Lab Exercise: Infrastructure as Code (IaC) with Terraform

Objective:

This lab will introduce you to the basic tasks involved in using **Terraform** to define and provision infrastructure as code (IaC). You will learn how to install Terraform, create a configuration file, and use Terraform to deploy infrastructure to a cloud provider (e.g., AWS). By the end of this lab, you'll understand how to define, plan, and apply changes to infrastructure using Terraform.

Step 1: Install Terraform

1. **Download Terraform**:
   - Go to the Terraform website and download the latest version for your operating system.
2. **Install Terraform**:
   - Follow the installation instructions for your operating system:
     - **Windows**: Extract the terraform.exe file and add it to your system's PATH.
     - **Linux/Mac**: Extract the file and move it to /usr/local/bin/.
3. **Verify the installation**:
   - Open a terminal and run the following command:

     ```
     terraform --version
     ```

   - You should see the installed version of Terraform.

Step 2: Set Up AWS Credentials

1. **Create an AWS IAM User**:
   - Log in to the AWS Management Console.
   - Navigate to **IAM** > **Users** and create a new user with programmatic access.
     - Go to the IAM dashboard

- Click Users
- Click Create User
- User name: TerraformDemo
- Click **Next**
- Select **Attach policies directly**
- Select the **AdministratorAccess** policy
- Click **Next**
- Click **Create User**
- Click on the user you just created
- Click **Security Credentials**
- Click **Create access key**
    1. Select **Command Line Interface**
    2. Click **Next**
    3. Description tag value: **TerraformDemo**
    4. Click **Create access key**
    5. Copy the Access key and Secret access key and paste them into a notepad file.

2. **Configure AWS credentials**:
    - On your local machine, configure the AWS credentials using the AWS CLI:

      ```
      aws configure
      ```

    - Enter the **Access Key ID** and **Secret Access Key** for the IAM user you created, along with the preferred region (e.g., us-east-1).

Step 3: Write a Basic Terraform Configuration

1. **Create a new directory**:
    - Create a directory to store your Terraform configuration files:

      ```
      mkdir terraform-lab
      cd terraform-lab
      ```

2. Open notepad and save the file as main.tf in the terraform-lab directory
3. Add the following configuration to main.tf to define an **AWS EC2 instance**:

      ```
      provider "aws" {
        region = "us-east-1"
      }

      resource "aws_instance" "example" {
      ```

```
  ami           = "ami-0c55b159cbfafe1f0"  # Amazon Linux 2 AMI
  instance_type = "t2.micro"

  tags = {
    Name = "Terraform-Example"
  }
}
```

- This configuration does the following:
  - Defines **AWS** as the provider and sets the region to us-east-1.
  - Creates an **EC2 instance** with the specified Amazon Machine Image (AMI) and instance type.

4. **Save the file** and exit the editor.

Step 4: Initialize Terraform

1. Initialize the Terraform project
   - Copy terraform.exe to the terraform-lab directory

```
terraform init
```

2. Terraform will download the necessary provider plugins (in this case, AWS) and prepare your environment for provisioning infrastructure.

Step 5: Plan and Apply the Terraform Configuration

1. **Plan the infrastructure changes**:
   - Run the following command to see what resources Terraform will create:

```
terraform plan
```

   - Terraform will show a detailed plan of the resources it intends to create, without making any actual changes yet. Verify that an EC2 instance will be created.

2. **Apply the configuration**:
   - To provision the infrastructure, run the following command:

```
terraform apply
```

   - Review the plan and type yes to confirm the changes.

3. **Verify the deployment**:
    - Once Terraform completes the apply process, log in to the **AWS Management Console**.
    - Navigate to **EC2** > **Instances** and verify that the instance has been created.

Step 6: Modify the Configuration

1. **Change the instance type**:
    - Open the main.tf file again and modify the instance_type to t2.small:

      ```
      instance_type = "t2.small"
      ```

2. **Apply the changes**:
    - Run the following commands to apply the changes:

      ```
      terraform plan
      terraform apply
      ```

    - Terraform will detect that the instance type has changed and update the instance accordingly.

Step 7: Destroy the Infrastructure

1. **Clean up the resources**:
    - Once you're done, it's important to destroy the infrastructure to avoid unnecessary costs.
    - Run the following command to destroy all the resources created by Terraform:

      ```
      terraform destroy
      ```

    - Review the plan and type yes to confirm the destruction.
2. You should also delete the IAM user that you created in step 2 of this lab.