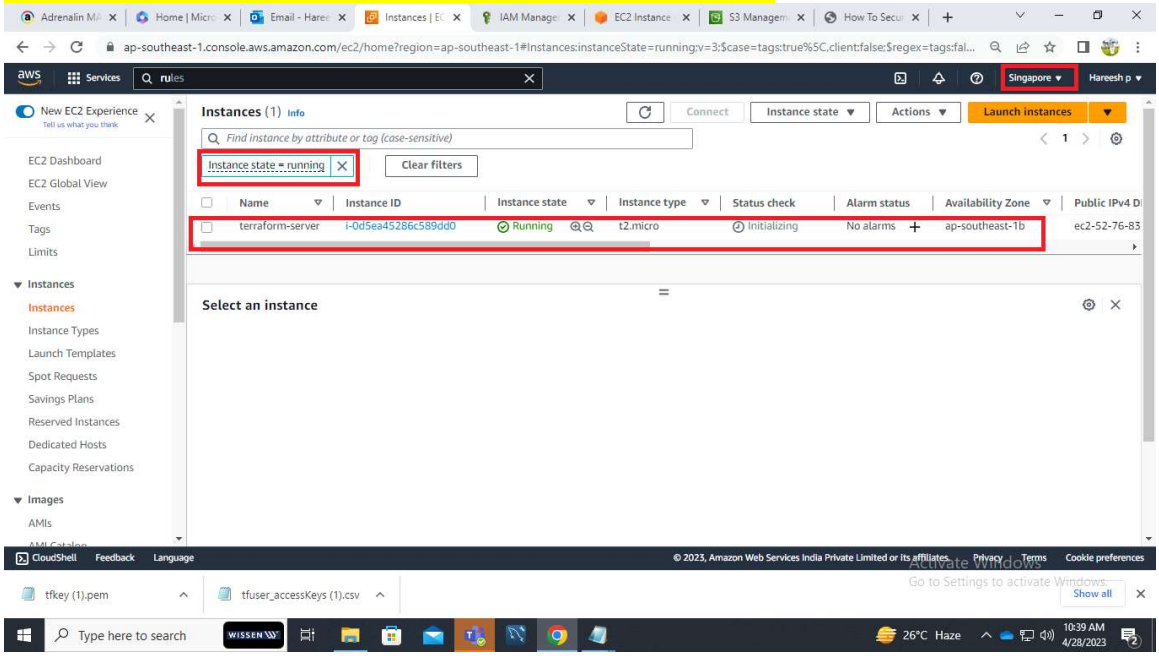
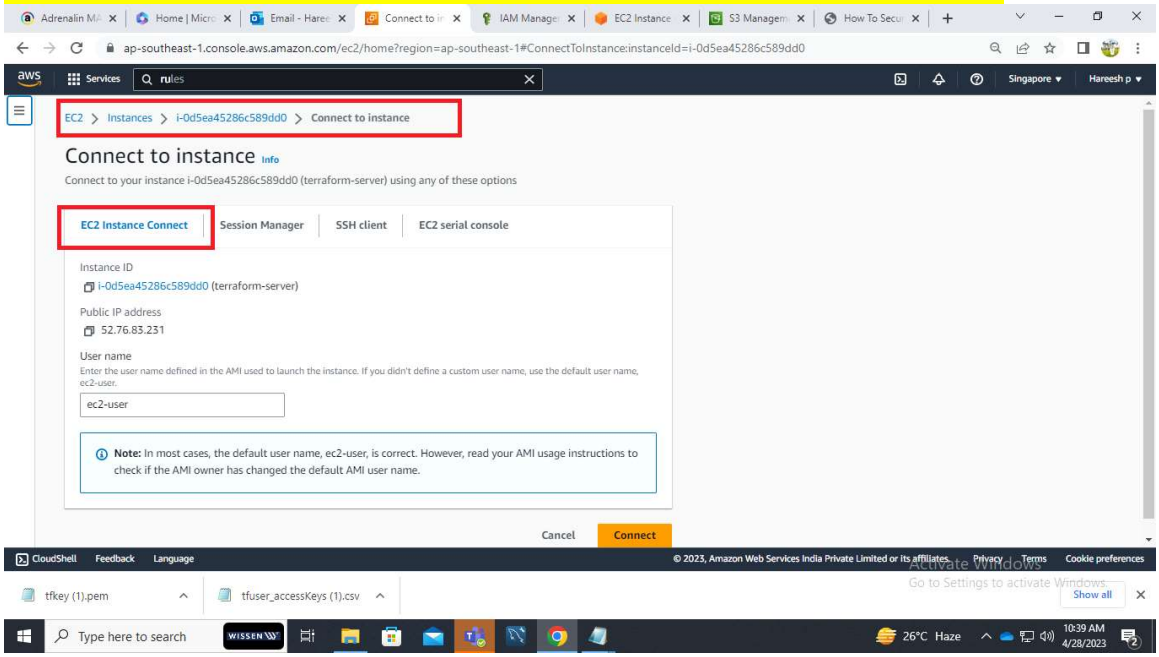
Ec2 instance creation using terraform

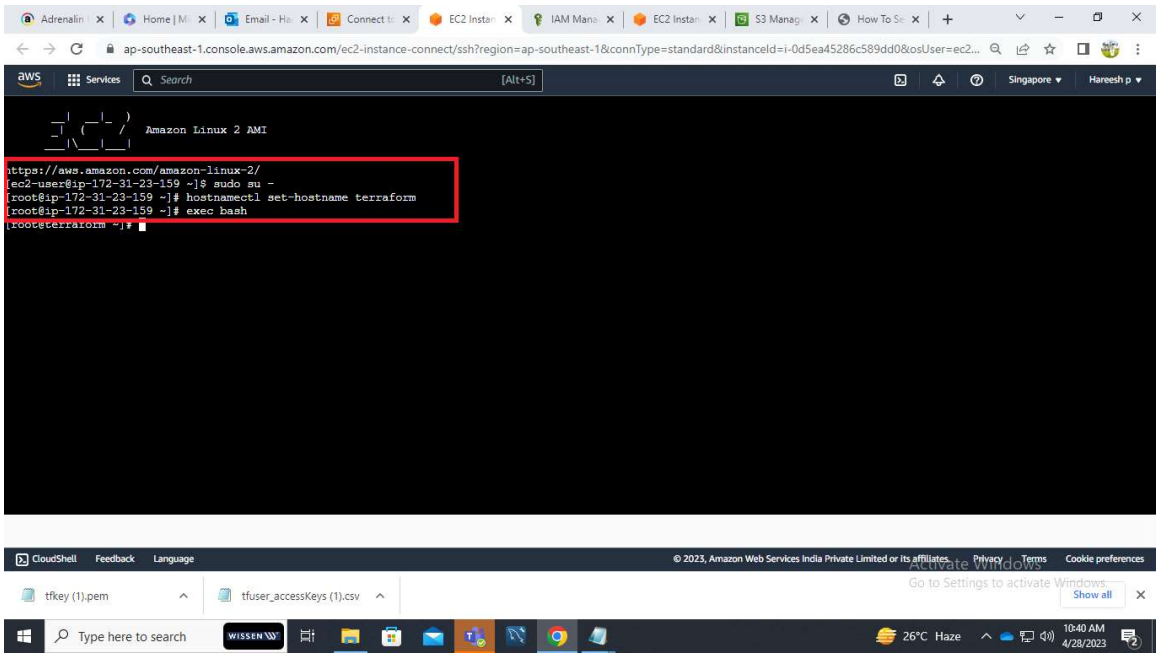
step-1: Login to aws management console create a ec2 instance



step-2: connect the terminal via ec2 connect (browser based) to run the commands



step-3: Login as a root user and create a hostname



- install the terraform in amazon linux machine follow the link and run those commands:

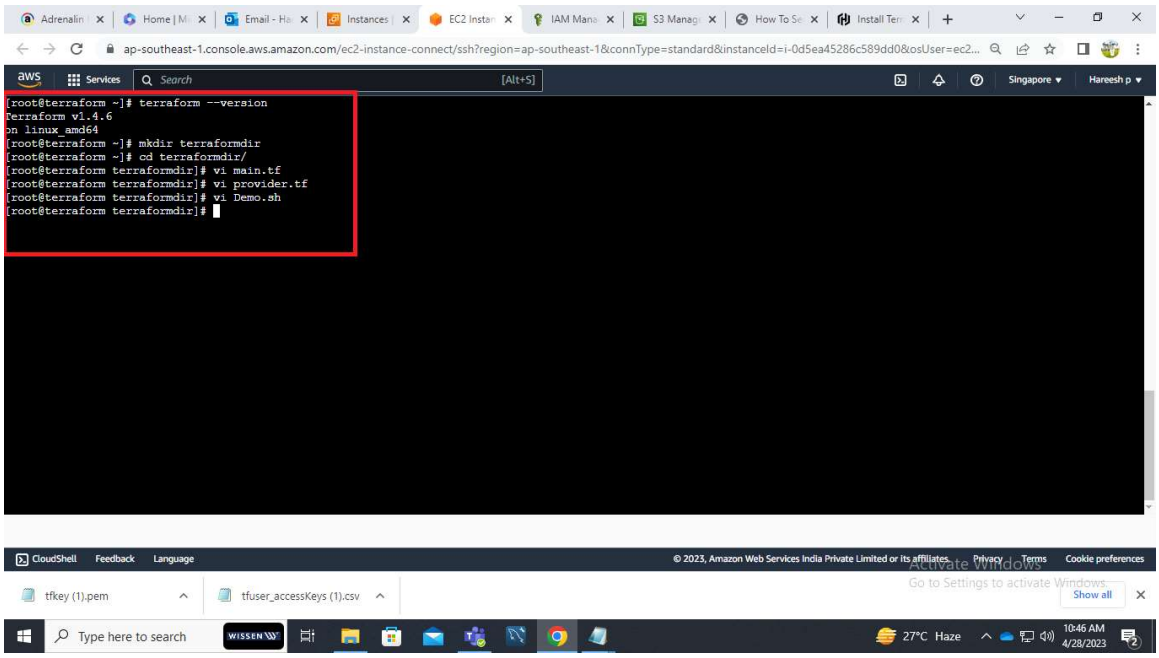
1. sudo yum install -y yum-utils

2.sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo

3.sudo yum -y install terraform

4.terraform --version

step-4: create a directory and change in to that directory



step-5: create a main.tf and provider.tf and Demo.sh files to run the below scripts

main.tf

```
resource "aws_instance" "myinstance" {

  ami = "ami-0f6ad051716c81af1"

  instance_type = "t2.micro"

  associate_public_ip_address = "true"

  availability_zone= ""

  key_name = "terraformkey"

  vpc_security_group_ids =["sg-01fb9122b6f7153ac"]

  subnet_id= "subnet-037a5d111edf24977"

  tags = {

    Name = "demo-instance "

  }

  count = "1"
```

```
  user_data = "${file("Demo.sh")}"
  # root disk
  root_block_device {
    volume_size          = "8"
    volume_type          = "gp2"
    encrypted            = "false"
    delete_on_termination = "false"
  }
  # data disk
  ebs_block_device {
    device_name          = "/dev/sdh"
    volume_size          = "8"
    volume_type          = "gp2"
    encrypted            = "false"
    delete_on_termination = "false"
  }
}
variable "aws_access_key" {}
variable "aws_secret_key" {}
```

provider.tf

```
provider "aws" {
  access_key = var.aws_access_key
  secret_key = var.aws_secret_key
  region     = "ap-southeast-2"
```

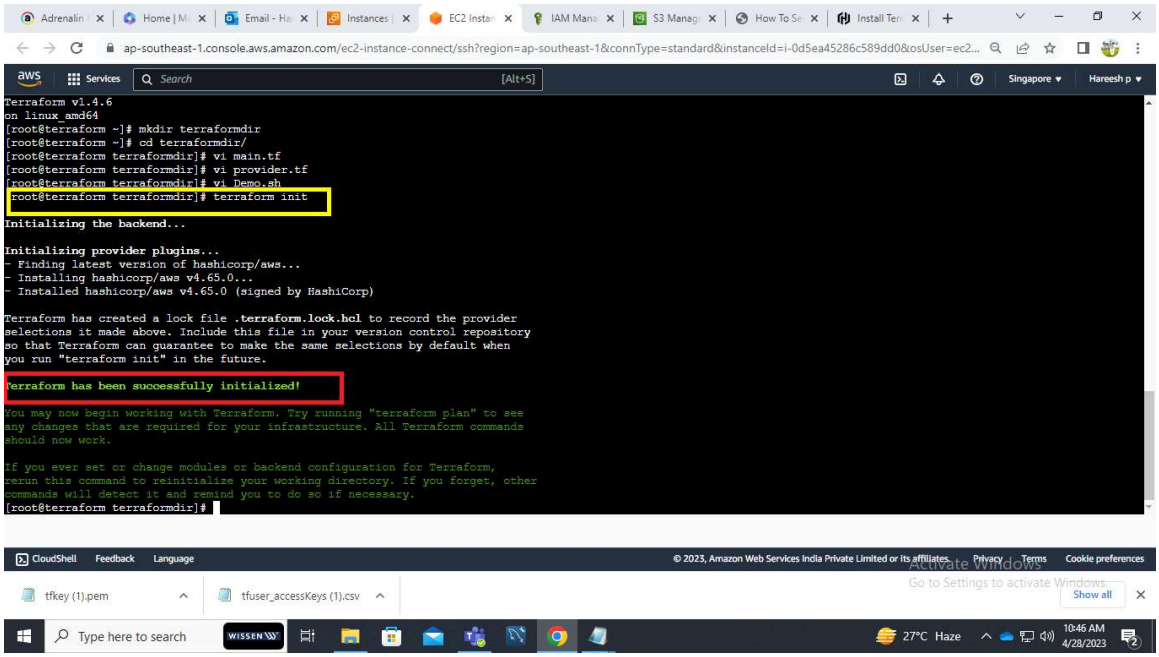
}

Demo.sh

sudo yum update -y

step-6: Apply the below commands to run the terraform

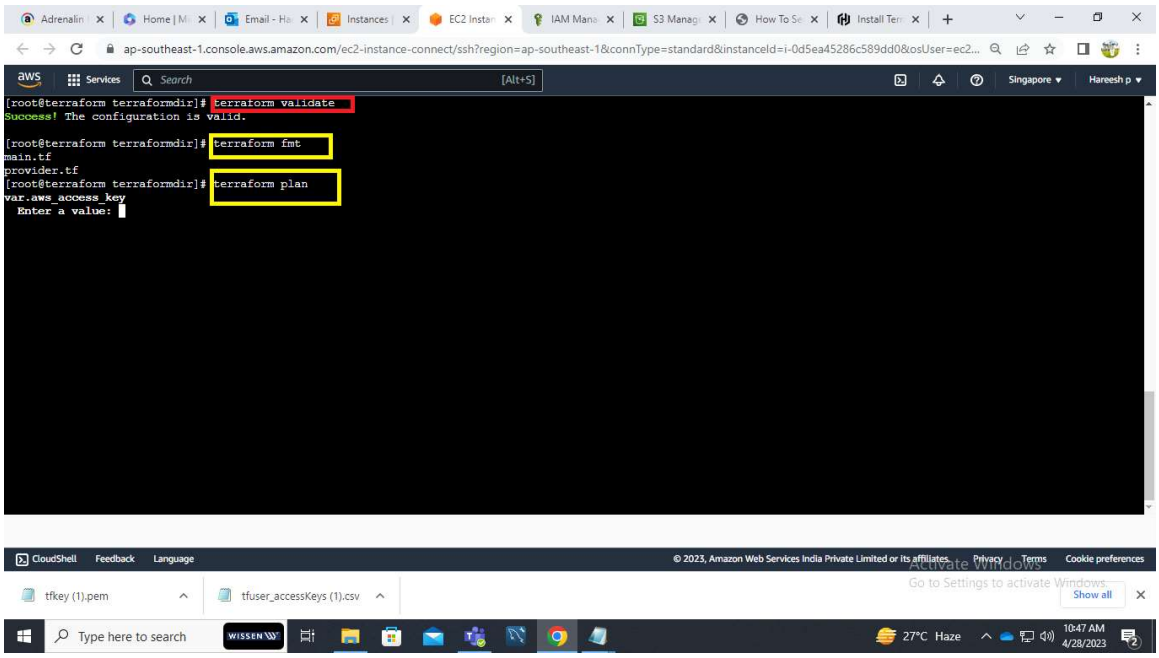
1. terraform init - Prepare your working directory for other commands



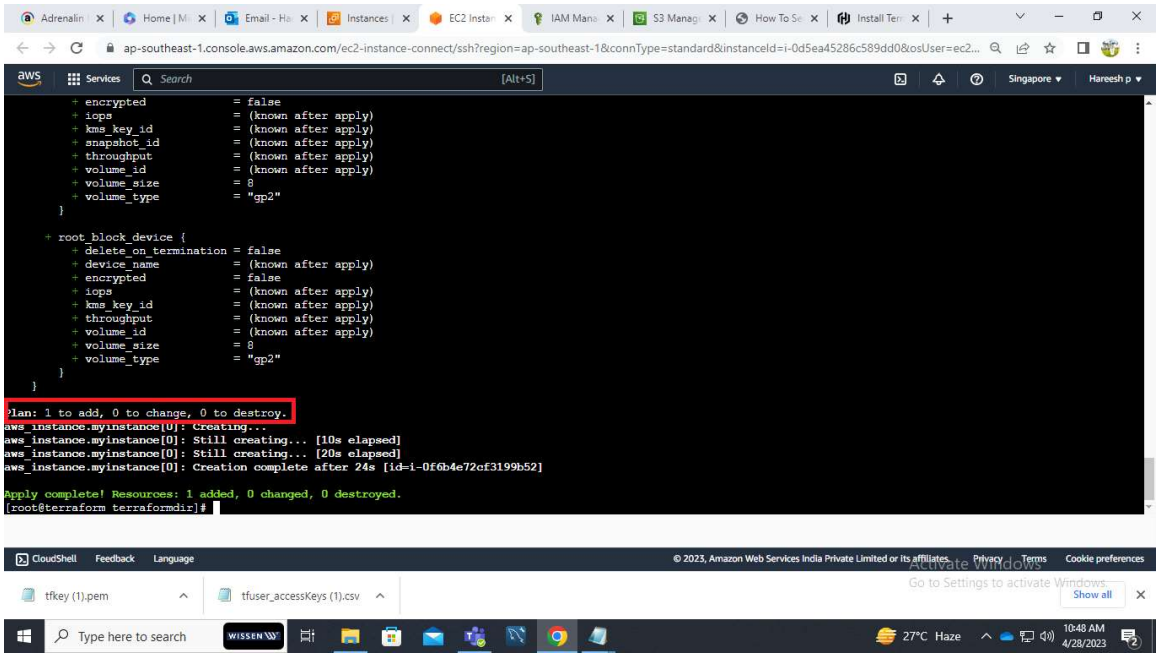
2. terraform validate - it will Check whether the configuration is valid

3. terraform fmt - it will reformat your configuration in the standard style

4. terrafprm plan -it will Show changes required by the current configuration

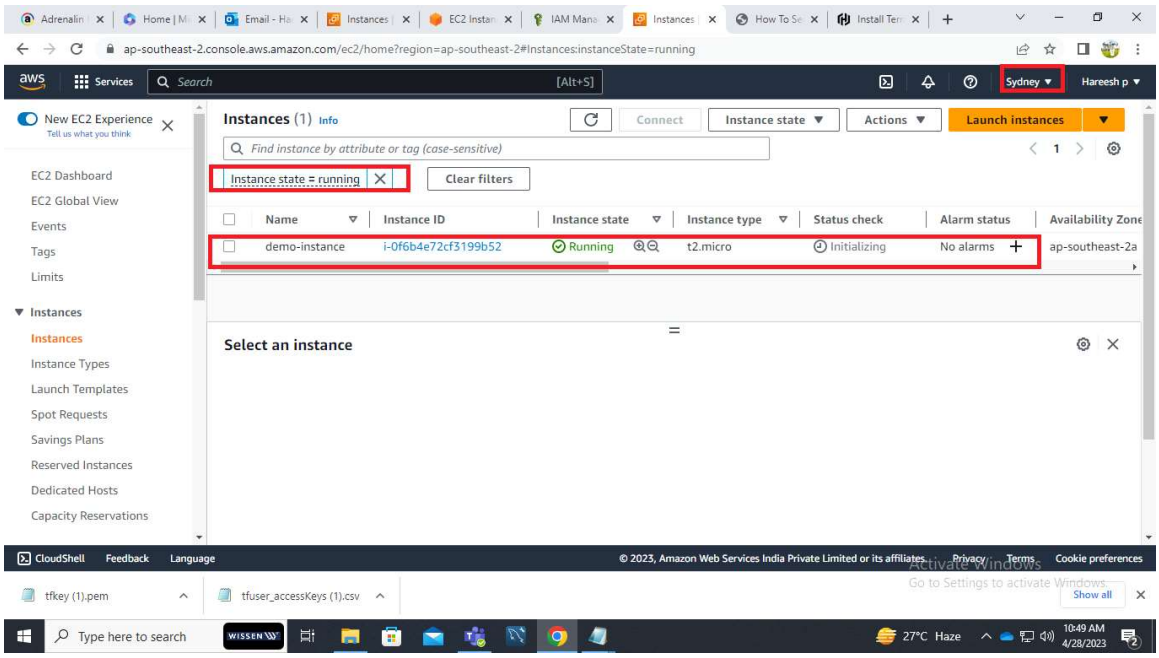


5. terraform apply -auto-approve - it will Create or update infrastructure

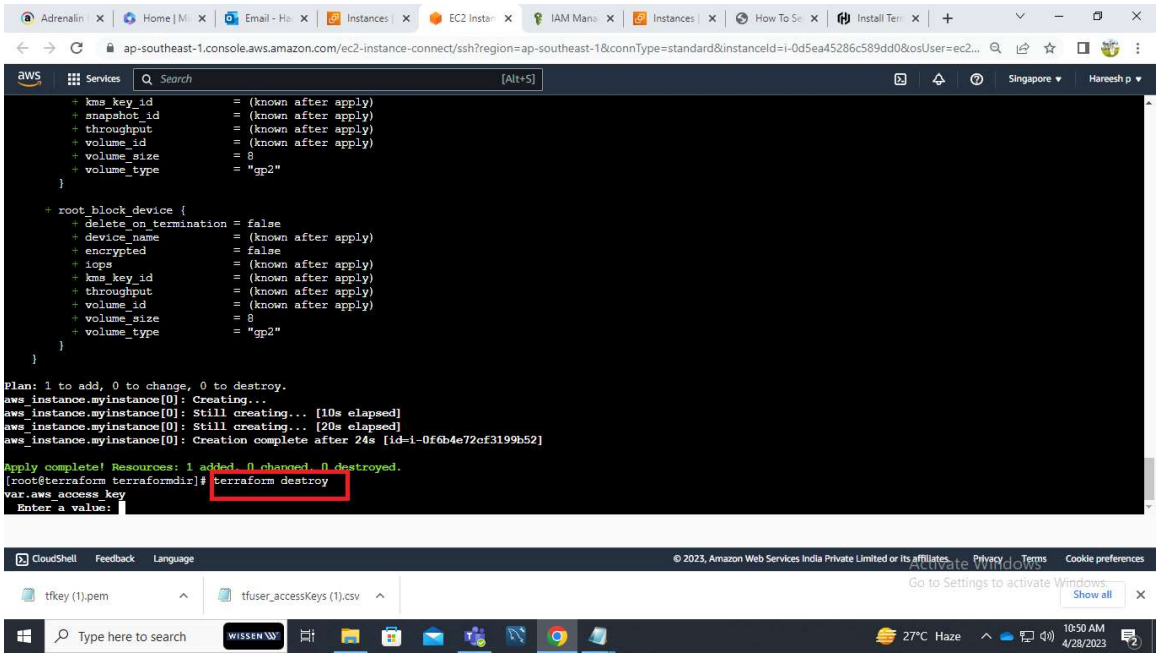


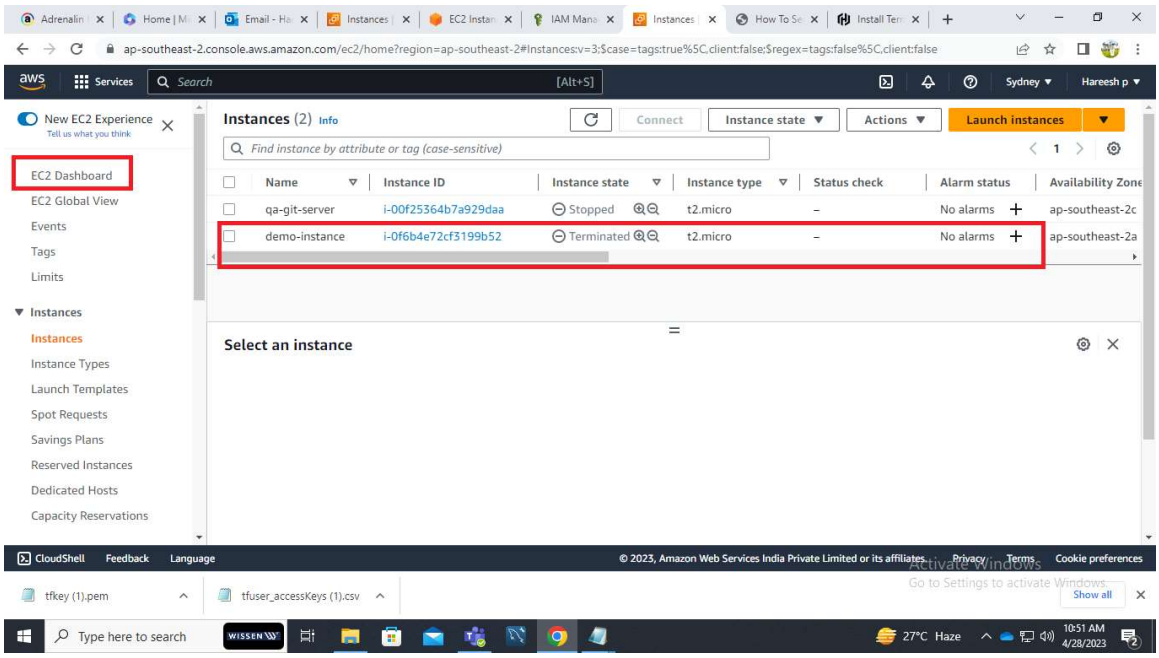
Navigate to ec2 dash board and check the resource created or Not

Here the Ec2 instance lauched successfully



- Apply terraform destroy - Destroy previously-created infrastructure




ap-southeast-2.console.aws.amazon.com/ec2/home?region=ap-southeast-2#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false
aws
Services
Search
[Alt+S]
Sydney
Hareesh p
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Instances (2) Info
Connect
Instance state
Actions
Launch instances
Find instance by attribute or tag (case-sensitive)
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
qa-git-server
i-00f25364b7a929daa
Stopped
t2.micro
–
No alarms
ap-southeast-2c
demo-instance
i-0f6b4e72cf3199b52
Terminated
t2.micro
–
No alarms
ap-southeast-2a
Select an instance
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
tfkey (1).pem
tfuser_accessKeys (1).csv
Show all
Type here to search
27°C Haze
10:51 AM
4/28/2023