**Extra Lab – Single Script for Terraform and Ansible Together**

**Objective:**

**Single Script for Terraform and Ansible Together**

This script provisions an **AWS EC2 instance** using Terraform and then automatically configures it with **Ansible** to install **Apache**.

**Single Script (deploy.sh)**

Create a new file deploy.sh and add the following content. This script will:

1. **Initialize and apply Terraform** to create an EC2 instance.
2. **Generate an Ansible inventory file dynamically**.
3. **Run Ansible playbook** to install Apache on the instance.

#!/bin/bash

# Step 1: Initialize Terraform

terraform init

# Step 2: Apply Terraform to create infrastructure

terraform apply -auto-approve

# Step 3: Retrieve the instance IP from Terraform output

INSTANCE\_IP=$(terraform output -raw instance\_ip)

# Step 4: Create Ansible inventory file

cat <<EOF > inventory

[web]

$INSTANCE\_IP ansible\_user=ubuntu ansible\_ssh\_private\_key\_file=~/.ssh/my-key.pem

EOF

# Step 5: Run Ansible Playbook

ansible-playbook -i inventory playbook.yml

**Terraform Configuration (main.tf)**

This Terraform script provisions an **EC2 instance** and generates an **Ansible inventory file** dynamically.

provider "aws" {

region = "us-east-1"

}

resource "aws\_instance" "web" {

ami = "ami-0c55b159cbfafe1f0" # Replace with a valid Ubuntu AMI ID

instance\_type = "t2.micro"

key\_name = "my-key" # Replace with your key pair name

tags = {

Name = "Terraform-Ansible-Instance"

}

provisioner "local-exec" {

command = <<EOT

echo "[web]" > inventory

echo "${self.public\_ip} ansible\_user=ubuntu ansible\_ssh\_private\_key\_file=~/.ssh/my-key.pem" >> inventory

EOT

}

}

output "instance\_ip" {

value = aws\_instance.web.public\_ip

}

**Ansible Playbook (playbook.yml)**

This playbook installs **Apache** on the EC2 instance.

- name: Configure Web Server

hosts: web

become: yes

tasks:

- name: Update packages

apt:

update\_cache: yes

- name: Install Apache

apt:

name: apache2

state: present

- name: Start Apache service

service:

name: apache2

state: started

enabled: yes

**Execution Steps**

1. Ensure you have Terraform, Ansible, and AWS CLI installed.
2. Save all files (deploy.sh, main.tf, playbook.yml) in the same directory.
3. Make the script executable:

chmod +x deploy.sh

1. Run the script to provision the infrastructure and configure it:

./deploy.sh

**Verification**

* Connect to the instance manually:

ssh -i ~/.ssh/my-key.pem ubuntu@$(terraform output -raw instance\_ip)

* Check if Apache is running:

systemctl status apache2