**Name – Nandan Shailesh Kasat**

**Batch – Jan-Feb 2024**

**Terraform Task\_2**

Q1. Install Terraform in local machine, configure AWS provider. Initialize Terraform configuration.

Create a VPC with CIDR block 192.168.0.0/16

Create two public subnets in different availability zones within the VPC.

Create an Internet Gateway and attach it to the VPC.

Create a route table and associate it with the public subnets, setting the default route to the Internet Gateway.

Create security groups allowing HTTP (port 80) and SSH (port 22) access.

Note: Provide access\_key, secret\_key and region, CIDR for vpc and subnets using input variable.

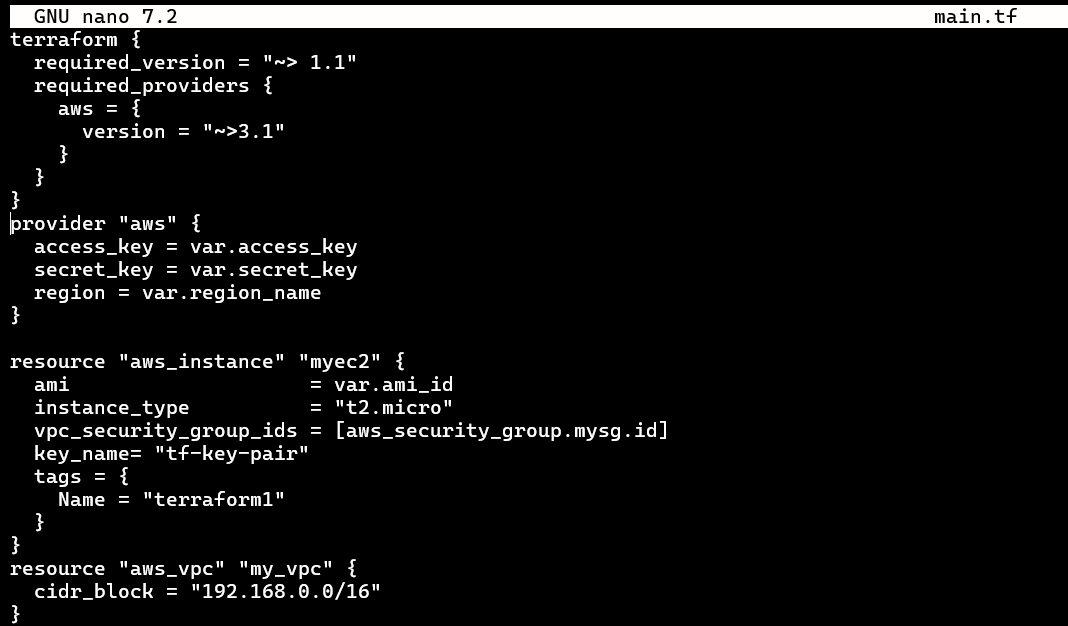
And display vpc-id, sg-id

Ans:

**1. Install Terraform and Configure AWS Provider**:

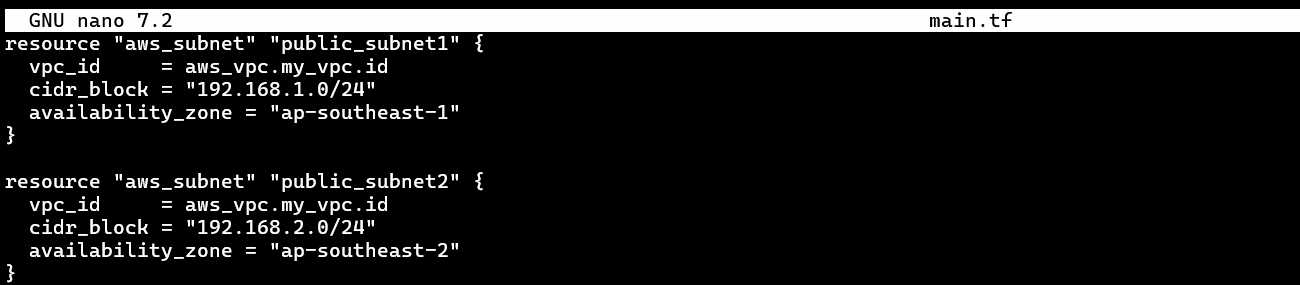
**2. Create a VPC**:

Define your VPC resource in Terraform. Specify the CIDR block (e.g., 192.168.0.0/16) for your VPC.



**3. Create Public Subnets**:

Create two public subnets in different availability zones (AZs) within your VPC.



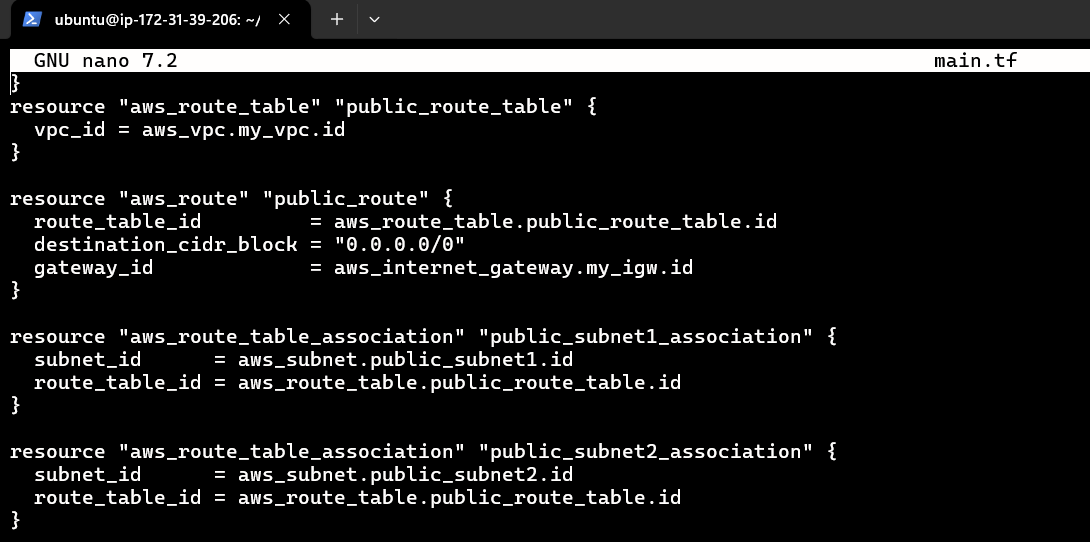
**4. Create an Internet Gateway (IGW)**:

Deploy an IGW and associate it with your VPC to enable internet traffic.



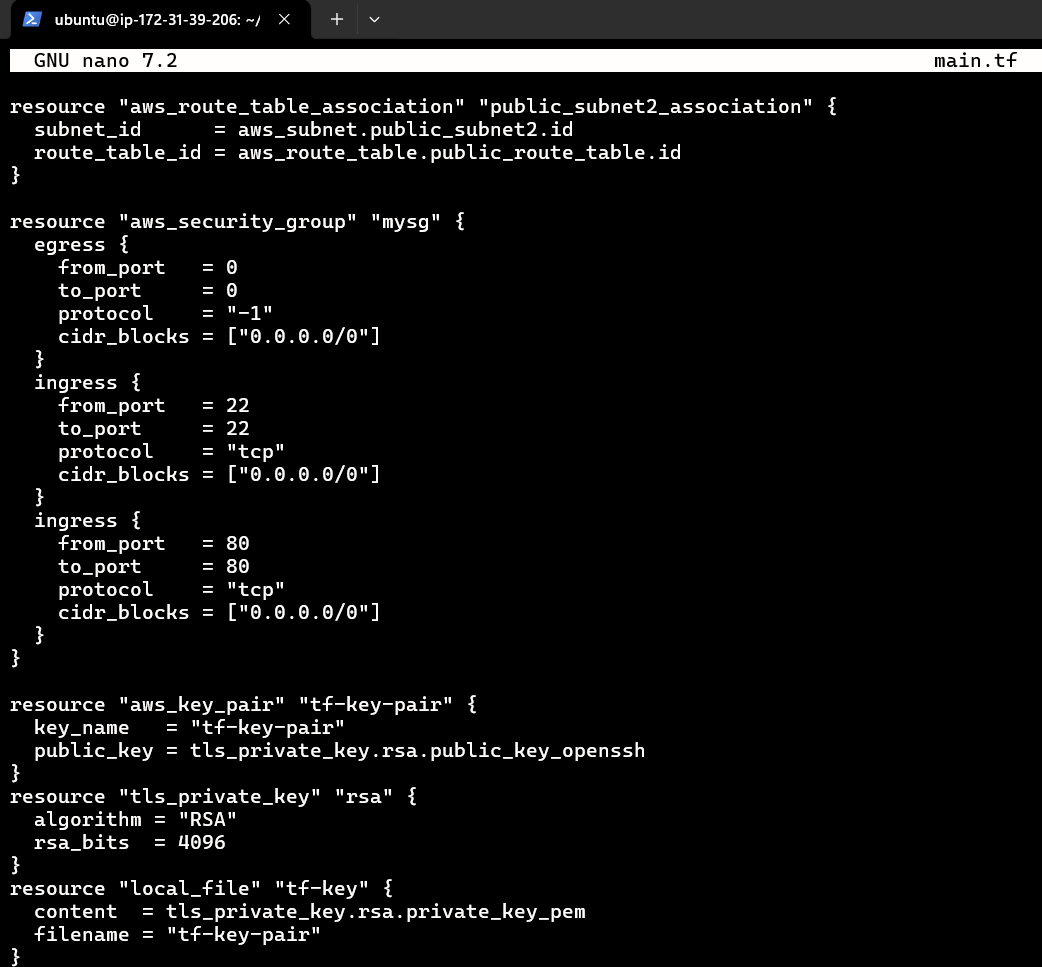
**5.** **Create a Route Table**:

Define a route table for your VPC. Associate the public subnets with this route table. Set the default route to the IGW.

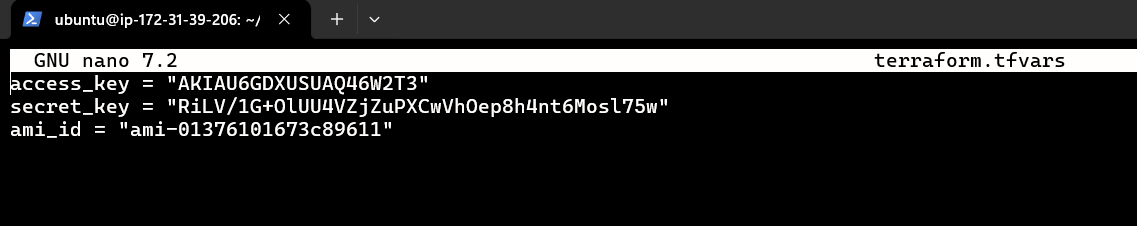


**6. Create Security Groups**:

Define security groups allowing HTTP (port 80) and SSH (port 22) access.

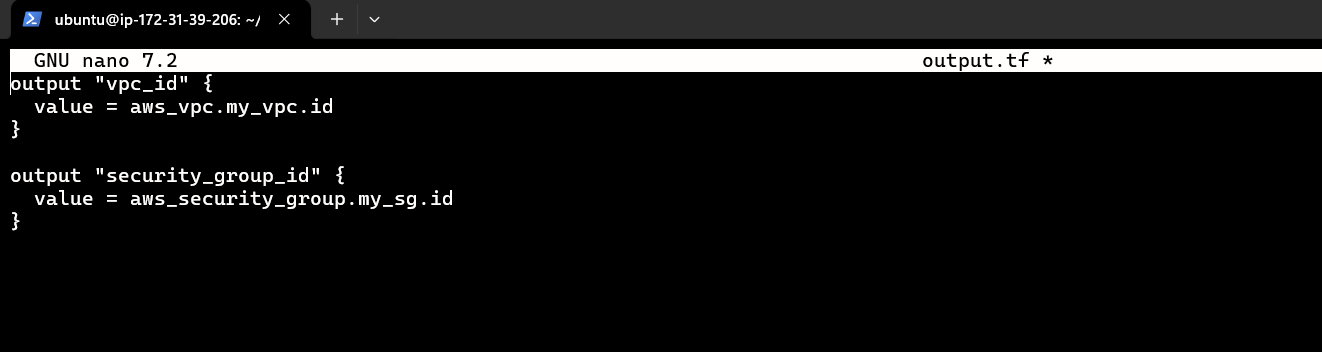


In Terraform.tfvars mention the Access Key, Secret Key and ami\_id.

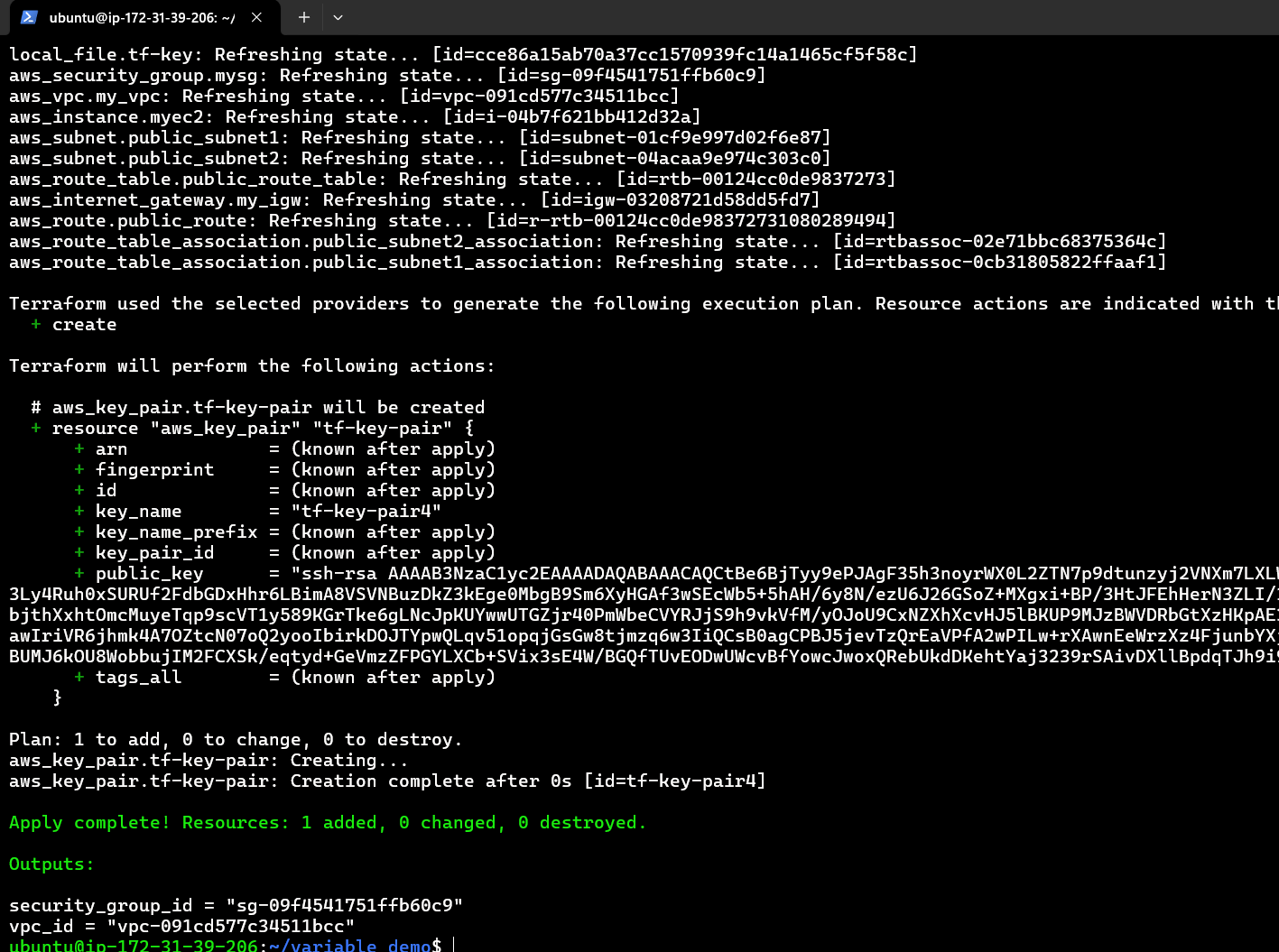


**7. Display VPC ID and Security Group ID**:

After applying your Terraform configuration, you can retrieve the VPC ID and security group ID using Terraform outputs.



**8. Final Output:**

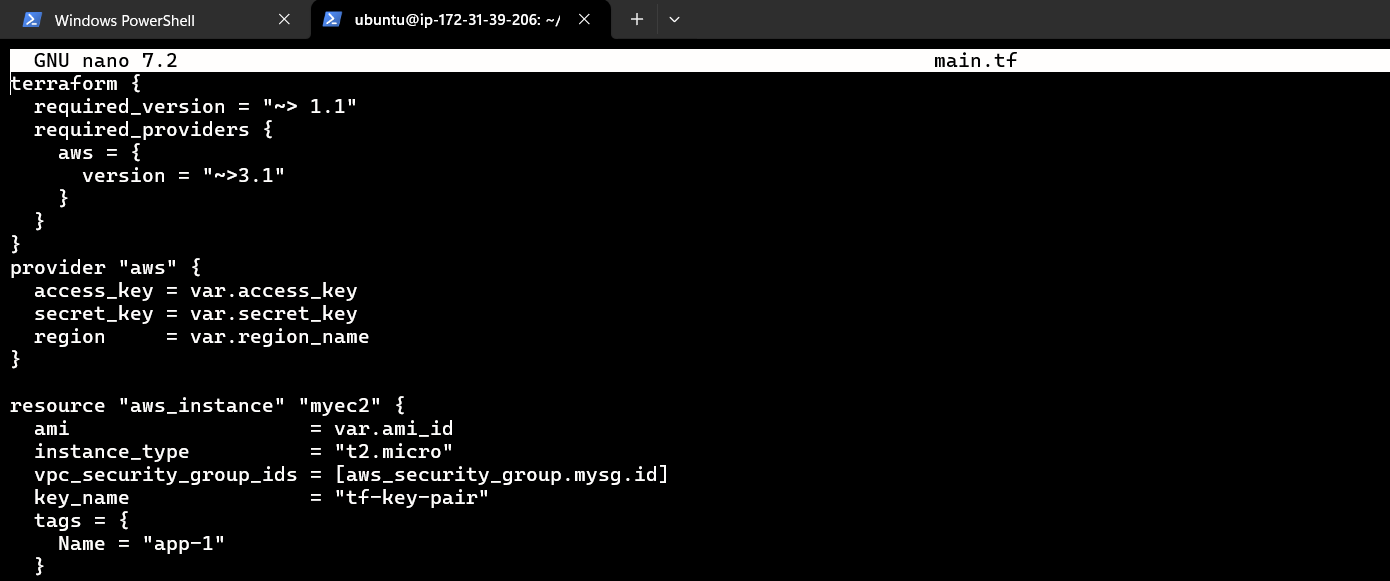


Q2. Launch an EC2 instances with names “app-1” and install apache, create two pages at its default location using provisioner block. Display webpages on browser.

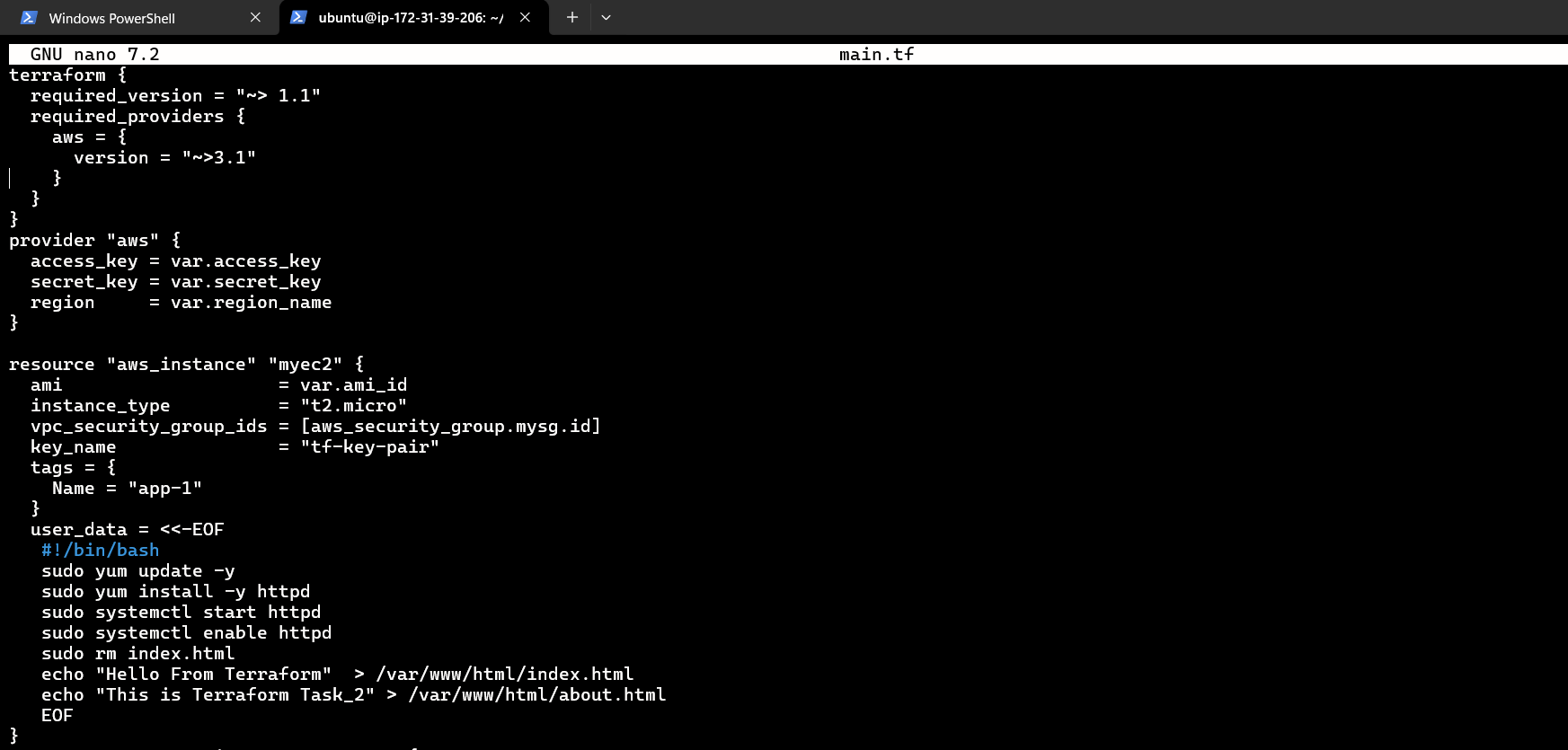
Ans:

**1. Launch an EC2 instance name “app\_1”**

Define your EC2 instance resource in Terraform. Set the instance name to “app-1” and choose an appropriate instance type (e.g., t2.micro).



**2. Install Apache**:

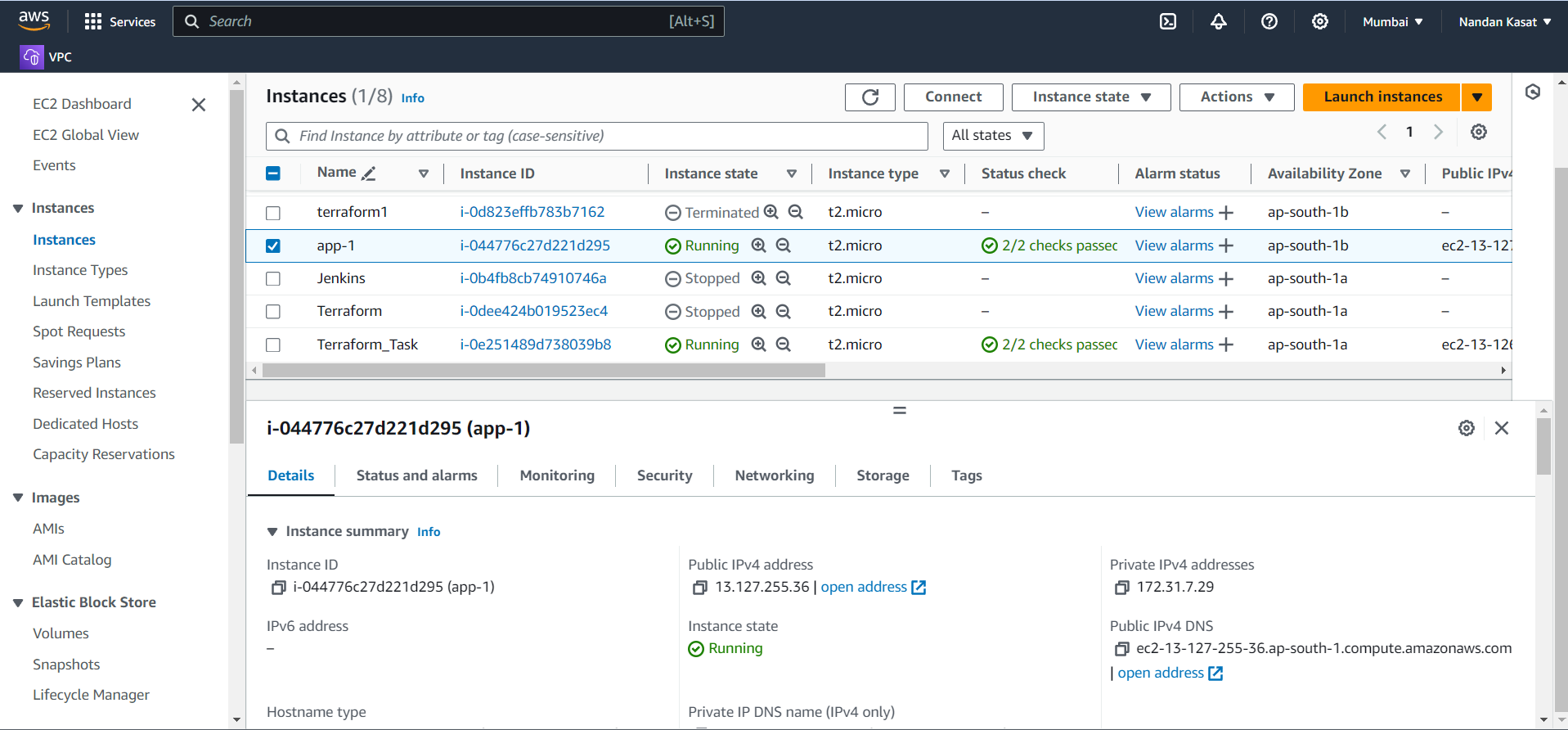


**3. Create Web Pages**:

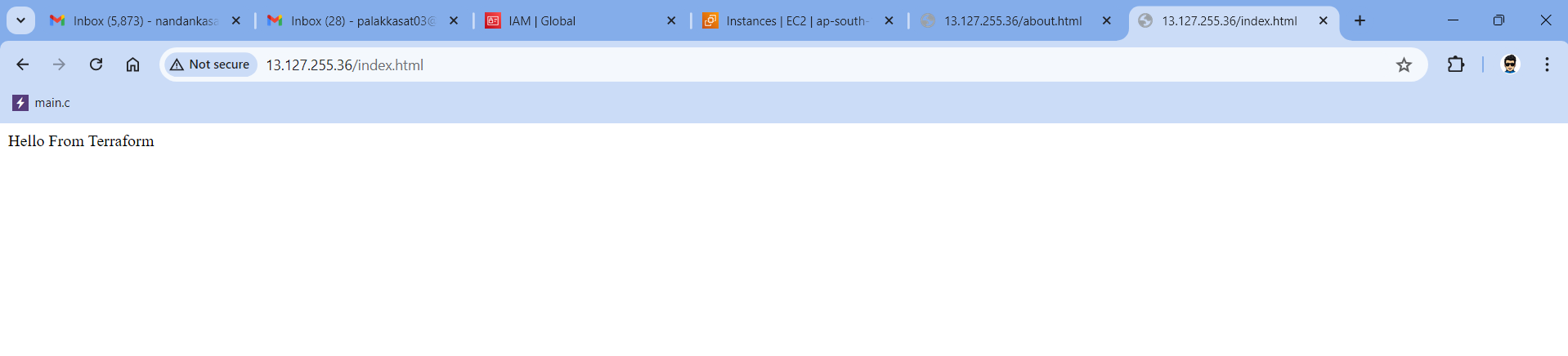
After installing Apache, create two HTML pages (e.g., index.html and about.html) in the default web server location (/var/www/html).



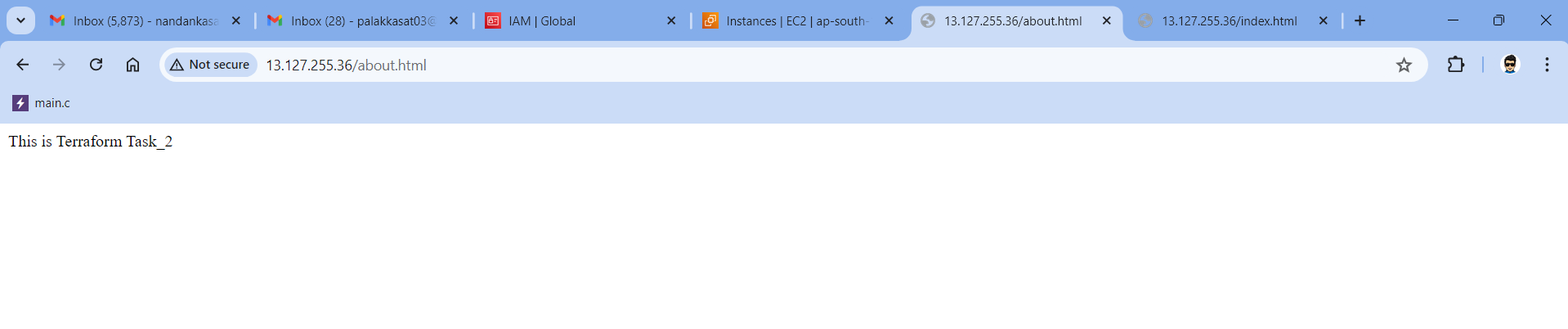
**4. You can see instance app\_1 on dashboard.**

****

**5. Index.html**

****

**6. About.html**

****

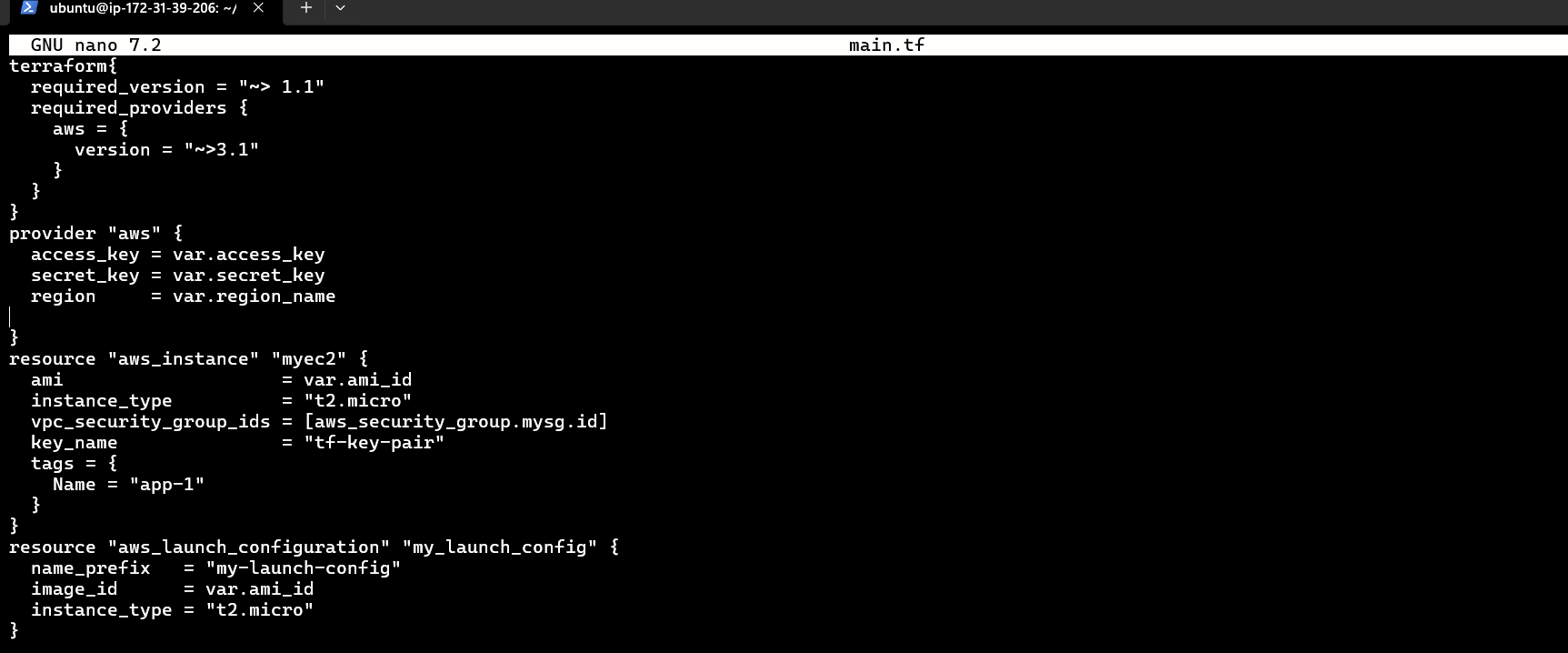
Q3. Create an Auto Scaling Group with a Launch Configuration to manage the EC2 instances, using Teeraform.

Ans:

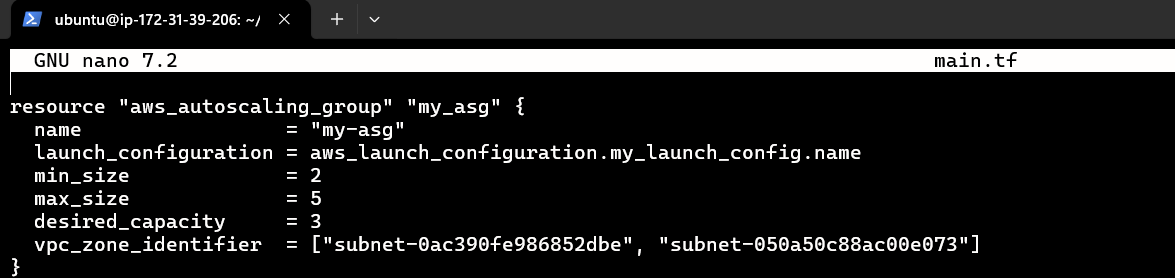
**1. Start the Instance:**

**2. Define a Launch Configuration:**

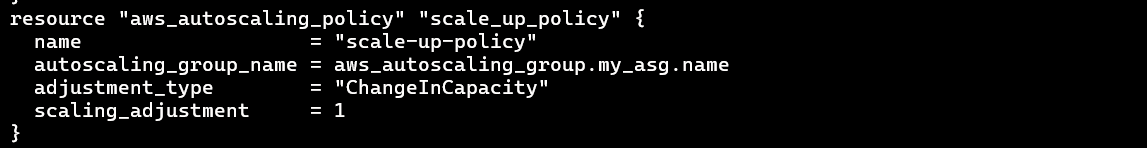
First, create a launch configuration that defines the instance specifications.

****

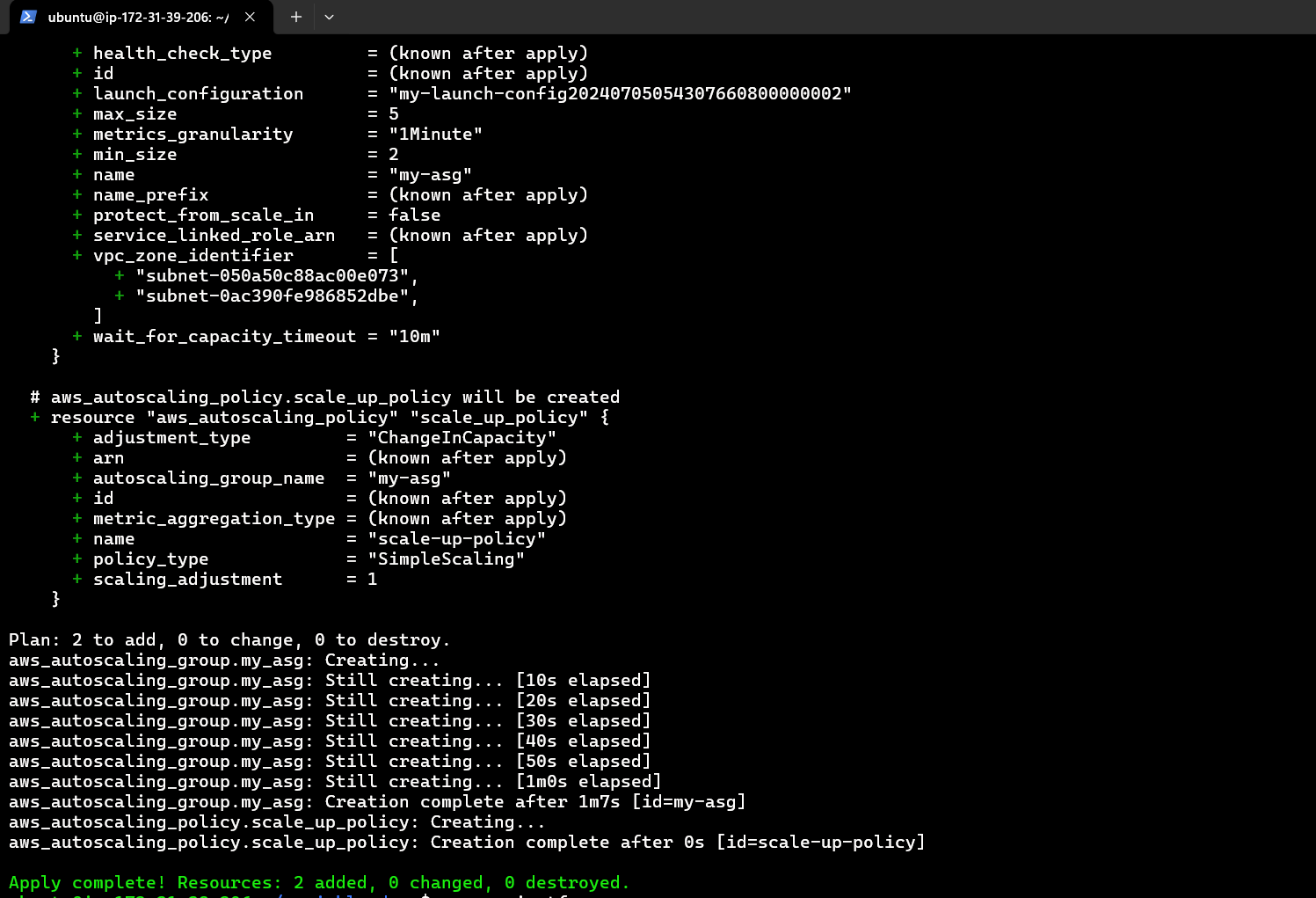
**3. Create an Auto Scaling Group:**Define your ASG resource, referencing the launch template.

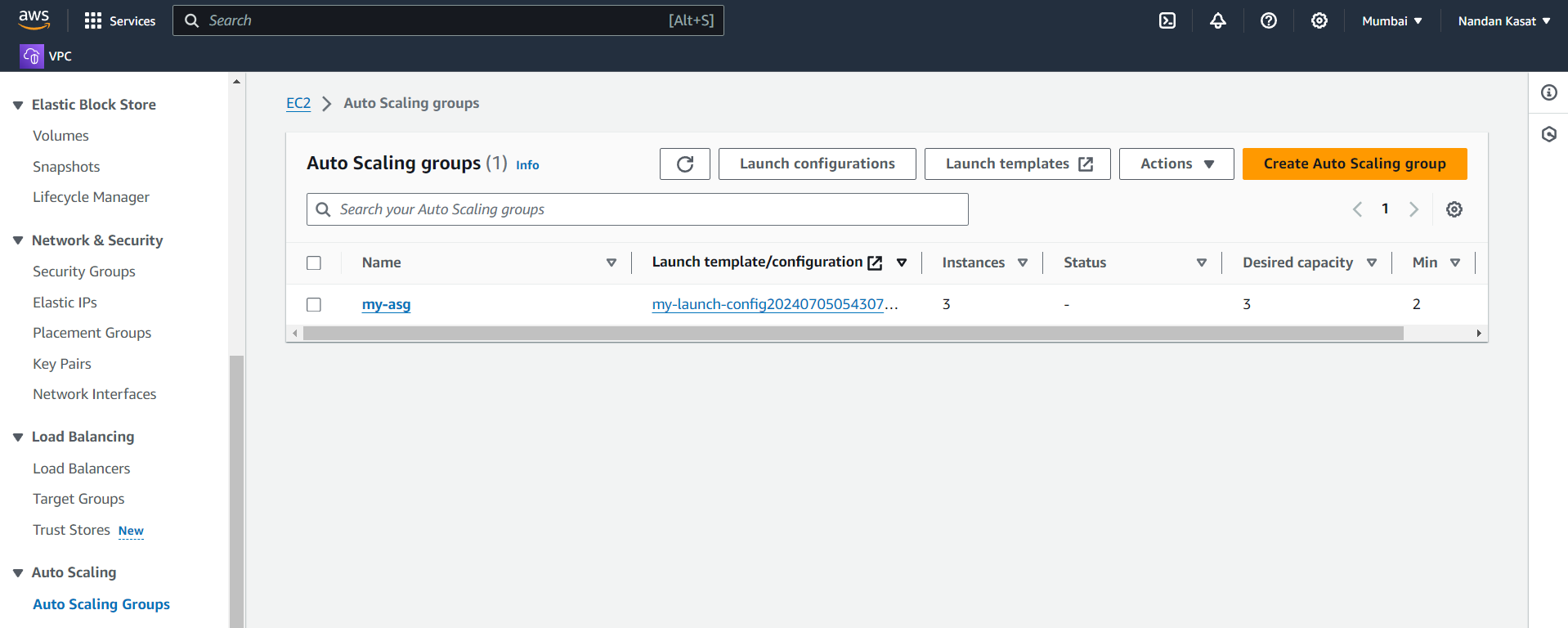
****

**4. Scaling Policies:**We can define scaling policies based on metrics like CPU utilization or custom metrics.Attach these policies to your ASG.

****

**5. The Auto Scaling Group is created on the Dashboard.**

****

****