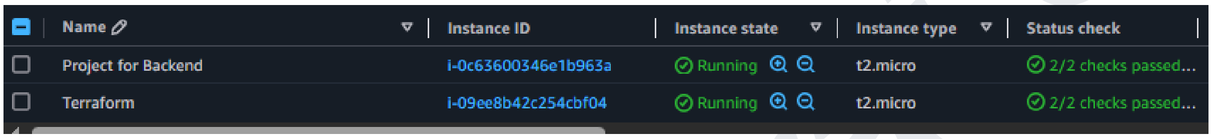
Terraform Project

Project Overview:

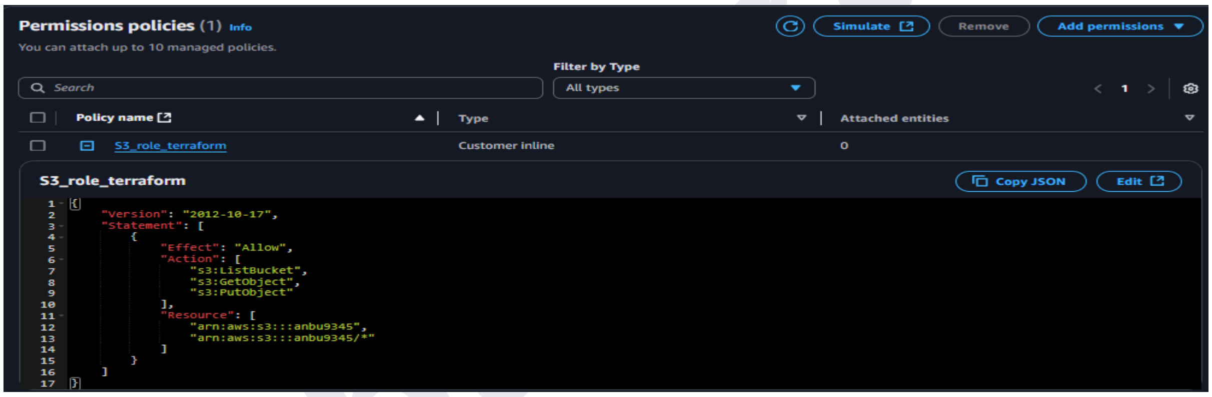
- **Initialize Terraform**: Set up a new Terraform project by creating a directory for your configuration files and initializing Terraform.
- **Backend Configuration**: Define the backend where Terraform will store its state (S3 for AWS).

Step : 1

Create an EC2 Linux instance and run a script file and install terraform on the server and create an IAM Policy for communication between EC2(Terraform instance) and S3.



Name	Instance ID	Instance state	Instance type	Status check
Project for Backend	i-0c63600346e1b963a	Running	t2.micro	2/2 checks passed...
Terraform	i-09ee8b42c254cbf04	Running	t2.micro	2/2 checks passed...



You can attach up to 10 managed policies.

Policy name	Type	Attached entities
S3_role_terraform	Customer inline	0

S3_role_terraform

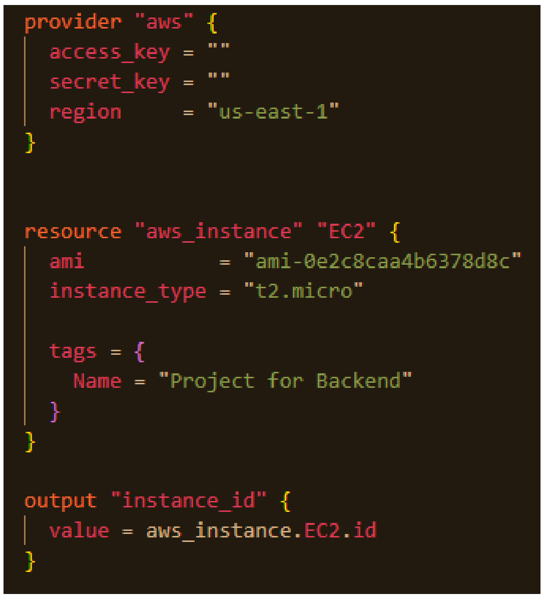
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::anbu9345",
                "arn:aws:s3:::anbu9345/*"
            ]
        }
    ]
}
```

Script File for terraform installation :

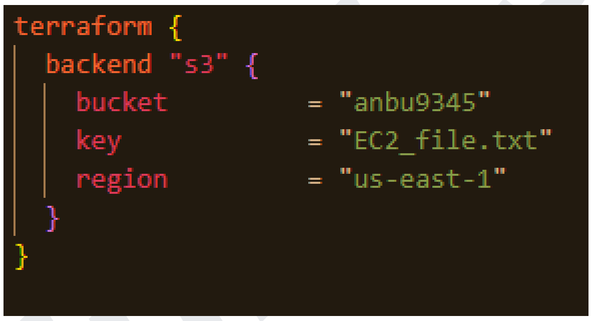
```
#!/bin/bash

mkdir terraform

cd terraform/

wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/hashicorp-archive-keyring.gpg

echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg]
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee
/etc/apt/sources.list.d/hashicorp.list

sudo apt update && sudo apt install terraform

terraform --version
```

Step 2:

Create a file in the server and save the below configured terraform code (Provider,resource and backend) on the server.



```
provider "aws" {
  access_key = ""
  secret_key = ""
  region     = "us-east-1"
}


resource "aws_instance" "EC2" {
  ami           = "ami-0e2c8caa4b6378d8c"
  instance_type = "t2.micro"

  tags = {
    Name = "Project for Backend"
  }
}

output "instance_id" {
  value = aws_instance.EC2.id
}
```



```
terraform {
  backend "s3" {
    bucket         = "anbu9345"
    key            = "EC2_file.txt"
    region         = "us-east-1"
  }
}
```

Step 3:

Initialization and Validation

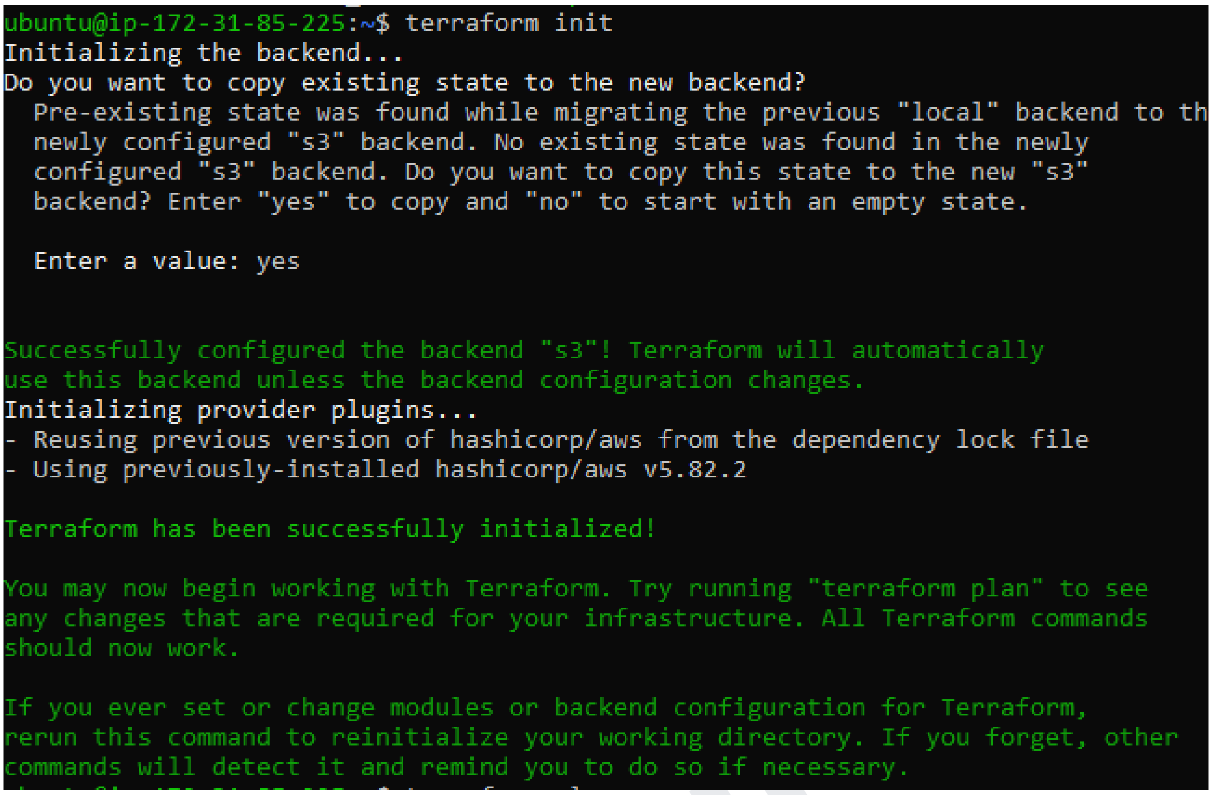
- **Initialize**: Run terraform init to initialize the project and download the necessary provider plugins.
- **Validation**: Validate your configuration using terraform validate to ensure there are no syntax errors or misconfigurations.

Planning and Applying

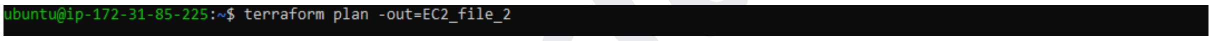
- **Plan**: Use terraform plan to generate an execution plan and review the changes Terraform will make.
- **Apply**: Run terraform apply to apply the changes and provision your infrastructure.

State Management

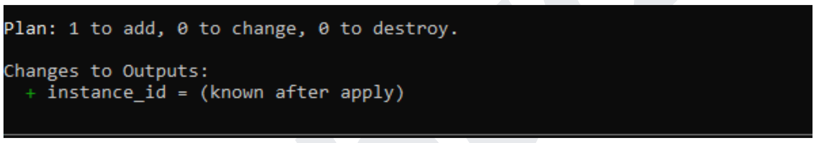
- **State File**: Terraform maintains a state file to keep track of the resources it manages. Ensure your state file is secure and backed up.(S3)



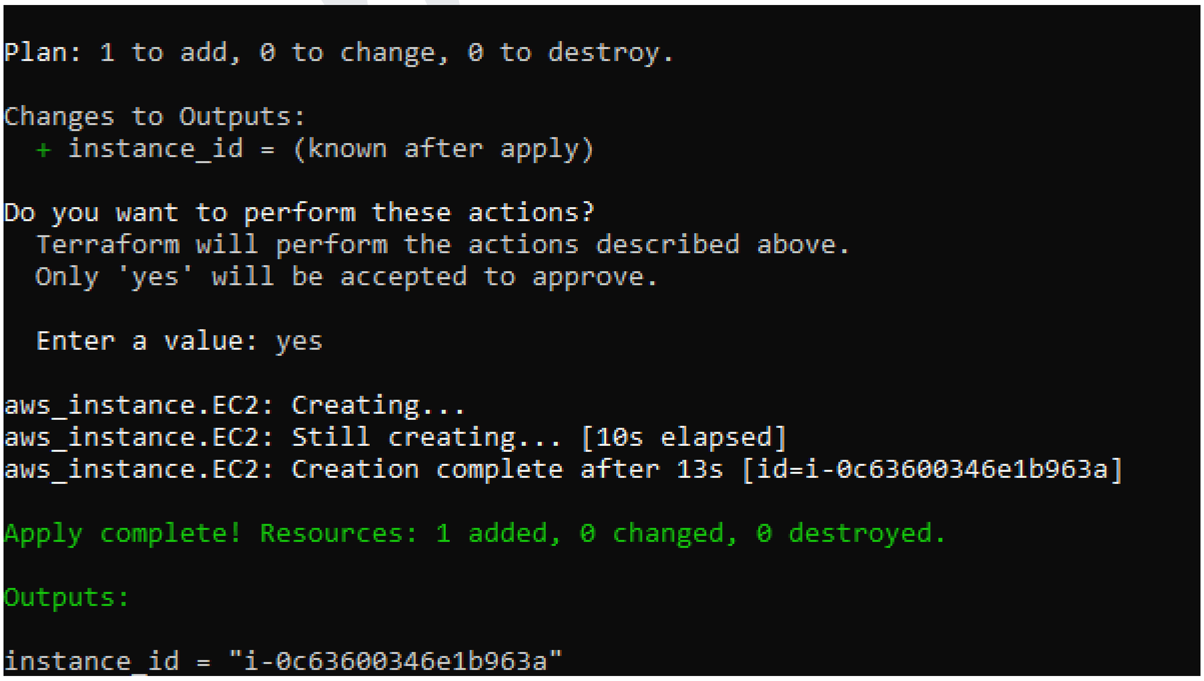
```
ubuntu@ip-172-31-85-225:~$ terraform init
Initializing the backend...
Do you want to copy existing state to the new backend?
  Pre-existing state was found while migrating the previous "local" backend to th
  newly configured "s3" backend. No existing state was found in the newly
  configured "s3" backend. Do you want to copy this state to the new "s3"
  backend? Enter "yes" to copy and "no" to start with an empty state.

  Enter a value: yes


Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.82.2

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-172-31-85-225:~$ terraform plan -out=EC2_file_2
```



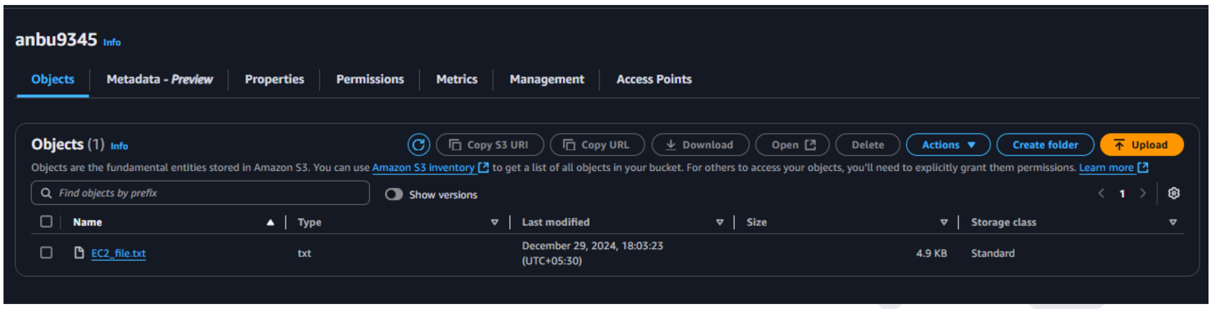
```
Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + instance_id = (known after apply)
```



```
Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + instance_id = (known after apply)

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_instance.EC2: Creating...
aws_instance.EC2: Still creating... [10s elapsed]
aws_instance.EC2: Creation complete after 13s [id=i-0c63600346e1b963a]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

instance_id = "i-0c63600346e1b963a"
```

Step 4 :

The Terraform file has been stored in S3 as per the configuration from the backend file .



anbu9345 Info

Objects (1) Info

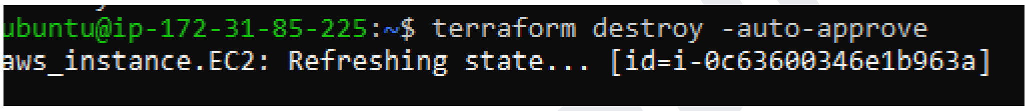
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
EC2_file.txt	txt	December 29, 2024, 18:03:23 (UTC+05:30)	4.9 KB	Standard

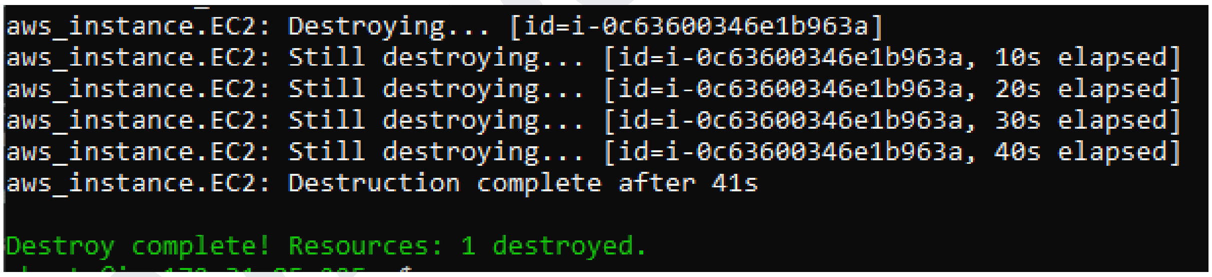
Step 5 :

Destroy the infrastructure: terraform destroy -autoapprove

Verify that all specified resources have been successfully destroyed.



```
ubuntu@ip-172-31-85-225:~$ terraform destroy -auto-approve
aws_instance.EC2: Refreshing state... [id=i-0c63600346e1b963a]
```



```
aws_instance.EC2: Destroying... [id=i-0c63600346e1b963a]
aws_instance.EC2: Still destroying... [id=i-0c63600346e1b963a, 10s elapsed]
aws_instance.EC2: Still destroying... [id=i-0c63600346e1b963a, 20s elapsed]
aws_instance.EC2: Still destroying... [id=i-0c63600346e1b963a, 30s elapsed]
aws_instance.EC2: Still destroying... [id=i-0c63600346e1b963a, 40s elapsed]
aws_instance.EC2: Destruction complete after 41s

Destroy complete! Resources: 1 destroyed.
```

Thank you ! Happy Learning !