CASE STUDY -CREATING AN ARCHITECTURE USING TERRAFORM ON AWS

You work as a DevOps Engineer in a leading Software Company. You have been asked to build an infrastructure safely and efficiently.

The company Requirements:
1. Use AWS cloud Provider and the software to be installed is Apache2
2. Use Ubuntu AMI
The company wants the Architecture to have the following services:
1. Create a template with a VPC, 2 subnets, and 1 instance in each subnet
2. Attach Security groups, internet gateway, and network interface to the instance

Answer

```
sudo mkdir casestudy
cd casestudy
cp ../assignment5/var.tf .
vim main.tf
```

```
provider "aws" {
  region = "us-east-1"
  secret_key = var.secret
  access_key = var.access
}

# Create VPC
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
  enable_dns_support = true
  enable_dns_hostnames = true
  tags = {
    Name = "my-vpc"
  }
}

# Create Internet Gateway
resource "aws_internet_gateway" "main" {
  vpc_id = aws_vpc.main.id
  tags = {
    Name = "my-igw"
  }
}
```

```
# Create Subnets
resource "aws_subnet" "subnet1" {
  vpc_id                  = aws_vpc.main.id
  cidr_block              = "10.0.1.0/24"
  availability_zone       = "us-east-1a"
  map_public_ip_on_launch = true
  tags = {
    Name = "subnet-1"
  }
}

resource "aws_subnet" "subnet2" {
  vpc_id                  = aws_vpc.main.id
  cidr_block              = "10.0.2.0/24"
  availability_zone       = "us-east-1b"
  map_public_ip_on_launch = true
  tags = {
    Name = "subnet-2"
  }
}

# Create Security Group
resource "aws_security_group" "instance_sg" {
  vpc_id = aws_vpc.main.id
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  tags = {
    Name = "instance-sg"
  }
}
```

```
# Create Network Interfaces
resource "aws_network_interface" "nic1" {
  subnet_id          = aws_subnet.subnet1.id
  security_group_ids = [aws_security_group.instance_sg.id]
}

resource "aws_network_interface" "nic2" {
  subnet_id          = aws_subnet.subnet2.id
  security_group_ids = [aws_security_group.instance_sg.id]
}

# Create Instances
resource "aws_instance" "instance1" {
  ami           = "ami-0c7217cdde317cfec"
  instance_type = "t2.micro"
  key_name      = "MyKeyPair-Ravi"

  network_interface_ids = [aws_network_interface.nic1.id]
  tags = {
    Name = "instance-1"
  }
}

resource "aws_instance" "instance2" {
  ami           = "ami-0c7217cdde317cfec"
  instance_type = "t2.micro"
  key_name      = "MyKeyPair-Ravi"

  network_interface_ids = [aws_network_interface.nic2.id]
  tags = {
    Name = "instance-2"
  }
}
```

```
sudo terraform init
sudo terraform plan
sudo terraform apply
```

Completed.