St. Francis Institute of Technology, Mumbai-400 103
Department of Information Technology

A.Y. 2025-2026
Class: TE-ITA/B, Semester: V

Experiment – 6: To understand and implement the basic lifecycle of Infrastructure as Code (IaC) and to Build, Apply, and Destroy an EC2 Instance on AWS using Terraform

Subject: **Advanced DevOps Lab**

1. **Aim:** To To understand and implement the basic lifecycle of Infrastructure as Code (IaC) using Terraform on Amazon Web Services (AWS)

2. **Objectives:** After study of this experiment, the student will be able to
   - Understand basic Terraform concepts
   - Write terraform scripts
   - Understand basic Terraform commands and concept of creating instance on EC2 using terraform.
3. **Lab objective mapped :** ITL504.3: To be familiarized with infrastructure as code for provisioning, compliance, and management of any cloud infrastructure and d service.
4. **Prerequisite:** Fundamentals of cloud computing and AWS account
5. **Requirements:** PC and Internet
6. **Pre-Experiment Exercise:**

Brief Theory:

Terraform

Terraform is a popular Infrastructure as Code (IaC) tool that enables cloud infrastructure to be described in configuration files and deployed consistently. In this experiment, we use Terraform to create, manage, and destroy an EC2 instance in AWS.

The Terraform workflow includes three main phases:

1. Build Phase:

Commands:

- terraform init: Initializes Terraform working directory and downloads the AWS provider plugin.
- terraform plan: Shows the execution plan and what resources will be created, without applying any changes.

Configuration Includes:

- Specifying the **provider** (AWS)
- Defining the **resource** (aws_instance)

- Setting required parameters (AMI ID, instance type, key pair, region, etc.)

---

2. Apply Phase:
Command:

- terraform apply: Executes the configuration and provisions the EC2 instance on AWS.

Terraform uses the configuration to interact with AWS APIs and launch the EC2 instance. The resource details are stored in the terraform.tfstate file, enabling Terraform to track resource changes.

---

3. Destroy Phase:
Command:

- terraform destroy: Removes all resources created by Terraform, in this case, the EC2 instance.

This phase is important to avoid incurring AWS charges for unused resources and to clean up after testing.

7. Laboratory Exercise

Step 1: First we will check that no instance is running on EC2.
Step 2: Create an IAM user with Programmatic Password, Administrator access and download access key and secret key from download.csv
IAM:
USERS:
ADD USER:
Give console access
I want to create user
custom pwd
Attach policy directly : Administrator Access
Review:
Create user
Clk on user name
Create access key
Command line interface
Create access key

Step 3: Now write a Terraform program in vs code, create new file with .tf extension

```
# Configure the AWS Provider
provider "aws" {
  access_key = ""
  secret_key = ""
  region     = "us-east-1"
```

```
}

# Launch an EC2 instance
resource "aws_instance" "terraformUser" {
  ami           = ""  # Replace with a region-specific valid AMI
  instance_type = "t3.micro"
  count = 3

  tags = {
Name = "praj"
}
}
```

In Launch instance, you will get ami : amazon machine image

For Instance type : t3.micro is freely available
Step 4: Now initialize the terraform …type c:\SfitApps> terraform init
Terraform has been initialized successfully.
Step 5: c:\sfitApps>terraform plan
Step 6: Check the instance on Ec2 before terraform apply
Instance is not yet created.
Step 7: Terraform apply
Step 8: Check terraform created instance on EC2…we have created 3 instances.
Step 9: Now destroy the instance from command prompt….c:\SfitApps> terraform destroy

8. **Post-Experiments Exercise**
   - **A. Extended Theory:**
     - Terraform Vs. Kubernetes (Soft copy)
     - Terraform Vs. Ansible (Soft copy)
     - How to create AWS S3 Bucket using Terraform? (Write only Terraform Code in hand)
   - **B. Questions:**(Soft copy)
     1. Name all version controls supported by Terraform.
     2. Name some major competitors of Terraform.
     3. Why is Terraform preferred as one of the DevOps tools?
   - **C. Conclusion:**
     A. Write what was performed in the experiment
     B. Mention few applications of what was studied.
     C. Write the significance of the studied topic
   - **D. References:**
     A. https://www.ibm.com/cloud/learn/terraform#toc-terraform--OoC-5III
     B. https://www.simplilearn.com/terraform-interview-questions-and-answers-article
     C. https://aws.amazon.com/microservices/
     D. https://www.monkeyvault.net/docker-vs-virtualization/
     E. https://cloudacademy.com/blog/docker-vs-virtualization/
     F. https://www.terraform.io/docs/language/values/variables.html