Name: Z.MD SHARUF RAJ
Batch: D015

Task-23: Write Terraform script to create highly available infrastructure in AWS. The infra should have1 vpc, 3 subnets setup in 3 different az and 2 instances setup in 2 different subnets.

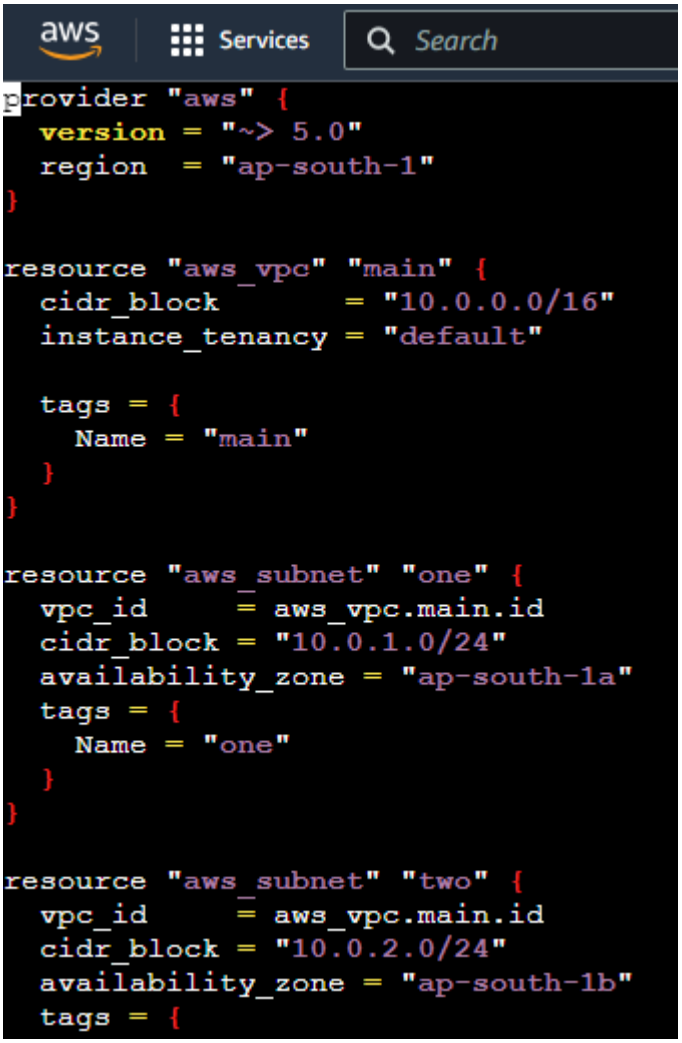
Solution:

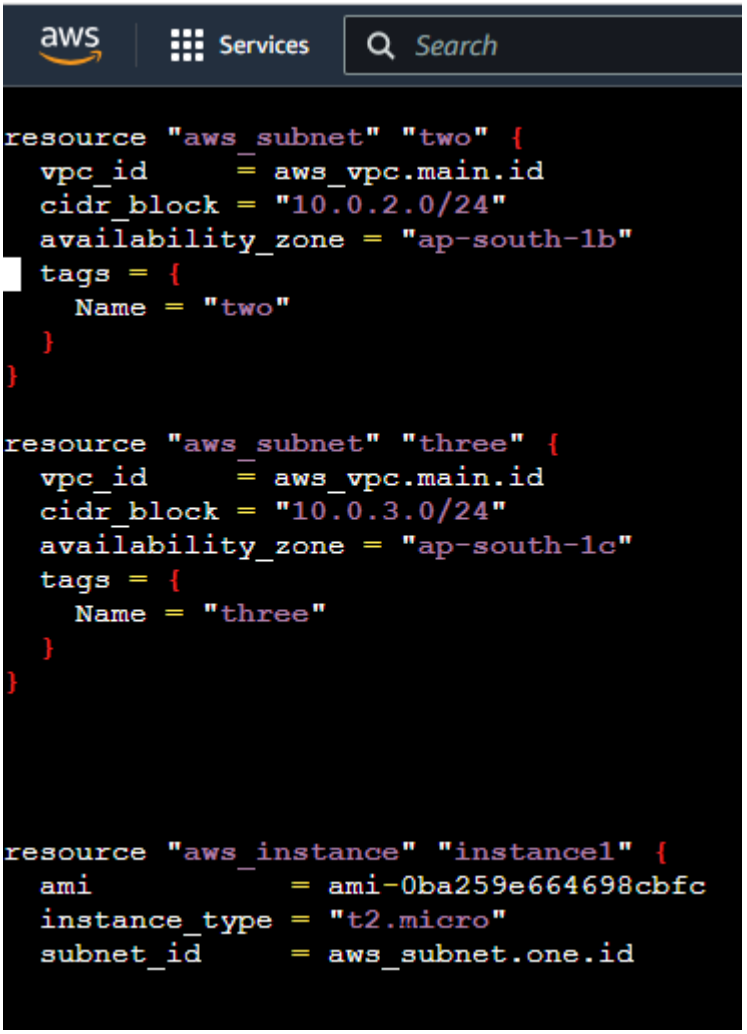
1.Creating a Terraform File in AWS:-

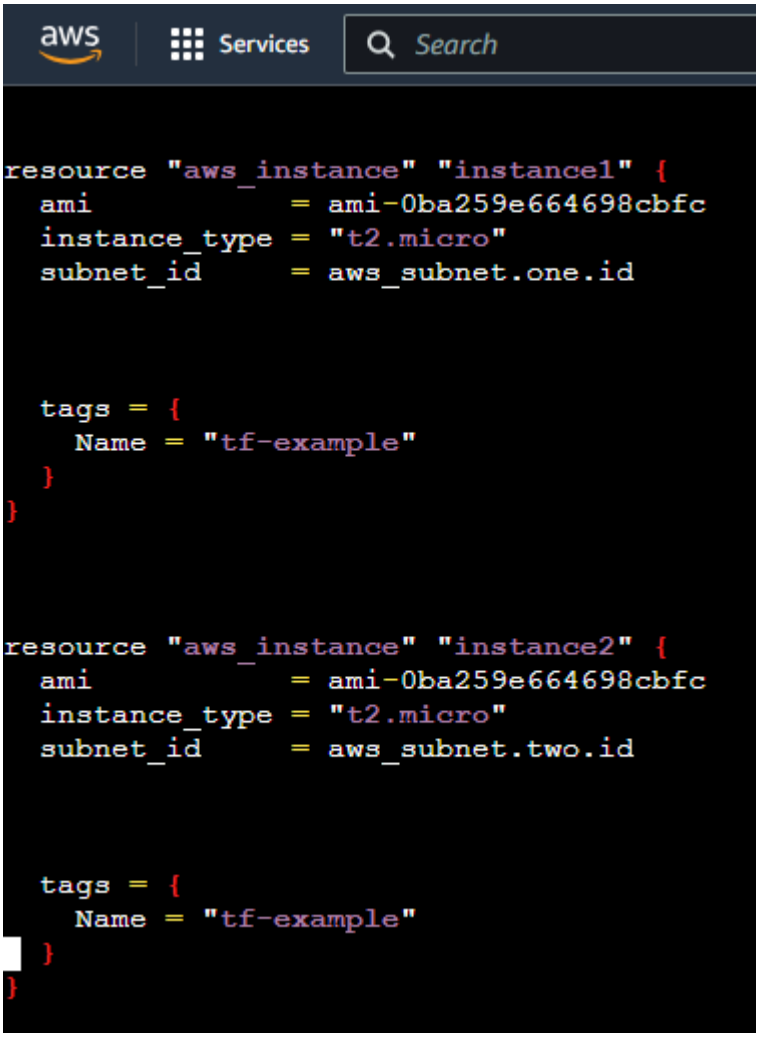
```
  System information as of Wed Mar 13 08:50:46 UTC 2024
ubuntu@ip-172-31-35-235:~$ sudo apt update -y
```

```
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com jammy main
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://apt.releases.hashicorp.com jammy/main amd64 Packages [122 kB]
Fetched 135 kB in 1s (169 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
39 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  terraform
0 upgraded, 1 newly installed, 0 to remove and 39 not upgraded.
```

2.Creating 1 VPC,3 Subnets in 3 different AZ and 2 instances setup in 2 different subnets:-


aws
Services
Search
provider "aws" {
version = "~> 5.0"
region = "ap-south-1"
}
resource "aws_vpc" "main" {
cidr_block = "10.0.0.0/16"
instance_tenancy = "default"
tags = {
Name = "main"
}
}
resource "aws_subnet" "one" {
vpc_id = aws_vpc.main.id
cidr_block = "10.0.1.0/24"
availability_zone = "ap-south-1a"
tags = {
Name = "one"
}
}
resource "aws_subnet" "two" {
vpc_id = aws_vpc.main.id
cidr_block = "10.0.2.0/24"
availability_zone = "ap-south-1b"
tags = {


aws
Services
Search
resource "aws_subnet" "two" {
vpc_id = aws_vpc.main.id
cidr_block = "10.0.2.0/24"
availability_zone = "ap-south-1b"
tags = {
Name = "two"
}
}
resource "aws_subnet" "three" {
vpc_id = aws_vpc.main.id
cidr_block = "10.0.3.0/24"
availability_zone = "ap-south-1c"
tags = {
Name = "three"
}
}
resource "aws_instance" "instance1" {
ami = ami-0ba259e664698cbfc
instance_type = "t2.micro"
subnet_id = aws_subnet.one.id



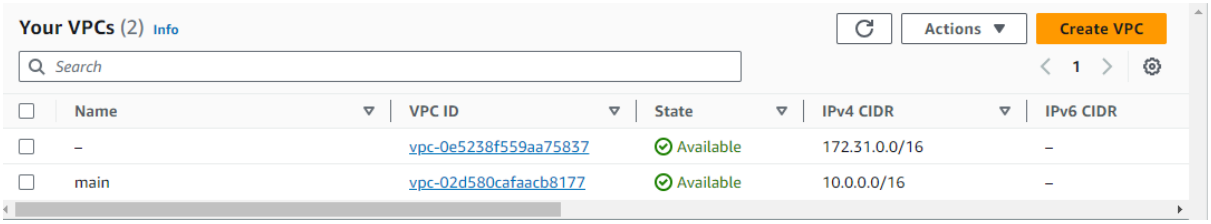
3.Doing Terraform Init:-

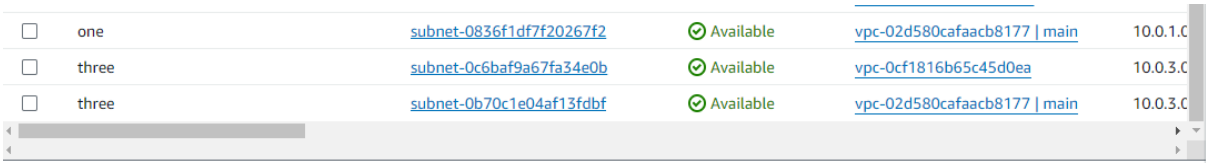
```
aws | Services | Search [Alt+S]
- Installed hashicorp/aws v5.40.0 (signed by HashiCorp)

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.


  Warning: Version constraints inside provider configuration blocks are deprecated

    on main.tf line 2, in provider "aws":
     2:    version = "~> 5.0"

  Terraform 0.13 and earlier allowed provider version constraints inside the provid
  be removed in a future version of Terraform. To silence this warning, move the pr

  (and one more similar warning elsewhere)


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.
ubuntu@ip-172-31-35-235:~$
```

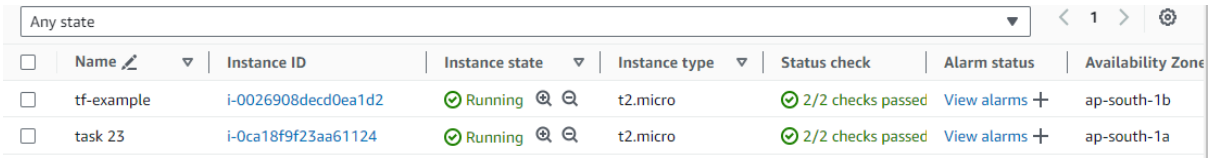
4.Terraform Plan and Apply:-

```
aws_vpc.main: Creating...
aws_vpc.main: Creation complete after 0s [id=vpc-02d580cafaacb8177]
aws_subnet.one: Creating...
aws_subnet.two: Creating...
aws_subnet.three: Creating...
aws_subnet.two: Creation complete after 1s [id=subnet-000bae1f683a267f6]
aws_instance.instance2: Creating...
aws_subnet.one: Creation complete after 1s [id=subnet-0836f1df7f20267f2]
aws_subnet.three: Creation complete after 1s [id=subnet-0b70c1e04af13fdbf]
aws_instance.instance1: Creating...
aws_instance.instance2: Still creating... [10s elapsed]
aws_instance.instance1: Still creating... [10s elapsed]
aws_instance.instance2: Still creating... [20s elapsed]
aws_instance.instance1: Still creating... [20s elapsed]
aws_instance.instance1: Creation complete after 21s [id=i-0476089ffb24fca30]
aws_instance.instance2: Creation complete after 21s [id=i-0026908decd0ea1d2]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-35-235:~$
```

5. Final Output 1Vpc, 3Subnets:-


Your VPCs (2) Info
Actions
Create VPC
Search
1
Name
VPC ID
State
IPv4 CIDR
IPv6 CIDR
–
vpc-0e5238f559aa75837
Available
172.31.0.0/16
–
main
vpc-02d580cafaacb8177
Available
10.0.0.0/16
–


one
subnet-0836f1df7f20267f2
Available
vpc-02d580cafaacb8177 | main
10.0.1.0
three
subnet-0c6baf9a67fa34e0b
Available
vpc-0cf1816b65c45d0ea
10.0.3.0
three
subnet-0b70c1e04af13fdbf
Available
vpc-02d580cafaacb8177 | main
10.0.3.0


Any state
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
tf-example
i-0026908decd0ea1d2
Running
t2.micro
2/2 checks passed
View alarms
ap-south-1b
task 23
i-0ca18f9f23aa61124
Running
t2.micro
2/2 checks passed
View alarms
ap-south-1a