UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

Experiment No. 2.2

Student Name: Rishav Kumar
UID: 22MCC20039
Branch: MCA - CCD
Section/Group: MCD-1/ Grp A
Semester: IV
Date of Performance: 04th Mar 24
Subject Name: CI/CD Pipelines
Subject Code: 22CAP-781

1. Aim/Overview of the practical:

a. Create AWS free tier account

b. Install Terraform in local system

c. Initialize terraform and create 3 instances on AWS.

d. Execute terraform plan and apply the changes using terraform apply.

e. After performing step c., destroy all the instances.

2. Code for practical:

a. Create AWS free tier account:

- Go to the AWS Free Tier website: AWS Free Tier.
- Click on the "Create an AWS Account" button and follow the instructions to set up AWS account.

b. Install Terraform in the local system:

- Download the Terraform binary for your operating system from the official Terraform website.
- Extract the downloaded archive to a directory.
- Add the directory containing the Terraform binary to your system's PATH.

c. Initialize Terraform and create 3 instances on AWS:

- Create a new directory for your Terraform configuration.
- Inside the directory, create a file named **main.tf** with the following content

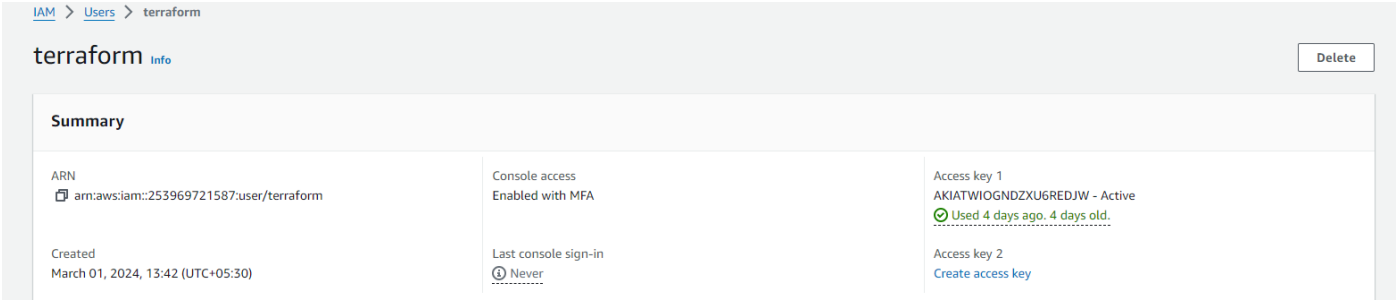
```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}
```

```
resource "aws_instance" "web" {
  ami           = "ami-00381a880aa48c6c6"
  instance_type = "t3.micro"

  tags = {
    Name = "Arkay"
  }
}

provider "aws" {
  region     = "eu-north-1"
  access_key = "AKIATWIOGNDZXU6REDJW"
  secret_key = "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"
}
```

- Go to IAM, and create a user and create access key and secret key.



- Open a terminal, navigate to the directory, and run the following commands:

terraform init

d. Execute Terraform plan and apply the changes:

terraform plan

```
ubuntu@ip-172-31-21-76:~/terraform$ 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.web will be created
  + resource "aws_instance" "web" {
      + ami                                  = "ami-00381a880aa48c6c6"
      + 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)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
```

terraform apply

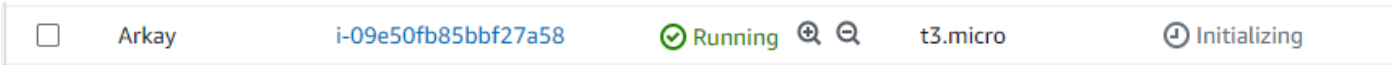
```
Plan: 1 to add, 0 to change, 0 to destroy.

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_instance.web: Creating...
aws_instance.web: Still creating... [10s elapsed]
aws_instance.web: Creation complete after 13s [id=i-09e50fb85bbf27a58]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-21-76:~/terraform$
```



e. Destroy all the instances:

terraform destroy

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

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_instance.web: Destroying... [id=i-09e50fb85bbf27a58]
aws_instance.web: Still destroying... [id=i-09e50fb85bbf27a58, 10s elapsed]
aws_instance.web: Still destroying... [id=i-09e50fb85bbf27a58, 20s elapsed]
aws_instance.web: Still destroying... [id=i-09e50fb85bbf27a58, 30s elapsed]
aws_instance.web: Still destroying... [id=i-09e50fb85bbf27a58, 40s elapsed]
aws_instance.web: Destruction complete after 40s

Destroy complete! Resources: 1 destroyed.
```