Create AWS resources via Terraform Challenge

By the end of this challenge, you should be able to spin up the following resources with the following specifications:

1. **VPC**
    a. Name of VPC: <your-name>-tf-vpc e.g. luqman-tf-vpc
    b. Choose an appropriate CIDR range e.g. 10.0.0.0/16
2. **Subnets**
    a. Choose appropriate CIDR ranges for the below 4 subnets
    b. Public Subnet 1 in AZ1
        i. Name of Subnet: <your-name>-tf-public-subnet-az1
        ii. Choose AZ as us-east-1a / us-east-1d
    c. Public Subnet 2 in AZ2
        i. Name of Subnet: <your-name>-tf-public-subnet-az2
        ii. Choose AZ as us-east-1b / us-east-1e
    d. Private Subnet 1 in AZ1
        i. Name of Subnet: <your-name>-tf-private-subnet-az1
        ii. Choose AZ as us-east-1a / us-east-1d
    e. Private Subnet 2 in AZ2
        i. Name of Subnet: <your-name>-tf-private-subnet-az2
        ii. Choose AZ as us-east-1b / us-east-1e
3. **Internet Gateway:**
    a. Name of IGW: <your-name>-tf-igw
4. **VPC Endpoint S3:**
    a. Name of VPCE-S3: <your-name>-tf-vpce-s3
5. **Route Tables:**
    a. Public Route Table for all Public Subnets
        i. Name of RTB: <your-name>-tf-public-rtb
    b. Private Route Table 1 for Private Subnet 1
        i. Name of RTB: <your-name>-tf-private-rtb-az1
    c. Private Route Table 2 for Private Subnet 2
        i. Name of RTB: <your-name>-tf-private-rtb-az2
6. **Security Group:**
    a. Allow the following ingress:
        i. HTTP from Anywhere

    ii. HTTPS from Anywhere
    iii. SSH from Anywhere
  b. Allow the following egress:
    i. All protocols from Anywhere
  c. Name of SG: <your-name>-tf-sg-allow-ssh-http-https
7. **EC2:**
  a. AMI: Amazon Linux 2023
  b. Enable Public IP
  c. VPC: Your newly created VPC from (1)
  d. Subnet: One of the public subnets from (2)
  e. Key Pair: A previously created key pair you own
  f. Security Group: From (6)
  g. Instance Type: t2.micro
  h. Name: <your-name>-tf-ec2

References for you:
VPC: https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/vpc
Subnet: https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/subnet
Route Tables: https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/route_table
Route Table Associations: https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/route_table_association
VPC Endpoint: https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/vpc_endpoint
EC2: https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/instance
Security Group: https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/security_group

🙂 Have Fun 🙂

Additional Challenge #1

Instead of using an existing key pair from the AWS console that you previously created, your first challenge today would be to create an EC2 key pair using Terraform, which downloads the key pair to your local machine for you to use to connect to the EC2 later on.

The outcome would be to successfully connect to your EC2 using a new key pair.

Update your existing code to achieve this.

Additional Challenge #2

Recall the activity where we were creating EC2 with a User Data script / bootstrap script.

Your second challenge would be to update the previously created EC2 with a user data script passed in. The outcome would be to convert your EC2 into a HTTPD web server.

The content of the script would be:

```
#!/bin/bash
yum update -y
yum install httpd -y
yum install docker -y
```

Upon successful creation of the EC2, you should see the 2 packages installed on your VM after SSH.

Verify that you are able to access the index.html from your browser. Follow this guide for information: https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-centos-7

Additional Challenge #3

We will be looking at Terraform modules soon.

Find out more about Terraform modules via this link: https://registry.terraform.io/namespaces/terraform-aws-modules and you can also refer to an existing Github repository with simple sample code: https://github.com/luqmannnn/simple-terraform-module

Your third challenge would be to convert the EC2, VPC, Subnets and other resources previously created in the main challenge from **Resources into Modules**.

Give it a try!