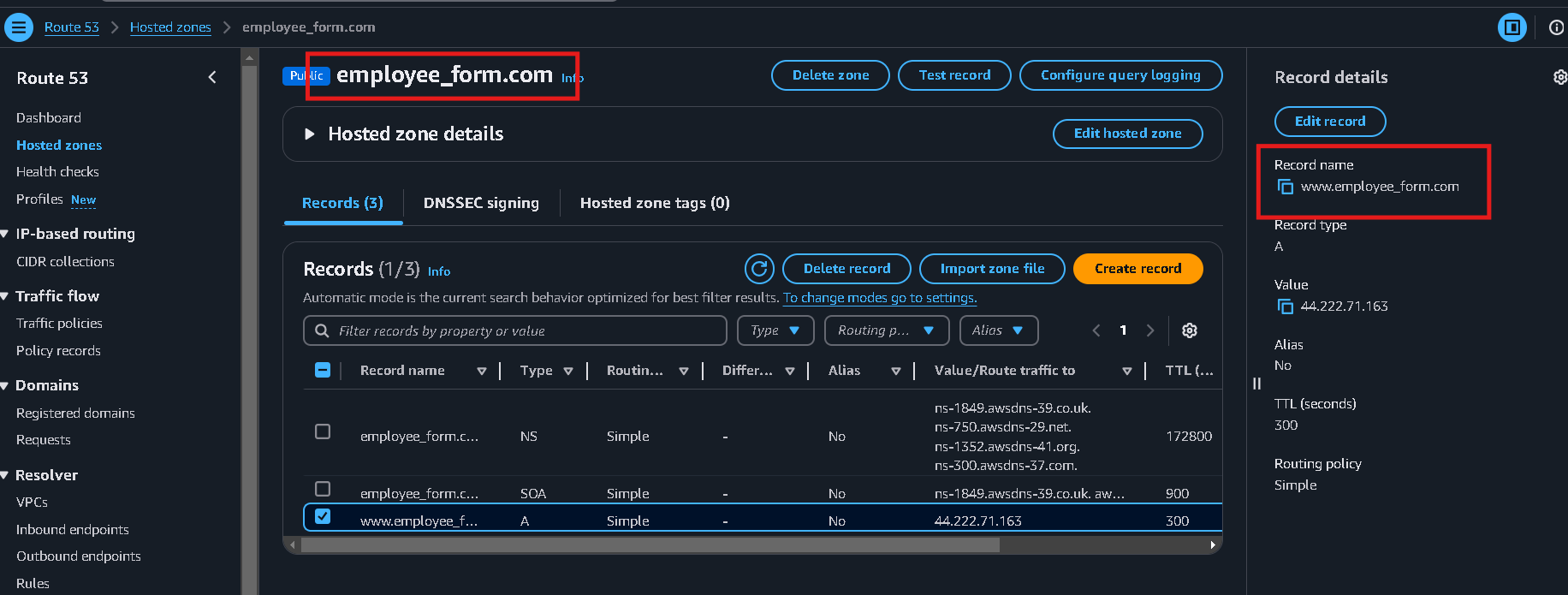
A **Hosted Zone** is required to manage DNS records for your domain.

1. **Go to the Route 53 Console**  
   Navigate to **Hosted Zones** in the Route 53 Dashboard.
2. **Create a Hosted Zone**
   * Click **Create Hosted Zone**.
   * Enter your domain name (e.g., example.com).
   * Set the **Type** to **Public Hosted Zone** (for public websites).
   * Add a description (optional).
   * Click **Create Hosted Zone**

3.Add DNS Records in the Hosted Zone

**Add an A Record for an EC2 Instance**

1. Click **Create Record** in the hosted zone.
2. Enter the following details:
   * **Record Name**: Leave blank for the root domain (e.g., example.com), or enter a subdomain (e.g., www).
   * **Record Type**: Select A - IPv4 address.
   * **Value**: Enter the public IPv4 address of your EC2 instance.
3. Click **Create Records**.



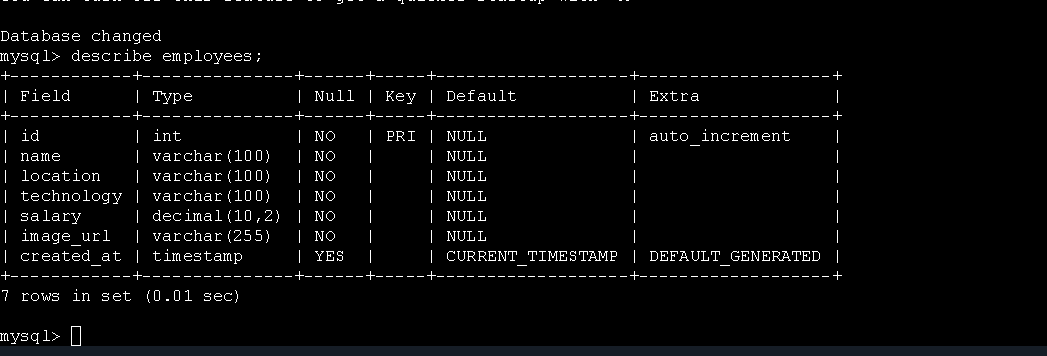
Step 7:

Now we will build application

Steps to Build

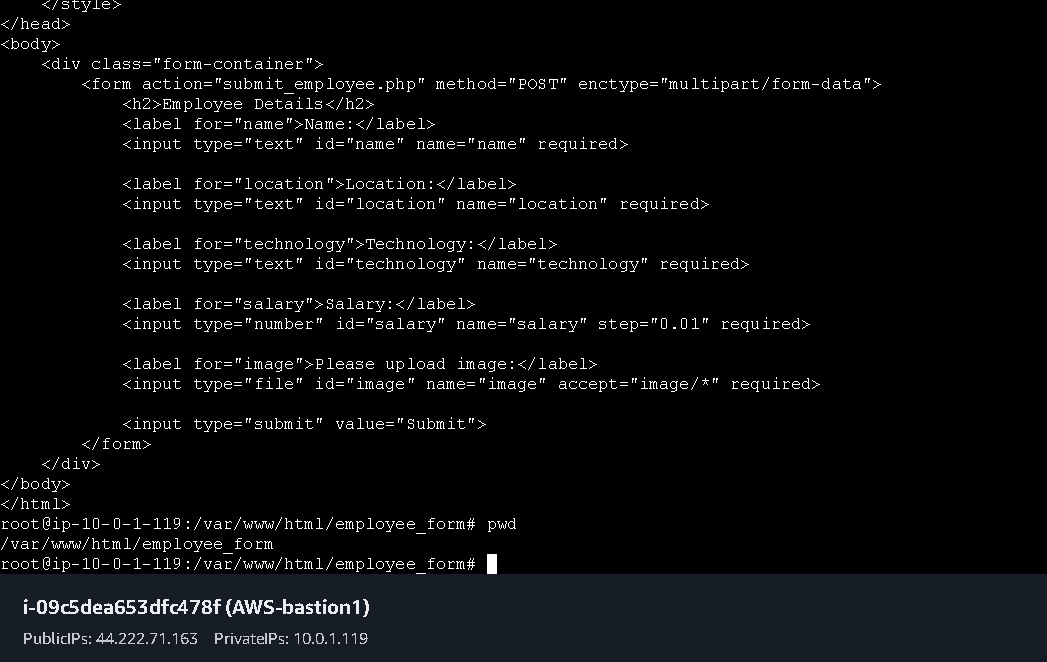
Step 1: Setup RDS Database st, you need to set up an **RDS database**.

* **Create a database** in RDS
* **Create a table** to store the user data with the following columns:
  + id (auto-increment primary key)
  + name
  + location
  + technology
  + salary
  + image\_url



**Step 2: HTML Form**

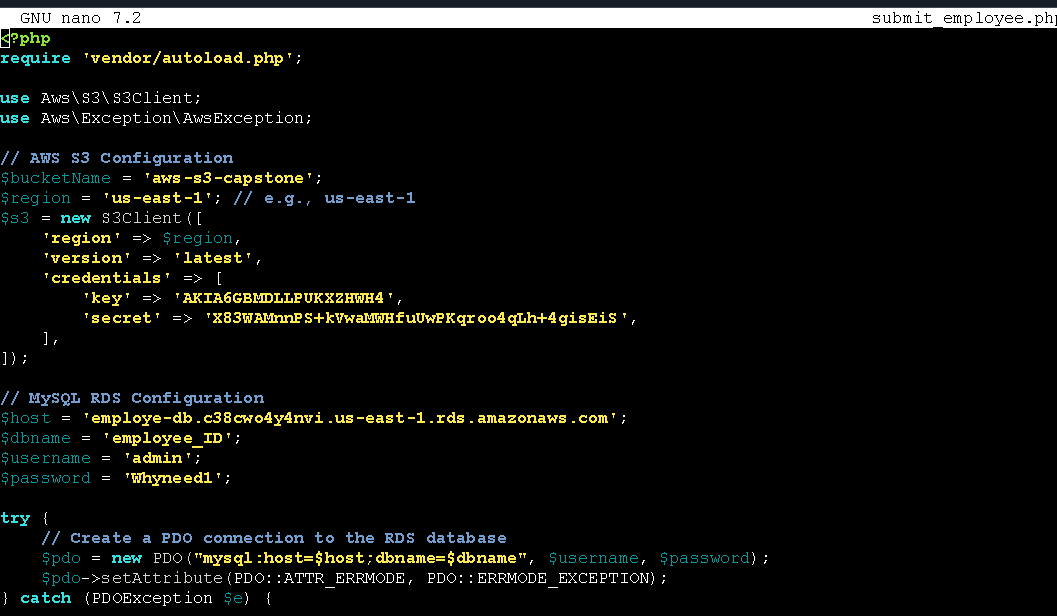
This is the HTML form that collect user inputs: Name, Location, Technology, Salary, and an image.

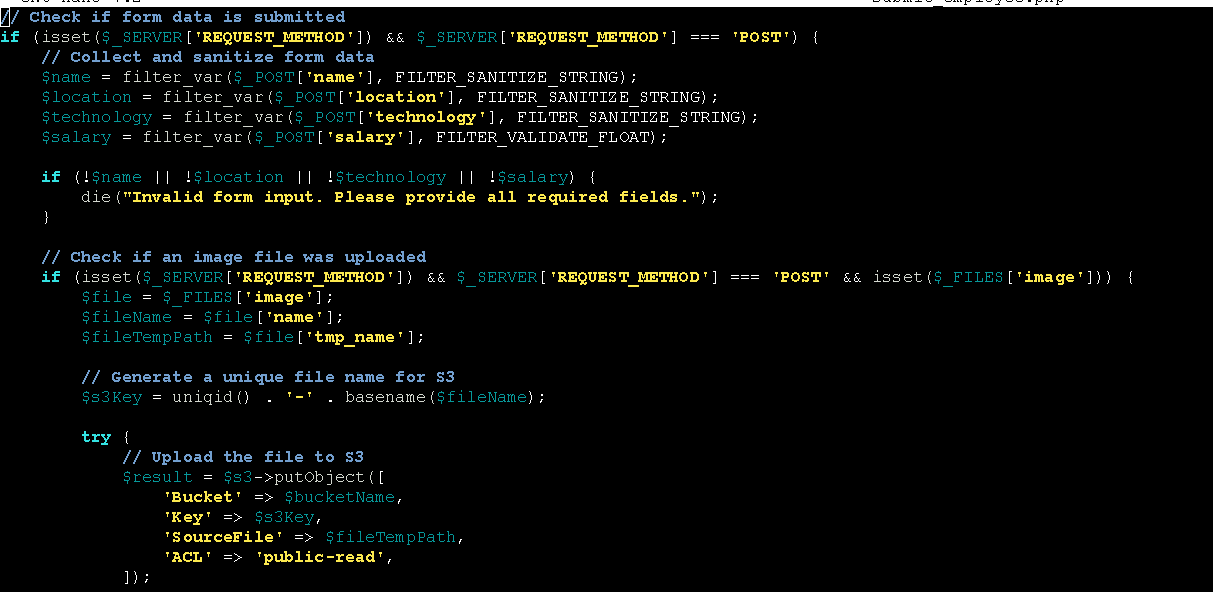


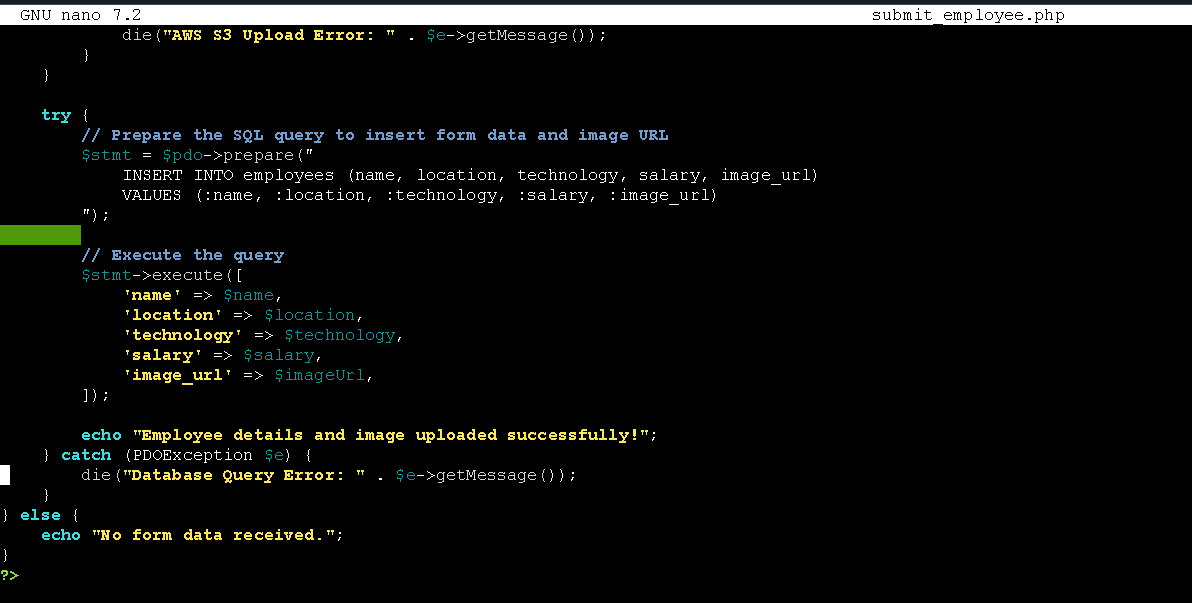
**Step 3: PHP Script to Handle Form Submission**

Now let’s create the PHP script to handle form submission and save the data to the RDS database.

1. **Database Connection:** We need to connect to the MySQL database on RDS. Update the connection parameters (host, dbname, username, password).
2. **File Upload Handling:** The script will process the uploaded image and store it on the server.
3. **Save Data in Database:** We will insert the form data (including the image URL) into the RDS database.





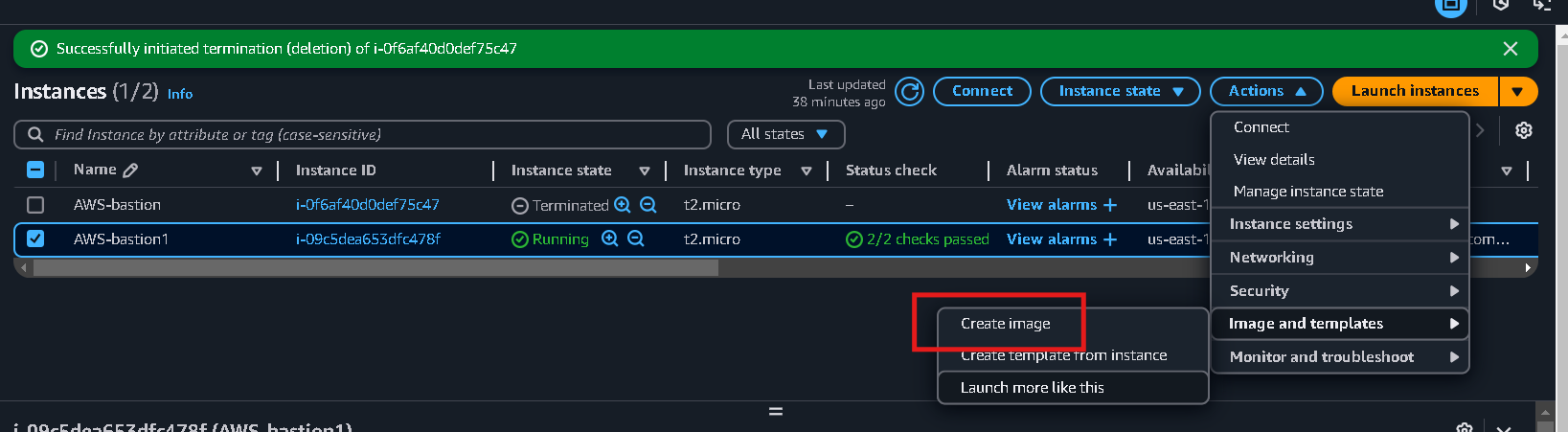


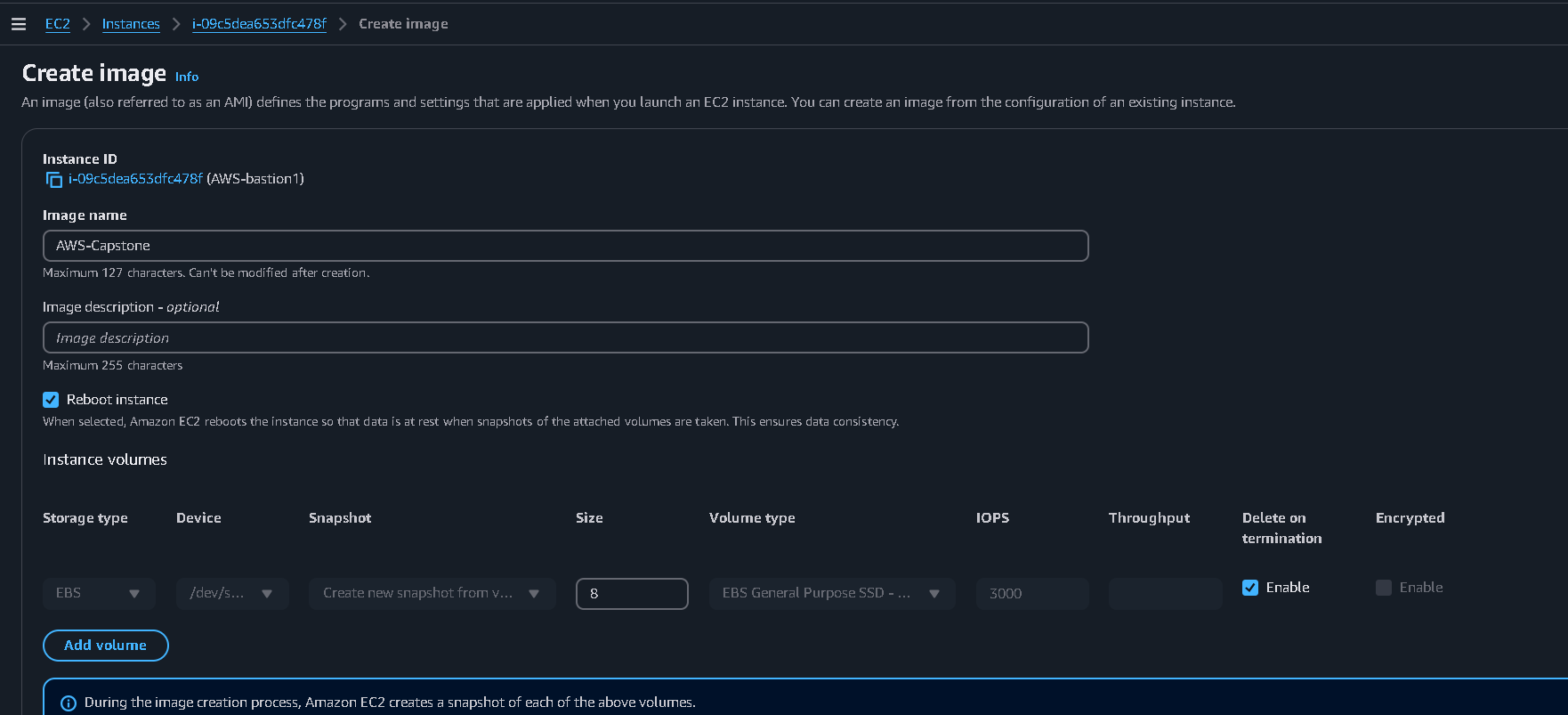
We will save above file and install all pre-required plugins.

**Step 8:**

Create AMI with application Login to a EC2 instance and setup application

Create an AMI from this EC2 instance to launch the template.





Click on create.

**Enable Auto Scaling on EC2 Instances**.

Create a Launch Template:

\* Go to EC2 Dashboard &gt; Launch Templates.

\* Click Create Launch Template.

\* Set up the template details:

o Template Name: Enter a name.

o AMI ID: Use the same AMI ID as your initial EC2 instance.

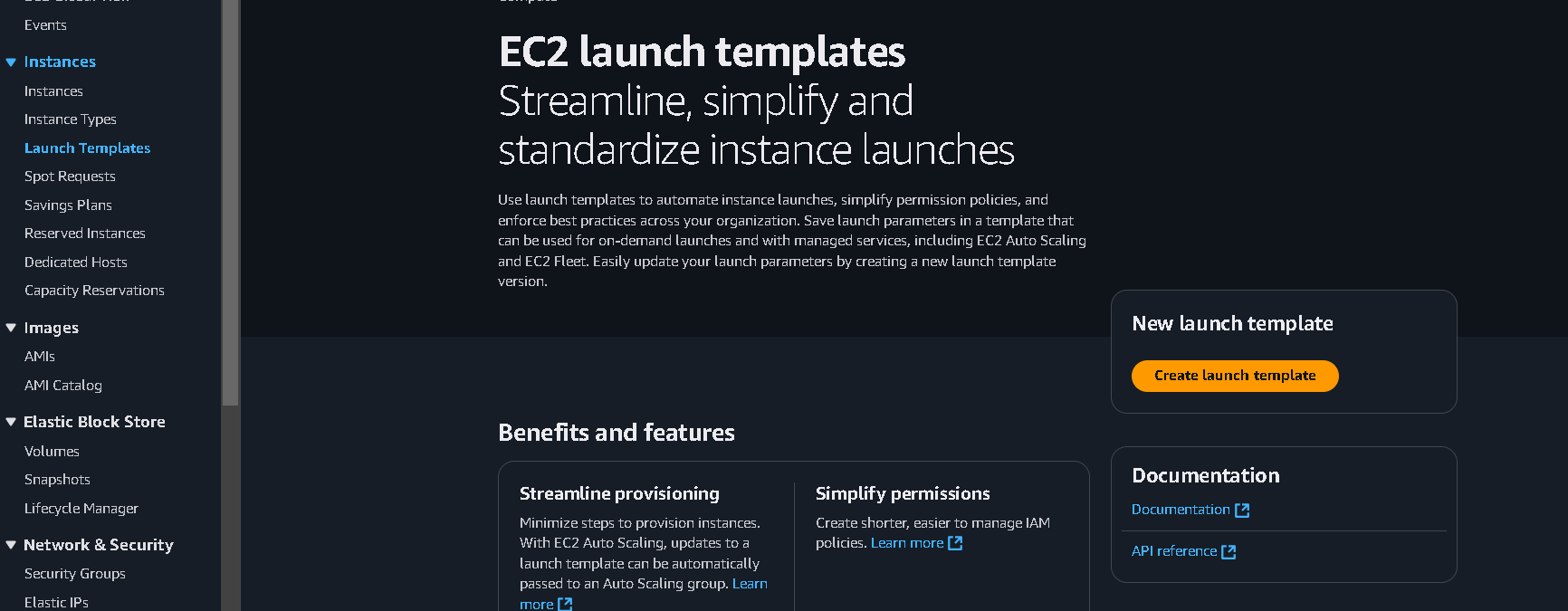
o Instance Type: Choose the same type as the first instance.

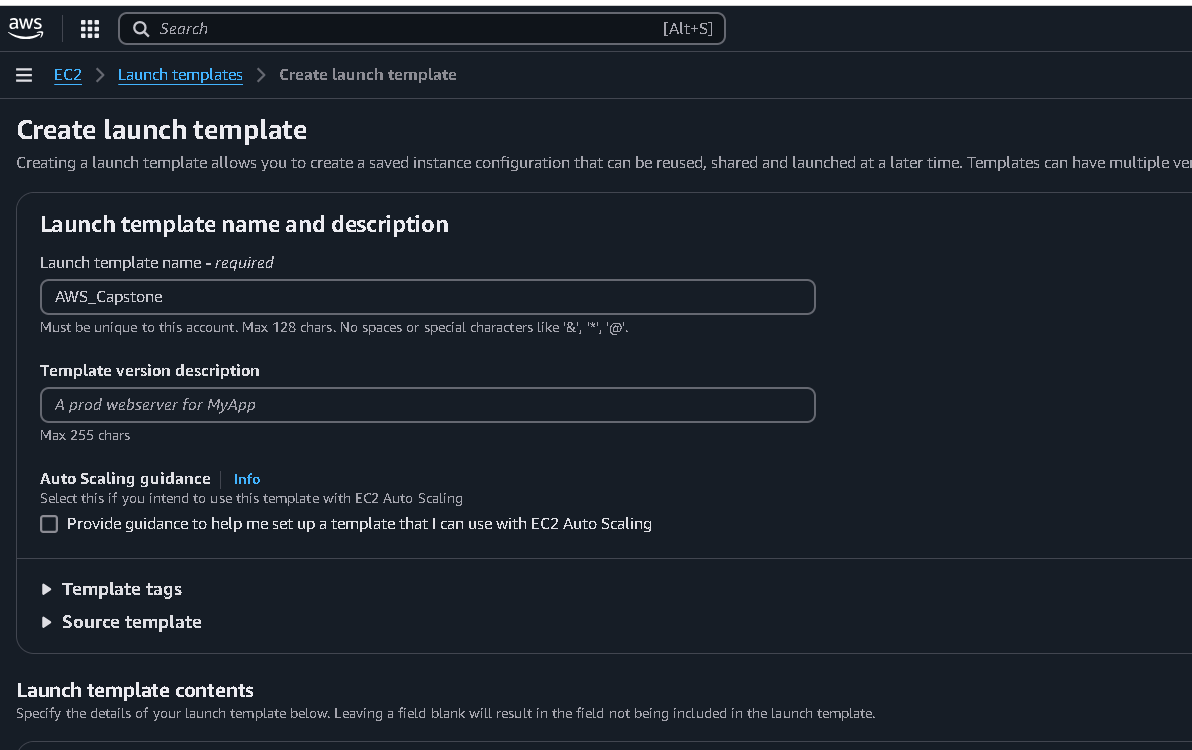
o Key Pair: Select the key pair used for the first instance.

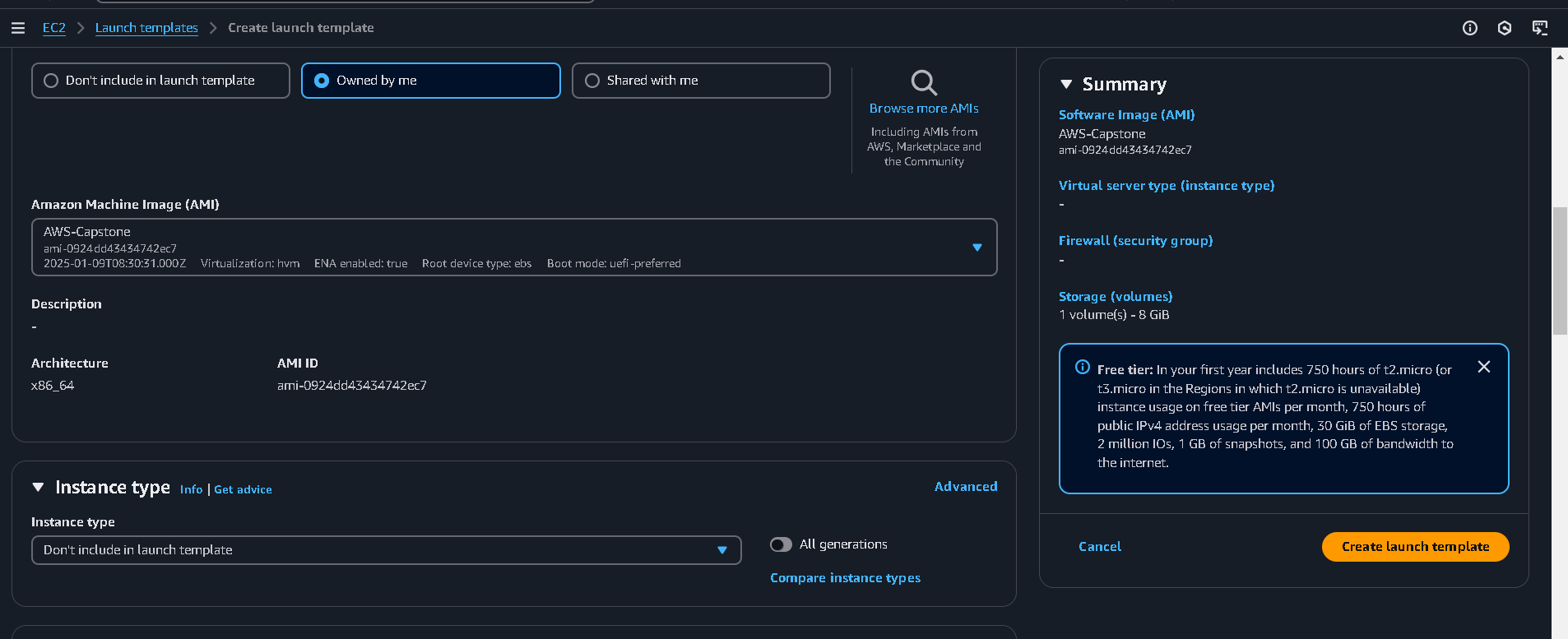
\* Network Settings:

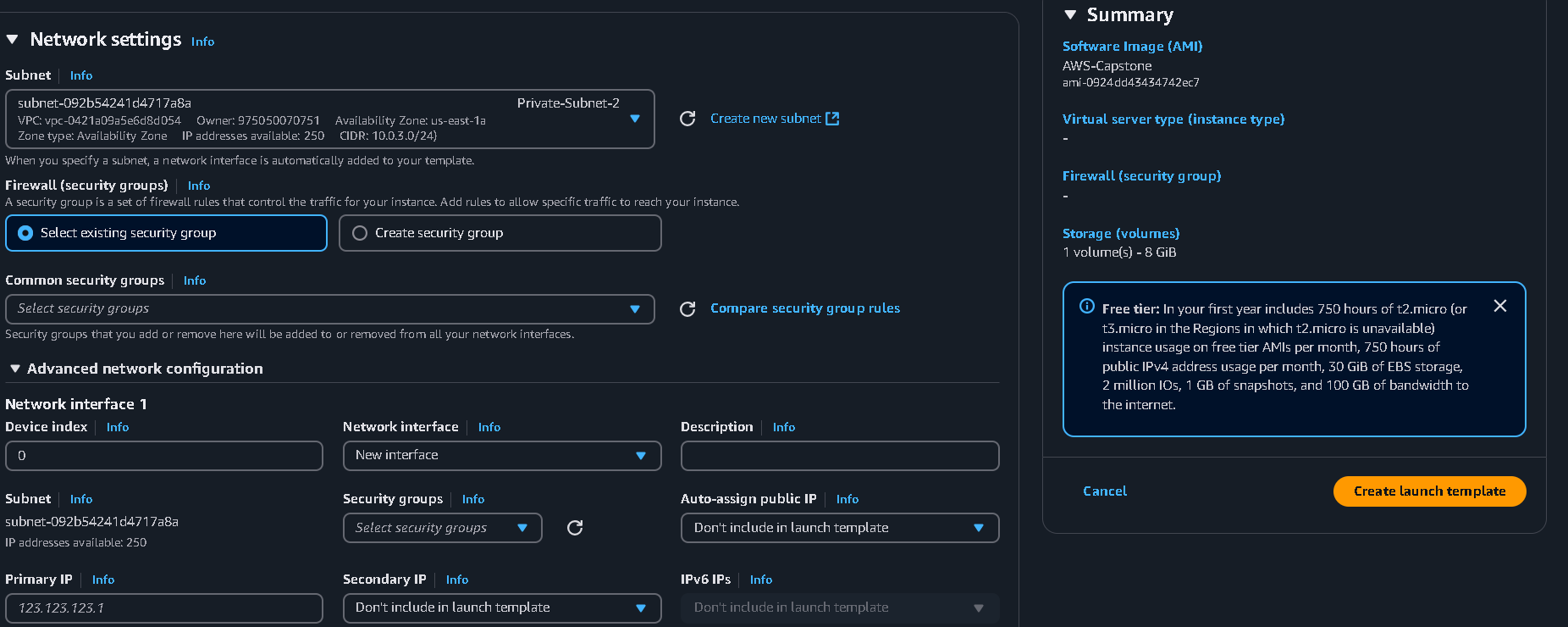
o Security Groups: Select the security group used for the initial instance.

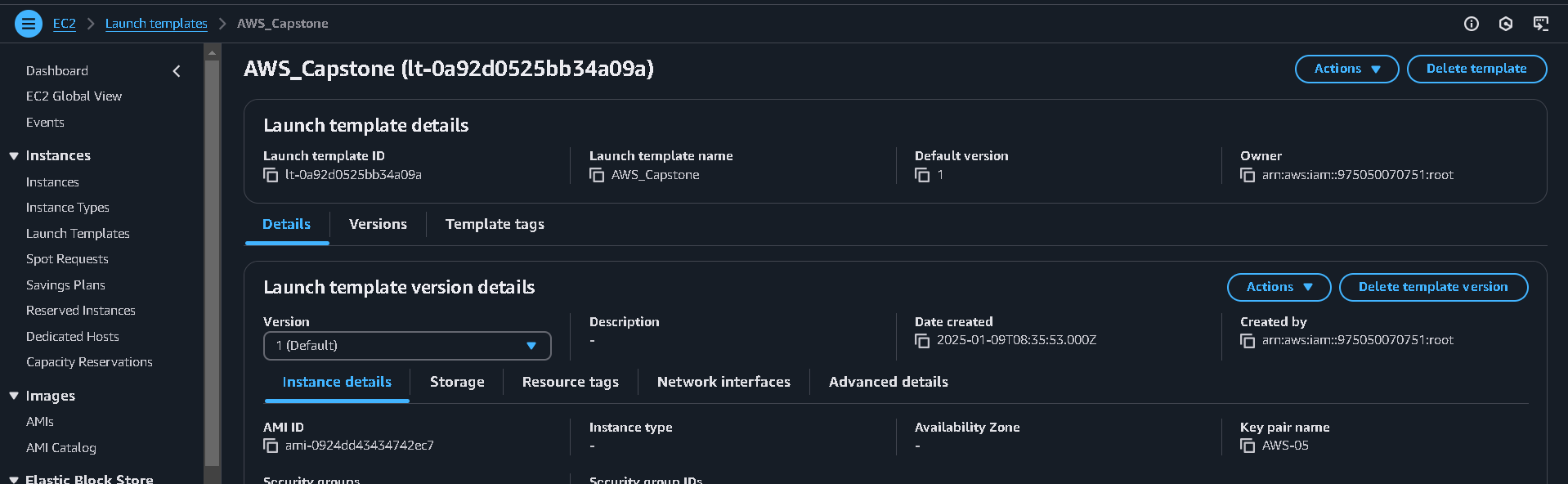
\* Click Create Launch Template.











**Enable Auto Scaling** on these instances (minimum 2):

Create an Auto Scaling Group.

Go to Auto Scaling Groups in the EC2 dashboard.

Click Create Auto Scaling Group.

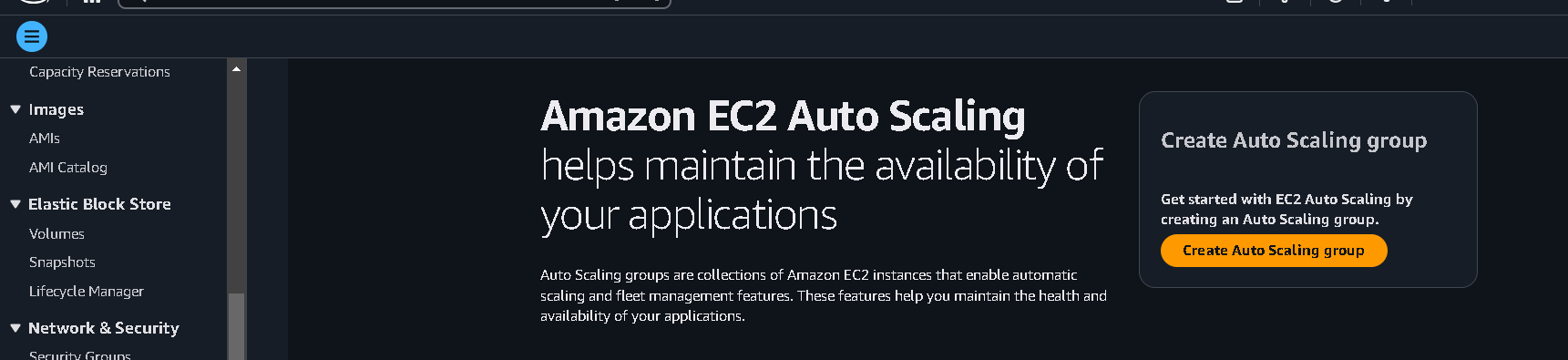
Choose the launch template created above.

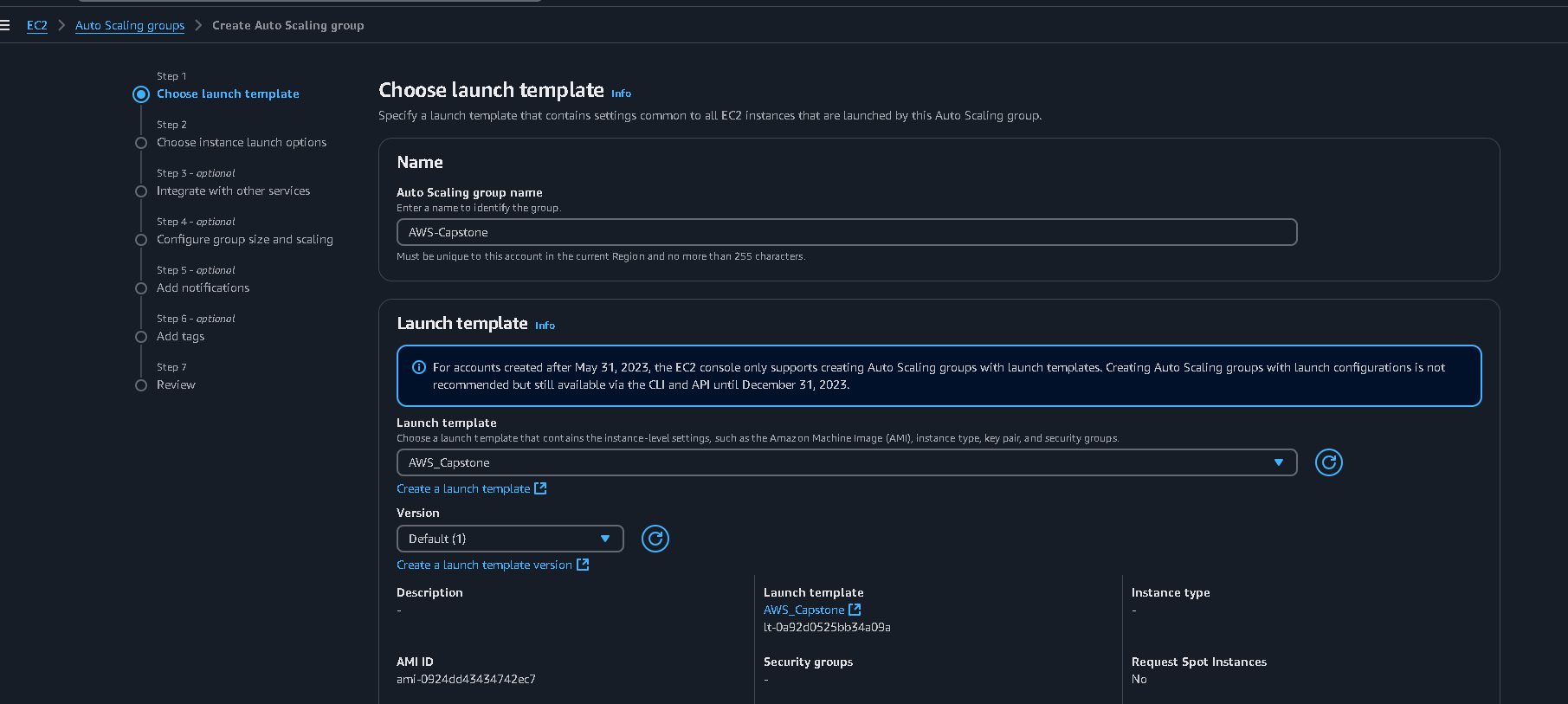
Configure Group Details:

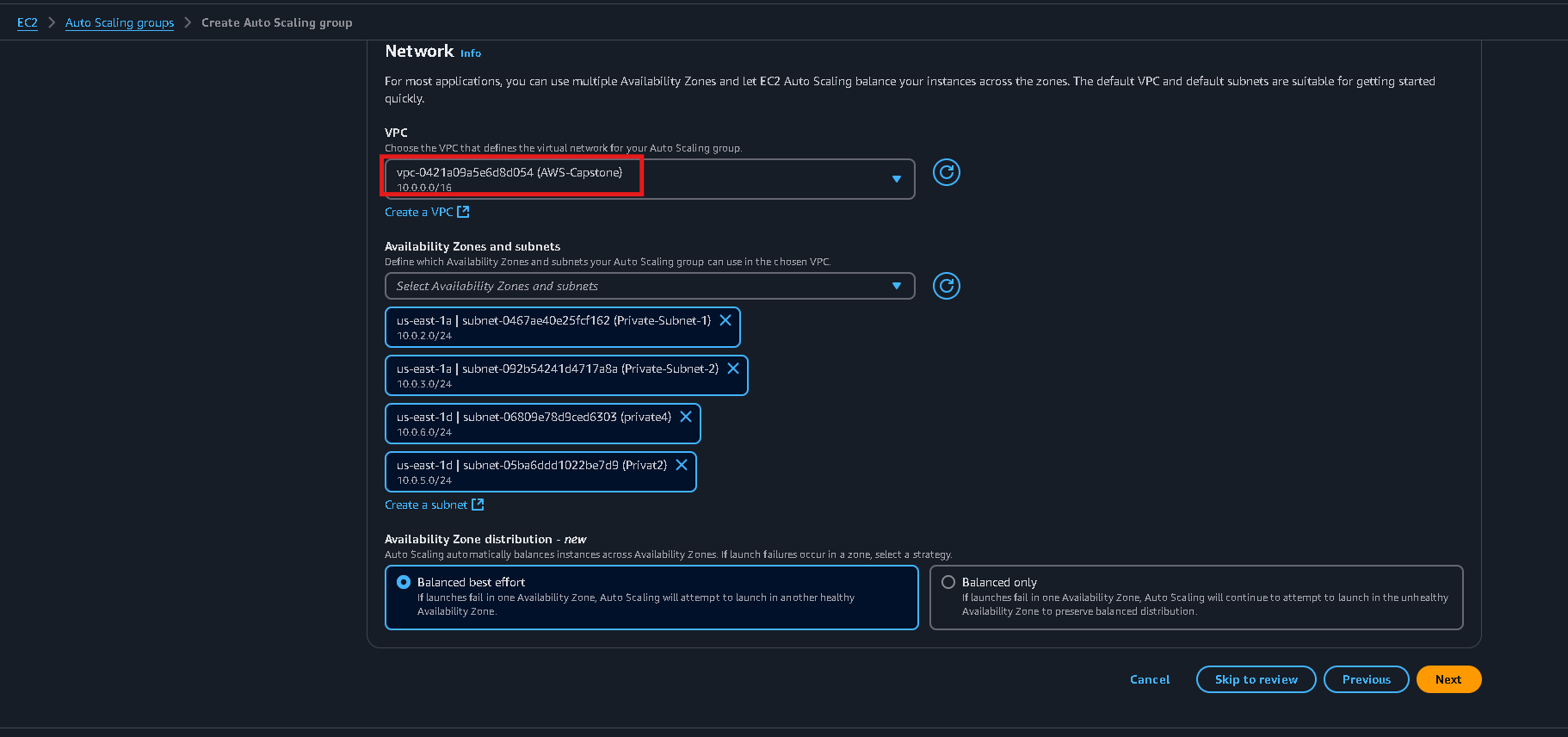
 Set Minimum Desired Capacity to 2 and Maximum Capacity to a higher value i.e. 3.

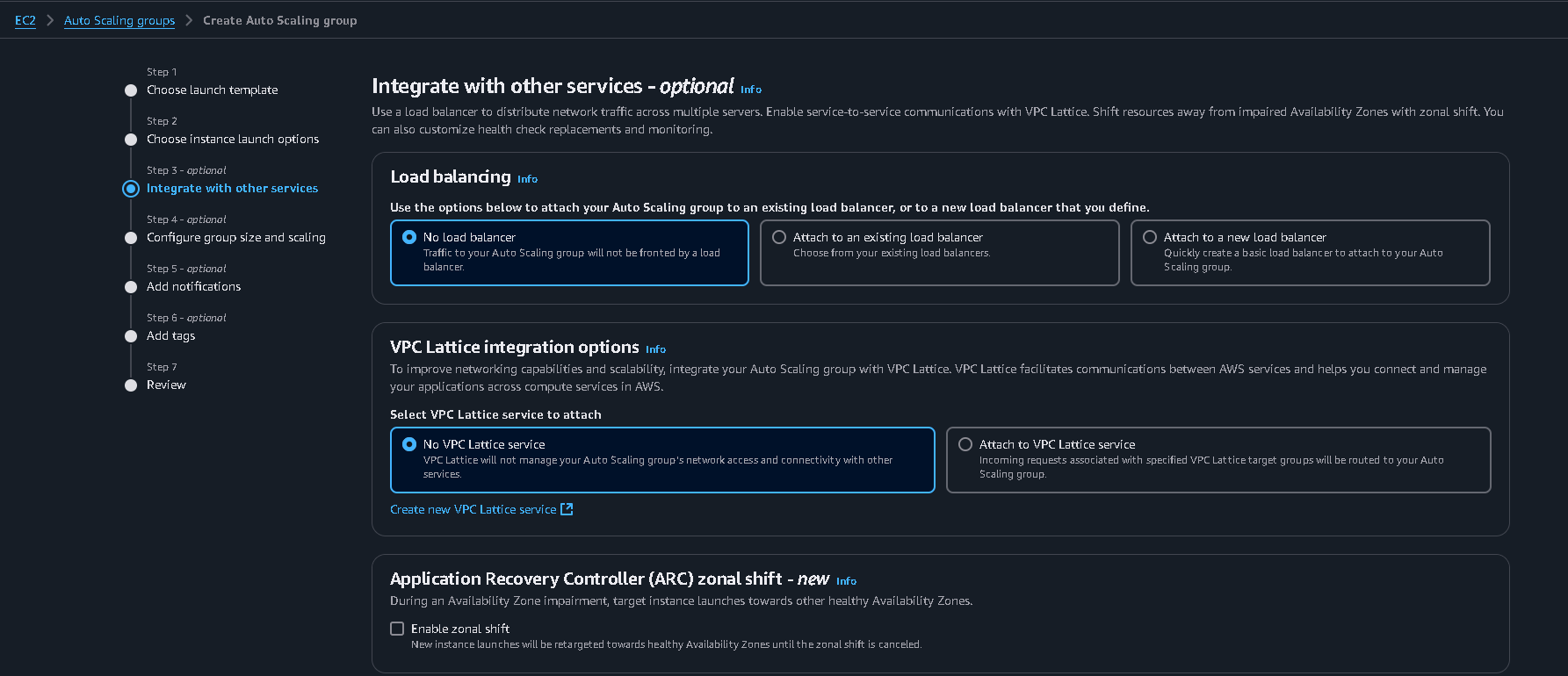
Configure Scaling Policies to manage scaling actions based on load.

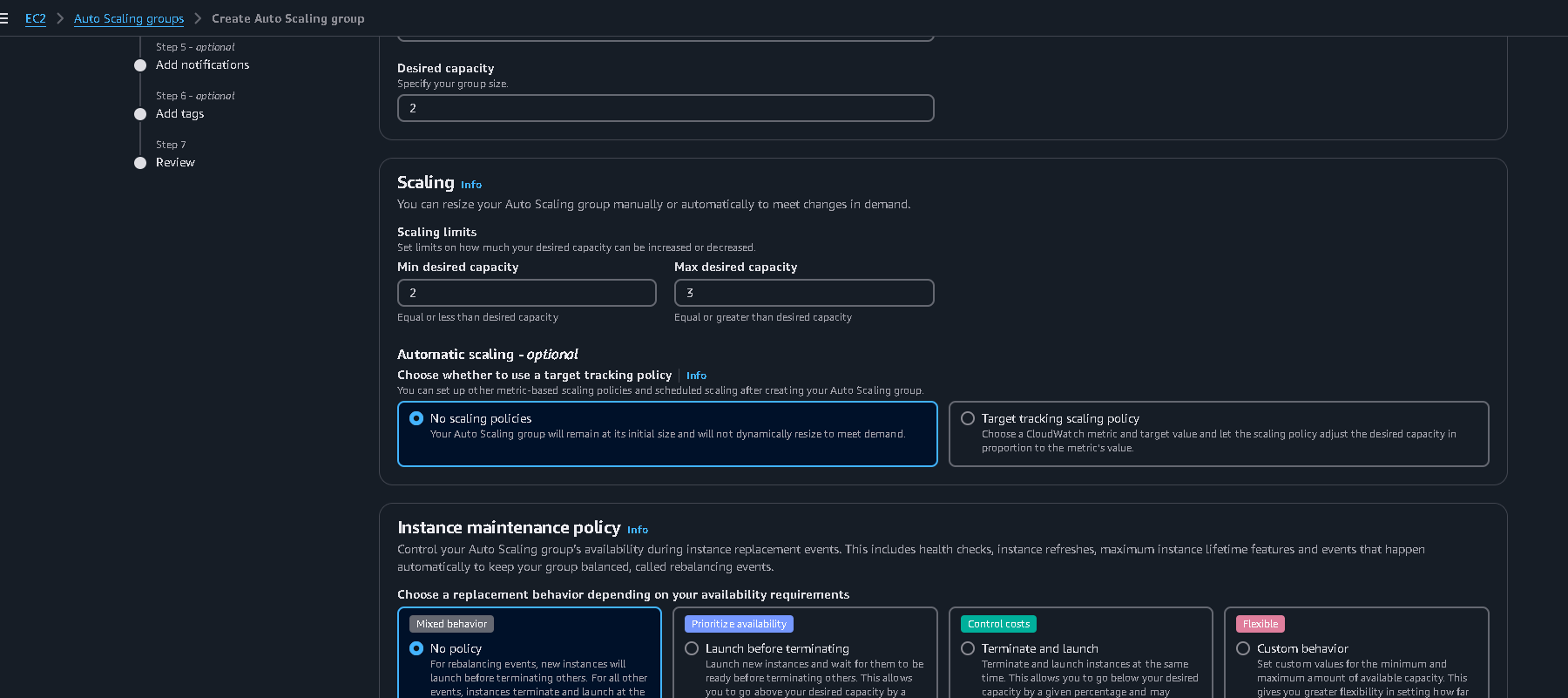
Review and create the Auto Scaling group.

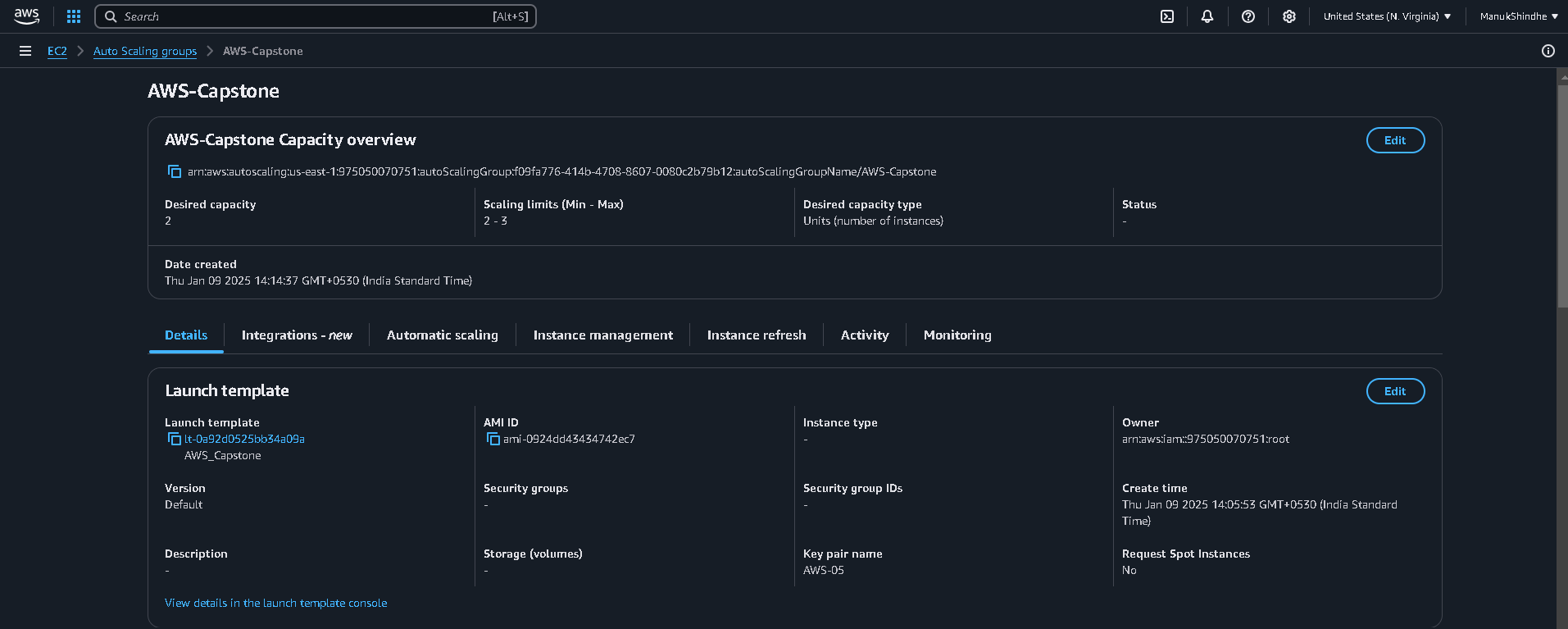




****

****

****

****

**Confirm Application Functionality.**

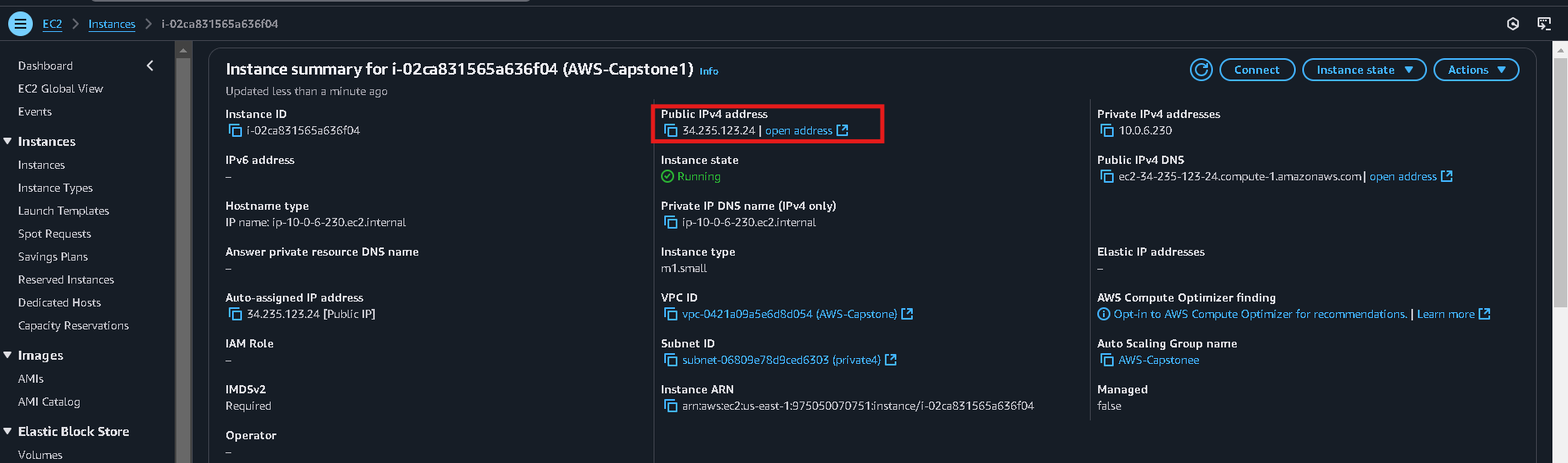
Access the website to confirm that PHP scripts on all instances can connect to the RDS

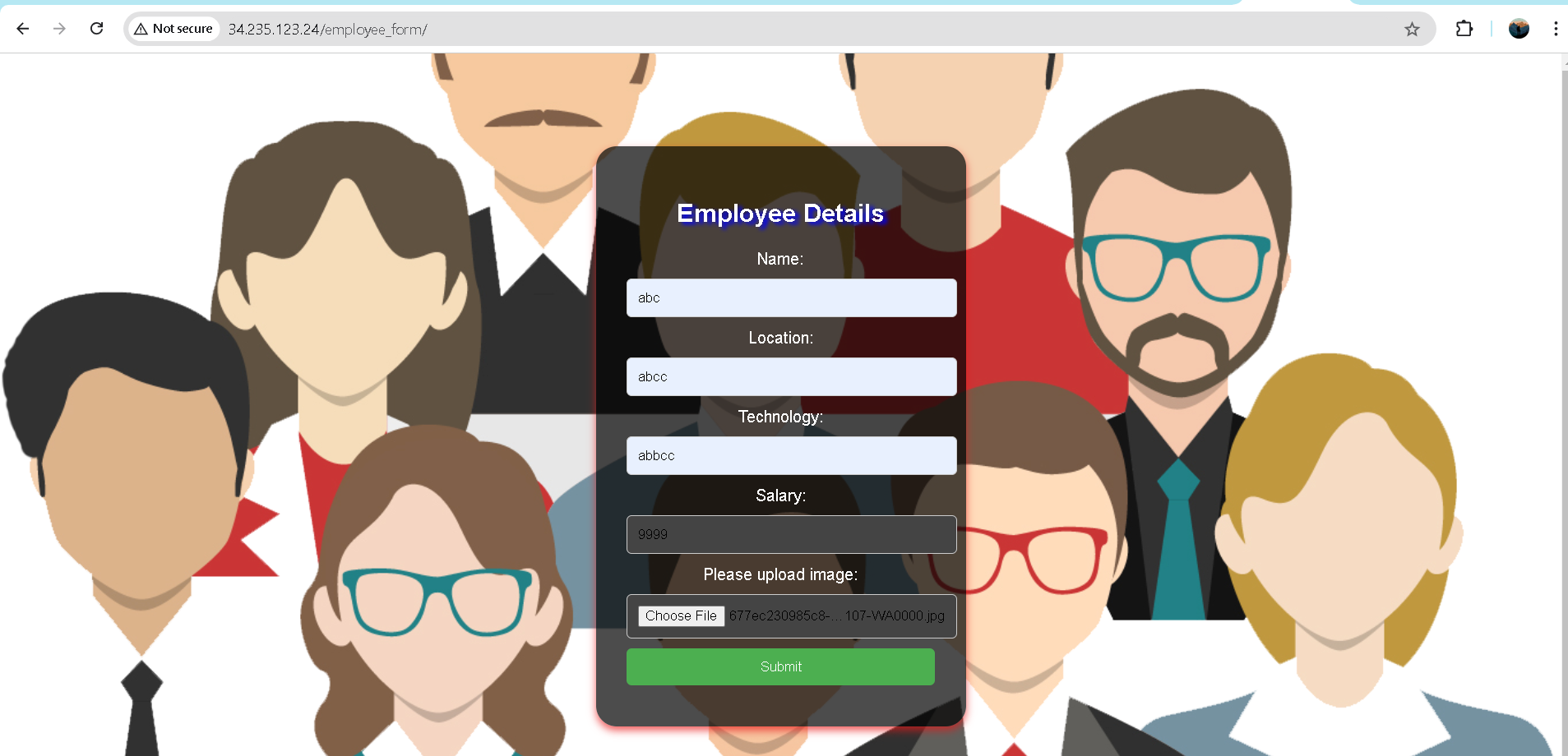
database and function correctly.

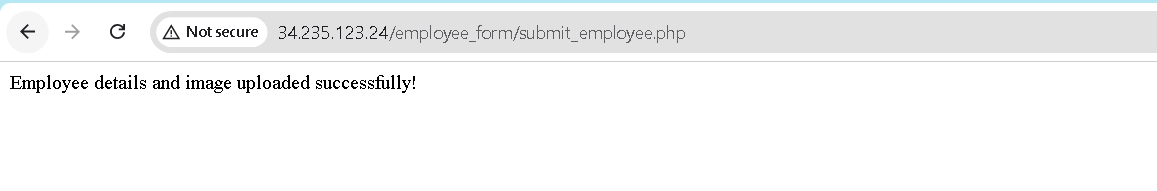
So copy the IP address of the newly created instances of Auto-Scaling group’s instance then

very the and try to add the data and verify in database.

1)

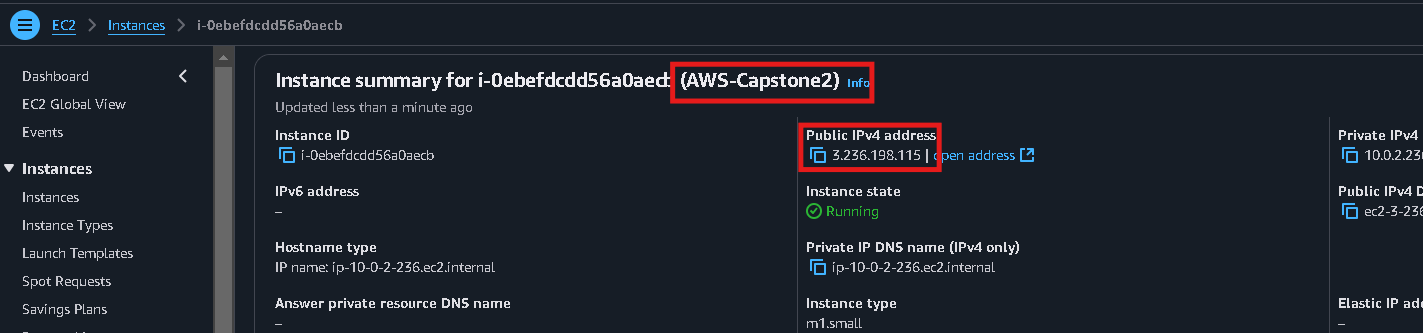
****

****

****

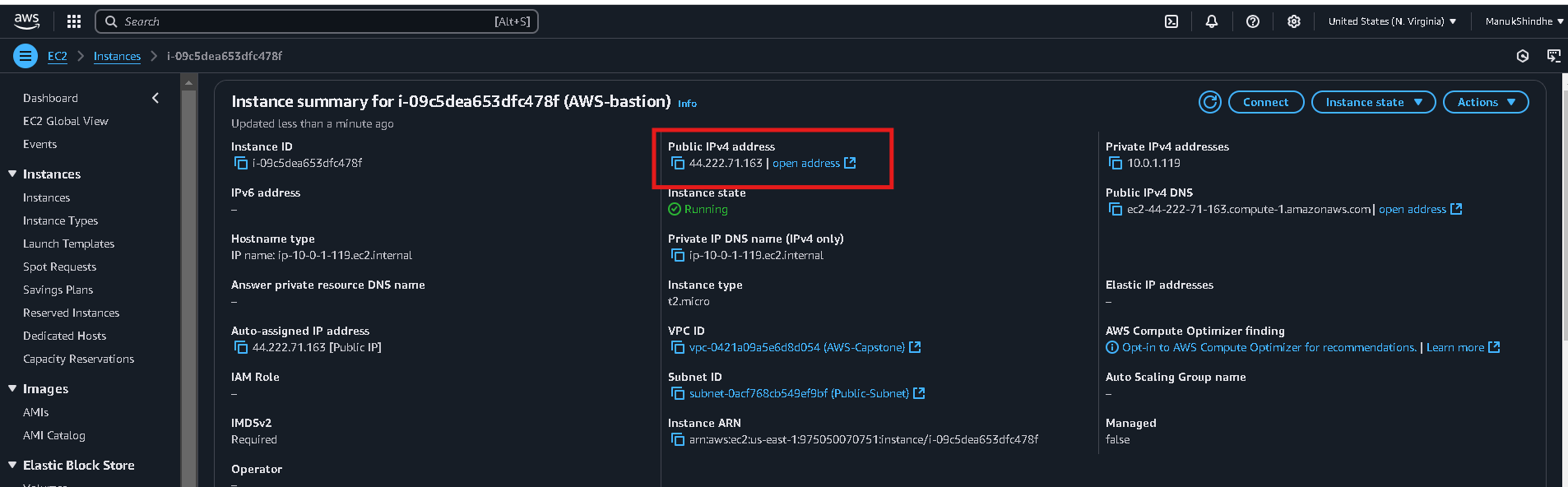
****

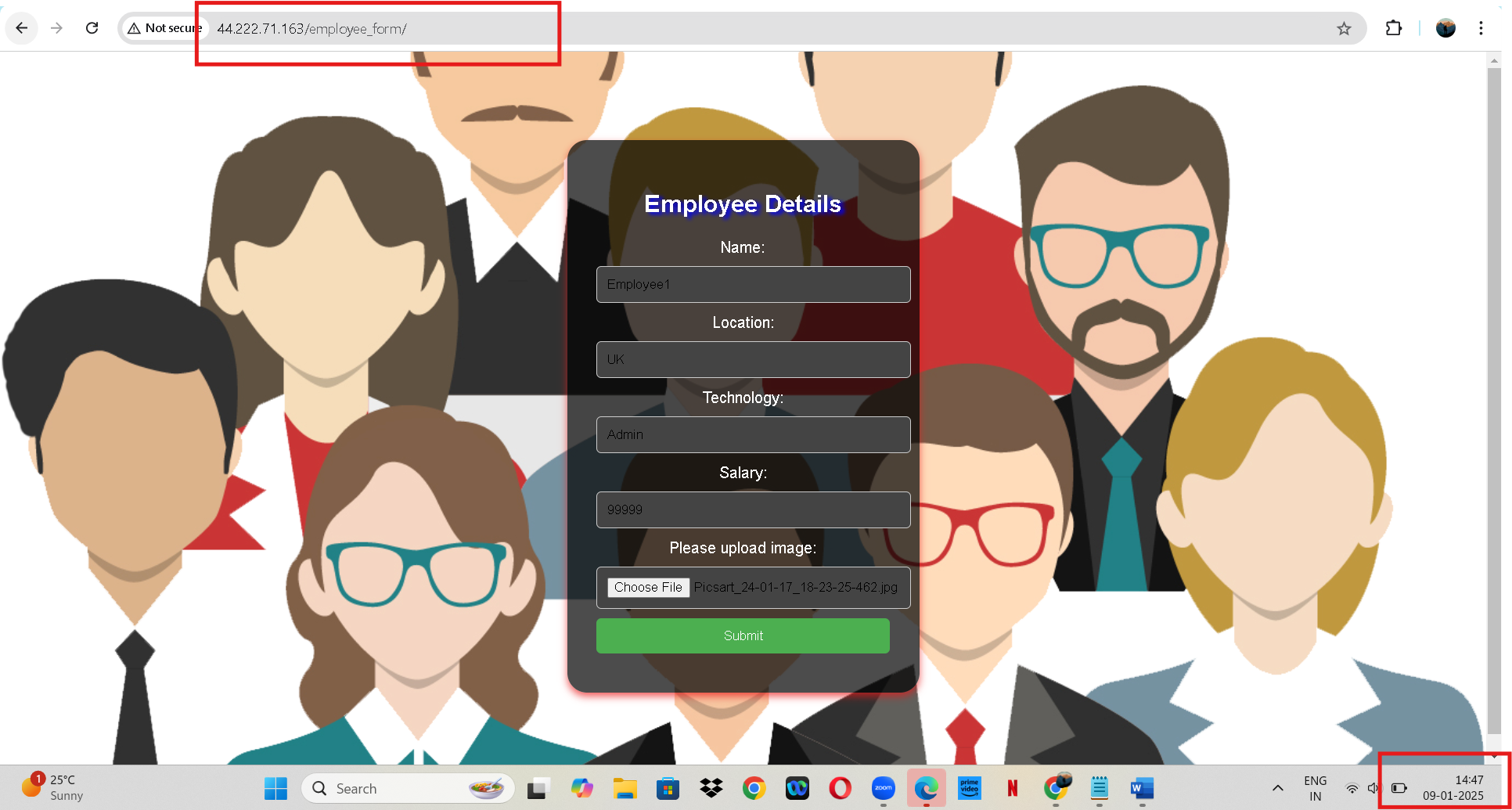
**2)**

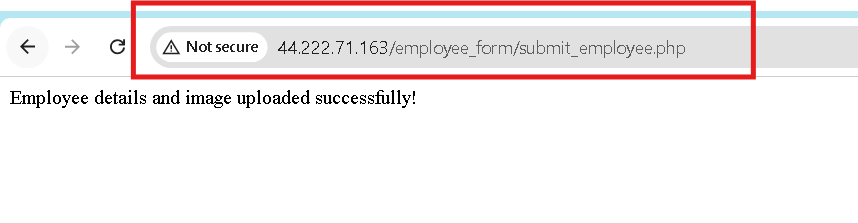
****

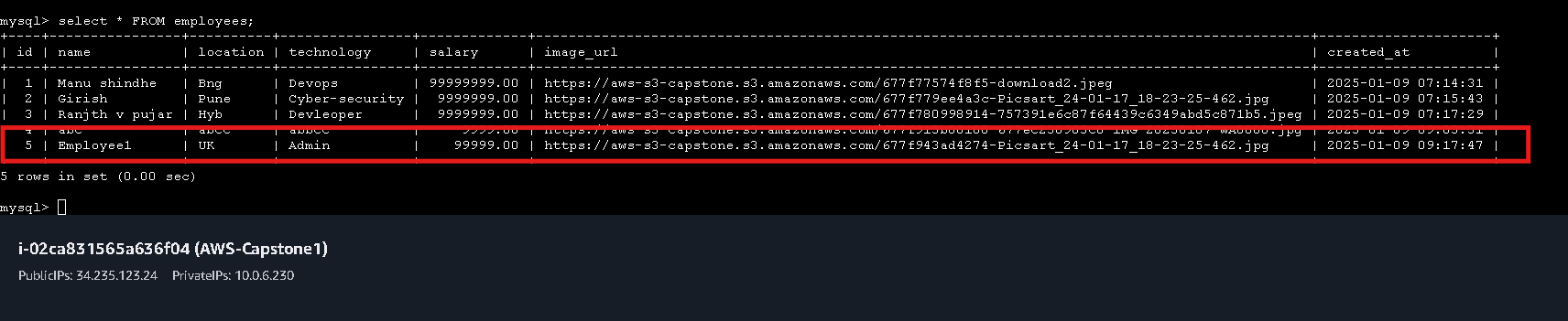
**Similarly, we will get same output for AWS-Capstone2 also. Once we information is provided over browser/internet by user this information will store backend in RDS table.**

**Now we will access AWS-bastion host**

****

****

****



**Step 10:**

Create S3 Event To send notification to SNS

Steps to Set up S3 Event Notification with SNS

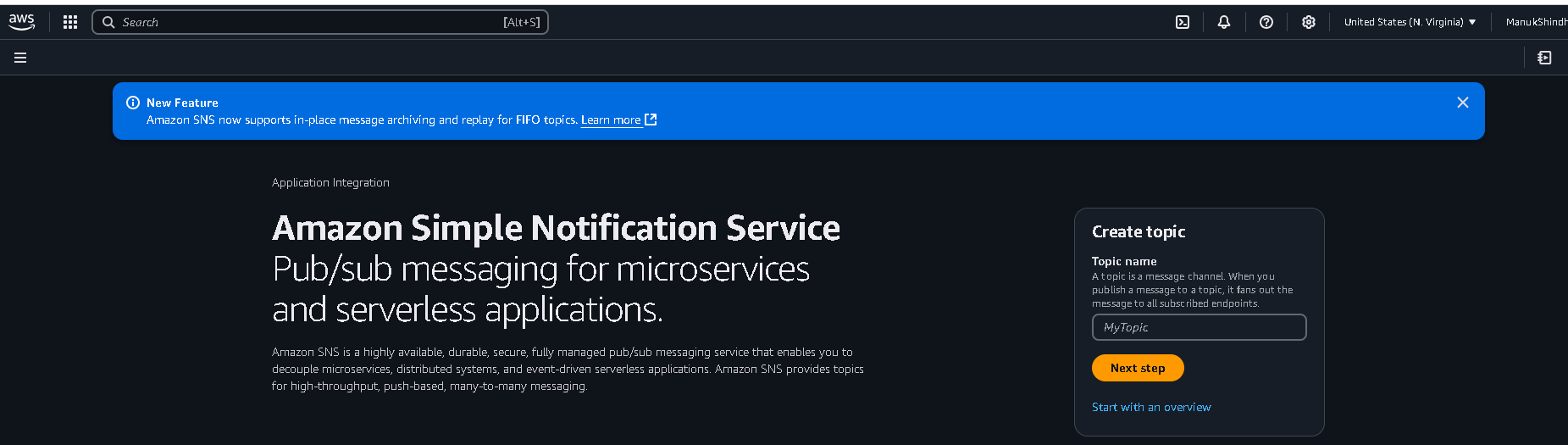
Step 1: Create an SNS Topic

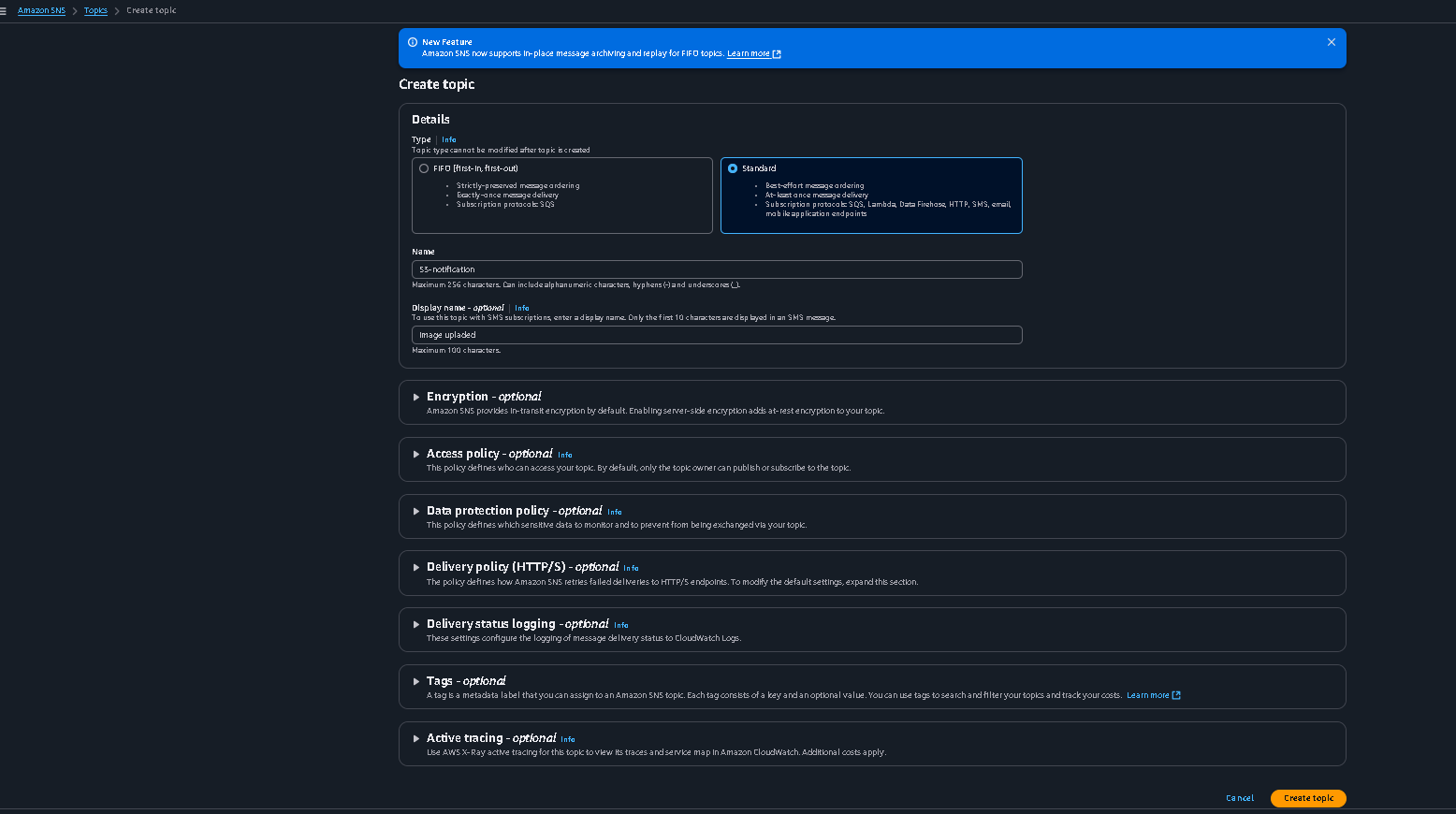
First, you need to create an SNS topic that will receive the notifications from your S3 bucket.

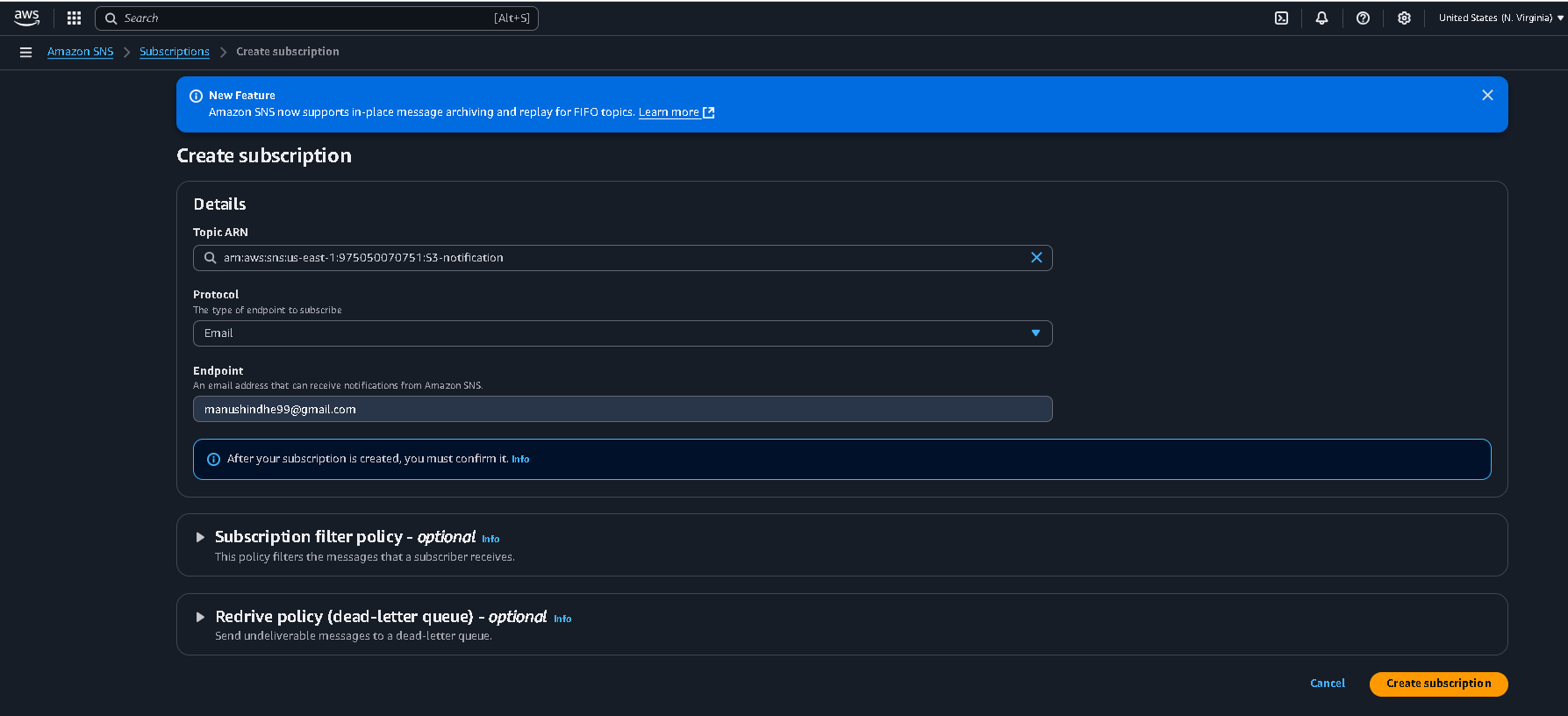
1. Log in to the AWS Management Console.
2. Navigate to SNS:
   * In the AWS Management Console, search for and select SNS (Simple Notification Service).
3. Create a Topic:
   * Click on Create topic.
   * Choose Standard as the type.
   * Give your topic a name
   * You can also add a display name if you like
   * Click Create topic to create the SNS topic.
4. Get the Topic ARN:
5. Step 2: Subscribe to the SNS Topic

Next, you'll want to subscribe to this SNS topic so that you can receive notifications.

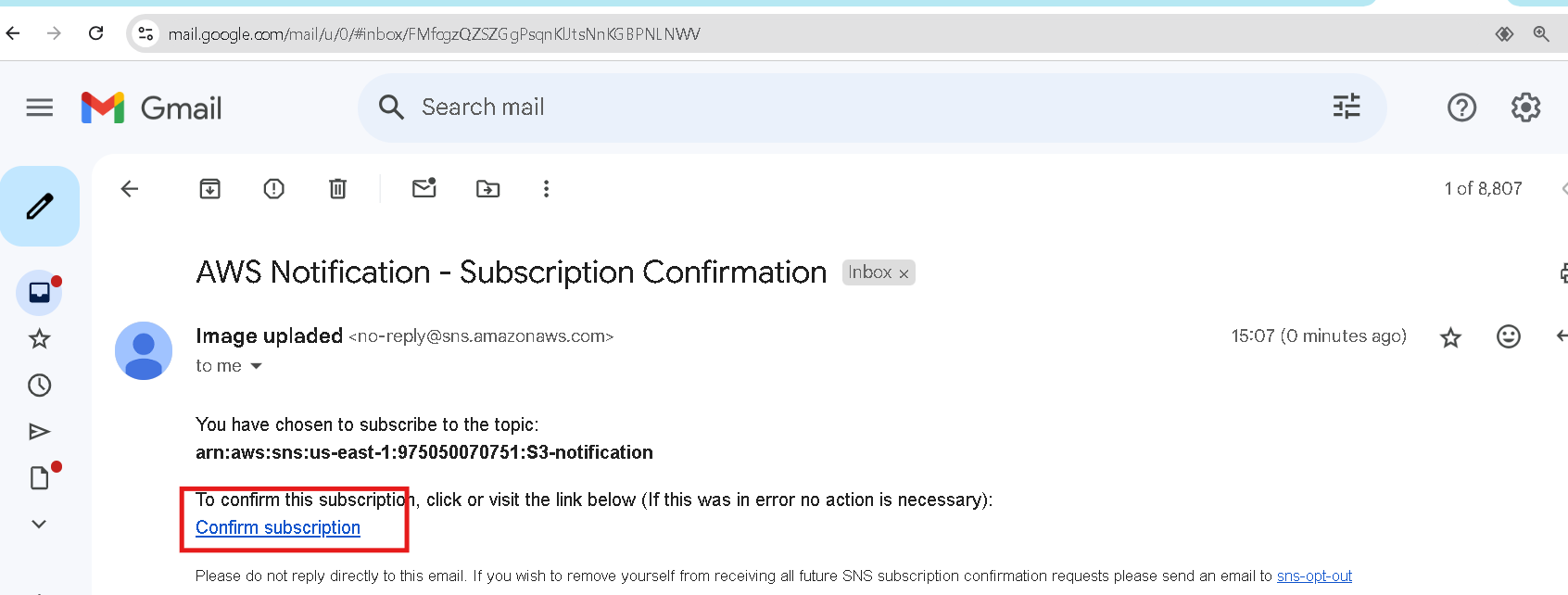
1. In the SNS dashboard, select the topic you just created.
2. Click on Create subscription.
3. Select the protocol (e.g., Email, SMS, Lambda, or HTTP/HTTPS) to receive the notifications.
   * For example, choose Email if you want to receive notifications via email.
4. Enter the endpoint (e.g., your email address for an email subscription).
5. Click Create subscription.
   * You will need to confirm the subscription if you chose Email or another endpoint that requires confirmation.
   * Check your email and click the confirmation link to confirm your subscription.



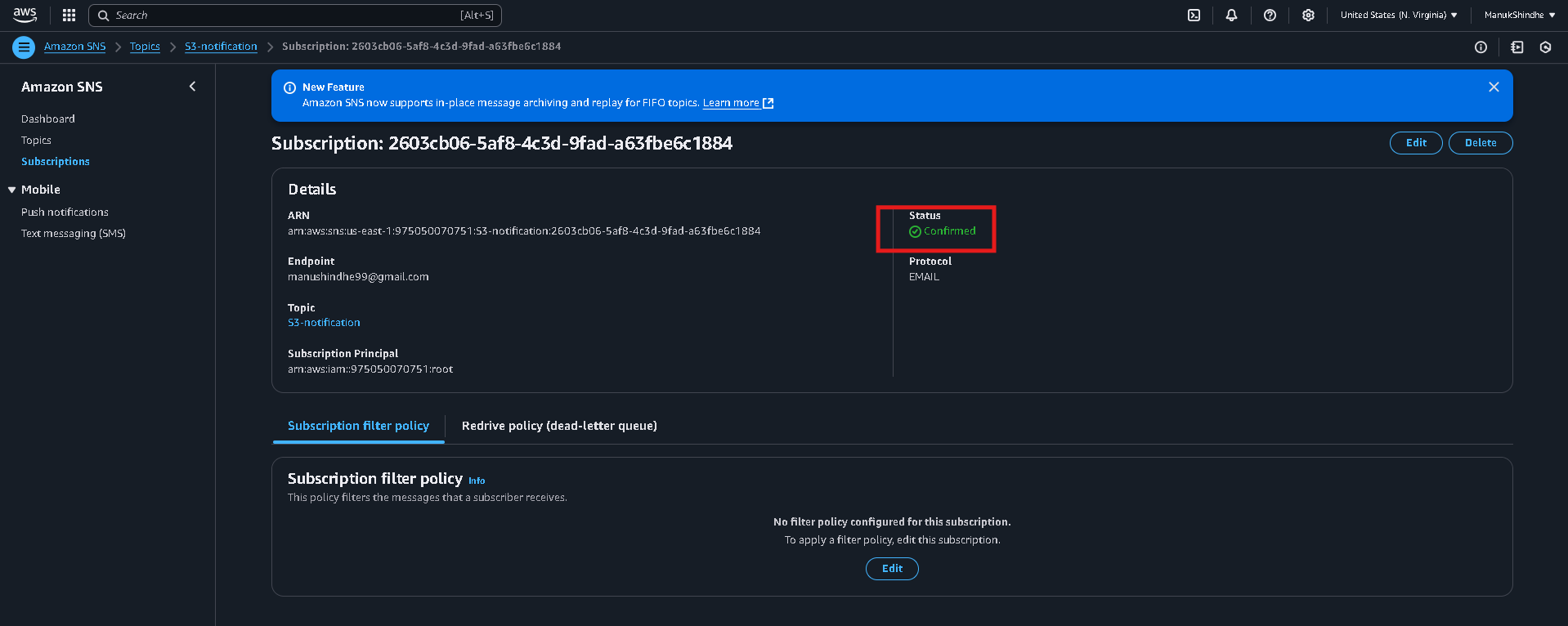




Will receive mail for subscription



Click on confirm



Step 3: Configure S3 Event Notification to Trigger SNS

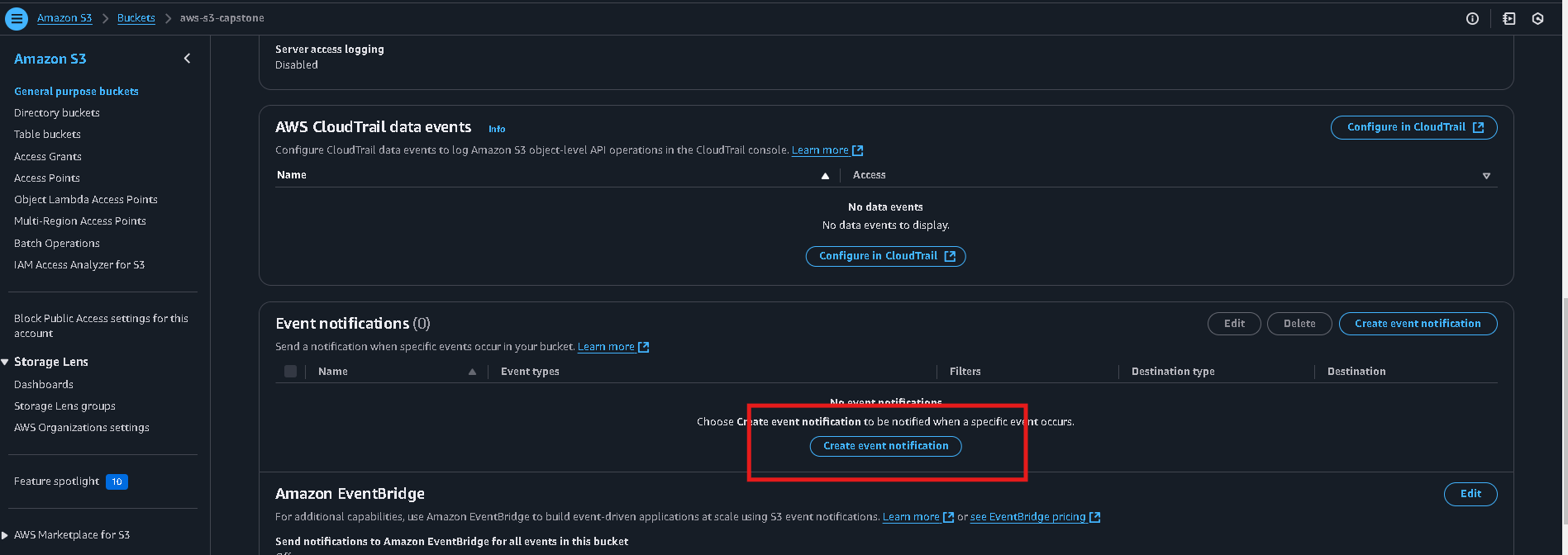
Now, you will configure the S3 bucket to send notifications to the SNS topic when specific events happen in the bucket (e.g., when an object is uploaded).

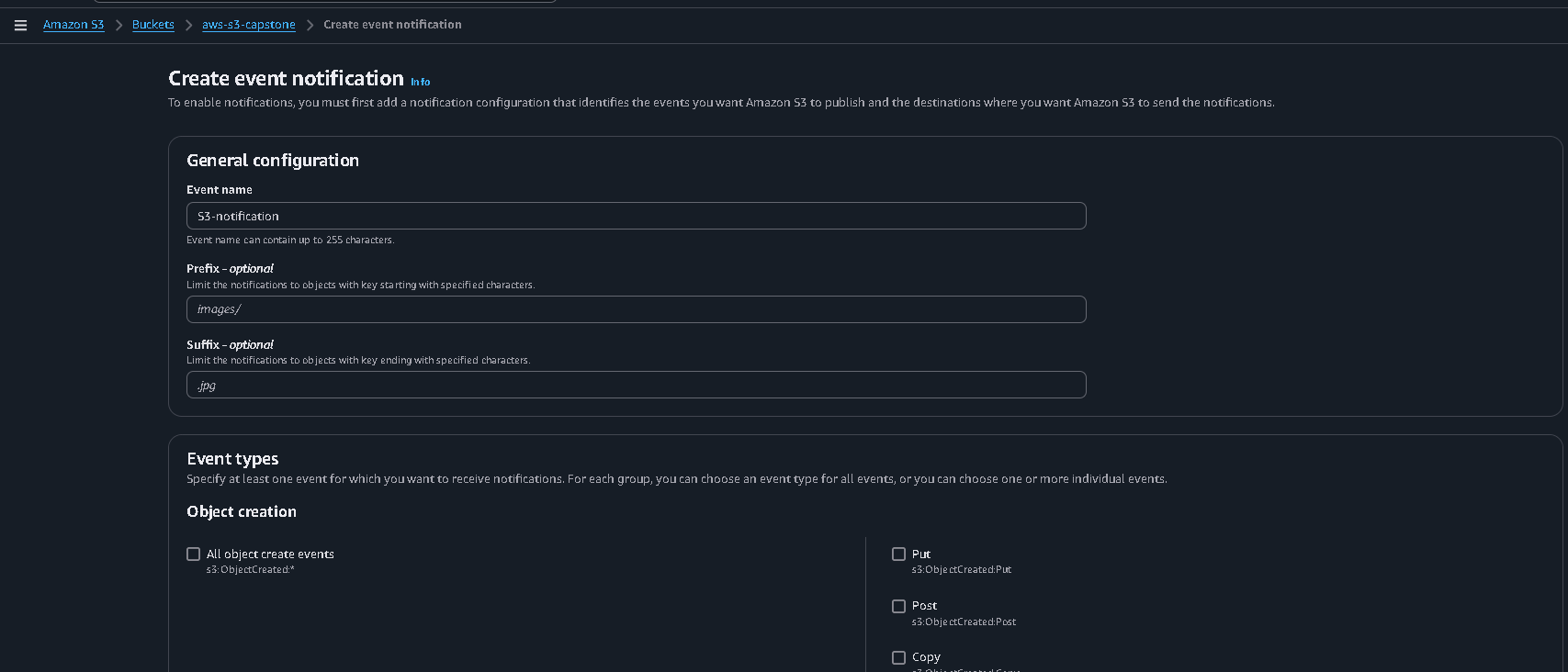
1. Navigate to S3:
   * In the AWS Management Console, search for and select S3.
2. Select Your S3 Bucket:
   * In the S3 console, select the bucket for which you want to configure the event notification.
3. Go to Properties:
   * In the bucket dashboard, go to the Properties tab.
4. Scroll to "Event Notifications":
   * Under the Event Notifications section, click Create event notification.
5. Configure the Event:
   * Event name: Give your event a name
   * Event types: Select the event type(s) you want to trigger the notification (e.g., ObjectCreated).
   * Prefix/Suffix (optional): You can specify a prefix (to filter by folder or object path) or a suffix (to filter by file extension). Leave blank for all objects.
   * Send to: Select SNS Topic.
   * SNS Topic ARN: Paste the SNS Topic ARN that you copied earlier.
6. Save the Notification:
   * After filling out the necessary fields, click Save changes.

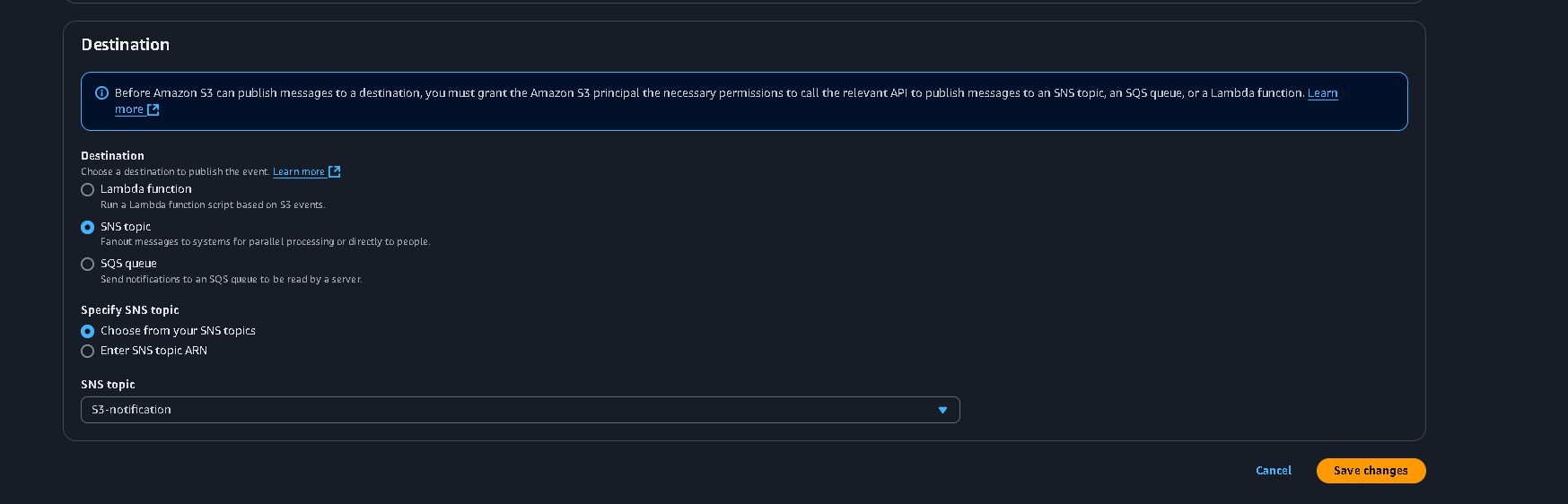
Step 4: Test the Setup

To test whether the S3 event notification to SNS is working:

1. Upload an object to your S3 bucket (either via the console or programmatically).
2. Check the SNS subscription (e.g., your email) to see if you receive a notification.







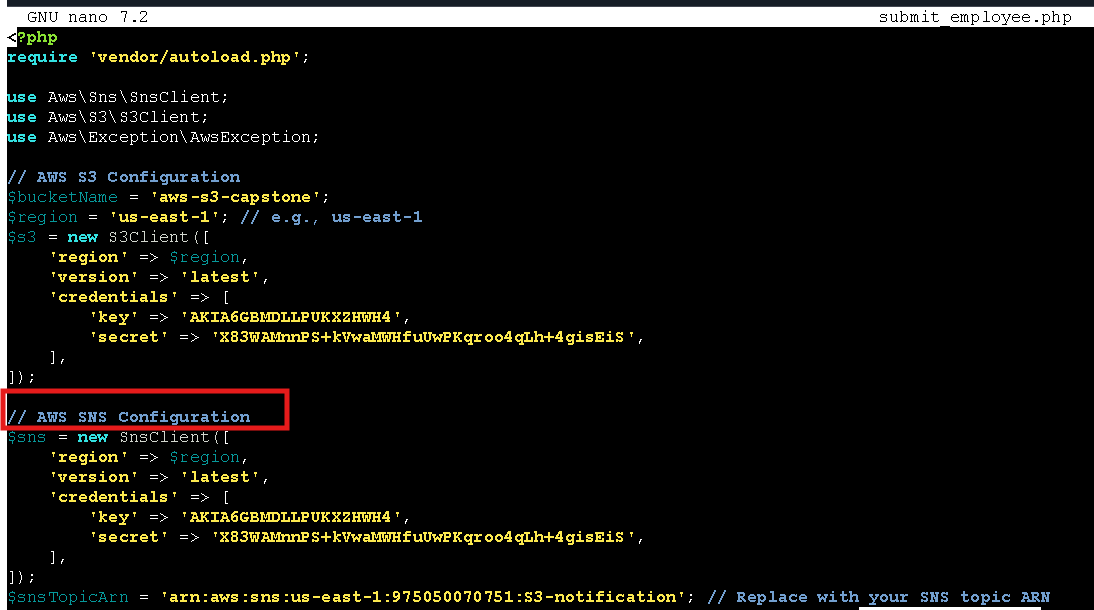
**Step 5: Update PHP Code to Publish to SNS**

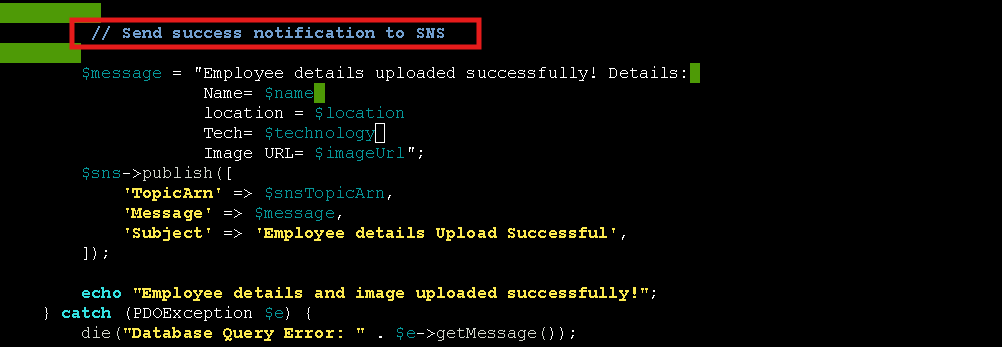
Now, modify your existing PHP code to send a notification to the SNS topic upon successful image upload to S3.

**Modify the PHP Code**:

In submit\_employee.php file, after successfully uploading the image to S3 and storing the data in your MySQL database, you will send a notification to the SNS topic.

Update code with the following to send a success message to the SNS topic:

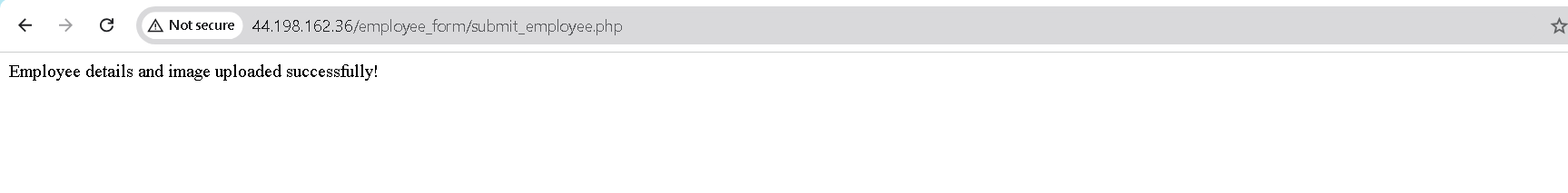


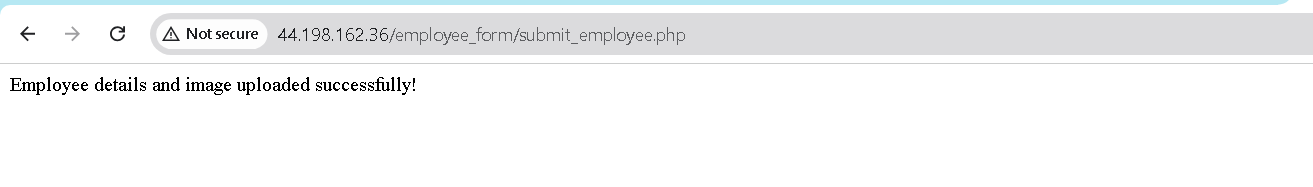


**Step 6: Test the Integration**

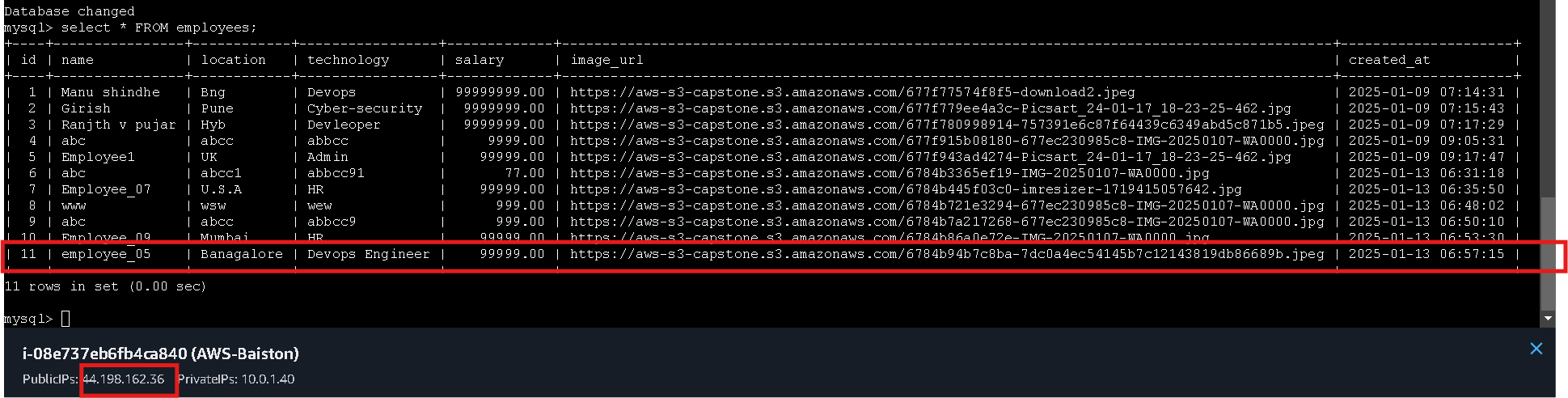
1. **Upload an image** and employee details using form.
2. Once the upload is successful, check email inbox for the notification from SNS.
3. If the email is successfully sent, you should see a message containing the Employee details, image URL and success notification.



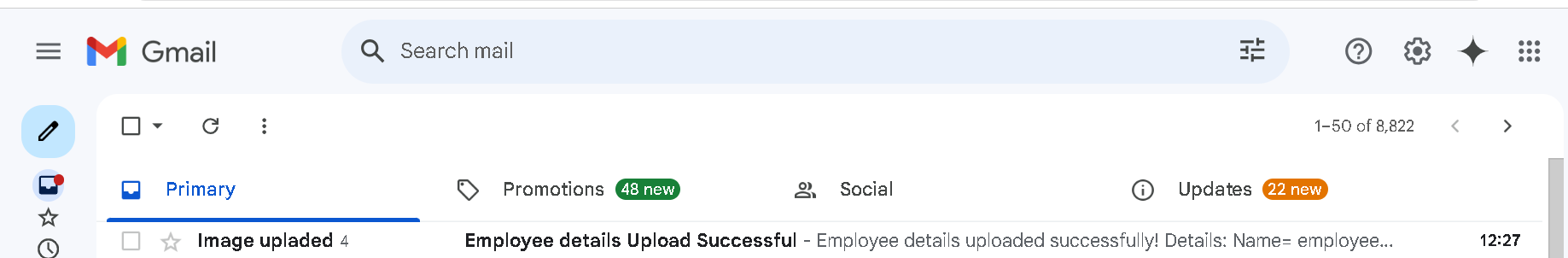


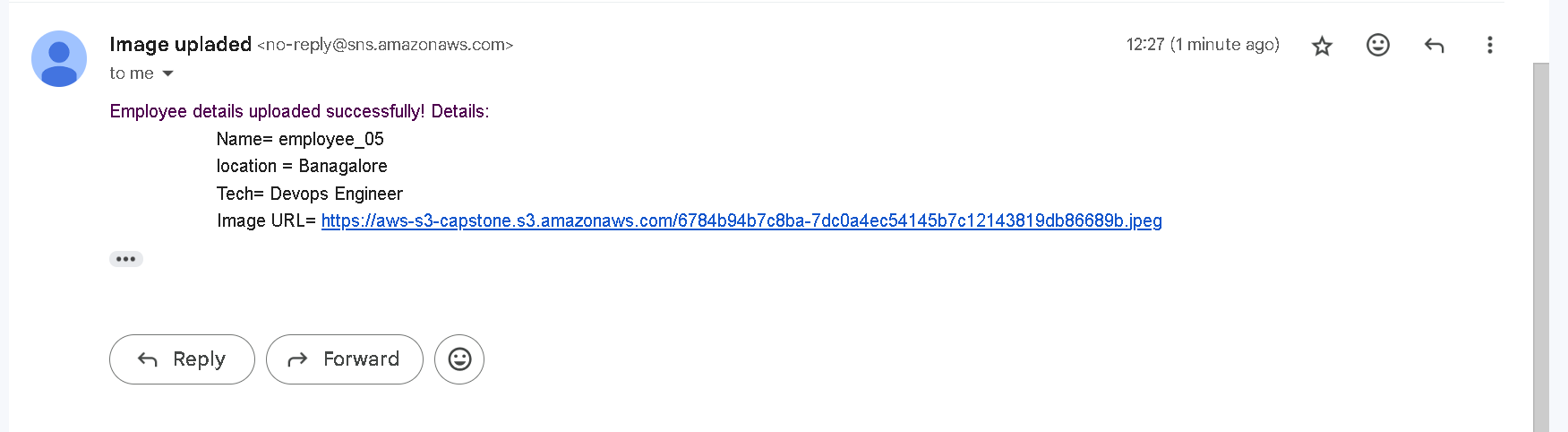


Now we will check in database whether details are stored.



Now we can verify email inbox for the notification from SNS.





This is how we can verify our application - – *New employees input their information and upload photos. Existing employees can get their information*.

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