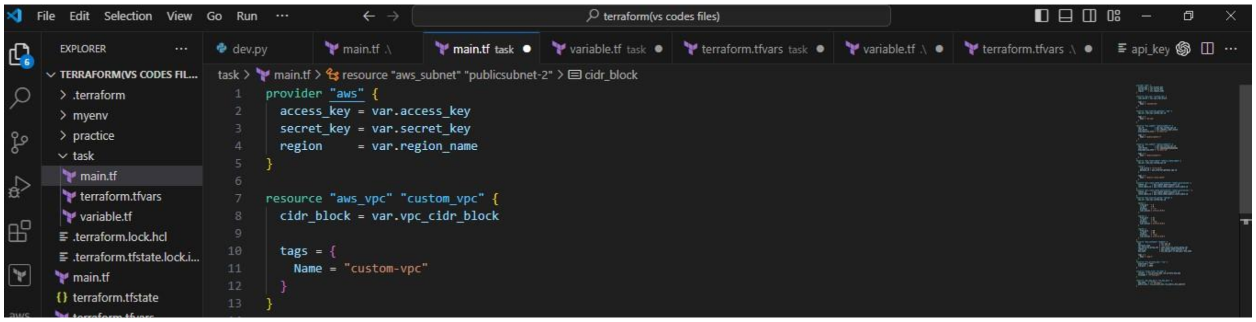
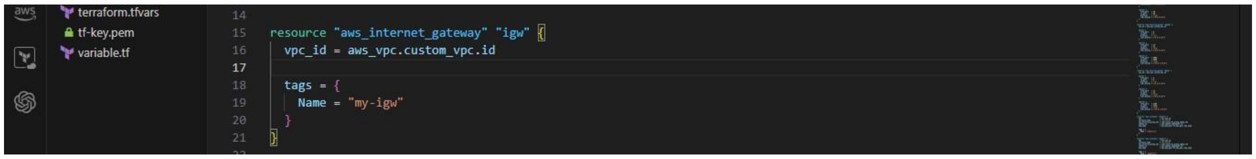
Name = Devendra Sanjay Sutar
Batch = 15 jan **Terraform = task 2**

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


File Edit Selection View Go Run
terraform(vs codes files)
EXPLORER
TERRAFORM(VS CODES FIL...
.terraform
myenv
practice
task
main.tf
terraform.tfvars
variable.tf
.terraform.lock.hcl
.terraform.tfstate.lock.i...
main.tf
terraform.tfstate
dev.py
task > main.tf > resource "aws_subnet" "publicsubnet-2" > cidr_block

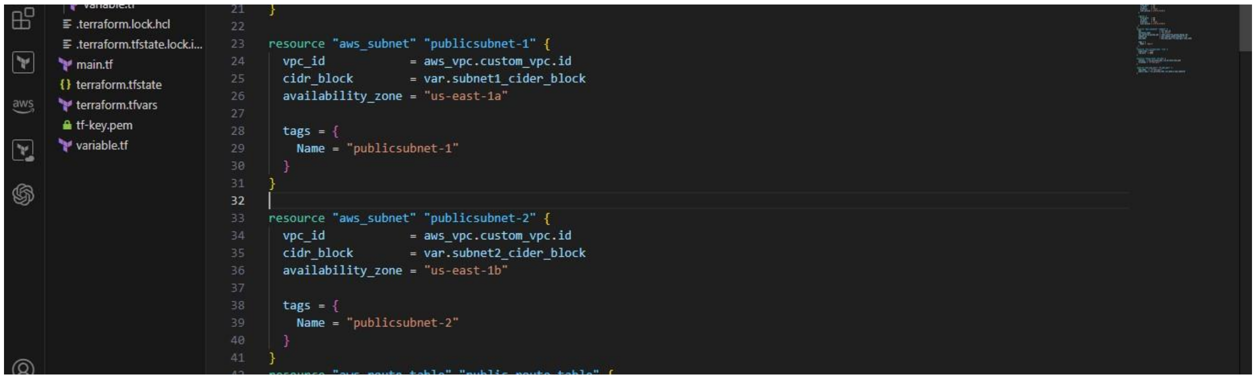
```
provider "aws" {
  access_key = var.access_key
  secret_key = var.secret_key
  region     = var.region_name
}

resource "aws_vpc" "custom_vpc" {
  cidr_block = var.vpc_cidr_block

  tags = {
    Name = "custom-vpc"
  }
}
```

- VPC code using terraform.



```

resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.custom_vpc.id

  tags = {
    Name = "my-igw"
  }
}
```

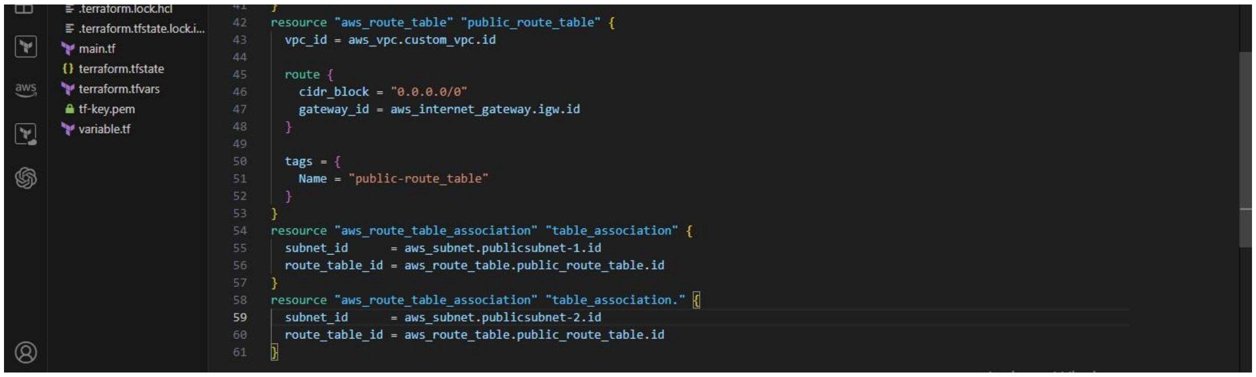
- Created an internet gateway using terraform.



```
}

resource "aws_subnet" "publicsubnet-1" {
  vpc_id            = aws_vpc.custom_vpc.id
  cidr_block        = var.subnet1_cider_block
  availability_zone = "us-east-1a"

  tags = {
    Name = "publicsubnet-1"
  }
}

resource "aws_subnet" "publicsubnet-2" {
  vpc_id            = aws_vpc.custom_vpc.id
  cidr_block        = var.subnet2_cider_block
  availability_zone = "us-east-1b"

  tags = {
    Name = "publicsubnet-2"
  }
}
```

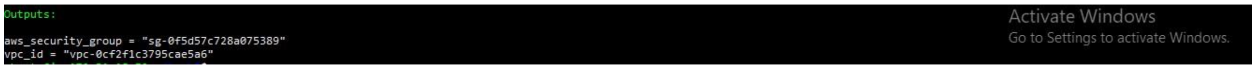
- Created a two public subnet in different zones and CIDR stores in variable

Name = Devendra Sanjay Sutar
Batch = 15 jan **Terraform = task 2**



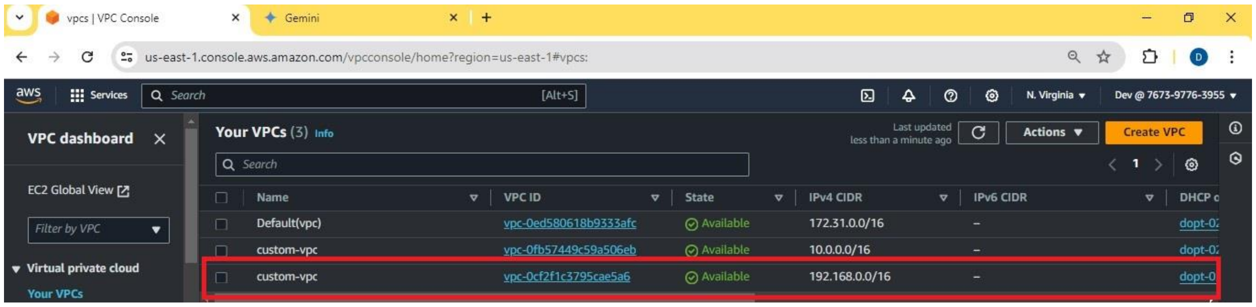
```
resource "aws_route_table" "public_route_table" {
  vpc_id = aws_vpc.custom_vpc.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.igw.id
  }

  tags = {
    Name = "public-route_table"
  }
}
resource "aws_route_table_association" "table_association" {
  subnet_id      = aws_subnet.publicsubnet-1.id
  route_table_id = aws_route_table.public_route_table.id
}
resource "aws_route_table_association" "table_association." {
  subnet_id      = aws_subnet.publicsubnet-2.id
  route_table_id = aws_route_table.public_route_table.id
}
```

- Created route table and associated with subnets.



```
Outputs:

aws_security_group = "sg-0f5d57c728a075389"
vpc_id = "vpc-0cf2f1c3795cae5a6"
```

- Successfully vpc-id and sg-id showing.

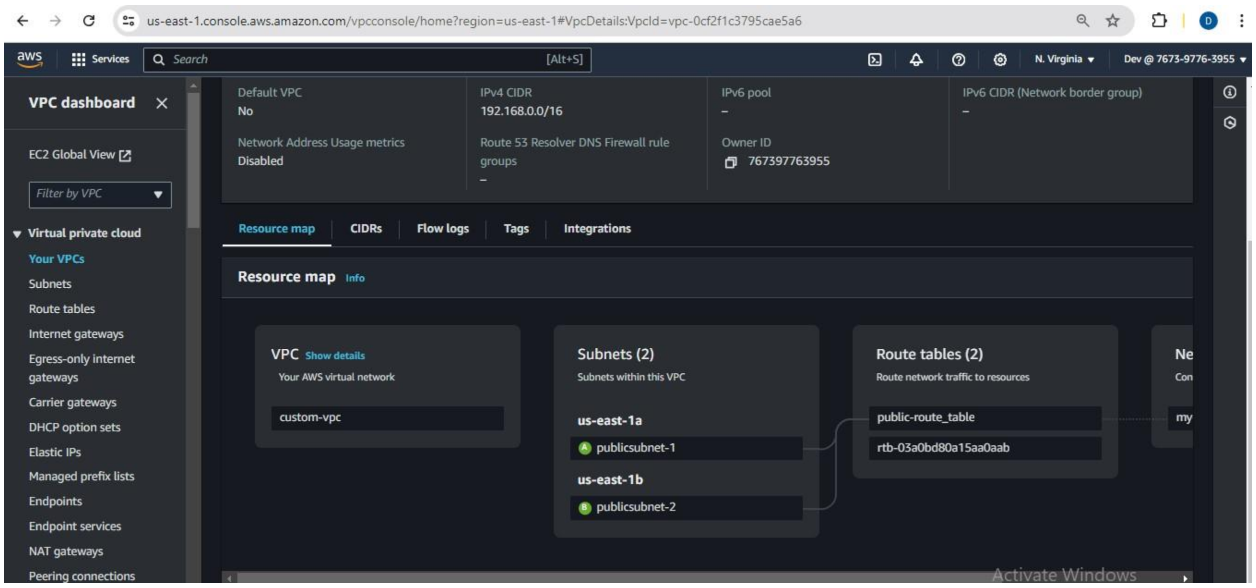


Your VPCs (3) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP o
Default(vpc)	vpc-0ed580618b9333afc	Available	172.31.0.0/16	–	dopt-0
custom-vpc	vpc-0fb57449c59a506eb	Available	10.0.0.0/16	–	dopt-0
custom-vpc	vpc-0cf2f1c3795cae5a6	Available	192.168.0.0/16	–	dopt-0

- VPC created successfully .

Name = Devendra Sanjay Sutar
Batch = 15 jan **Terraform = task 2**


us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#VpcDetails:VpcId=vpc-0cf2f1c3795cae5a6

Default VPC
No

IPv4 CIDR
192.168.0.0/16

IPv6 pool
–

IPv6 CIDR (Network border group)
–

Network Address Usage metrics
Disabled

Route 53 Resolver DNS Firewall rule groups
–

Owner ID
767397763955

Resource map

VPC — Your AWS virtual network
custom-vpc

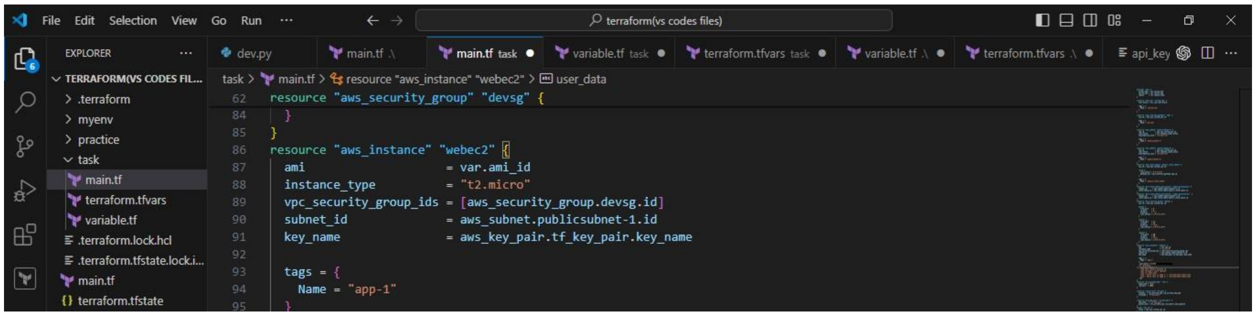
Subnets (2) — Subnets within this VPC
us-east-1a
publicsubnet-1
us-east-1b
publicsubnet-2

Route tables (2) — Route network traffic to resources
public-route_table
rtb-03a0bd80a15aa0aab

- Two public subnet created successfully in two different zones and attached with public route table.

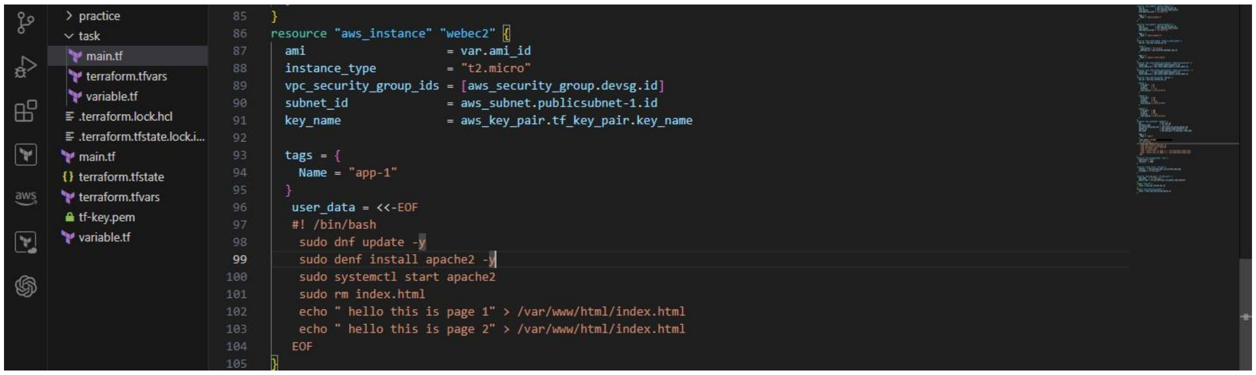
Name = Devendra Sanjay Sutar
Batch = 15 jan **Terraform = task 2**

2 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.



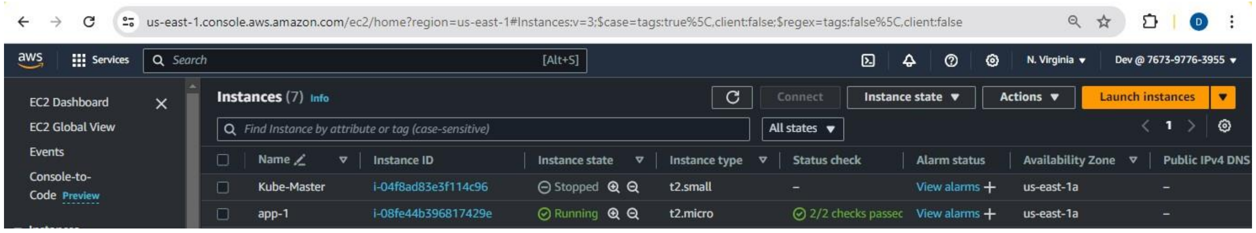
```
resource "aws_security_group" "devsg" {
  }
}
resource "aws_instance" "webec2" {
  ami                    = var.ami_id
  instance_type          = "t2.micro"
  vpc_security_group_ids = [aws_security_group.devsg.id]
  subnet_id              = aws_subnet.publicsubnet-1.id
  key_name               = aws_key_pair.tf_key_pair.key_name

  tags = {
    Name = "app-1"
```

- created a code for instance with name “app-1”.



```
}
resource "aws_instance" "webec2" {
  ami                    = var.ami_id
  instance_type          = "t2.micro"
  vpc_security_group_ids = [aws_security_group.devsg.id]
  subnet_id              = aws_subnet.publicsubnet-1.id
  key_name               = aws_key_pair.tf_key_pair.key_name

  tags = {
    Name = "app-1"
  }
   user_data = <<-EOF
   #! /bin/bash
    sudo dnf update -y
    sudo denf install apache2 -y
    sudo systemctl start apache2
    sudo rm index.html
    echo " hello this is page 1" > /var/www/html/index.html
    echo " hello this is page 2" > /var/www/html/index.html
   EOF
}
```

- in provisioner block created code to install apache and created s two pages .



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false

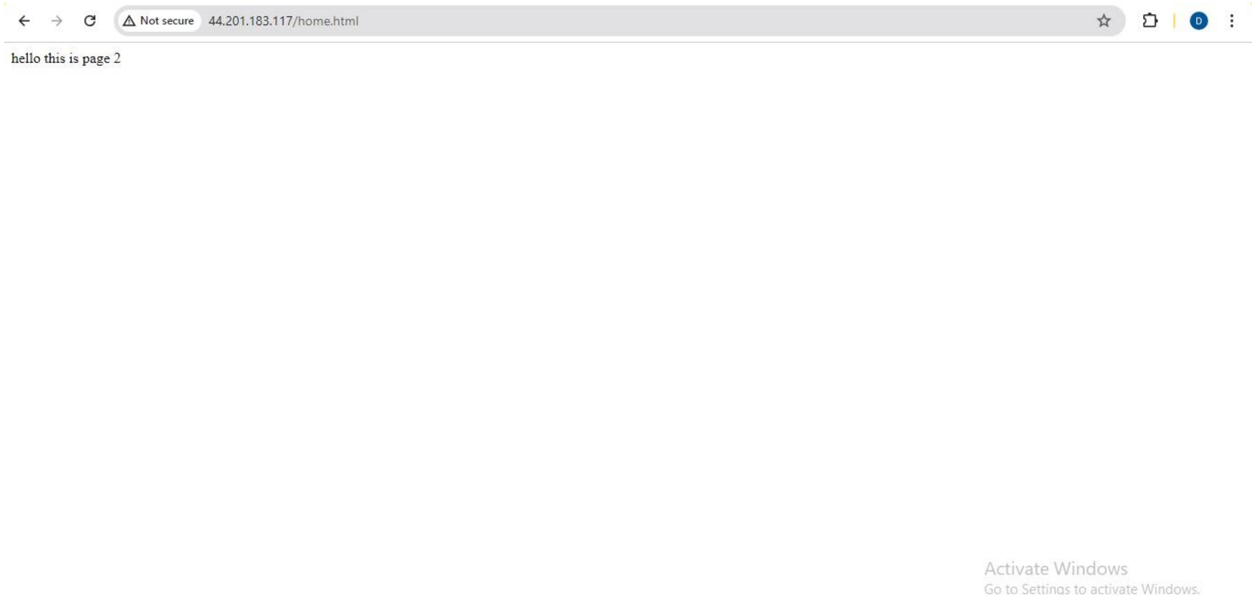
Instances (7)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
	Kube-Master	i-04f8ad83e3f114c96	Stopped	t2.small	–	View alarms	us-east-1a	–
	app-1	i-08fe44b396817429e	Running	t2.micro	2/2 checks passec	View alarms	us-east-1a	–

- Successfully launched ec2 “app-1”.

Name = Devendra Sanjay Sutar

Batch = 15 jan **Terraform = task 2**

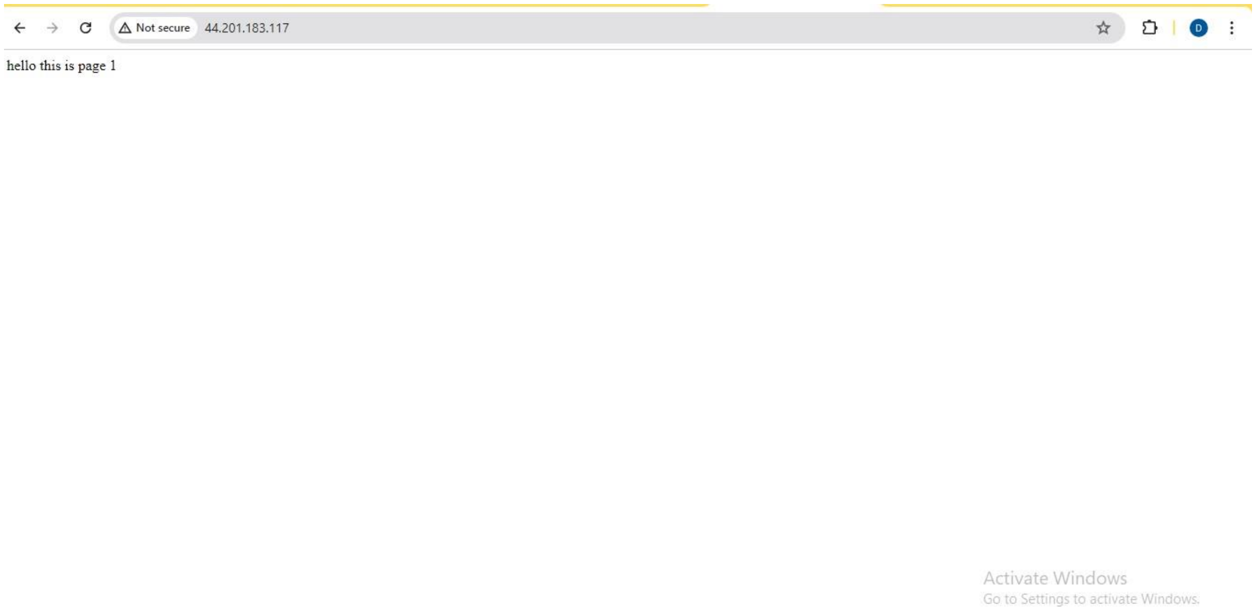


Not secure 44.201.183.117/home.html

hello this is page 2

Activate Windows
Go to Settings to activate Windows.

- page first Successfully visible.



Not secure 44.201.183.117

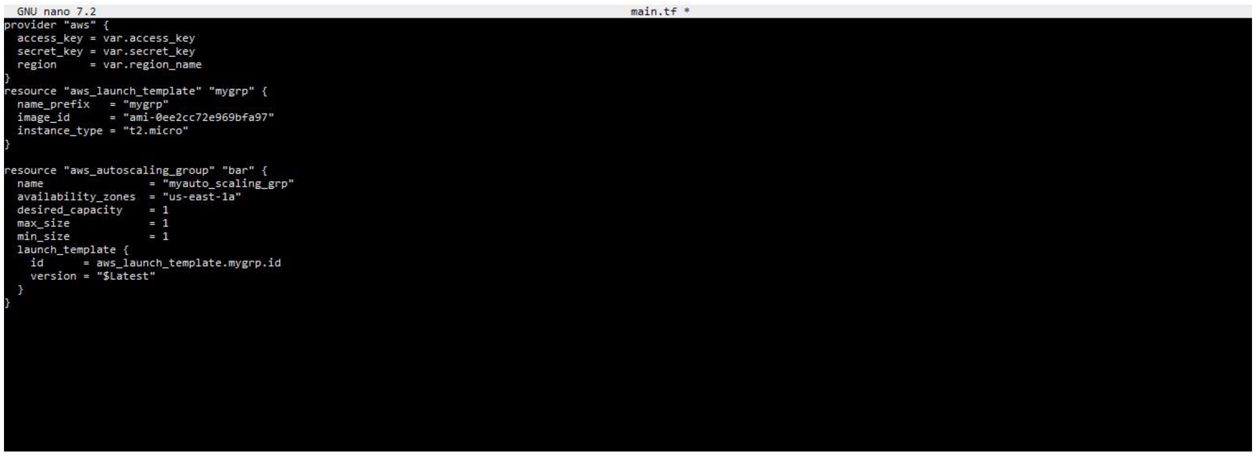
hello this is page 1

Activate Windows
Go to Settings to activate Windows.

- page second also visible Successfully .

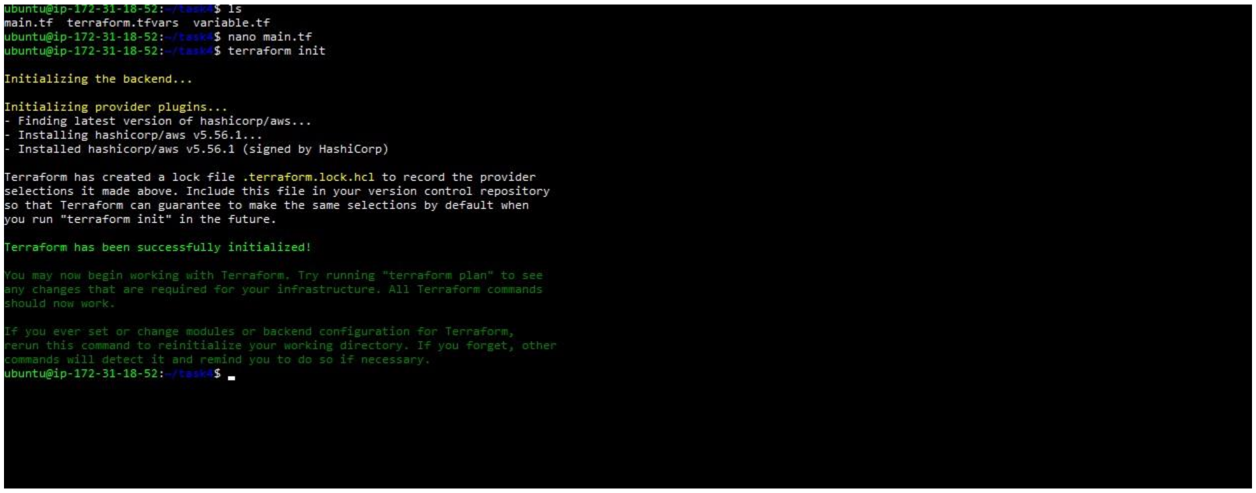
Name = Devendra Sanjay Sutar
Batch = 15 jan **Terraform = task 2**

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



```
  GNU nano 7.2                                                     main.tf *
provider "aws" {
  access_key = var.access_key
  secret_key = var.secret_key
  region     = var.region_name
}
resource "aws_launch_template" "mygrp" {
  name_prefix   = "mygrp"
  image_id      = "ami-0ee2cc72e969bfa97"
  instance_type = "t2.micro"
}

resource "aws_autoscaling_group" "bar" {
  name                  = "myauto_scaling_grp"
  availability_zones    = "us-east-1a"
  desired_capacity      = 1
  max_size              = 1
  min_size              = 1
  launch_template {
    id      = aws_launch_template.mygrp.id
    version = "$Latest"
  }
}
```

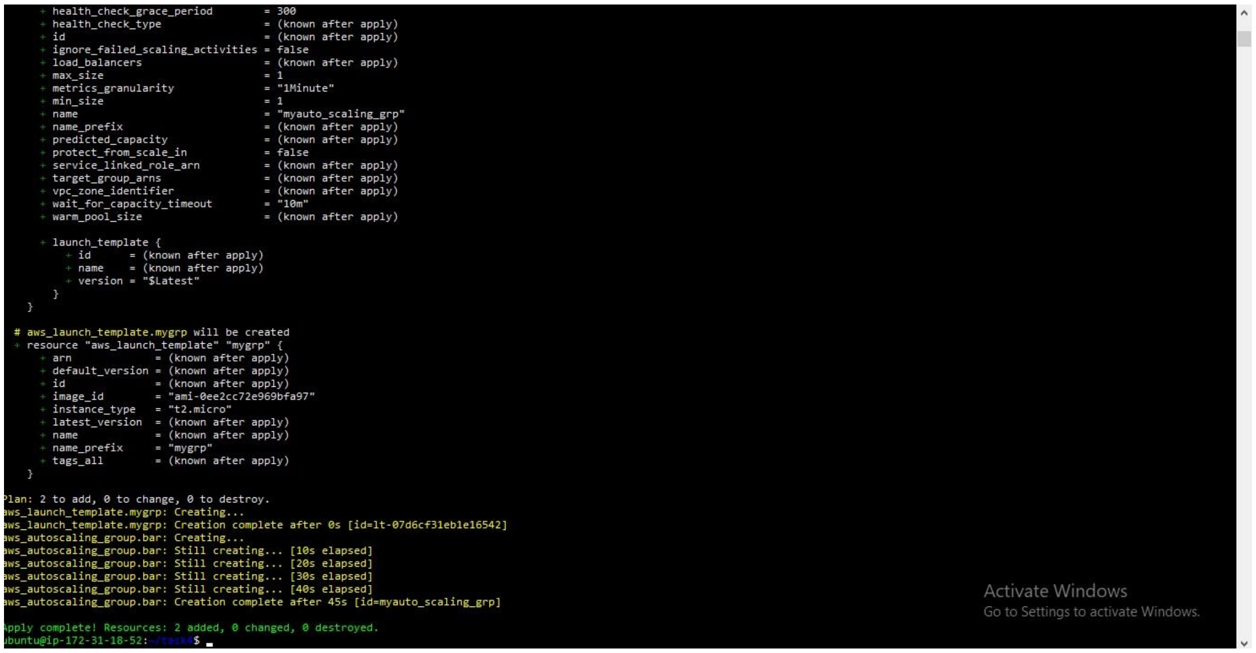
- Code for autoscaling group and launch template.



```
ubuntu@ip-172-31-18-52:~/task4$ ls
main.tf  terraform.tfvars  variable.tf
ubuntu@ip-172-31-18-52:~/task4$ nano main.tf
ubuntu@ip-172-31-18-52:~/task4$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.56.1...
- Installed hashicorp/aws v5.56.1 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-18-52:~/task4$ 
```

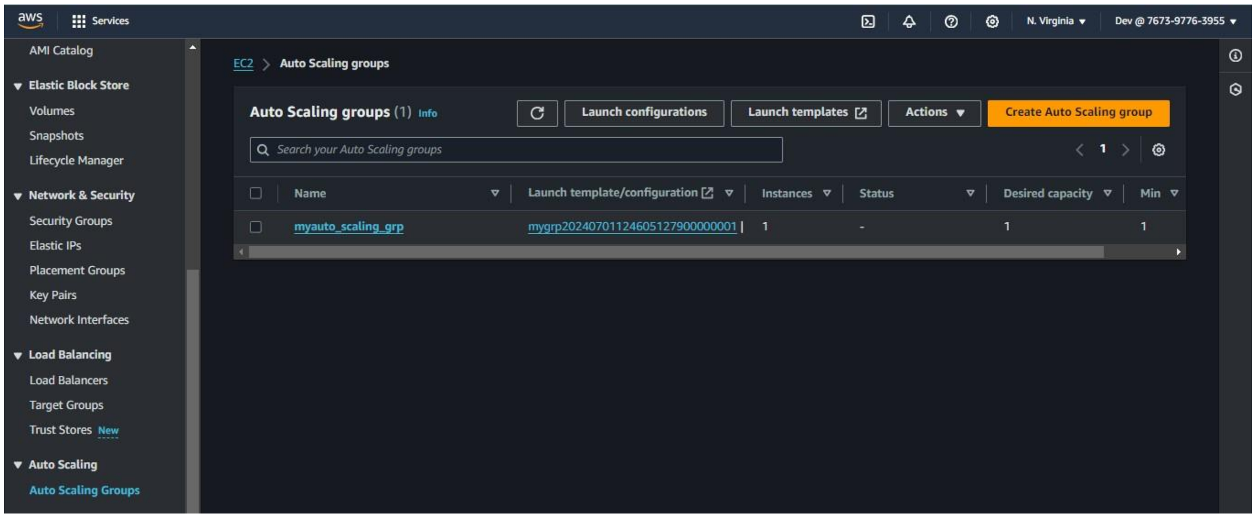
- Succesfully initialized.

Name = Devendra Sanjay Sutar
Batch = 15 jan **Terraform = task 2**



```
      + health_check_grace_period        = 300
      + health_check_type                = (known after apply)
      + id                               = (known after apply)
      + ignore_failed_scaling_activities = false
      + load_balancers                   = (known after apply)
      + max_size                         = 1
      + metrics_granularity              = "1Minute"
      + min_size                         = 1
      + name                             = "myauto_scaling_grp"
      + name_prefix                      = (known after apply)
      + predicted_capacity               = (known after apply)
      + protect_from_scale_in            = false
      + service_linked_role_arn          = (known after apply)
      + target_group_arns                = (known after apply)
      + vpc_zone_identifier              = (known after apply)
      + wait_for_capacity_timeout        = "10m"
      + warm_pool_size                   = (known after apply)

      + launch_template {
          + id      = (known after apply)
          + name    = (known after apply)
          + version = "$Latest"
        }
    }

  # aws_launch_template.mygrp will be created
  + resource "aws_launch_template" "mygrp" {
      + arn             = (known after apply)
      + default_version = (known after apply)
      + id              = (known after apply)
      + image_id        = "ami-0ee2cc72e969bfa97"
      + instance_type   = "t2.micro"
      + latest_version  = (known after apply)
      + name            = (known after apply)
      + name_prefix     = "mygrp"
      + tags_all        = (known after apply)
    }

Plan: 2 to add, 0 to change, 0 to destroy.
aws_launch_template.mygrp: Creating...
aws_launch_template.mygrp: Creation complete after 0s [id=lt-07d6cf31eb1e16542]
aws_autoscaling_group.bar: Creating...
aws_autoscaling_group.bar: Still creating... [10s elapsed]
aws_autoscaling_group.bar: Still creating... [20s elapsed]
aws_autoscaling_group.bar: Still creating... [30s elapsed]
aws_autoscaling_group.bar: Still creating... [40s elapsed]
aws_autoscaling_group.bar: Creation complete after 45s [id=myauto_scaling_grp]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-18-52:~/task4$ 
```

- Successfully applied.



Auto Scaling groups (1) Info

	Name	Launch template/configuration	Instances	Status	Desired capacity	Min
☐	myauto_scaling_grp	mygrp20240701124605127900000001	1	-	1	1

- **Successfully created .**