

Lab Exercise 5–Provisioning an S3 Bucket on AWS

Name- Manish kumar

SAP ID- 500119723

Batch- 2

Exercise Steps:

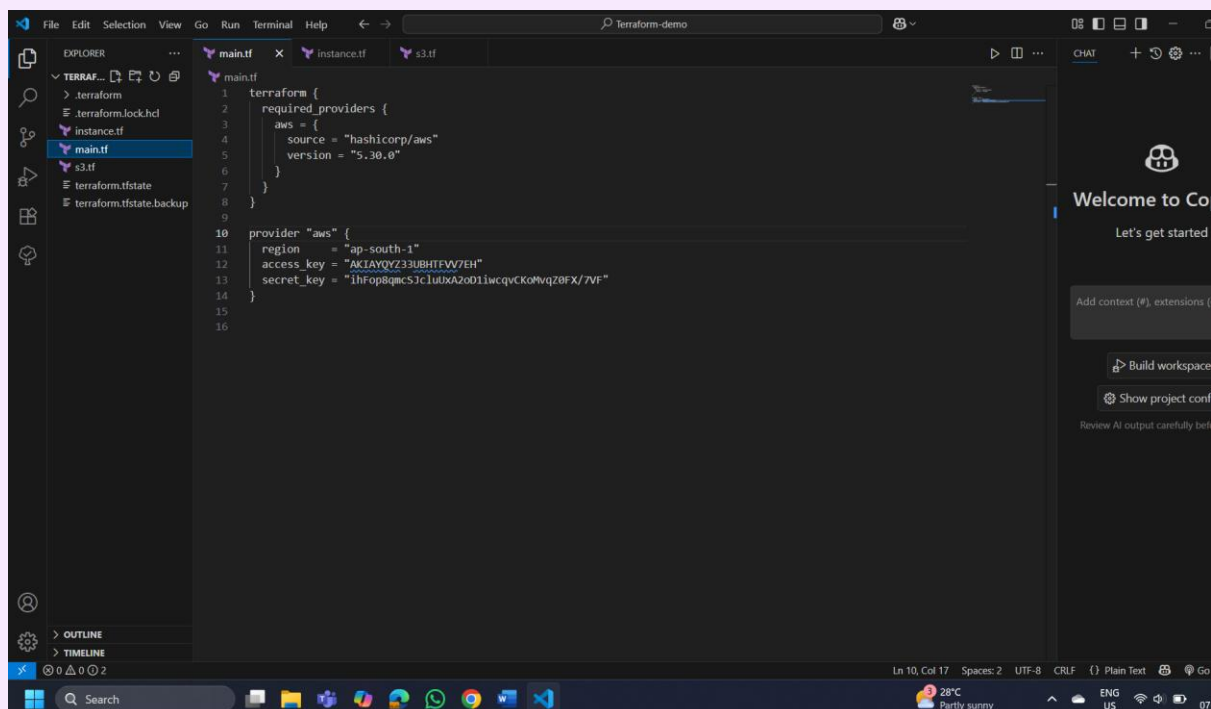
Step 1: Create a New Directory:

Create a new directory to store your Terraform configuration:

```
mkdir Terraform-S3-Demo
cd Terraform-S3-Demo
```

Step 2: Create the Terraform Configuration File (main.tf):

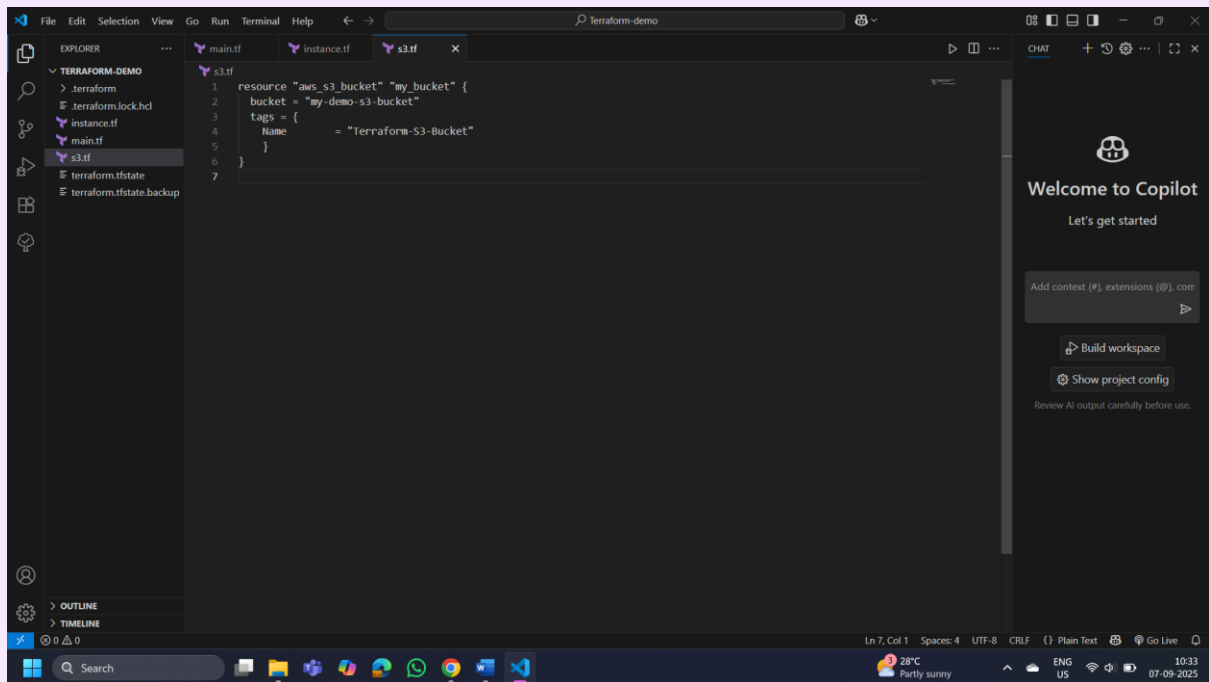
Create a file named main.tf with the following content:



```
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.30.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   region = "ap-south-1"
12   access_key = "AKIAVQVZ33UBHTEVW7EH"
13   secret_key = "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"
14 }
15
16
```

This file sets up the Terraform AWS provider.

Step 3: Create a Terraform Configuration File for the S3 Bucket (s3.tf):



This file provisions an S3 bucket with a unique name using a random string suffix.

Step 4: Initialize Terraform:

Run the following command to initialize your Terraform working directory:

```
terraform init
```

```
Command Prompt
(c) Microsoft Corporation. All rights reserved.

C:\Users\Misha>cd Terraform
The system cannot find the path specified.

C:\Users\Misha>cd "C:\Terraform\Terraform-demo"

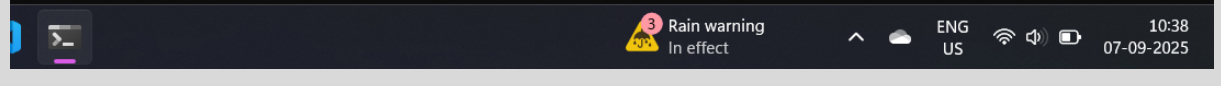
C:\Terraform\Terraform-demo>terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.30.0

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:\Terraform\Terraform-demo>|
```



Step 5: Review the Plan:

Preview the changes Terraform will make:

```
terraform plan
```

```
Command Prompt
C:\Terraform\Terraform-demo>terraform plan

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.My-instance will be created
+ resource "aws_instance" "My-instance" {
  + ami                        = "ami-03f4878755434977f"
  + 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              = "t3.micro"
  + ipv6_address_count         = (known after apply)
}
```

Step 6: Apply the Changes:

Create the resources:

```
terraform apply
```

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

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_s3_bucket.my_bucket: Creating...
aws_s3_bucket.my_bucket: Creation complete after 4s [id=demo-s3-bucket-misha-12072005]

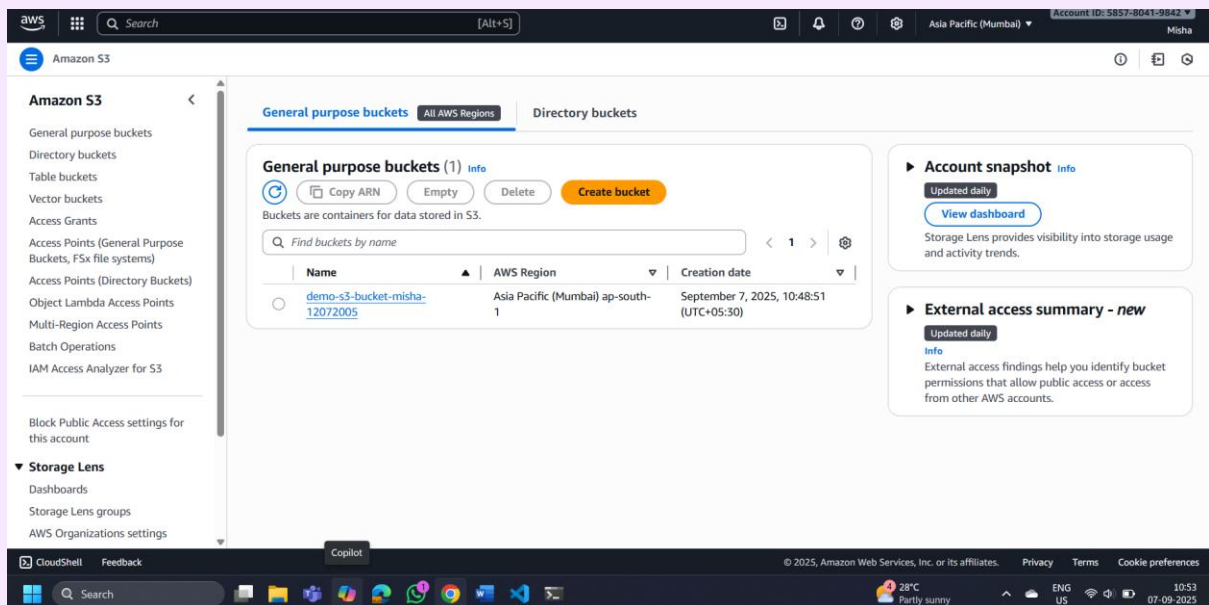
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

C:\Terraform\Terraform-demo>
```

When prompted, type yes to confirm.

Step 7: Verify Resources:

1. Log in to your AWS Management Console.
2. Navigate to the **S3** dashboard.
3. Verify that the S3 bucket has been created with the specified configuration.



Step 8: Cleanup Resources:

To remove the resources created, run the following command:

```
terraform destroy
```

```
Enter a value: yes

aws_s3_bucket.my_bucket: Destroying... [id=demo-s3-bucket-misha-12072005]
aws_instance.My-instance: Destroying... [id=i-01d82deebfcf1aa35]
aws_s3_bucket.my_bucket: Destruction complete after 1s
aws_instance.My-instance: Still destroying... [id=i-01d82deebfcf1aa35, 00m10s elapsed]
aws_instance.My-instance: Still destroying... [id=i-01d82deebfcf1aa35, 00m20s elapsed]
aws_instance.My-instance: Still destroying... [id=i-01d82deebfcf1aa35, 00m30s elapsed]
aws_instance.My-instance: Still destroying... [id=i-01d82deebfcf1aa35, 00m40s elapsed]
aws_instance.My-instance: Still destroying... [id=i-01d82deebfcf1aa35, 00m50s elapsed]
aws_instance.My-instance: Destruction complete after 51s

Destroy complete! Resources: 2 destroyed.

C:\Terraform\Terraform-demo>
```

When prompted, type yes to confirm.
