**Overview**

In this assignment, you’ll design and implement a Terraform configuration that provisions a simple, secure AWS network and compute environment:

1. **VPC & Subnets**
2. **Security Groups**
3. **RDS MySQL**
4. **EC2 Instance**

You will demonstrate your understanding of Terraform modules, resource dependencies, and AWS best practices for isolating public and private resources.

**Objectives**

* **Network Layer**
  + Create a custom VPC and both public & private subnets across two Availability Zones (AZs).
* **Security Layer**
  + Define and attach security groups to control inbound traffic.
* **Data Layer**
  + Launch a MySQL RDS instance inside private subnets using the default VPC security group.
* **Compute Layer**
  + Launch a Linux EC2 instance in a public subnet, secured by two security groups (custom + default).

**Prerequisites**

* AWS account with appropriate IAM permissions for VPC, EC2, RDS, Security Groups, Subnets, Internet Gateway, etc.
* Terraform ≥ 1.0 installed and configured (terraform init working).
* AWS CLI configured with default credentials.

**Tasks**

1. **Module Structure**
   * Organize your code into modules (vpc, security-groups, rds, ec2) under a root configuration.
   * Each module should have its own main.tf, variables.tf, and outputs.tf.
2. **VPC Module**
   * Create a VPC with a customizable CIDR (e.g., 10.0.0.0/16 via variable).
   * Define two public subnets and two private subnets, each pair spread across two AZs (e.g., ap-south-1a, ap-south-1b).
3. **Security Groups Module**
   * **Custom SG** (sg\_public\_ec2):
     + Inbound rules:
       - SSH (TCP/22) from 0.0.0.0/0
       - HTTP (TCP/80) from 0.0.0.0/0
     + Outbound: allow all.
   * **Default SG**: use the VPC’s default security group (no extra rules).
4. **RDS Module**
   * Launch an **RDS MySQL** instance:
     + Engine version: latest MySQL LTS.
     + Instance class: db.t3.micro.
     + Storage: 20 GB.
     + Subnet group referencing your private subnets.
     + **Security group:** attach only the VPC’s **default** security group.
   * No public accessibility.
5. **EC2 Module**
   * Launch an **EC2 Linux** instance (e.g., Amazon Linux 2) in one of your **public** subnets.
   * Attach **two** security groups:
     + The **custom SG** (sg\_public\_ec2).
     + The **default SG** of the VPC.
   * Assign a public IP.
   * Use a key pair variable for SSH access.
6. **Outputs & Documentation**
   * Output the following after terraform apply:
     + VPC ID
     + Public subnet IDs
     + Private subnet IDs
     + RDS endpoint
     + EC2 public IP
   * Include a **README.md** with:
     + Setup instructions (terraform init, plan, apply)
     + How to verify:
       - SSH into EC2 (ssh -i <key> ec2-user@<public-ip>)
       - From EC2, connect to the RDS endpoint via the MySQL client.

**Deliverables**

* A Zip file with your name containing:
  + Root Terraform files (main.tf, variables.tf, outputs.tf, backend.tf if used).
  + README.md with clear execution & verification steps.