**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**

**Steps:**

Creating a script .tf file and named as main.tf

**Scripts to create two EC2 instances in eu-west-1 :**

provider "aws" {

region = "eu-west-1"

}

resource "aws\_vpc" "tf\_vpc" {

cidr\_block = "10.0.0.0/16"

tags = {

Name = "My-first-vpc"

}

}

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

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

tags = {

Name = "My-first-igw"

}

}

resource "aws\_route\_table" "tf\_rt" {

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

route {

cidr\_block = "0.0.0.0/0"

gateway\_id = aws\_internet\_gateway.tf\_igw.id

}

tags = {

Name = "My-first-rt"

}

}

resource "aws\_route\_table" "tf\_rt1" {

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

route {

cidr\_block = "0.0.0.0/0"

gateway\_id = aws\_internet\_gateway.tf\_igw.id

}

tags = {

Name = "My-first-rt"

}

}

resource "aws\_route\_table\_association" "tf\_subnet\_assosiation1" {

subnet\_id = aws\_subnet.tf\_subnet1.id

route\_table\_id = aws\_route\_table.tf\_rt.id

}

resource "aws\_route\_table\_association" "tf\_subnet\_assosiation2" {

subnet\_id = aws\_subnet.tf\_subnet2.id

route\_table\_id = aws\_route\_table.tf\_rt.id

}

resource "aws\_subnet" "tf\_subnet1" {

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

cidr\_block = "10.0.1.0/24"

availability\_zone = "eu-west-1a"

}

resource "aws\_subnet" "tf\_subnet2" {

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

cidr\_block = "10.0.30.0/24"

availability\_zone = "eu-west-1b"

}

resource "aws\_subnet" "tf\_subnet3" {

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

cidr\_block = "10.0.70.0/24"

availability\_zone = "eu-west-1c"

}

resource "aws\_security\_group" "tf\_sg" {

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

name = "terraform-sg1"

description = "SG used for SSH"

# Inbound rule for ssh (port 22)

ingress {

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

# Inbound rule for HTTP (port 80)

ingress {

from\_port = 80

to\_port = 80

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

}

resource "aws\_instance" "terraform-instance1" {

ami = "ami-07355fe79b493752d"

instance\_type = "t2.micro"

subnet\_id = aws\_subnet.tf\_subnet1.id

vpc\_security\_group\_ids = ["sg-09eedc87ceb7361c2"]

tags = {

Name = "My-first-tf-instance"

}

}

resource "aws\_instance" "terraform-instance2" {

ami = "ami-07355fe79b493752d"

instance\_type = "t2.micro"

subnet\_id = aws\_subnet.tf\_subnet2.id

vpc\_security\_group\_ids = ["sg-09eedc87ceb7361c2"]

tags = {

Name = "My-Second-tf-instance"

}

}

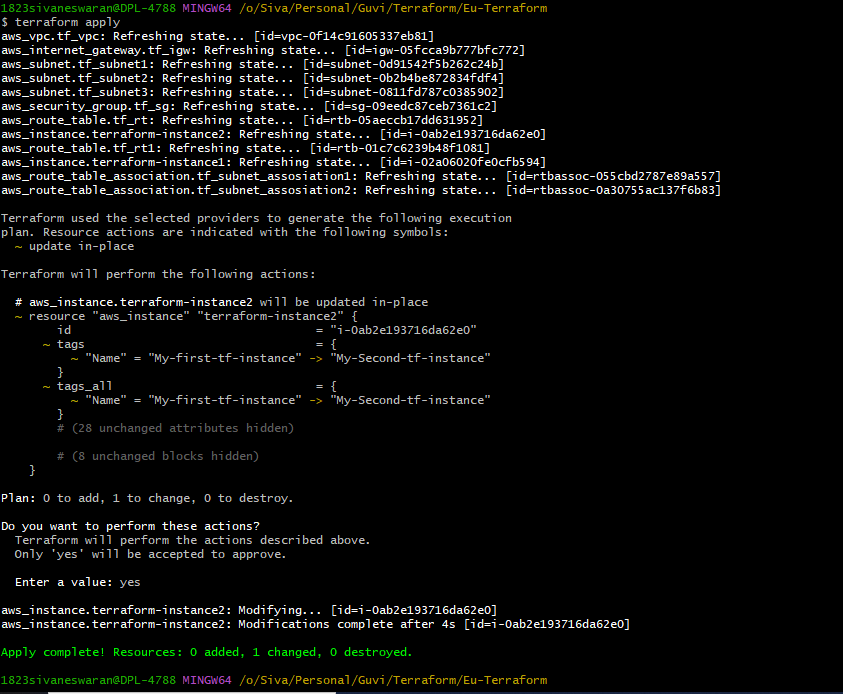
Commands:

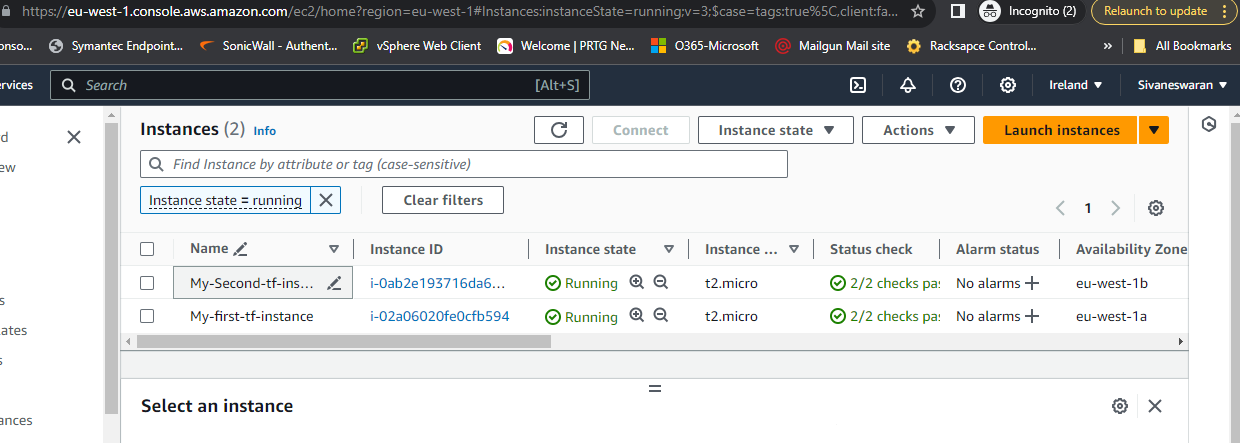
Terraform init

Terraform validate

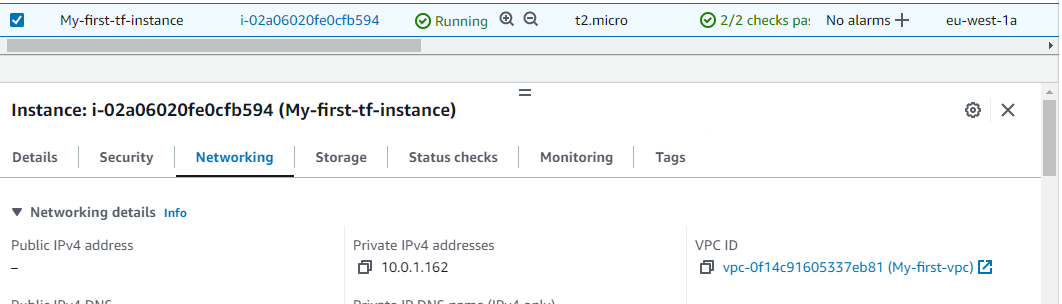
Terraform plan

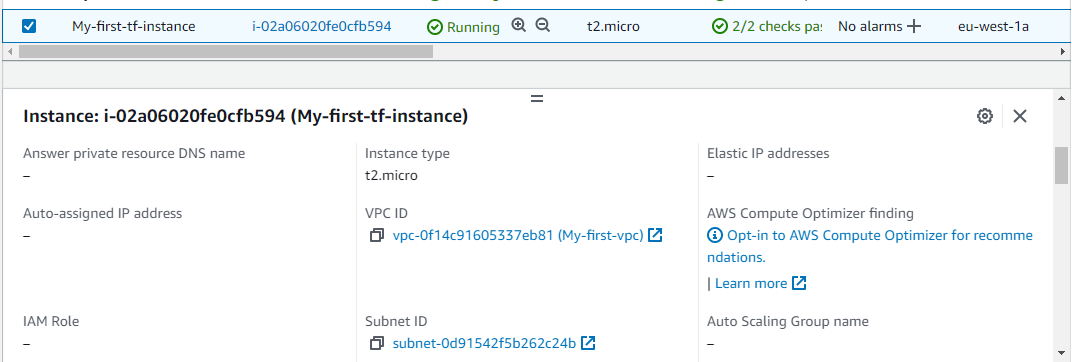
Terraform apply





First EC2: subnet status





Second EC2: subnet status

