**TERRAFORM TASK 7**

**1) Create VPC**

**2) Create Internet gateway**

**3) Create Custom Route Table**

**4) Create Subnet**

**5) Associate subnet with Route Table**

**6) Create Security Group to allow port 22.80,443**

**7) Create a network interface with an ip in the subnet that was created in step 4**

**8) Assign an elastic IP to the network interface created in step 7**

**9) Create Ubuntu server and install/enable apache2 Note:**

**1) Create single main.tf which will be created the above resources and do not hardcode the id's.**

**2) Configure s3 as backend and dynamo db locking for multi user execution.**

resource "aws\_vpc" "main-vpc"{

    cidr\_block = "172.68.0.0/24"

    enable\_dns\_hostnames = true

    enable\_dns\_support = true

    tags = {

        Name = "VPC-Terra"

    }

}

resource "aws\_internet\_gateway" "gateway"{

    vpc\_id = aws\_vpc.main-vpc.id

    tags = {

        Name = "IGW-Terra"

    }

}

resource "aws\_route\_table" "route"{

    vpc\_id = aws\_vpc.main-vpc.id

    tags = {

        Name = "RTB-Terra"

    }

}

resource "aws\_subnet" "subnet"{

    vpc\_id = aws\_vpc.main-vpc.id

    cidr\_block = "172.68.0.0/28"

    map\_public\_ip\_on\_launch = true

    availability\_zone = "us-west-2a"

    tags = {

        Name = "subnet-terra"

    }

}

resource "aws\_route\_table\_association" "sr"{

    subnet\_id = aws\_subnet.subnet.id

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

}

resource "aws\_security\_group" "security" {

    vpc\_id = aws\_vpc.main-vpc.id

    tags = {

        Name = "SG-Terra"

    }

}

resource "aws\_security\_group\_rule" "tcp"{

    type = "ingress"

    from\_port = 22

    to\_port = 22

    protocol = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

    security\_group\_id = aws\_security\_group.security.id

}

resource "aws\_security\_group\_rule" "https"{

    type = "ingress"

    from\_port = 443

    to\_port = 443

    protocol = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

    security\_group\_id = aws\_security\_group.security.id

}

resource "aws\_security\_group\_rule" "http"{

    type = "ingress"

    from\_port = 80

    to\_port = 80

    protocol = "tcp"

    cidr\_blocks = ["0.0.0.0/0"]

    security\_group\_id = aws\_security\_group.security.id

}

resource "aws\_network\_interface" "NAT-GW"{

    subnet\_id = aws\_subnet.subnet.id

    tags = {

        Name = "NAT-GW"

    }

}

resource "aws\_eip" "elastic"{

    network\_interface = aws\_network\_interface.NAT-GW.id

}

resource "aws\_instance" "server"{

    ami = "ami-05134c8ef96964280"

    instance\_type = "t2.micro"

    subnet\_id = aws\_subnet.subnet.id

tags ={

    Name = "server-terra"

}

    user\_data = <<EOF

    #!/bin/bash

    sudo apt update -y

    sudo apt install -y httpd

    sudo systemctl enable httpd

    sudo systemctl start httpd

    sudo systemctl status httpd

    EOF

}

resource "aws\_s3\_bucket" "bucket"{

    bucket = "s3-back-terra"

    acl = "private"

}

resource "aws\_dynamodb\_table" "dynamodb\_table"{

    name = "dynamodb-table-terra1"

    hash\_key = "LockID"

    read\_capacity = 20

    write\_capacity =20

    attribute {

        name = "LockID"

            type = "S"

        }

    }

    terraform {

        backend "s3"{

            bucket = "s3-back-terra"

            dynamodb\_table = "dynamodb-table-terra1"

            key = "terraform.tfstate"

            region = "us-west-2"

        }

    }











