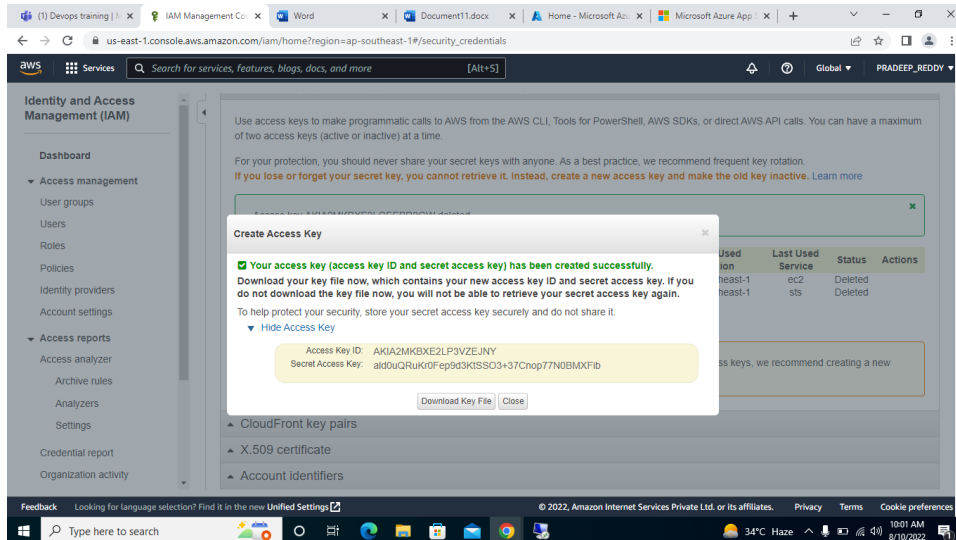


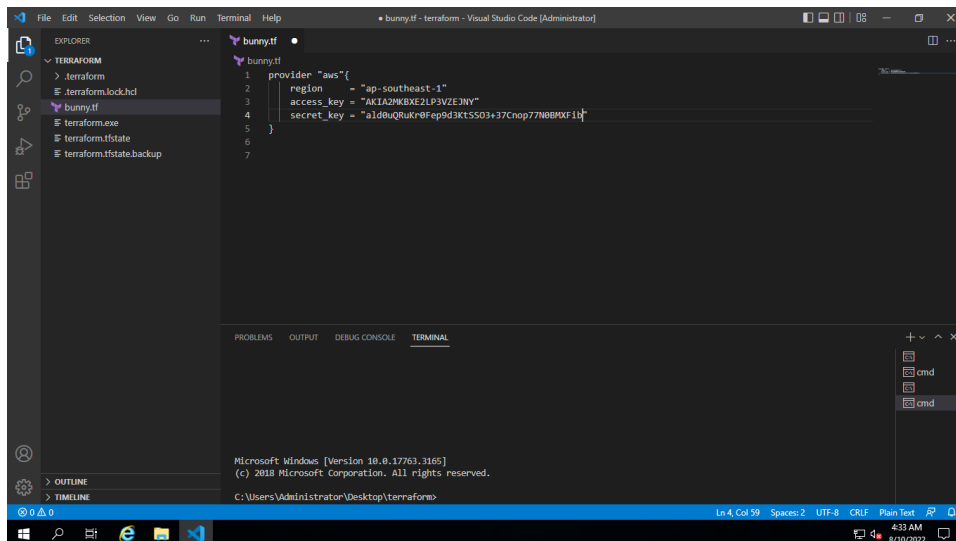
Launching S3 Bucket on AWS via Terraform

1. Generate access key and secret key

Security credentials >> create new access key

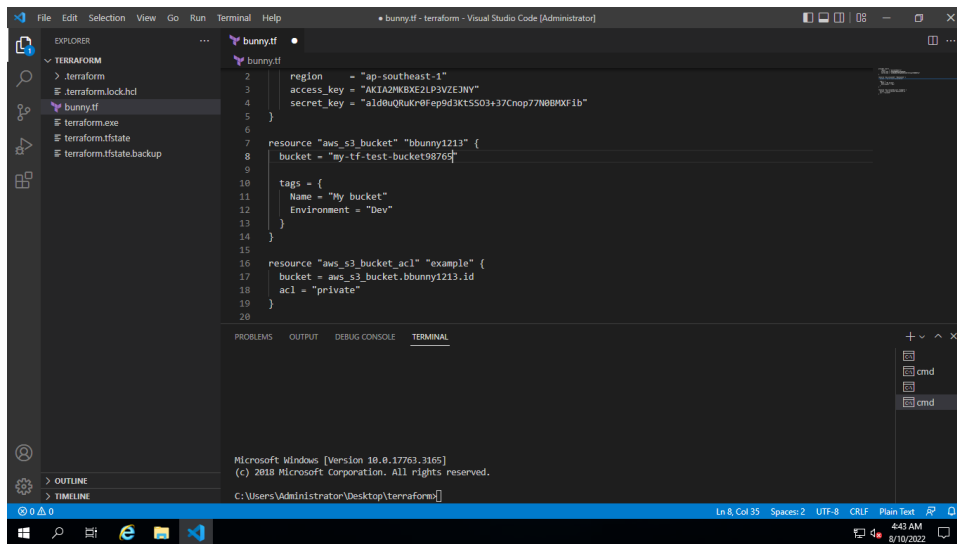


Provide keys >> region

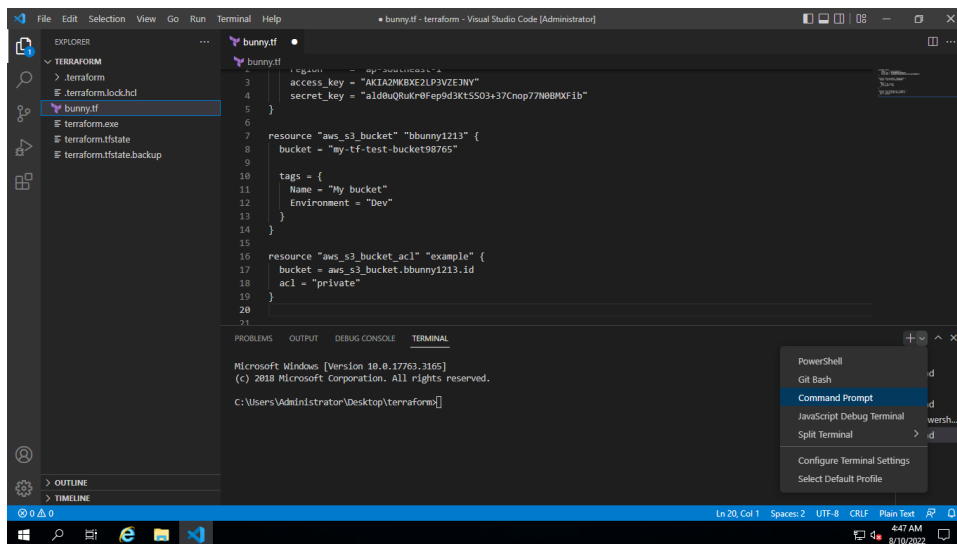


2. Give any name to resource >> and bucket >> tags >> ACL as private >> save

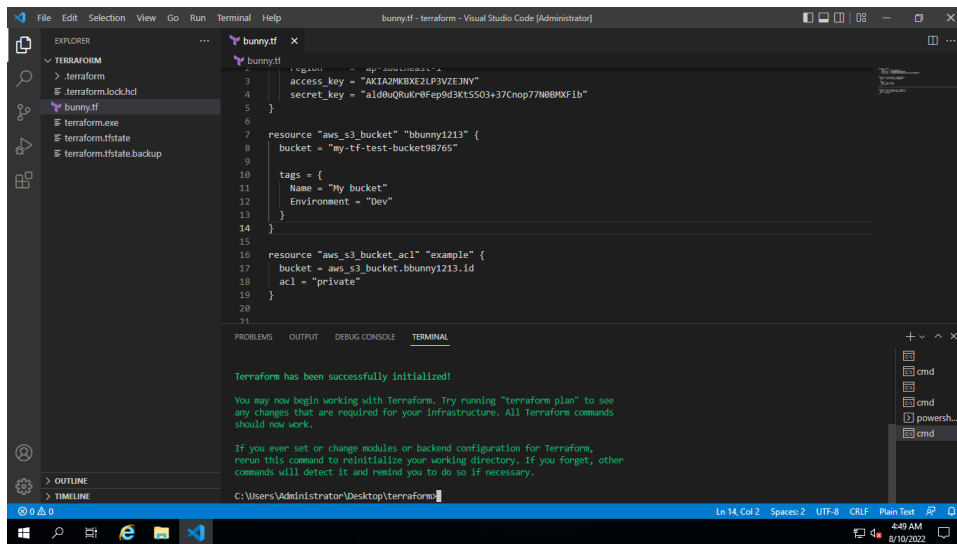
ACL (access control list)



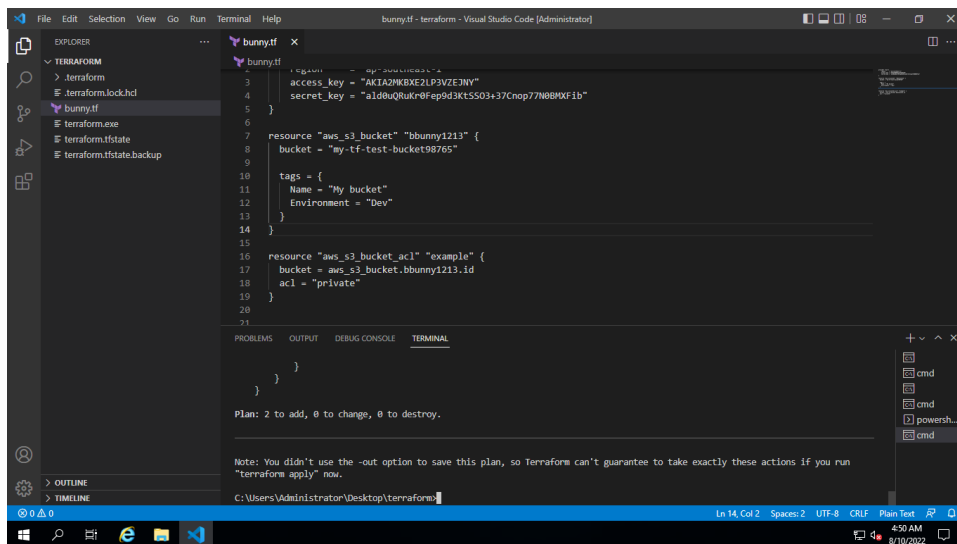
3. Terminal >> plus >> command prompt



In cmd prompt >> command >> terraform init



Terraform plan



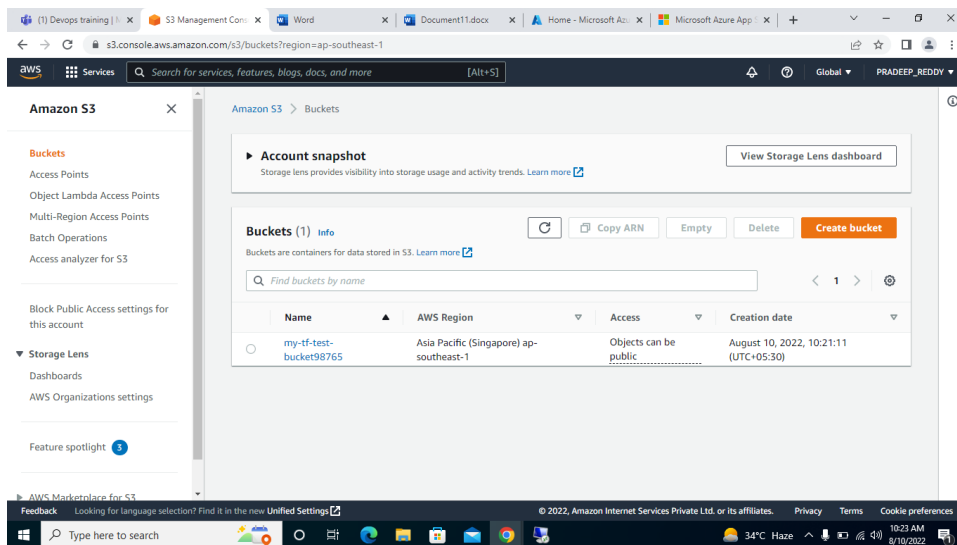
Terraform apply >> give YES

The screenshot shows a Visual Studio Code window with a Terraform configuration file named 'bunny.tf'. The configuration defines two AWS S3 buckets: 'bunny1213' and 'example'. The 'bunny1213' bucket is created with a specific access key and secret key, and the 'example' bucket is created with the same access key and secret key. The terminal output shows the successful execution of the Terraform plan and apply commands, resulting in the creation of both buckets.

```
1 provider "aws" {
2   access_key = "AKIA2PHKXE2LP3VZEJNY"
3   secret_key = "a1d0uQrKr0Fep9d3Kt5503+37Cnop77N08MXf1b"
4 }
5
6
7 resource "aws_s3_bucket" "bunny1213" {
8   bucket = "my-tf-test-bucket98765"
9 }
10
11 tags = {
12   Name = "My bucket"
13   Environment = "Dev"
14 }
15
16 resource "aws_s3_bucket_acl" "example" {
17   bucket = aws_s3_bucket.bunny1213.id
18   acl = "private"
19 }
20
21
```

Only 'yes' will be accepted to approve.
Enter a value: yes
aws_s3_bucket.bunny1213: Creating...
aws_s3_bucket.bunny1213: Creation complete after 3s [id=my-tf-test-bucket98765]
aws_s3_bucket_acl.example: Creating...
aws_s3_bucket_acl.example: Creation complete after 0s [id=my-tf-test-bucket98765,private]
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

U can see bucket created in AWS



If u want to destroy we can destroy it by using command >> terraform destroy

