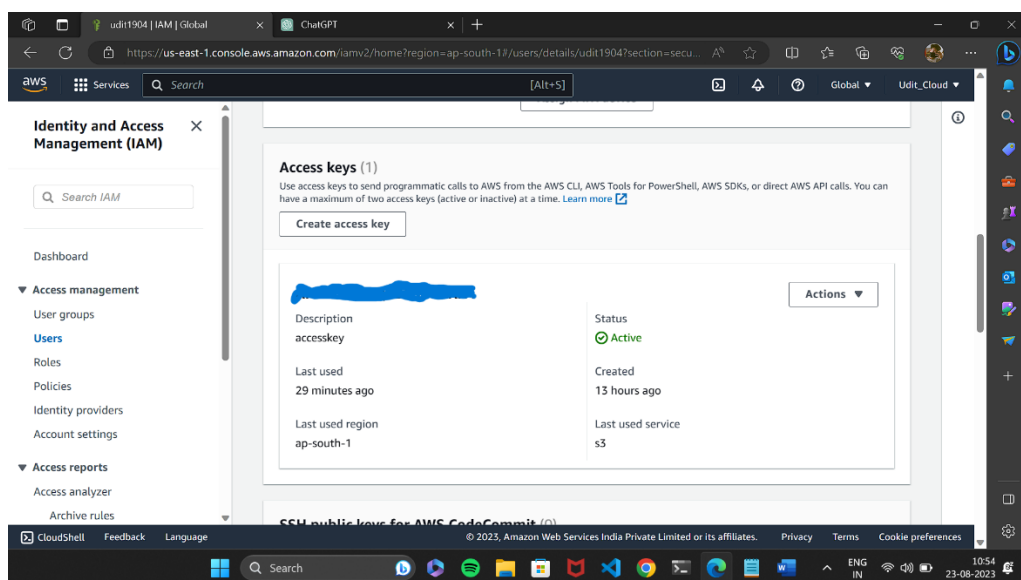
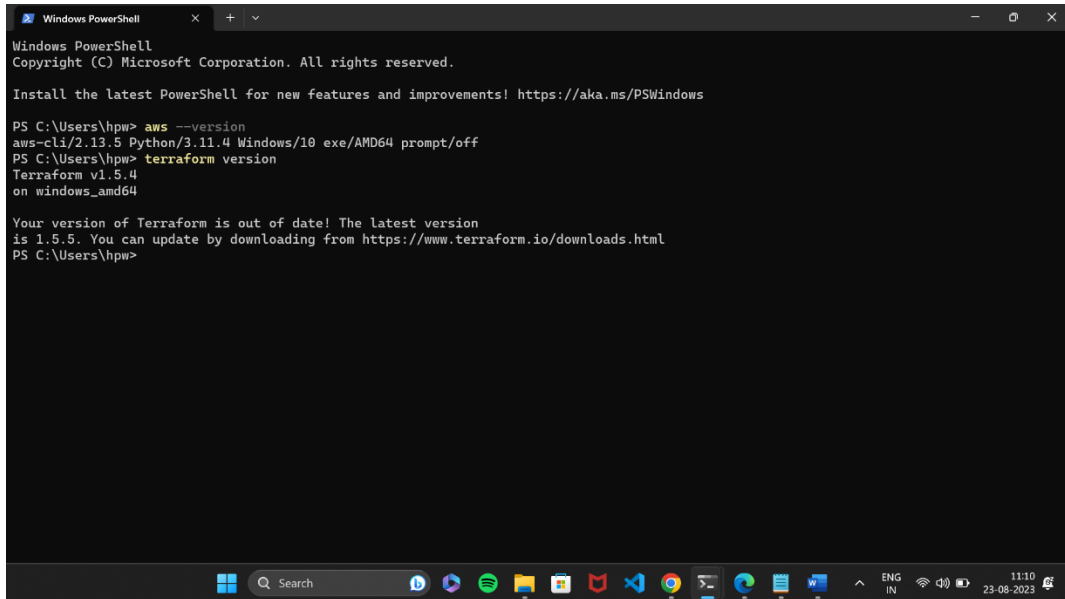


# Infrastructure as Code (IaaC)

1. Install [Terraform](#)
2. Install [AWS CLI](#)
3. Create IAM Role From Root Account and Create Access Key and Secret Key (Console > IAM > Users > (Select Created User) > Security Credentials > Access Keys) Save the Keys.



4. Goto Command Prompt/ PowerShell / Terminal and make sure aws cli and terraform is installed in your machine. You can do it by giving command like - **“aws --version”** and **“terraform version”**



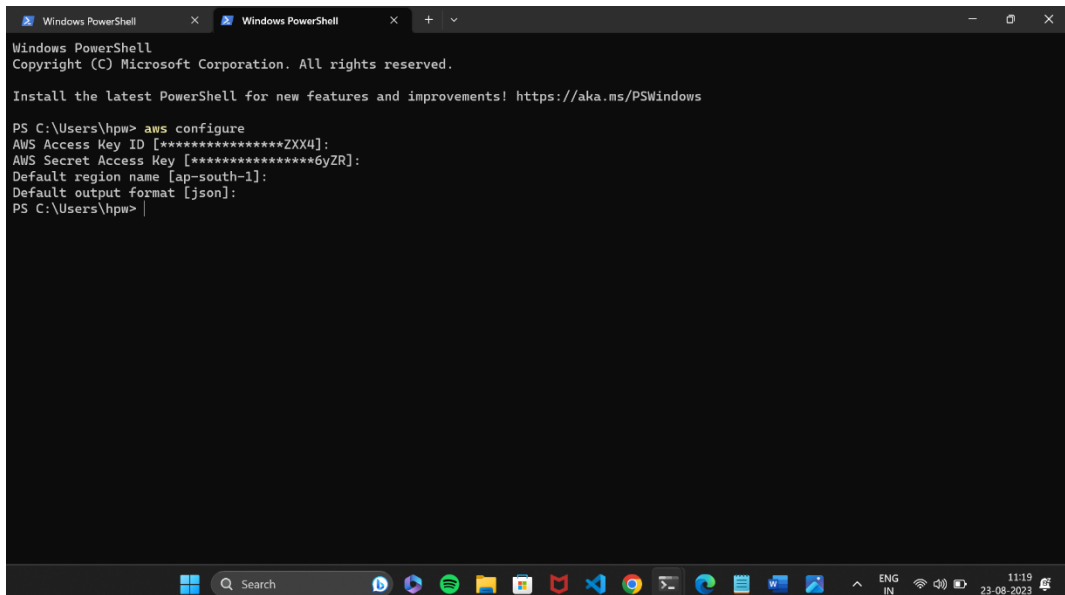
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\hwp> aws --version
aws-cli/2.13.5 Python/3.11.4 Windows/10 exe/AMD64 prompt/off
PS C:\Users\hwp> terraform version
Terraform v1.5.4
on windows_amd64

Your version of Terraform is out of date! The latest version
is 1.5.5. You can update by downloading from https://www.terraform.io/downloads.html
PS C:\Users\hwp>
```

5. Now Configure aws credentials, give command “**aws configure**” and give your access key and secret key.



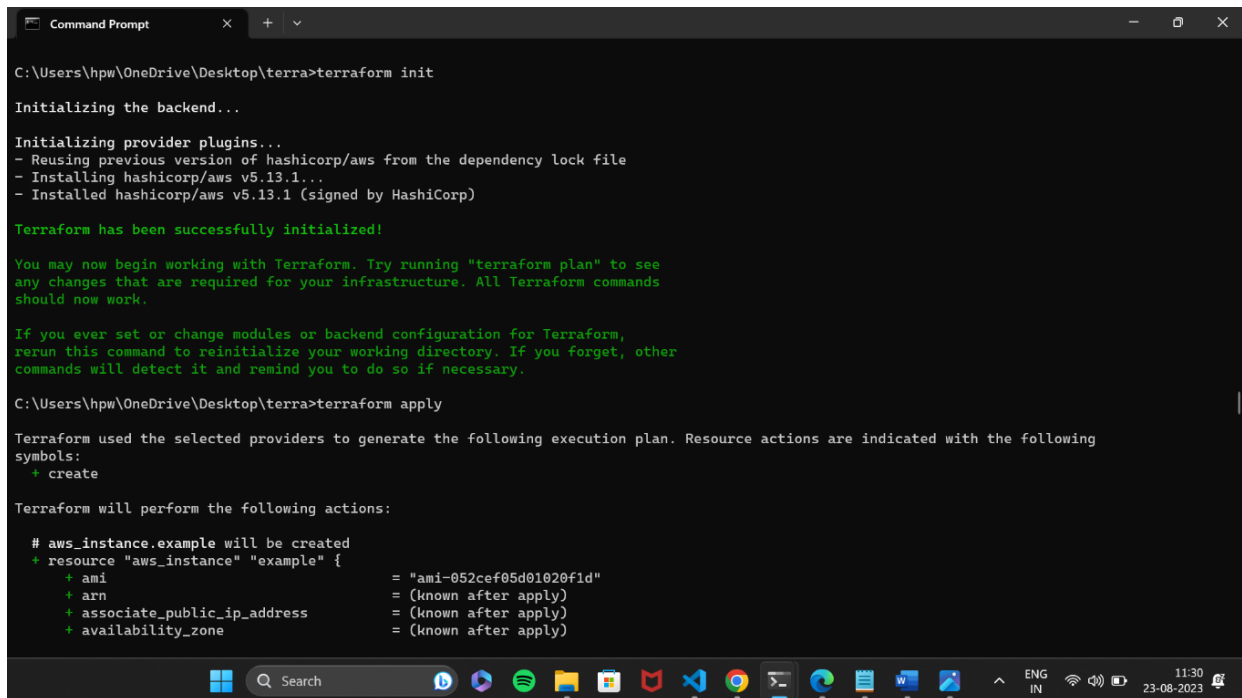
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\hwp> aws configure
AWS Access Key ID [*****ZXX4]:
AWS Secret Access Key [*****6yZR]:
Default region name [ap-south-1]:
Default output format [json]:
PS C:\Users\hwp>
```

6. Now Create Terraform configuration File and name it according to your preference common name can be [main.tf](#)

7. Now Set PATH where you have saved terraform config file and run command “**terraform init**”.



```
C:\Users\hpw\OneDrive\Desktop\terra>terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Installing hashicorp/aws v5.13.1...
- Installed hashicorp/aws v5.13.1 (signed by HashiCorp)

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.

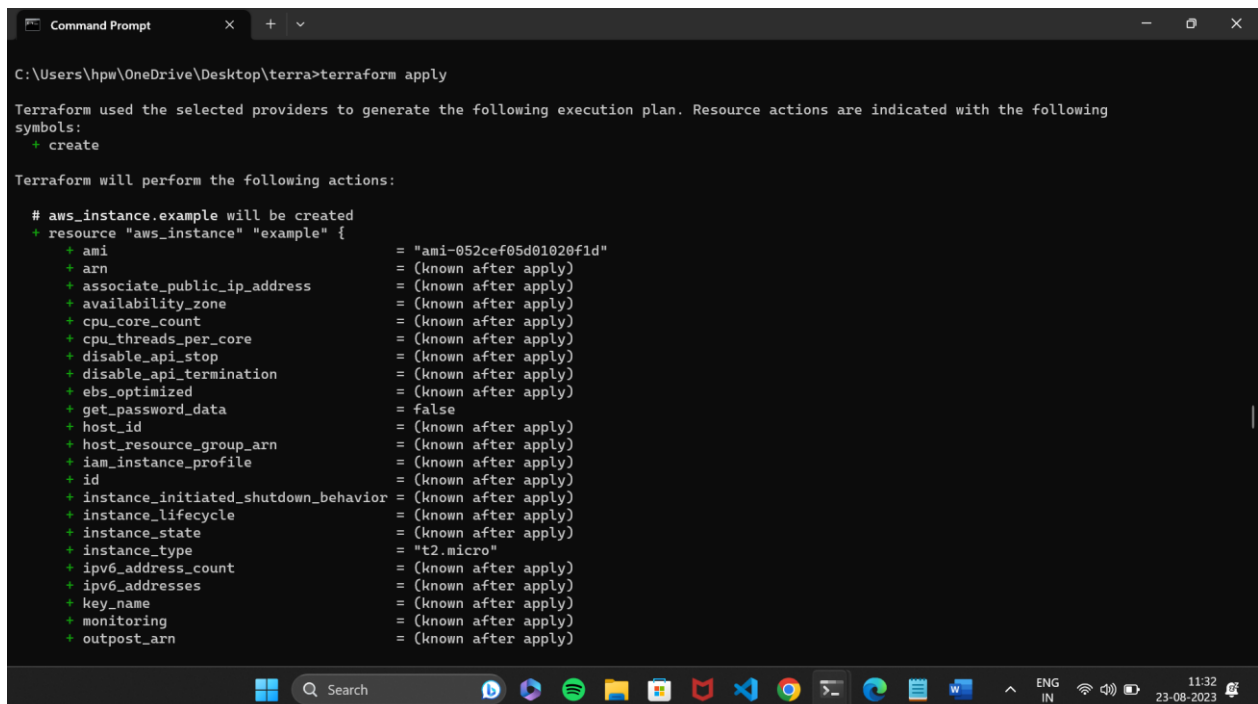
C:\Users\hpw\OneDrive\Desktop\terra>terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
  + create

Terraform will perform the following actions:

# aws_instance.example will be created
+ resource "aws_instance" "example" {
  + ami           = "ami-052cef05d01020f1d"
  + arn           = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone = (known after apply)
}
```

8. Then run command terraform apply



```
C:\Users\hpw\OneDrive\Desktop\terra>terraform apply

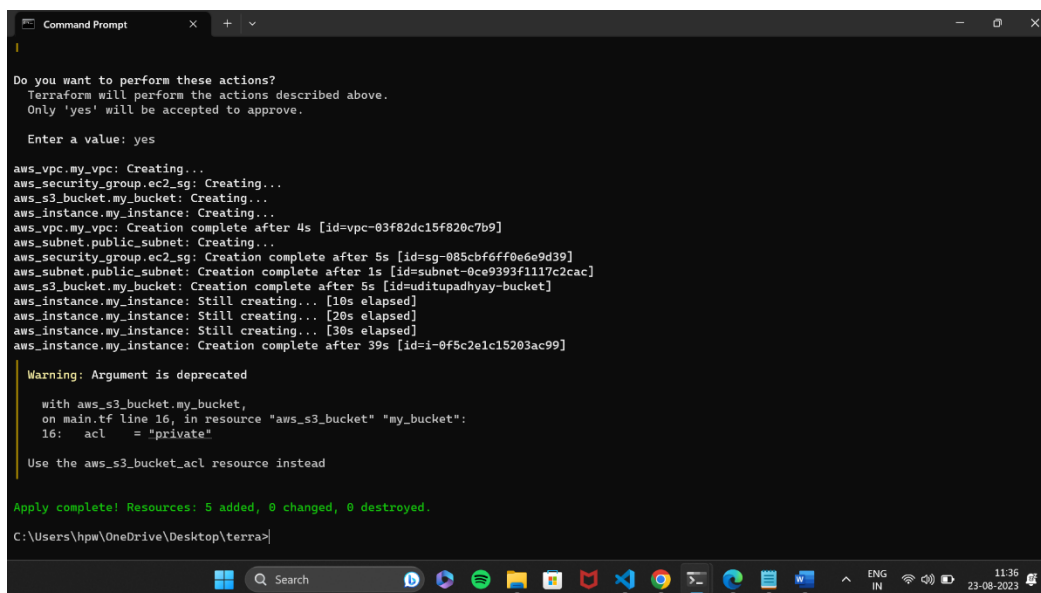
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
  + create

Terraform will perform the following actions:

# aws_instance.example will be created
+ resource "aws_instance" "example" {
  + ami           = "ami-052cef05d01020f1d"
  + arn           = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone = (known after apply)
  + cpu_core_count = (known after apply)
  + cpu_threads_per_core = (known after apply)
  + disable_api_stop = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized   = (known after apply)
  + get_password_data = false
  + host_id         = (known after apply)
  + host_resource_group_arn = (known after apply)
  + iam_instance_profile = (known after apply)
  + id             = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle = (known after apply)
  + instance_state  = (known after apply)
  + instance_type   = "t2.micro"
  + ipv6_address_count = (known after apply)
  + ipv6_addresses   = (known after apply)
  + key_name        = (known after apply)
  + monitoring       = (known after apply)
  + outpost_arn      = (known after apply)
}
```

9. You will Get Confirmation message to perform action which you put in terraform config and then type “yes” to execute them.

10. You will see message appear on prompt that “Apply Complete!” with added resource, changes and destruction of resources.



```
Command Prompt
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_vpc.my_vpc: Creating...
aws_security_group.ec2_sg: Creating...
aws_s3_bucket.my_bucket: Creating...
aws_instance.my_instance: Creating...
aws_vpc.my_vpc: Creation complete after 4s [id=vpc-03f82dc15f820c7b9]
aws_subnet.public_subnet: Creating...
aws_security_group.ec2_sg: Creation complete after 5s [id=sg-085cbf6ff0e6e9d39]
aws_subnet.public_subnet: Creation complete after 1s [id=subnet-0ce9393f1117c2cac]
aws_s3_bucket.my_bucket: Creation complete after 5s [id=uditupadhyay-bucket]
aws_instance.my_instance: Still creating... [10s elapsed]
aws_instance.my_instance: Still creating... [20s elapsed]
aws_instance.my_instance: Still creating... [30s elapsed]
aws_instance.my_instance: Creation complete after 39s [id=i-0f5c2e1c15203ac99]

Warning: Argument is deprecated

  with aws_s3_bucket.my_bucket,
  on main.tf line 16, in resource "aws_s3_bucket" "my_bucket":
   16:   acl      = "private"

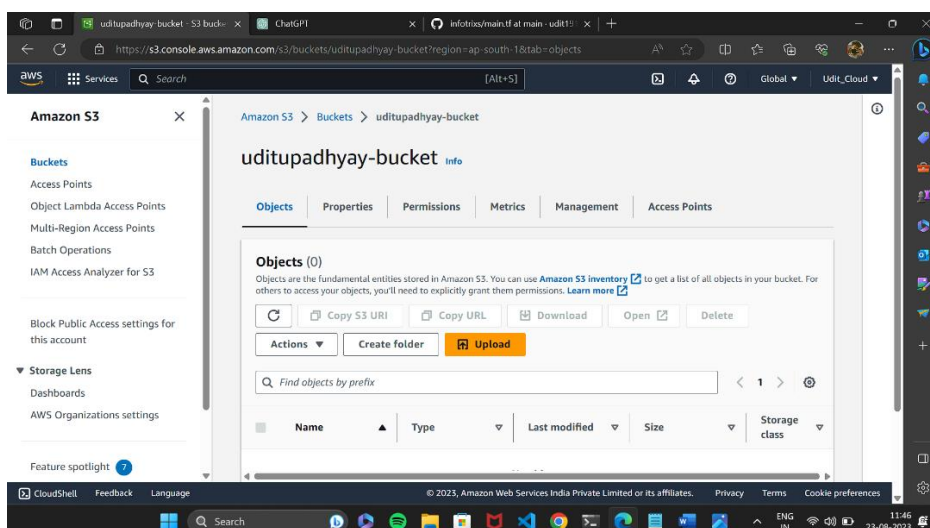
Use the aws_s3_bucket_acl resource instead

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.

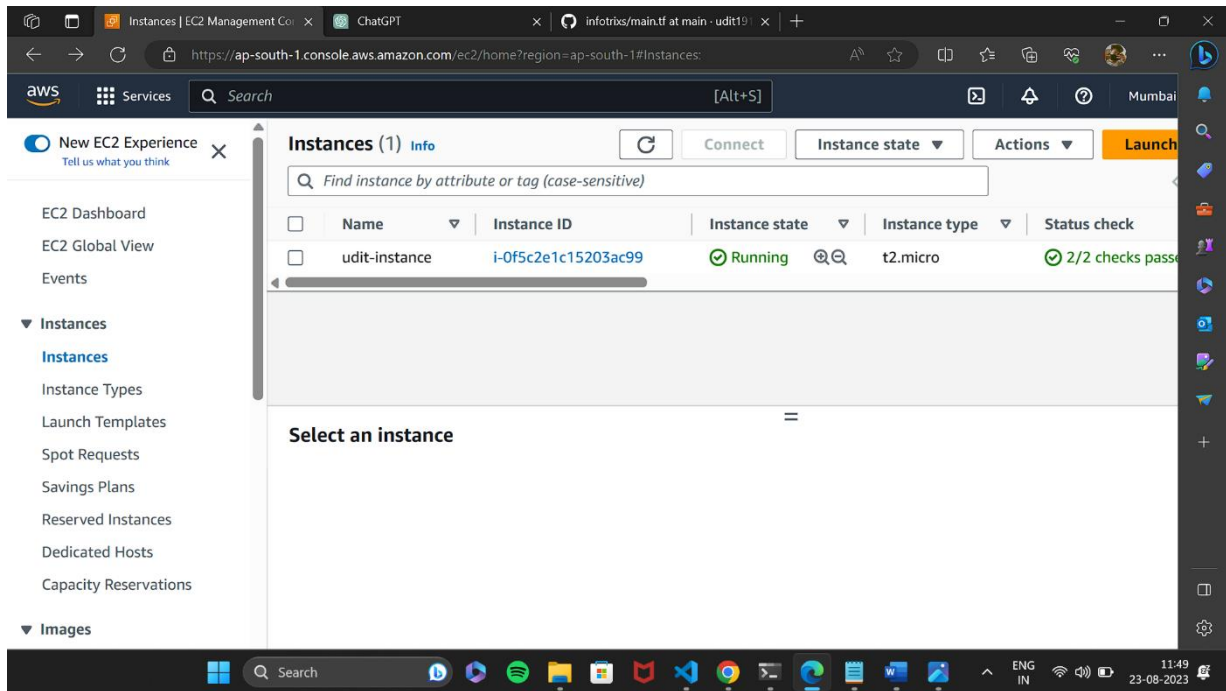
C:\Users\hpn\OneDrive\Desktop\terra>
```

11. Now Goto AWS Management Console and Check Resources which you created are visible in aws or not.

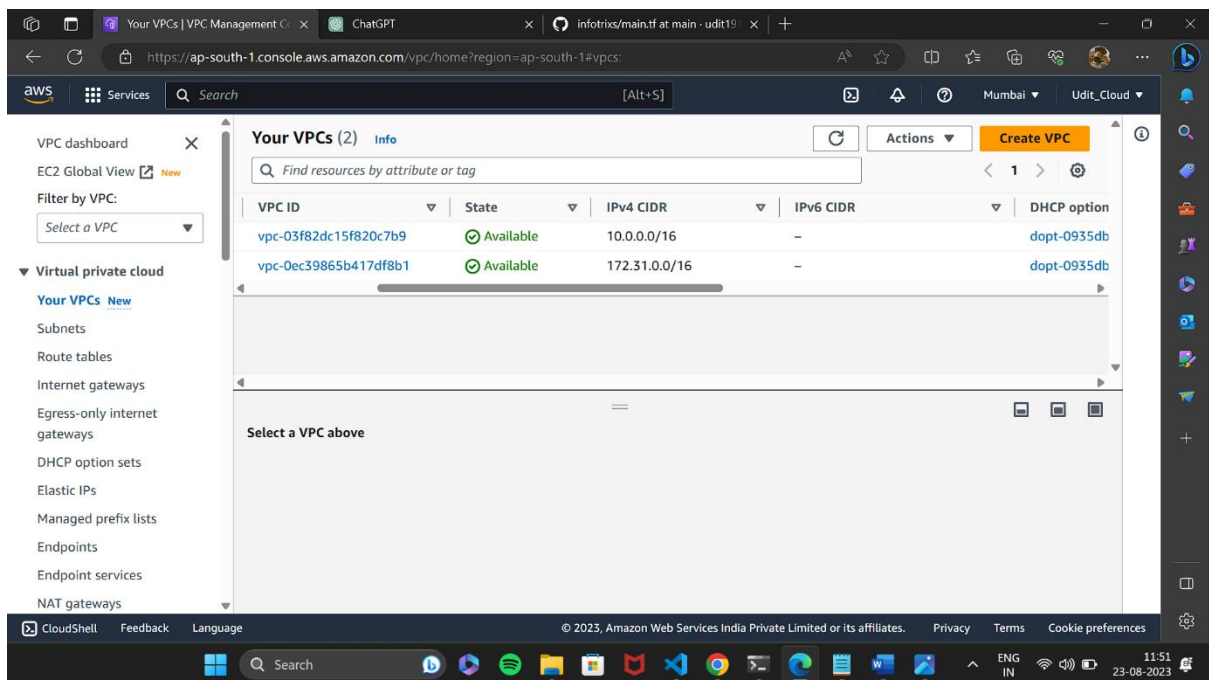
S3:



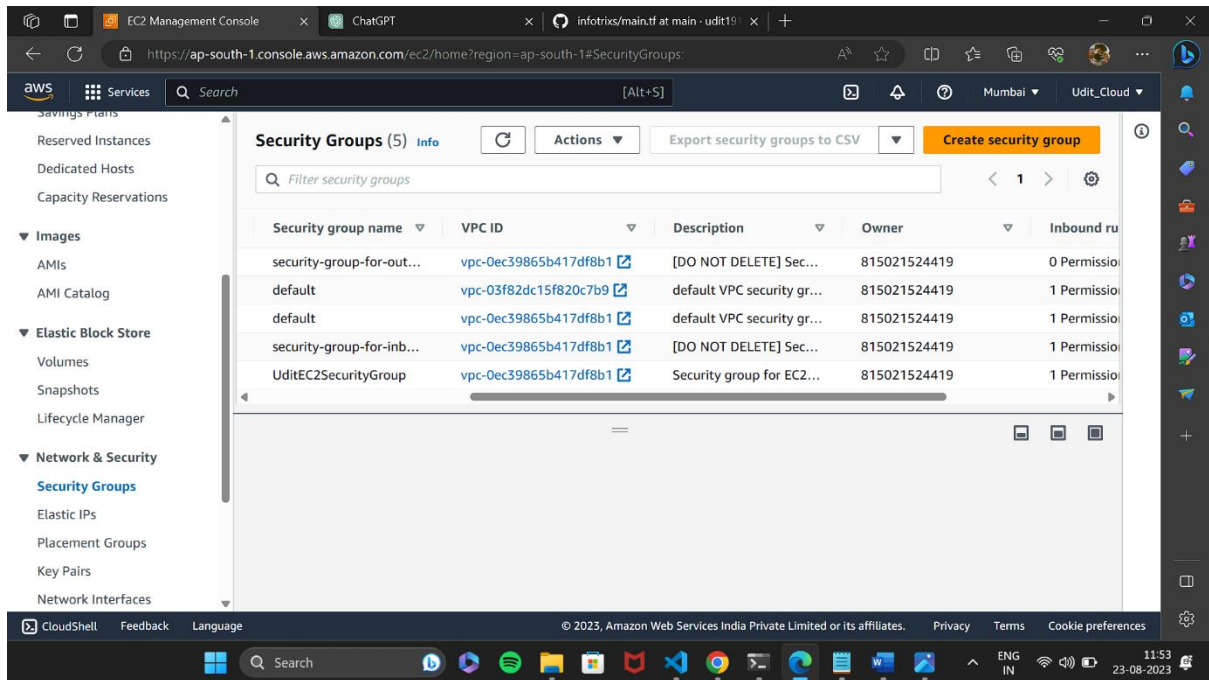
## EC2:



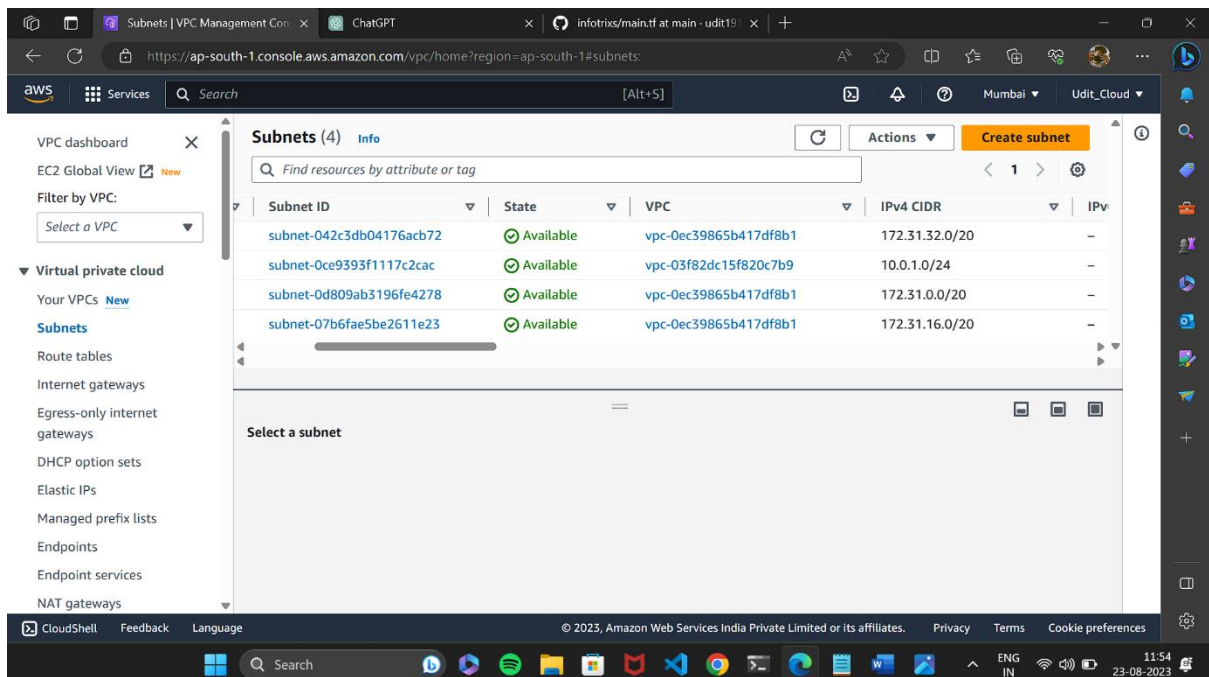
## VPC:



# SECURITY GROUP:

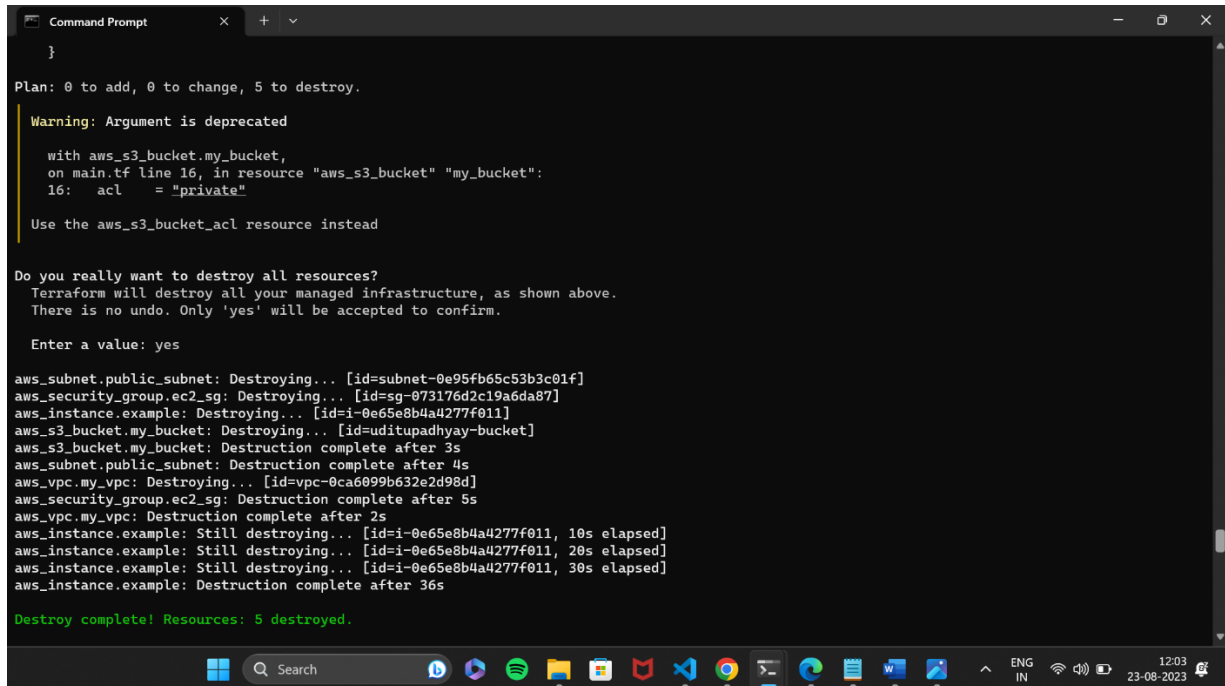


# SUBNET:



# OPTIONAL

You Can Also Delete All Resources Created By Terraform By Just Running Command “**Terraform destroy**”



```
Command Prompt
}

Plan: 0 to add, 0 to change, 5 to destroy.

Warning: Argument is deprecated
  with aws_s3_bucket.my_bucket,
  on main.tf line 16, in resource "aws_s3_bucket" "my_bucket":
  16:   acl      = "private"

Use the aws_s3_bucket_acl resource instead

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_subnet.public_subnet: Destroying... [id=subnet-0e95fb65c53b3c01f]
aws_security_group.ec2_sg: Destroying... [id=sg-073176d2c19a6da87]
aws_instance.example: Destroying... [id=i-0e65e8b4a4277f011]
aws_s3_bucket.my_bucket: Destroying... [id=uditupadhyay-bucket]
aws_s3_bucket.my_bucket: Destruction complete after 3s
aws_subnet.public_subnet: Destruction complete after 4s
aws_vpc.my_vpc: Destroying... [id=vpc-0ca6099b632e2d98d]
aws_security_group.ec2_sg: Destruction complete after 5s
aws_vpc.my_vpc: Destruction complete after 2s
aws_instance.example: Still destroying... [id=i-0e65e8b4a4277f011, 10s elapsed]
aws_instance.example: Still destroying... [id=i-0e65e8b4a4277f011, 20s elapsed]
aws_instance.example: Still destroying... [id=i-0e65e8b4a4277f011, 30s elapsed]
aws_instance.example: Destruction complete after 36s

Destroy complete! Resources: 5 destroyed.
```