Activity 1: Deploying an Apache Server on AWS EC2 Using Terraform

Objective: The objective of this activity is to familiarize students with Terraform by creating an AWS EC2 instance and setting up an Apache2 web server using a user data script. Students will also create a security group to allow HTTP access to the application.

Instructions:

1.Prerequisites:

- Have access to an AWS account with the necessary permissions to create EC2 instances, security groups, and other required resources. For that you will need to create an IAM user with the policy mentioned in the file `iam_terraform_policy.json` present in the labDirectory folder. Keep the `Access key ID` and `Secret access key` of the IAM user with you for future reference.

2. **Specification**

a. **Security Group Specification:**

- Attach the security group to any VPC which will allow deploying EC2 and allow traffic for the application(you can also use the **default** VPC)
- Ingress Rule:
  - Allow inbound HTTP traffic on port 80 from all IPv4 (`0.0.0.0/0`) addresses
- Egress Rule:
  - Allow all outbound traffic to all IPv4 addresses.
- Tag the security group with a Name tag of `TF_SG`.

a. **EC2 Instance Specification:**

- The operating system should be **Ubuntu Server 24.04 LTS (HVM)**
- The instance type should be **t2.micro**
- Attach a security group to the EC2 instance using the ID of the security group created in the Terraform code.
- Use the `install_apache.sh` script as the `user_data` to configure the instance during initialization.
- Tag the EC2 instance with a Name tag of `example-ec2`.

3. activity Setup:

- **Content of install_apache.sh:** The following script present in the `install_apache.sh` file helps us to install apache 2 in the EC2 instance:

```
Unset
#! /bin/bash
sudo apt-get update
sudo apt-get install -y apache2
sudo systemctl start apache2
sudo systemctl enable apache2
echo "<h1>Welcome : Apache installed with the help of user_data
and Terraform</h1>" | sudo tee /var/www/html/index.html
```

- **Content of Terraform Files:** You need to write the content for the following Terraform files:

  a. **`main.tf`:** Define the EC2 instance and security group.
    - Use the `install_apache.sh` script present in labDirectory as the `user_data` for the EC2 instance using the filename of the script.

  b. **`terraform.tfvars`:** Assign values to the variables defined in `variables.tf`.

  c. **`outputs.tf`:** Define outputs to display the public IP, Instance ID of the EC2 instance and Security Group ID of the Security Group after deployment. For

    A. Instance ID of the EC2 instance use variable name “instance_id”
    B. Public IP Address of the EC2 instance use variable name “public-ip-address”
    C. Security group ID use variable name “securitygroup”

  d. **`provider.tf`:** Configure the AWS provider with the appropriate region and credentials.

  e. **`variables.tf`:** Define all the variables here. For

    A. Access key ID use variable name “access_key_value”
    B. Secret Access Key use variable name “secret_key_value”
    C. Region use variable name “region_value”
    D. AMI ID use variable name “ami_id_value”

E. Instance Type use variable name “instance_type_value”
F. VPC ID use variable name “vpc_id_value”

- **Steps to Complete the activity:**

a. Initialize Terraform: Run the command `terraform init` inside labDirectory to initialize Terraform.

b. Plan the Terraform deployment: Execute `terraform plan` to preview the changes that Terraform will make.

c. Apply the Terraform deployment: Run `terraform apply` and confirm the action when prompted. This will create the EC2 instance and security group.

- **Verification:**

  - Once the EC2 instance is running, open a web browser and enter the public IP address of the instance. You should see a webpage with the message: **"Welcome: Apache installed with the help of user_data and Terraform"**.
  - To find the public IP, you can:
    - Check the output displayed after running `terraform apply`.
    - Go to the AWS Management Console, navigate to the EC2 dashboard, and locate your instance.

- **Clean-Up:**

➢ To avoid incurring charges after completing the activity:
➢ Run **terraform destroy** inside labDirectory and confirm the action.

- Evaluation:

After **terraform destroy** click on Evaluate. It may take 1-2 minutes for evaluation.