**Case Study-4:**

**Problem Statement:** Client wants to provision an AWS S3 to provide **shared storage** across 3 AWS EC2 instances. The EC2 instances must exist in the same VPC with each instance in separate availability zone. Implement a solution to ensure that all the 3 EC2 instances can **mount the S3 bucket**. All these EC2 instances have **docker installed** on it. Use appropriate IaC (Infrastructure as code) tool to develop AWS infrastructure.

SOLUTION:

Terraform script automates the setup of an AWS infrastructure, creating a VPC, subnets, EC2 instances, S3 bucket, IAM roles, and security configurations.

1. VPC and Networking  
   It creates a VPC, an Internet Gateway (gw) for external access, and three subnets (az\_subnets) across availability zones. A route table (rt) ensures internet access, and subnets are linked to it.
2. Security Group  
   A security group (instance\_sg) allows SSH (port 22) access from anywhere while permitting outbound traffic to all destinations.
3. S3 Bucket and Access Control  
   It provisions an S3 bucket (shared\_bucket), ensuring instances have full access using IAM Role (ec2\_s3\_access). This role allows EC2 instances to read/write from S3.
4. EC2 Instances Setup  
   It launches three EC2 instances (ec2), assigning them subnets and IAM profiles for secure S3 interaction. The instances run a user-data script that:
   * Installs necessary packages like s3fs for S3 access
   * Mounts S3 to /mnt/s3bucket for data storage
   * Installs Docker, Git, and Unzip for additional capabilities

SCREENSHOT:

