Problem statement:

This Terraform project deploys a highly available 3-tier web application architecture on AWS, using modules for better organization. It includes Amazon EC2 instances, Auto Scaling, Elastic Load Balancing, and a Virtual Private Cloud (VPC).

❖ Terraform Infrastructure setup:

1. Backend Configuration (terraform block):

- Configure the Terraform backend to store state information in an S3 bucket named "my-terraform-bucket2912."
- Specify the S3 bucket key and region for state storage.
- Use a DynamoDB table named "my-DynamoDB-table2912" for state locking and consistency.

2. Modules (module blocks):

- Define and configure modules "EC2-instance" and "VPC," each with its specific configuration files (main.tf, variables.tf, outputs.tf) and variables.

3. VPC and Internet Gateway:

- Create a Virtual Private Cloud (VPC) named "my_vpc" with a specified CIDR block, DNS support, and DNS hostnames settings.
- Create an internet gateway and attach it to the VPC for internet connectivity.

4. Subnets:

- Create public and private subnets across multiple availability zones within the VPC.
- Ensure that public subnets have `map_public_ip_on_launch` set to true for direct internet access.

5. Security Group (aws_security_group):

- Create a security group named "my-tf-sg" to allow incoming traffic on ports 22 (SSH), 80 (HTTP), and 443 (HTTPS) from any source (0.0.0.0/0).

6. Elastic IP (aws_eip) and NAT Gateway (aws_nat_gateway):

- Create an Elastic IP and associate it with the VPC.

- Create a NAT gateway that uses the Elastic IP for outbound internet access from private subnets.

7. Route Tables (aws_route_table and aws_route_table_association):

- Create route tables for public and private subnets.
- Associate the public subnets with a route table that routes traffic through the internet gateway.
- Associate the private subnets with a route table for routing through the NAT gateway.

8. Application Load Balancer (ALB) and Target Group:

- Create an Application Load Balancer (ALB) named "my-tf-alb"
- Ensure that it is not an internal ALB and is of type "application"
- Associate the ALB with the subnets defined for private application subnets.
- Create a target group named "my-tf-tg" to route traffic to targets on port 80.

9. EC2 Instance (aws_instance):

- Create an Amazon EC2 instance using the specified AMI and instance type.
- Associate the instance with a key pair named "terraform-key" created using aws_key_pair.
- Use user data from the "userdata.sh" script.
- Tag the instance with the name "mt-tf-instance."

10. Key Pair (aws_key_pair):

- Create an AWS key pair named "terraform-key" for secure SSH access to the EC2 instance.