Steps to perform:

**Step 1: Create the main.tf File**

This file contains the main Terraform configuration.

**Step 2: Create the variables.tf File**

This file defines input variables for the configuration.

**Step 3: Create the outputs.tf File**

This file defines outputs to display after deployment.

**Step 4: Create the README.md File**

This file provides instructions for using the Terraform configuration.

**Usage**

1. **Clone the Repository**

git clone <repository-url>

cd terraform-eks-cluster

1. **Initialize Terraform** Initialize Terraform to download necessary modules and providers:

terraform init

1. **Set Variables** Update variables.tf with your SSH key name (key\_name) or provide it as a CLI argument.
2. **Plan the Infrastructure** Review the changes Terraform will make:

terraform plan -var "key\_name=<your-ssh-key>"

1. **Apply the Configuration** Deploy the infrastructure:

terraform apply -var "key\_name=<your-ssh-key>"

1. **Access the Cluster** Configure your kubectl to use the EKS cluster:

aws eks --region <your-region> update-kubeconfig --name eks-cluster

1. **Destroy the Infrastructure** To delete all resources:

terraform destroy -var "key\_name=<your-ssh-key>"

**Notes**

* Ensure the SSH key (key\_name) exists in your AWS account before applying the configuration.
* Terraform will output the cluster details, including the endpoint and kubeconfig.

---

### \*\*Step 5: Initialize and Deploy\*\*

Run the following commands to deploy the infrastructure:

terraform init

terraform plan -var "key\_name=<your-ssh-key>"

terraform apply -var "key\_name=<your-ssh-key>"

**Step 6: Push to a Public Repository**

1. Initialize a Git repository:

git init

git add .

git commit -m "Initial commit for Terraform EKS cluster"

git remote add origin <repository-url>

git push -u origin main

1. repository link shared -