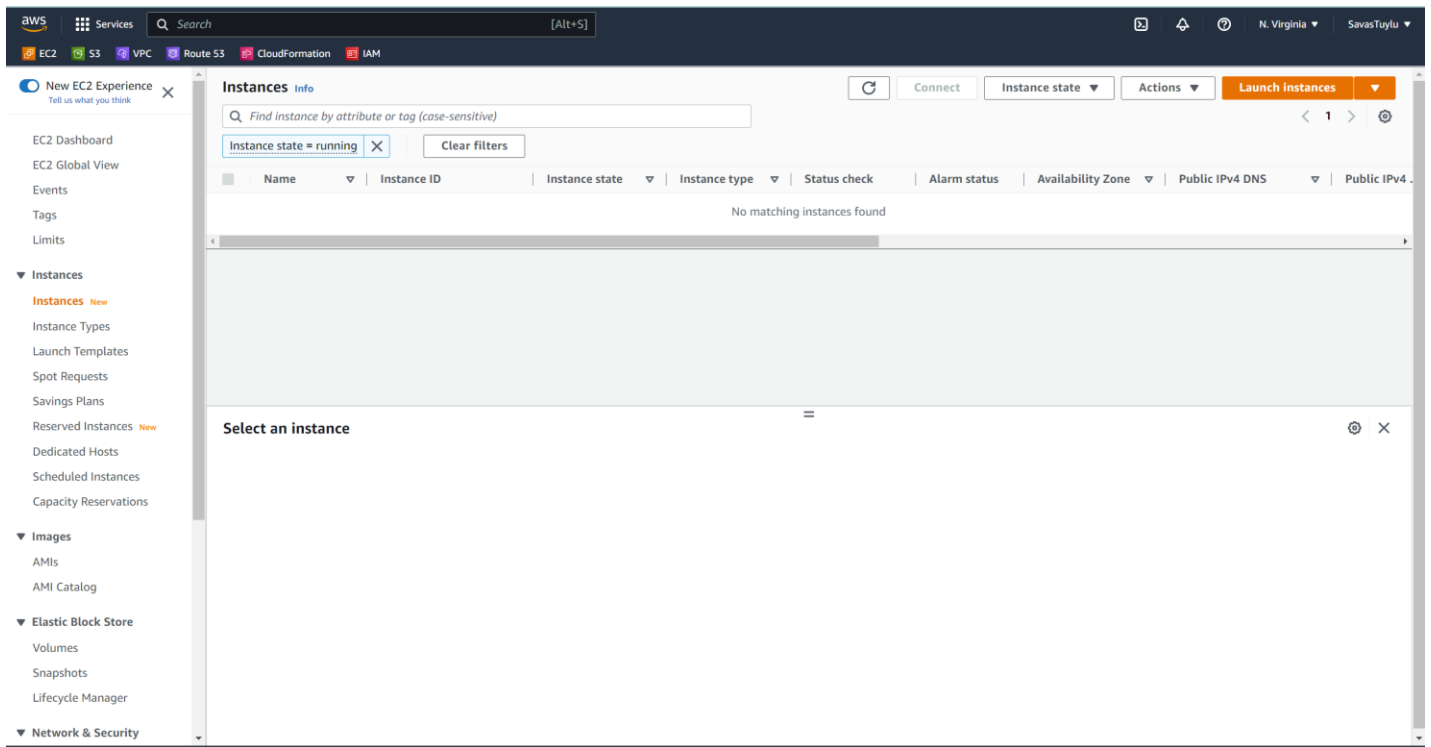
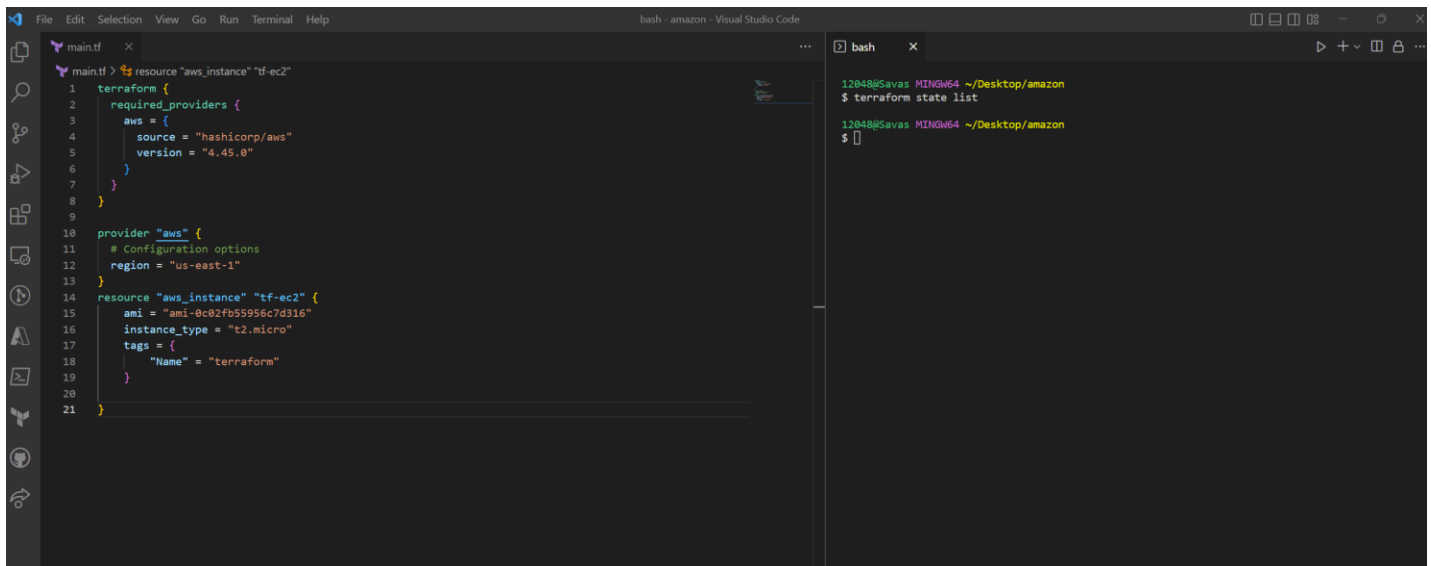


Terraform-Resources:-aws_ec2_instance & aws_s3_bucket

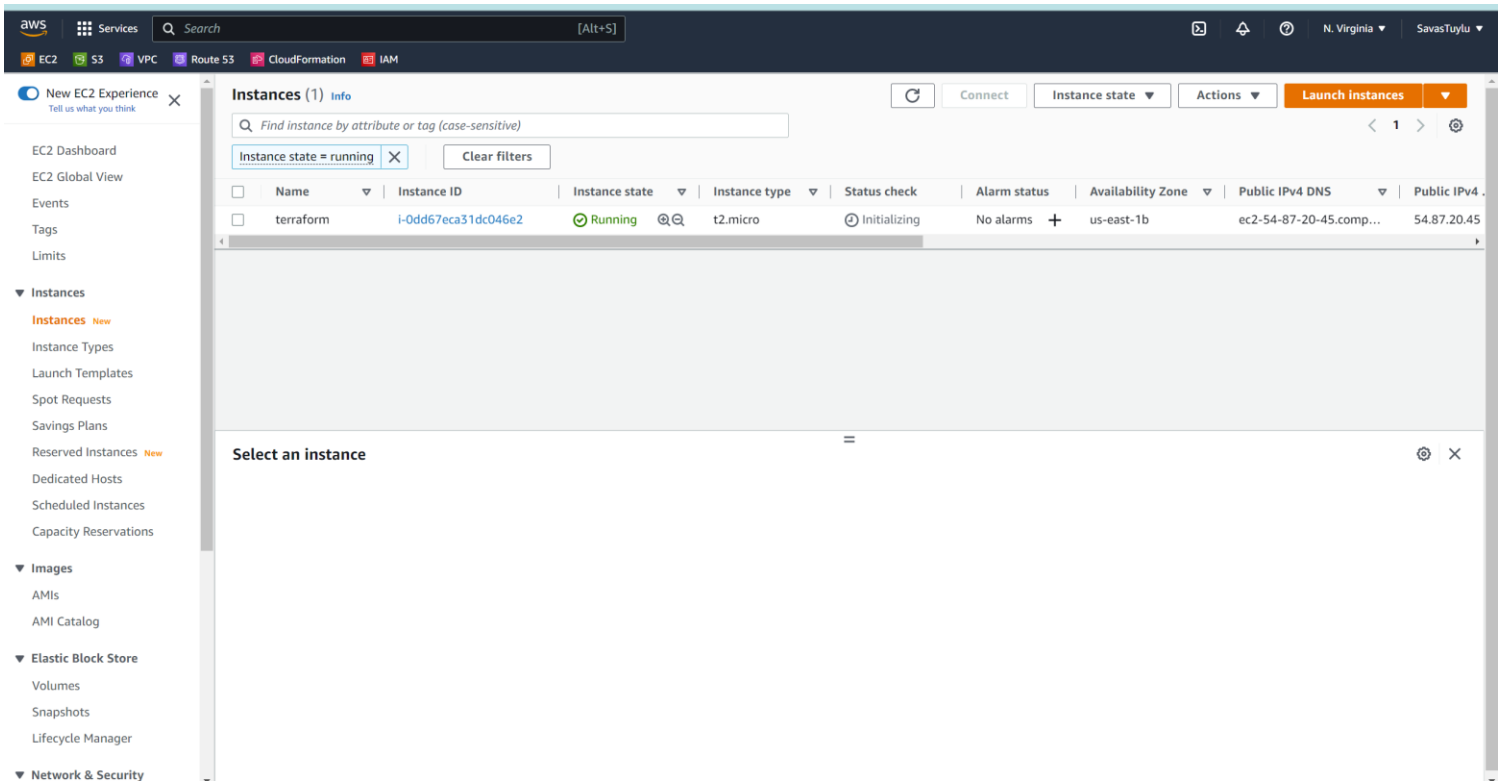
1A-Before Deployment of EC2



1B-Before Deployment in VS Code



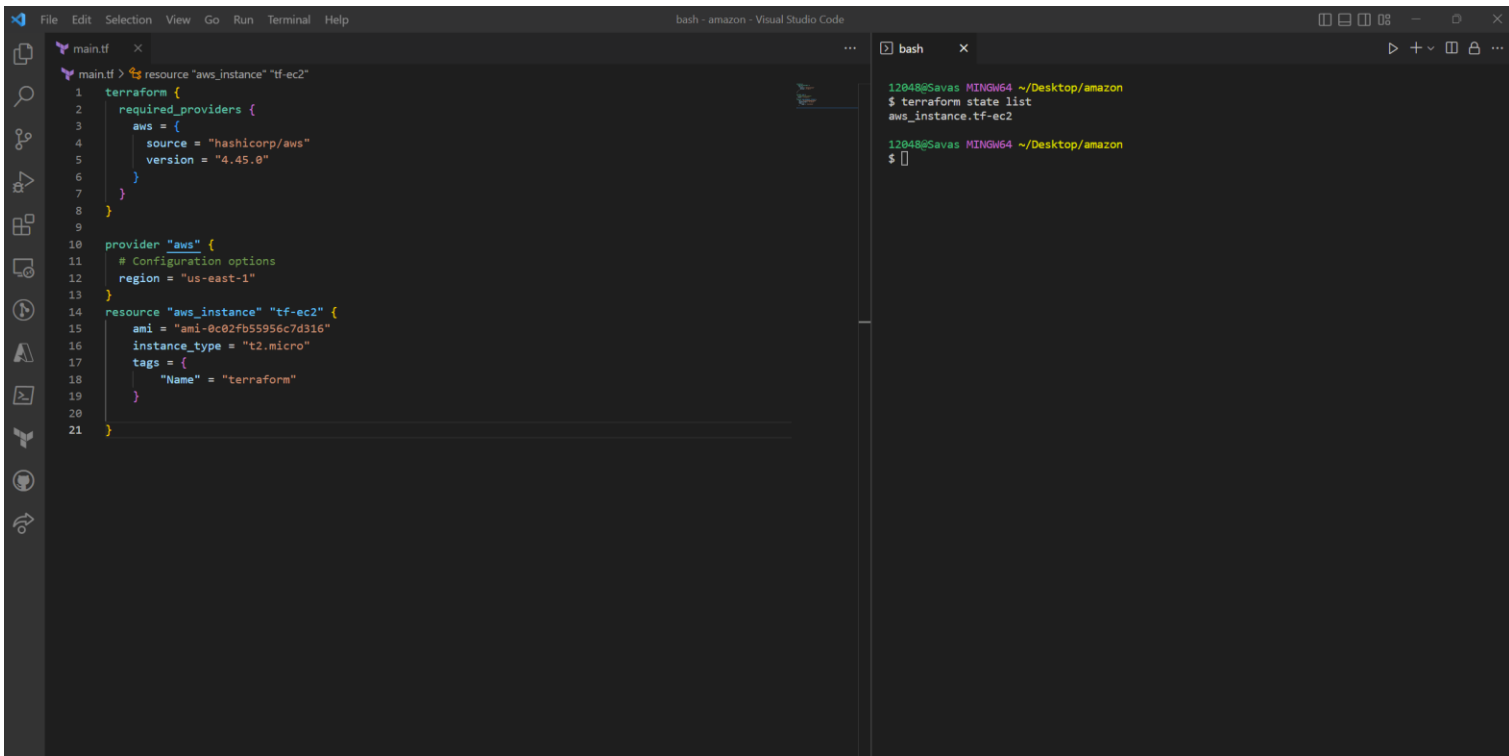
2A-After Deployment of EC2



The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, a search bar, and the user's name 'SavasTuytu'. The left sidebar contains a navigation menu with categories like 'New EC2 Experience', 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances', 'Images', 'Elastic Block Store', and 'Network & Security'. The main content area displays the 'Instances (1) Info' page. A table lists the instance 'terraform' with ID 'i-0dd67eca31dc046e2', state 'Running', type 't2.micro', and other details. Below the table, a 'Select an instance' dialog is visible.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
terraform	i-0dd67eca31dc046e2	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-54-87-20-45.comp...	54.87.20.45

2B-Before Deployment in VS Code



The screenshot shows a Visual Studio Code editor with a Terraform configuration file named 'main.tf' and a terminal window. The configuration file contains the following code:

```
1 resource "aws_instance" "tf-ec2"
2   terraform {
3     required_providers {
4       aws = {
5         source = "hashicorp/aws"
6         version = "4.45.0"
7       }
8     }
9   }
10  provider "aws" {
11    # Configuration options
12    region = "us-east-1"
13  }
14  resource "aws_instance" "tf-ec2" {
15    ami = "ami-0c82fb595956c7d316"
16    instance_type = "t2.micro"
17    tags = {
18      "Name" = "terraform"
19    }
20  }
21 }
```

The terminal window shows the following commands and output:

```
12048@Savas MINGW64 ~/Desktop/amazon
$ terraform state list
aws_instance.tf-ec2

12048@Savas MINGW64 ~/Desktop/amazon
$
```

3A-Deployment of EC2-Count=5

The screenshot displays the AWS Management Console interface. On the left, the navigation menu includes sections for EC2 Dashboard, Events, Tags, Limits, Instances (with a sub-menu for Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, and Capacity Reservations), Images (with a sub-menu for AMIs and AMI Catalog), and Elastic Block Store (with a sub-menu for Volumes, Snapshots, and Lifecycle Manager). The main content area is titled 'Instances (5) Info'. It features a search bar with the placeholder text 'Find instance by attribute or tag (case-sensitive)' and a filter dropdown set to 'Instance state = running'. Below this is a table listing five EC2 instances, all of which are in the 'Running' state. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv4 Address. The instances are named 'terraform' and have various Instance IDs. The Status check column shows '2/2 checks passed' for one instance and 'Initializing' for the others. The Alarm status column shows 'No alarms' for all instances. The Availability Zone is 'us-east-1b' for all instances. The Public IPv4 DNS and Public IPv4 Address columns show the respective values for each instance. At the bottom of the console, there is a 'Select an instance' dialog box.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 Address
terraform	i-0786392d53650cde4	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-34-226-197-82.co...	34.226.197.82
terraform	i-0cf47bd1789def9d9	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-54-172-251-25.co...	54.172.251.25
terraform	i-0dd67eca31dc046e2	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-87-20-45.comp...	54.87.20.45
terraform	i-046059d6e54e3b6a8	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-107-21-75-177.co...	107.21.75.177
terraform	i-071f8e80ea553122a	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-3-90-103-57.comp...	3.90.103.57

3B-Deployment in VS Code- Count=5

The screenshot shows a Visual Studio Code editor with a Terraform configuration file named 'main.tf' and a terminal window displaying the output of the 'terraform state list' command. The Terraform configuration file contains the following code:

```
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "4.45.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   # Configuration options
12   region = "us-east-1"
13 }
14 resource "aws_instance" "tf-ec2" {
15   ami           = "ami-0c02fb5956c7d316"
16   instance_type = "t2.micro"
17   tags = {
18     "Name" = "terraform"
19   }
20   count = 5
21 }
22
23
```

The terminal window shows the output of the 'terraform state list' command, listing the five instances created:

```
12B48@Savas MINGW64 ~/Desktop/amazon
$ terraform state list
aws_instance.tf-ec2[0]
aws_instance.tf-ec2[1]
aws_instance.tf-ec2[2]
aws_instance.tf-ec2[3]
aws_instance.tf-ec2[4]

12B48@Savas MINGW64 ~/Desktop/amazon
$
```

4A-Before Deployment of S3

The screenshot shows the AWS S3 console landing page. The top navigation bar includes the AWS logo, a search bar, and links to various services like EC2, S3, VPC, Route 53, CloudFormation, and IAM. A blue banner at the top contains a message about improving the S3 console and a 'Provide feedback' button. The main content area features the 'Amazon S3' logo and the tagline 'Store and retrieve any amount of data from anywhere'. Below this, a video titled 'Introduction to Amazon S3' is displayed. To the right, there are three sections: 'Create a bucket' with a 'Create bucket' button, 'Pricing' with information about fees and a 'View pricing details' link, and 'Resources' with links to the 'User guide' and 'API reference'.

Storage

Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

[Create bucket](#)

Pricing

With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.

Estimate your monthly bill using the [AWS Simple Monthly Calculator](#)

[View pricing details](#)

Resources

- [User guide](#)
- [API reference](#)

How it works

[Introduction to Amazon S3](#)

[Copy link](#)

4A-After Deployment of S3

The screenshot shows the AWS S3 console after deployment. The left sidebar is expanded, showing the 'Amazon S3' section with a 'Buckets' link. The main content area displays the 'Buckets' page. At the top, there is a blue banner with a message about improving the S3 console and a 'Provide feedback' button. Below this, there is a section for 'Account snapshot' with a 'View Storage Lens dashboard' button. The 'Buckets (1)' section shows a table with one bucket, 'my-tf-test-bucket345353', in the 'US East (N. Virginia) us-east-1' region. The table has columns for Name, AWS Region, Access, and Creation date. A 'Create bucket' button is visible in the top right of the buckets section.

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- AWS Organizations settings

Feature spotlight 3

[AWS Marketplace for S3](#)

Account snapshot

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

Buckets (1) [Info](#)

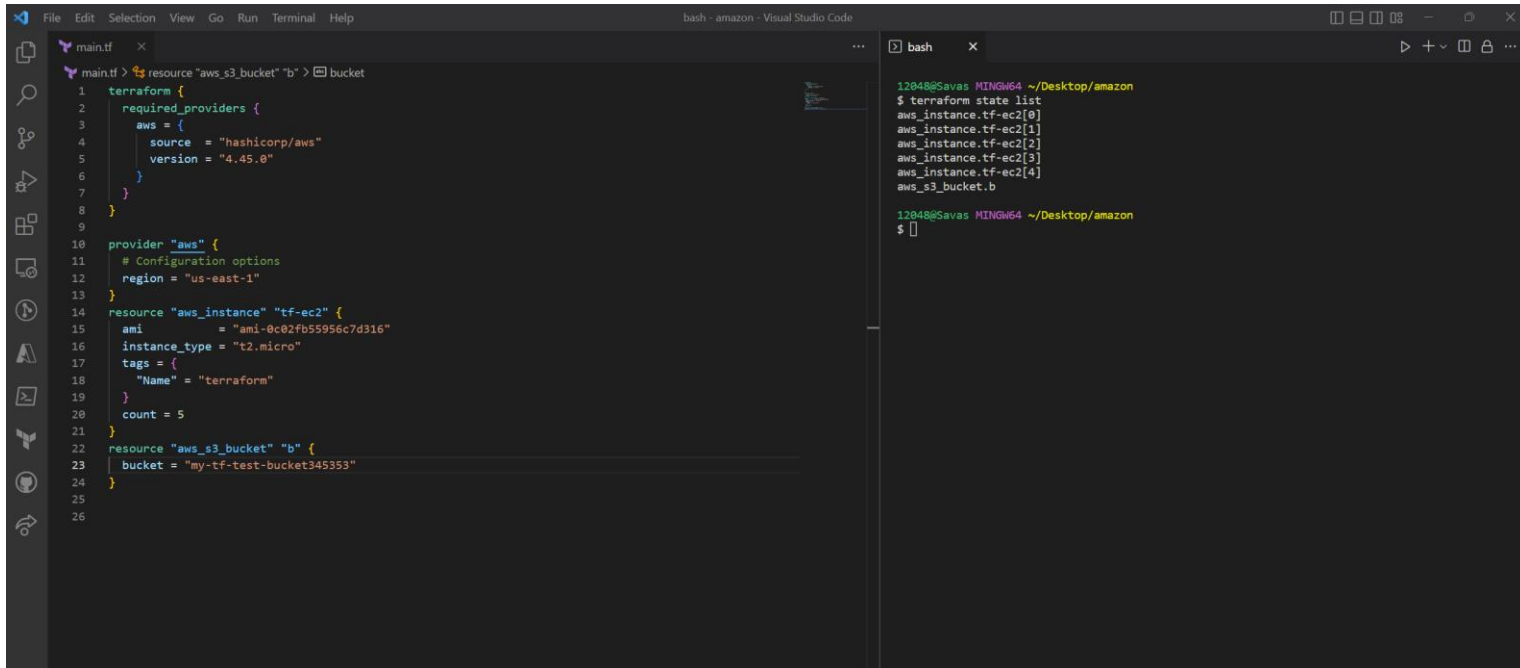
Buckets are containers for data stored in S3. [Learn more](#)

[Find buckets by name](#)

	Name	AWS Region	Access	Creation date
<input type="radio"/>	my-tf-test-bucket345353	US East (N. Virginia) us-east-1	Objects can be public	December 7, 2022, 20:03:29 (UTC-05:00)

[Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

24-Before Deployment in VS Code



The screenshot shows the Visual Studio Code interface with a Terraform configuration file named `main.tf` and a terminal window.

main.tf:

```
1 terraform {  
2   required_providers {  
3     aws = {  
4       source = "hashicorp/aws"  
5       version = "4.45.0"  
6     }  
7   }  
8 }  
9  
10 provider "aws" {  
11   # Configuration options  
12   region = "us-east-1"  
13 }  
14 resource "aws_instance" "tf-ec2" {  
15   ami           = "ami-0c02fb55956c7d316"  
16   instance_type = "t2.micro"  
17   tags = {  
18     "Name" = "terraform"  
19   }  
20   count = 5  
21 }  
22 resource "aws_s3_bucket" "b" {  
23   bucket = "my-tf-test-bucket345353"  
24 }  
25  
26
```

Terminal:

```
12048@Savas MINGW64 ~/Desktop/amazon  
$ terraform state list  
aws_instance.tf-ec2[0]  
aws_instance.tf-ec2[1]  
aws_instance.tf-ec2[2]  
aws_instance.tf-ec2[3]  
aws_instance.tf-ec2[4]  
aws_s3_bucket.b  
  
12048@Savas MINGW64 ~/Desktop/amazon  
$
```