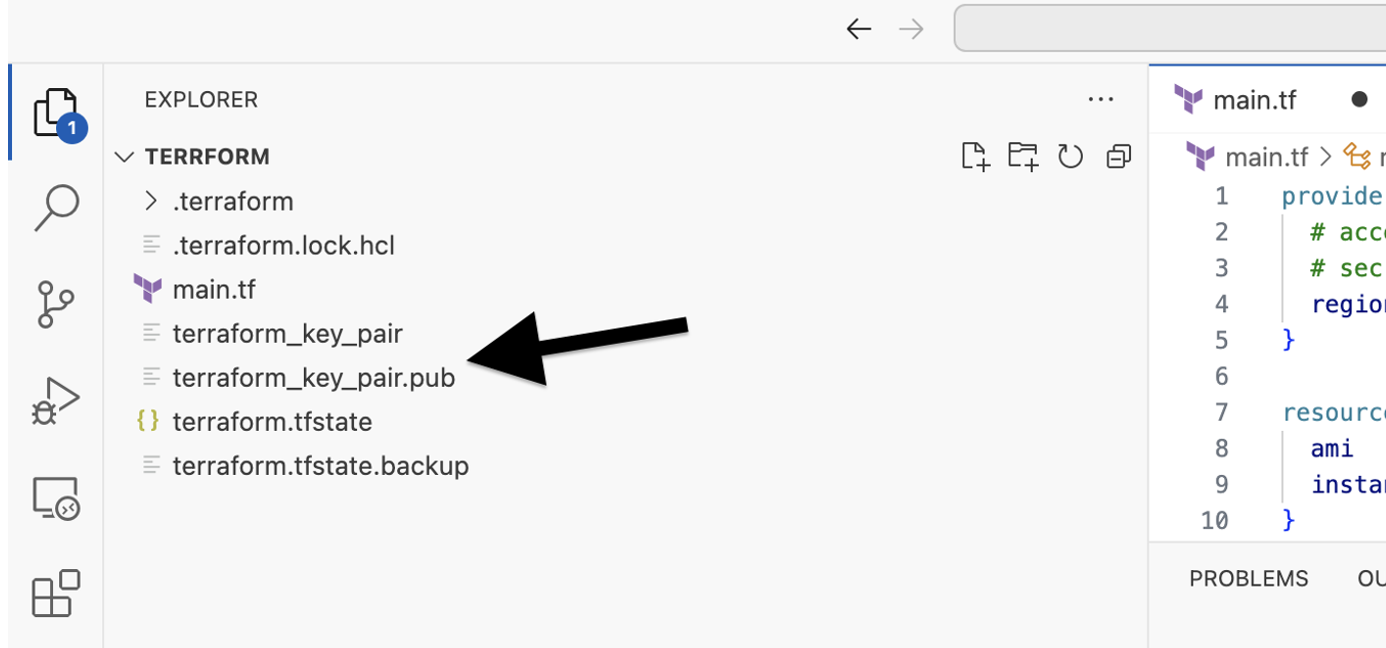
Task: Create an Instance, assign SG, key\_pair, deploy the static website using the user datya and access it by using public IP.

Steps:

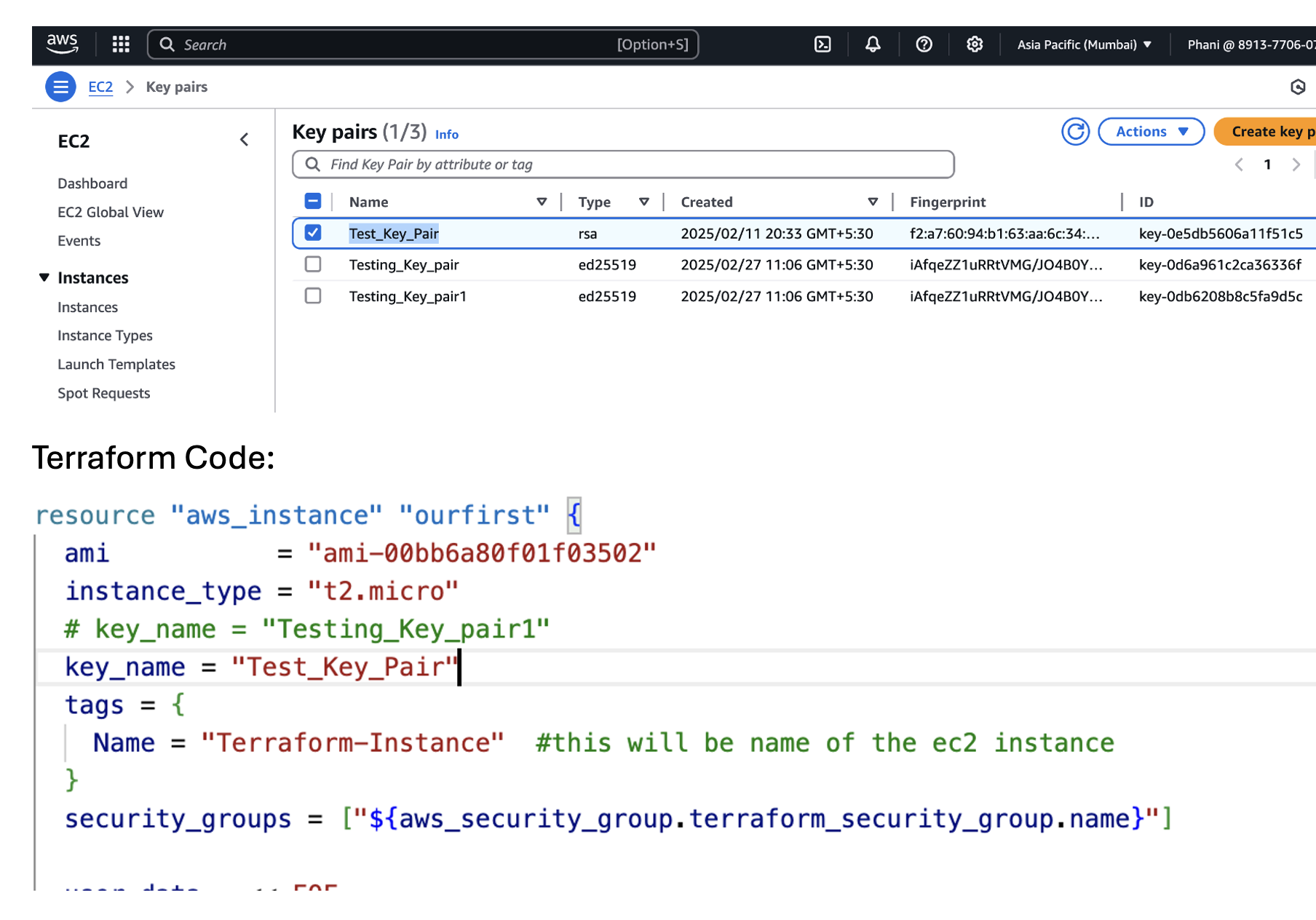
1. Create a resource for key value pair to access the instance from our local machine
2. Create a security group with ssh and http inbound rules
3. Create an instance and add key value pair, security group
4. Write a script in user data to download the static website and move the files to the apache server(Here we will be using apache2)  
   <https://www.free-css.com/free-css-templates>

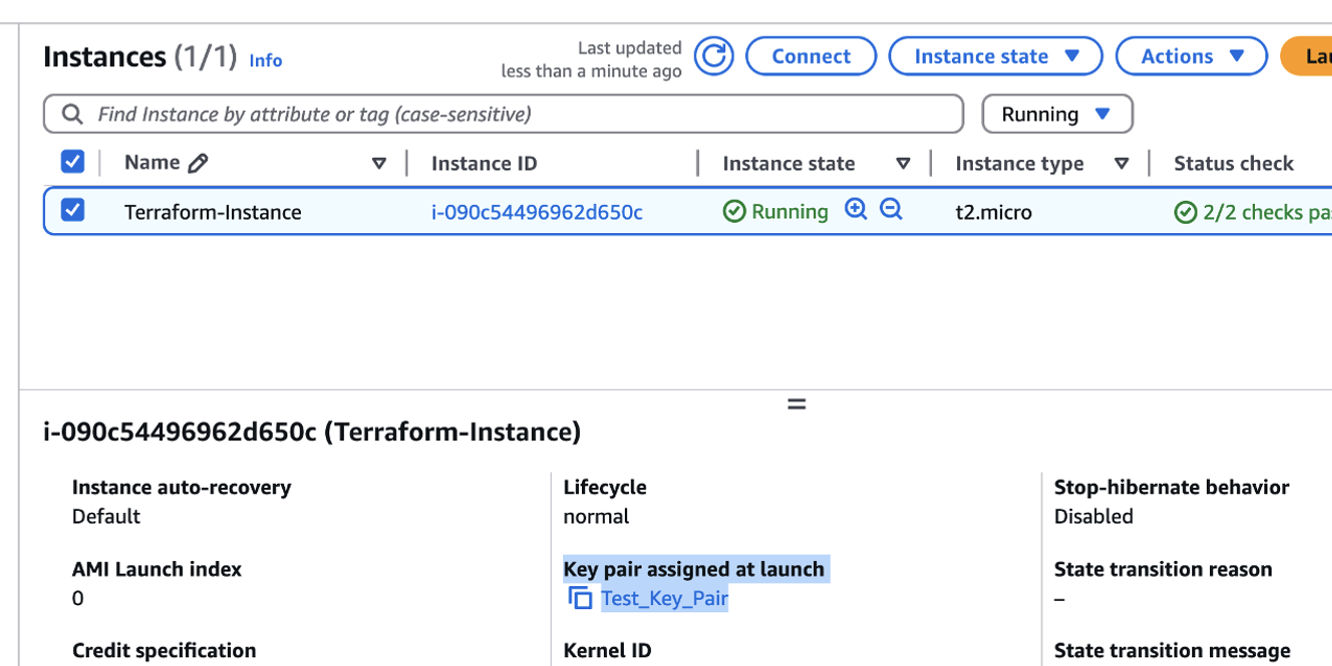
Commands:

* terraform plan
* terraform validate
* terraform apply
* terraform destroy

* Create key pairs
  1. <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/key_pair>
  2. we can manually pass the key value pairs or we can store them in a file and we can use that file
  3. we can use the below command to generate the ssh key and store it in a file in the current directory.  
     **ssh-keygen -f terraform\_key\_pair  
       
       
     **
  4. Syntax for passing the values directly and from the file

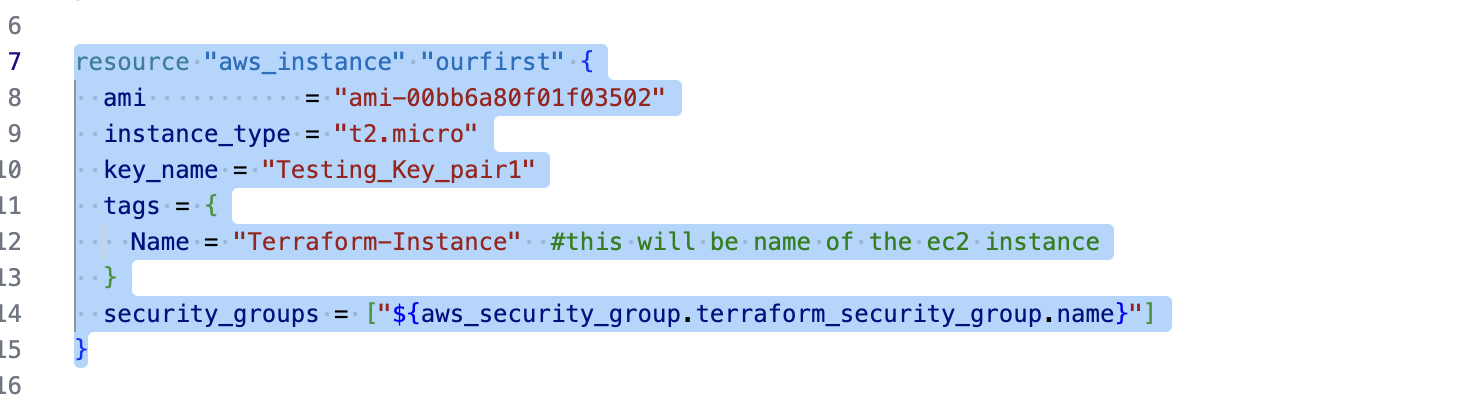
|  |
| --- |
| resource "aws\_instance" "ourfirst" {  ami = "ami-00bb6a80f01f03502"  instance\_type = "t2.micro"  key\_name = "Testing\_Key\_pair1"  tags = {  Name = "Terraform-Instance" #this will be name of the ec2 instance  }  }  #passing the key pair values directly  resource "aws\_key\_pair" "terrform\_key\_pair" {  key\_name = "Testing\_Key\_pair"  public\_key = "ssh-ed25519 hjdfbhd"  }  #passing the key pair values from the file  resource "aws\_key\_pair" "terrform\_key\_pair1" {  key\_name = "Testing\_Key\_pair1"  public\_key = "${file("terraform\_key\_pair.pub")}"  } |

* 1. Here we can also make use of the key pairs which are already present in the aws.  
     



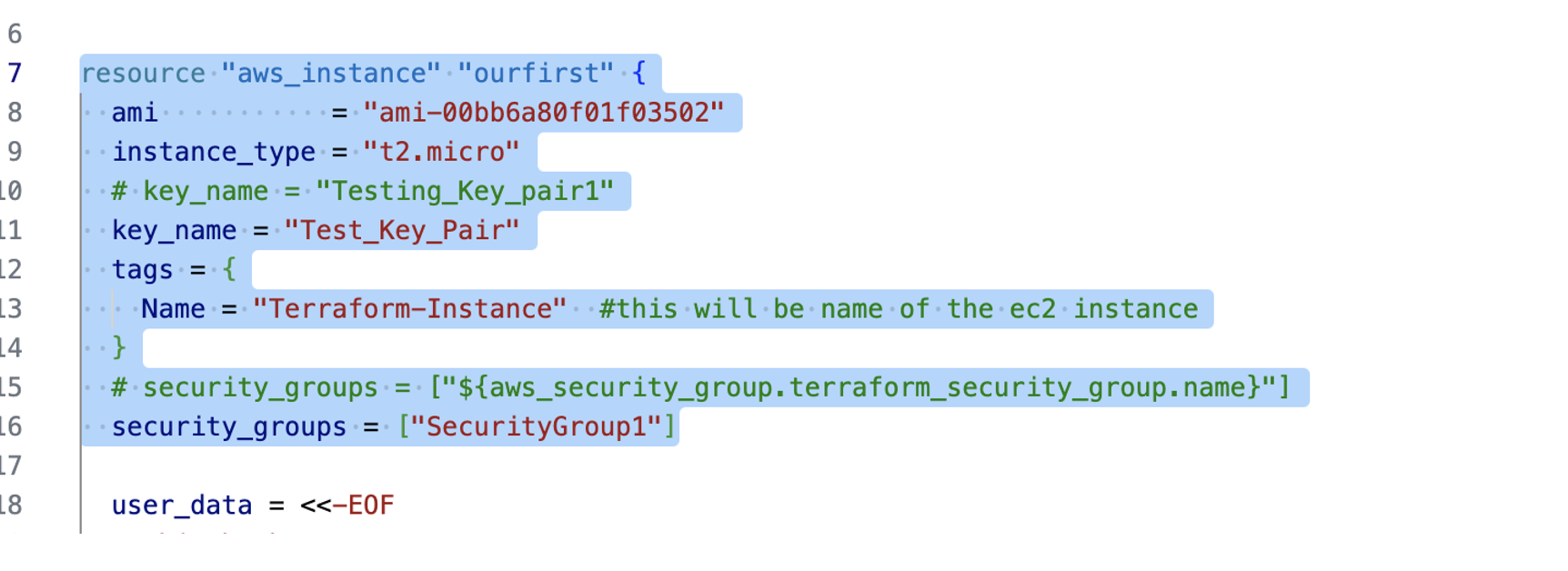
* Create security group
  1. <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/security_group>
  2. ingress are the inbound rules
  3. egress are the outbound rules
  4. From the console  
     A screenshot of a computer

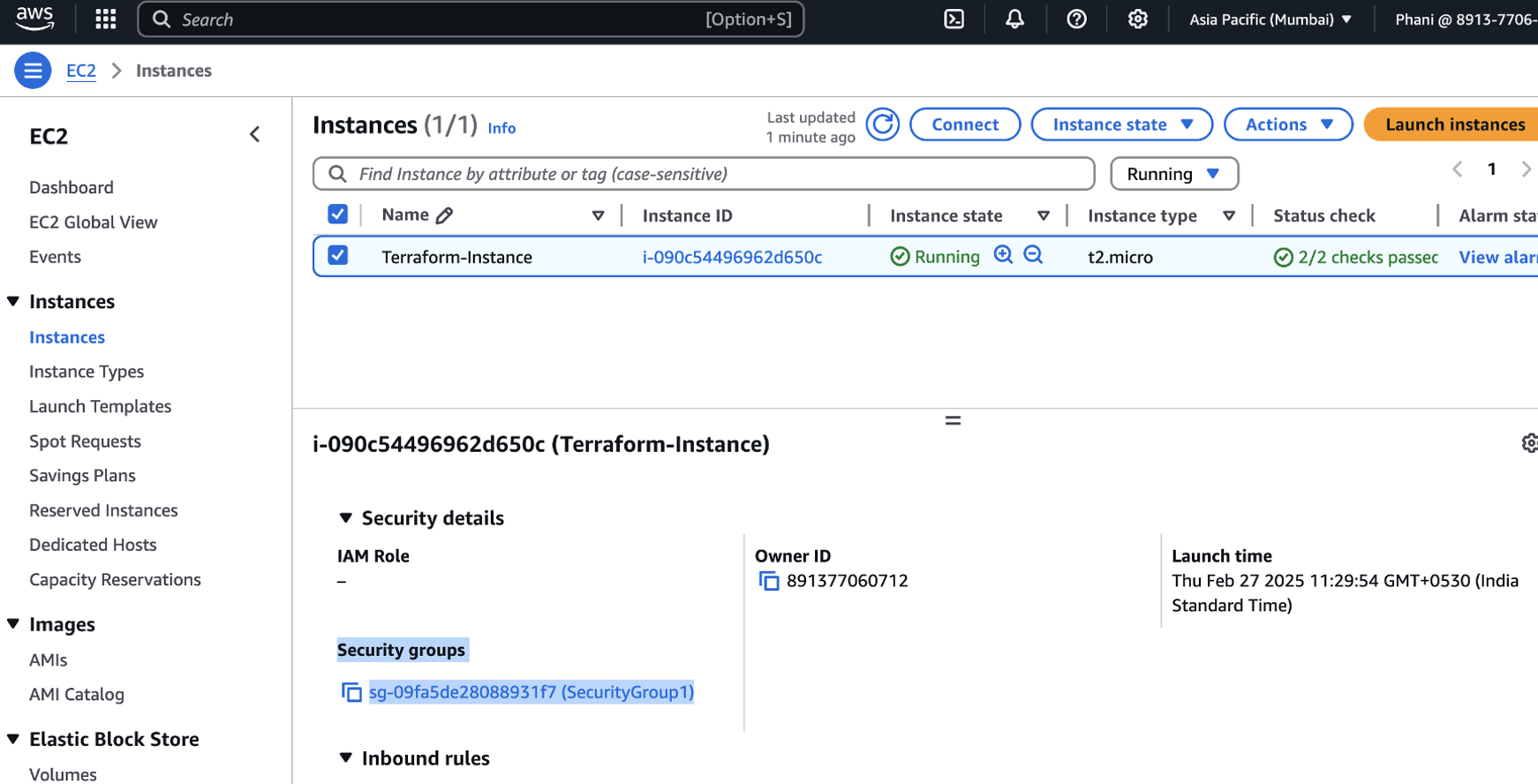
     AI-generated content may be incorrect.
  5. Terraform code  
     A screenshot of a computer program

     AI-generated content may be incorrect.  
       
     

* 1. Here we can also make use of the security group which are already present in the aws.  
     A screenshot of a computer

     AI-generated content may be incorrect.





* User data
  1. user data code:

|  |
| --- |
| user\_data = <<-EOF  #!/bin/bash  sudo apt update -y  sudo apt install zip -y  sudo apt install apache2 -y  cd /var/www/html/  sudo rm index.html  sudo wget https://www.free-css.com/assets/files/free-css-templates/download/page296/neogym.zip  sudo unzip neogym.zip  sudo cp -r neogym-html/\* .  EOF |

* 1. Complete instance code  
     
* Complete code:

|  |
| --- |
| provider "aws" {  # access\_key = "AKIA47C"  # secret\_key = "R8OwUh"  region = "ap-south-1"  }  resource "aws\_instance" "ourfirst" {  ami = "ami-00bb6a80f01f03502"  instance\_type = "t2.micro"  # key\_name = "Testing\_Key\_pair1"  key\_name = "Test\_Key\_Pair"  tags = {  Name = "Terraform-Instance" #this will be name of the ec2 instance  }  # security\_groups = ["${aws\_security\_group.terraform\_security\_group.name}"]  security\_groups = ["SecurityGroup1"]  user\_data = <<-EOF  #!/bin/bash  sudo apt update -y  sudo apt install zip -y  sudo apt install apache2 -y  cd /var/www/html/  sudo rm index.html  sudo wget https://www.free-css.com/assets/files/free-css-templates/download/page296/neogym.zip  sudo unzip neogym.zip  sudo cp -r neogym-html/\* .  EOF  }  #passing the key pair values directly  resource "aws\_key\_pair" "terrform\_key\_pair" {  key\_name = "Testing\_Key\_pair"  public\_key = "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILW25lLS+wWjtcJVMHUsXtZl/f1esl7ryi+5SmptCeh8 phaninandigam@FVFHL0LQQ05D"  }  #passing the key pair values from the file  resource "aws\_key\_pair" "terrform\_key\_pair1" {  key\_name = "Testing\_Key\_pair1"  public\_key = "${file("terraform\_key\_pair.pub")}"  }  #security group creation  resource "aws\_security\_group" "terraform\_security\_group" {  name = "Testing\_Security\_Group"  description = "Allowing SSH and HTTP ports"    ingress { #allowing ssh port  from\_port = 22  to\_port = 22  protocol = "tcp"  cidr\_blocks = ["0.0.0.0/0"] #from anywhere  }  ingress { #allowing http port  from\_port = 80  to\_port = 80  protocol = "tcp"  cidr\_blocks = ["0.0.0.0/0"] #from anywhere  }  egress{  from\_port = 0  to\_port = 0  protocol = "-1" # any port  cidr\_blocks = ["0.0.0.0/0"] # anywhere  }  } |

* NOTE:
  1. If we are making any changes to the resources, then it will destroy the existing one and create a new one
  2. If we are trying to make any changes to the items in the resource, then it won’t destroy the resources
  3. Example: if we are trying to change the security group which is a resource in the aws\_instance resource, then it will destroy the instance and create a new one.
  4. Same works for the key value pair
  5. If we try to make any changes to the user\_data or instance type, it won’t destroy the instance.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.  
  
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

