**Task Description:**

As part of your application process, we would like you to complete a practical task that

simulates a common scenario in a DevOps role involving multiple remote machines. Please

follow the instructions below and submit your solution within the provided time frame.

**Scenario:**

You have been assigned to set up a private network for multiple remote machines that need

to communicate with each other securely. Your task is to:

Set Up a Private Network:

1. Use AWS VPC (Virtual Private Cloud) to create a private network.

2. Ensure that the network is secure and isolated from the public internet.

3. Create subnets and configure routing tables as needed.

4. Provision Remote Machines:

5. Launch two EC2 instances within the VPC:One instance for a web application (e.g.,

a Node.js or Python Flask application).

6. One instance for a PostgreSQL database.

7. Ensure that the instances are in the same private subnet and can communicate with

each other.

**Application and Database Configuration:**

1. Install and configure the web application on the first instance.

2. Install and configure PostgreSQL on the second instance.

3. Ensure the web application can connect to the PostgreSQL database.

4. Web application is routed by ALB and setup a dns for your application.

**Secure Communication:**

1. Implement security groups and network ACLs to allow only necessary traffic between

the instances.

2. Set up a bastion host to allow secure SSH access to the instances from outside the

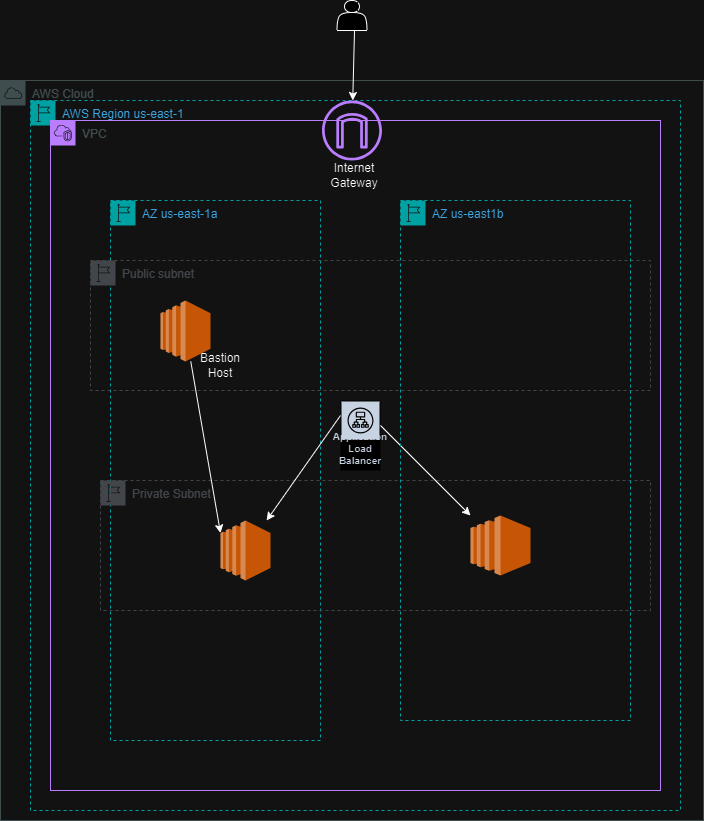
VPC.

**Automate the Setup:**

1. Use Terraform to automate the setup of the VPC, subnets, security groups, and EC2

instances.

2. Provide a Terraform configuration file that can be used to recreate the entire setup.



**Prerequisites**

1. An AWS account with secret access key
2. Knowledge of AWS Management Console
3. Knowledge of EC2 instances, VPC structure, ALB, Security Group,database, etc.
4. Knowledge of Linux, SSH, and Shell Scripting
5. Access to the command-line tool.

**AWS Authentication**

Run the command in the VS Code terminal to configure programmatic access.

aws configure

AWS Access Key ID:

Secret Access Key:

Default region name: **us-east-1**

**Clone Repo**

Open your chosen terminal or code editor. I’ll be using Visual Studio Code.

git clone https://github.com/Rahulpandya11/Terraform\_Task.git

To set up the working directory, use**terraform init**.

**The Terraform Plan** shows the execution strategy and the modifications that Terraform intends to make.

The plan presented in Terraform Plan is put into action via **Terraform Apply.**

There will be a prompt; enter “yes.”

**Clear up**

**Terraform**destroy  
all produced resources.

Where it prompts a response, enter yes