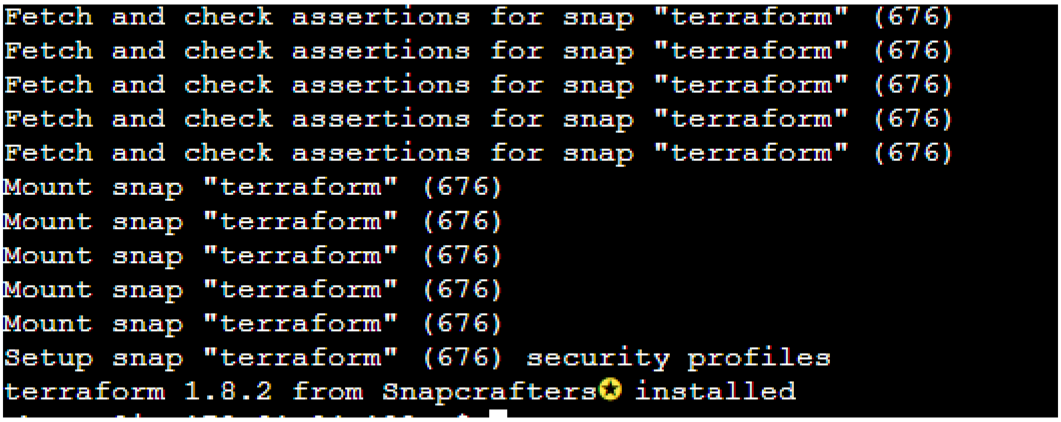
Name: Z.MD SHARUF RAJ
Batch: D015

Task-28:
Question: Launch an ec2 instance under a default subnet and VPC using terraform template

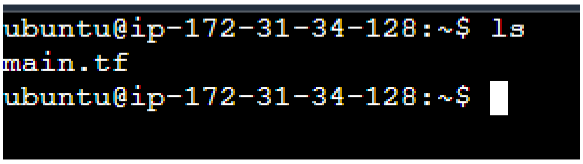
Solution:

1. Installing Terraform:



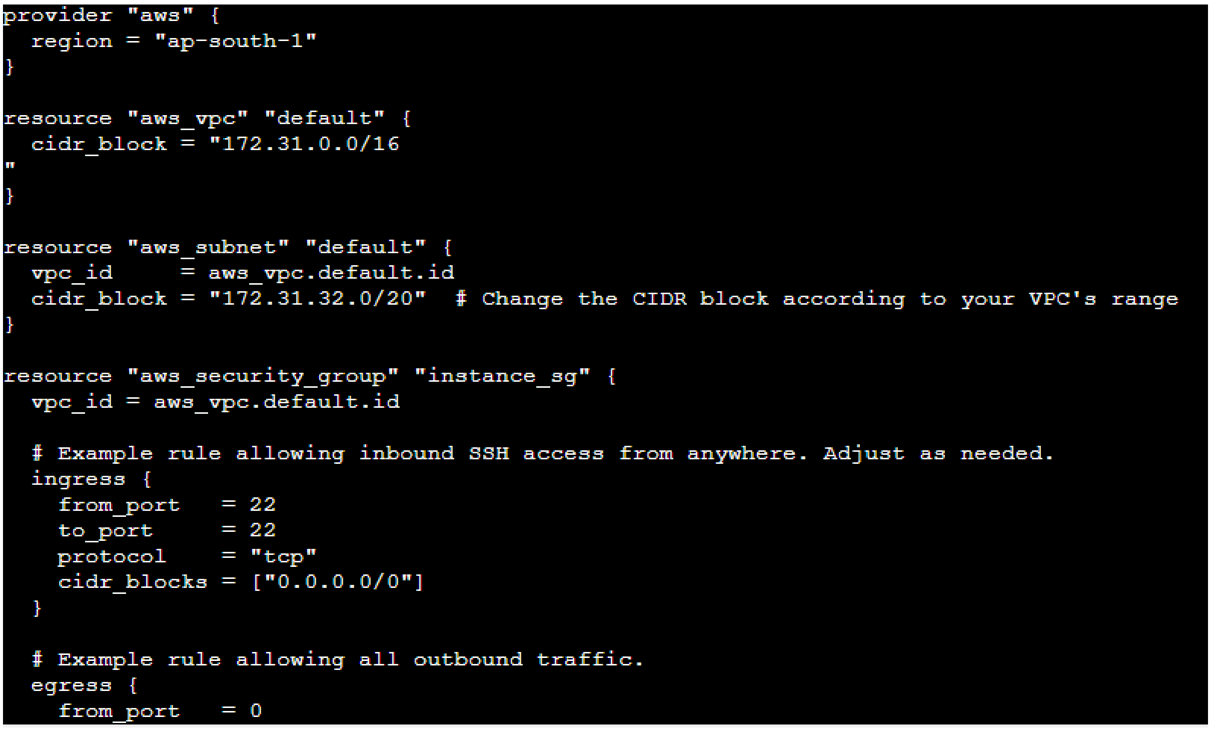
```
Fetch and check assertions for snap "terraform" (676)
Fetch and check assertions for snap "terraform" (676)
Fetch and check assertions for snap "terraform" (676)
Fetch and check assertions for snap "terraform" (676)
Fetch and check assertions for snap "terraform" (676)
Mount snap "terraform" (676)
Mount snap "terraform" (676)
Mount snap "terraform" (676)
Mount snap "terraform" (676)
Mount snap "terraform" (676)
Setup snap "terraform" (676) security profiles
terraform 1.8.2 from Snapcrafters✪ installed
```

2.Creating main.tf:



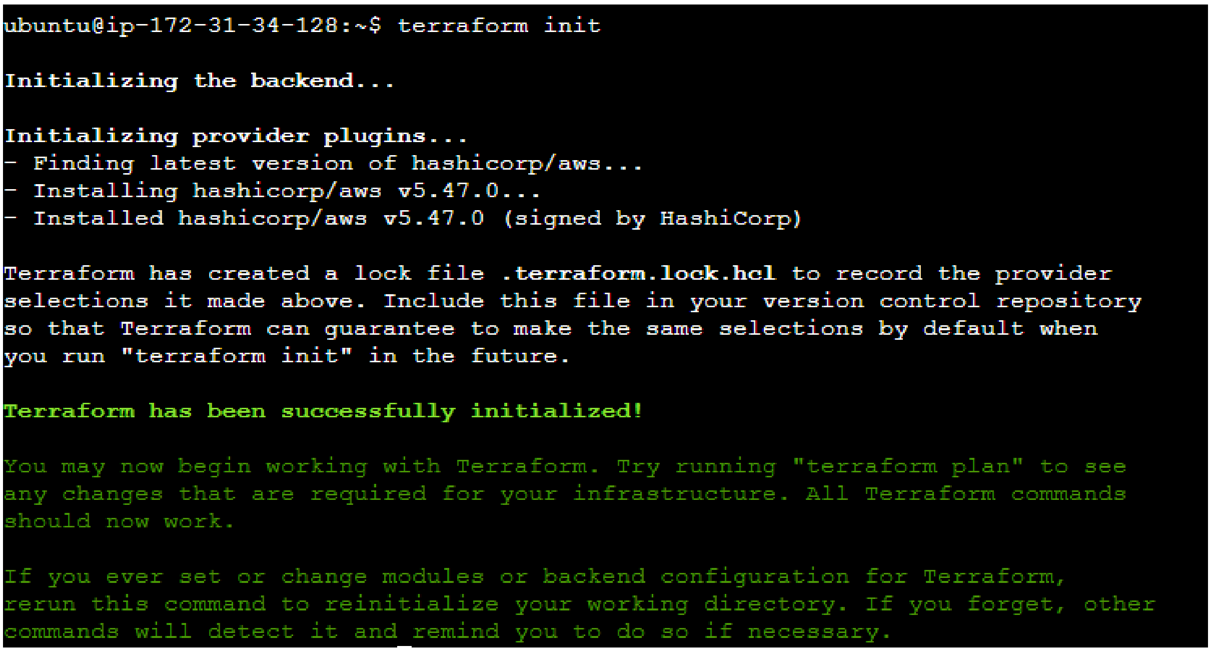
```
ubuntu@ip-172-31-34-128:~$ ls
main.tf
ubuntu@ip-172-31-34-128:~$ █
```

3.Editing the main.tf file:



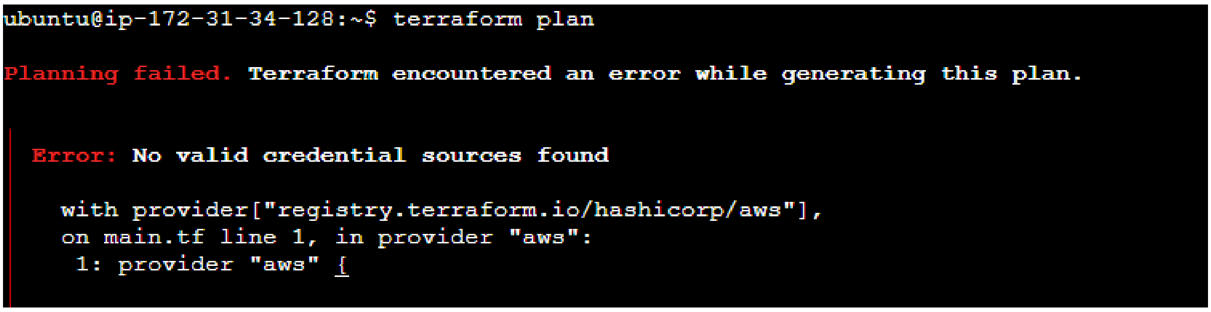
```
provider "aws" {
  region = "ap-south-1"
}

resource "aws_vpc" "default" {
  cidr_block = "172.31.0.0/16
"
}

resource "aws_subnet" "default" {
  vpc_id     = aws_vpc.default.id
  cidr_block = "172.31.32.0/20"  # Change the CIDR block according to your VPC's range
}

resource "aws_security_group" "instance_sg" {
  vpc_id = aws_vpc.default.id

  # Example rule allowing inbound SSH access from anywhere. Adjust as needed.
  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  # Example rule allowing all outbound traffic.
  egress {
    from_port   = 0
```

4.Doing terraform init:



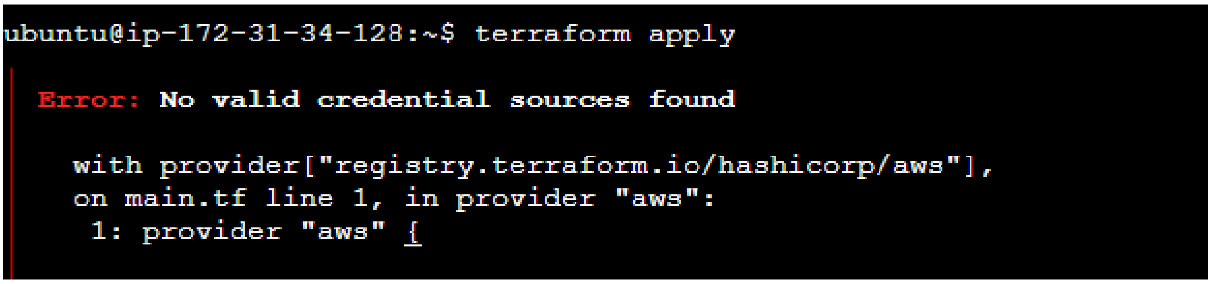
```
ubuntu@ip-172-31-34-128:~$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.47.0...
- Installed hashicorp/aws v5.47.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

5. Terraform Plan:



```
ubuntu@ip-172-31-34-128:~$ terraform plan

Planning failed. Terraform encountered an error while generating this plan.


  Error: No valid credential sources found

    with provider["registry.terraform.io/hashicorp/aws"],
    on main.tf line 1, in provider "aws":
     1: provider "aws" {
```

6.Terraform Apply:



```
ubuntu@ip-172-31-34-128:~$ terraform apply

  Error: No valid credential sources found

    with provider["registry.terraform.io/hashicorp/aws"],
    on main.tf line 1, in provider "aws":
     1: provider "aws" {
```