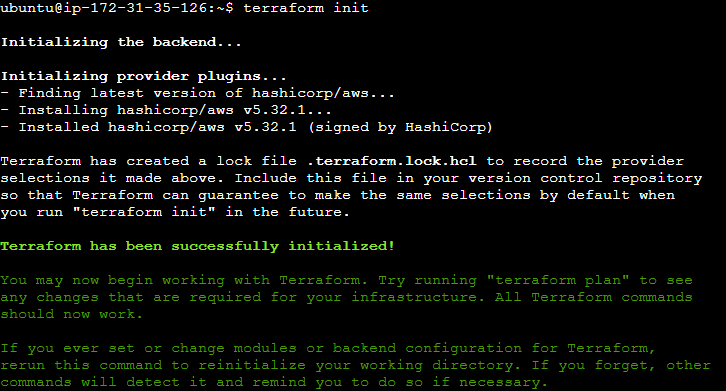
**Terraform Case Study**

1. Install Terraform in your Linux Machine (visit hashicorp site for installation method)
2. Once installed, build the code as per the requirement in main.tf file and push ‘terraform init’ command.



1. After the init, push terraform plan followed by terraform apply to execute the code.
2. Find the main.tf file content below :

# Configure the AWS provider

provider "aws" {

region = "ap-south-1"

access\_key = ""

secret\_key = ""

}

# Create a VPC

resource "aws\_vpc" "my\_vpc" {

cidr\_block = "10.0.0.0/16"

enable\_dns\_support = true

enable\_dns\_hostnames = true

tags = {

Name = "my-vpc"

}

}

# Create two subnets in the VPC

resource "aws\_subnet" "subnet\_a" {

vpc\_id = aws\_vpc.my\_vpc.id

cidr\_block = "10.0.1.0/24"

availability\_zone = "us-east-1a"

map\_public\_ip\_on\_launch = true

tags = {

Name = "subnet-a"

}

}

resource "aws\_subnet" "subnet\_b" {

vpc\_id = aws\_vpc.my\_vpc.id

cidr\_block = "10.0.2.0/24"

availability\_zone = "us-east-1b"

map\_public\_ip\_on\_launch = true

tags = {

Name = "subnet-b"

}

}

# Create an Internet Gateway

resource "aws\_internet\_gateway" "my\_igw" {

vpc\_id = aws\_vpc.my\_vpc.id

tags = {

Name = "my-igw"

}

}

# Create a security group allowing inbound traffic on port 80

resource "aws\_security\_group" "my\_security\_group" {

name = "my-security-group"

description = "Allow inbound traffic on port 80"

vpc\_id = aws\_vpc.my\_vpc.id

ingress {

from\_port = 80

to\_port = 80

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

tags = {

Name = "my-security-group"

}

}

# Create a network interface with a private IP address

resource "aws\_network\_interface" "my\_network\_interface" {

subnet\_id = aws\_subnet.subnet\_a.id

security\_group\_ids = [aws\_security\_group.my\_security\_group.id]

}

# Launch an EC2 instance in each subnet

resource "aws\_instance" "instance\_a" {

ami = "ami-0c55b159cbfafe1f0" # Ubuntu 20.04 LTS AMI ID

instance\_type = "t2.micro"

subnet\_id = aws\_subnet.subnet\_a.id

key\_name = "MumbaiKeyPair"

network\_interface\_ids = [aws\_network\_interface.my\_network\_interface.id]

}

resource "aws\_instance" "instance\_b" {

ami = "ami-0c55b159cbfafe1f0" # Ubuntu 20.04 LTS AMI ID

instance\_type = "t2.micro"

subnet\_id = aws\_subnet.subnet\_b.id

key\_name = "MumbaiKeyPair"

network\_interface\_ids = [aws\_network\_interface.my\_network\_interface.id]

}

1. The above code should be able to create the infrastructure by running terraform apply command.