**Task 3:**

Write a script to automate infrastructure provisioning on a cloud platform (e.g., AWS, Azure, or GCP).

This script should create virtual machines, configure networking, and install necessary software.

**Solution:**

Below I am trying to provision the infra using AWS provider.

**# main.tf**

|  |
| --- |
| # AWS provider  provider "aws" {  region = "us-east-1"  }  # Create an EC2 instance  resource "aws\_instance" "example" {  ami = "ami-11111111" # AMI ID  instance\_type = "t2.micro"    tags = {  Name = "ExampleInstance"  }  }  # Create a security group allowing SSH access  resource "aws\_security\_group" "example" {  name = "example\_sg"  description = "Allow SSH inbound traffic"  ingress {  from\_port = 22  to\_port = 22  protocol = "tcp"  cidr\_blocks = ["0.0.0.0/0"]  }  tags = {  Name = "ExampleSecurityGroup"  }  } |

Need to run below commands in tf CLI

***terraform init # Initialize Terraform in the directory***

***terraform plan # Generate an execution plan***

***terraform apply # Apply the changes to create resources on AWS***

To install necessary software, need to use provisioners in Terraform or use a configuration management tool like Ansible.

Below is an example of using the **remote-exec** provisioner in Terraform to execute commands on the EC2 instance

|  |
| --- |
| # Add this block to main.tf to install software using remote-exec provisioner  provisioner "remote-exec" {  inline = [  "sudo apt-get update",  "sudo apt-get install -y <package\_name>", # Replace <package\_name> with the desired software package  ]  connection {  type = "ssh"  user = "ec2-user"  private\_key = file("<path\_to\_private\_key\_file>")  host = aws\_instance.example.public\_ip  }  } |