```
provider "aws" {
  region = "us-east-1" # Change as needed
}

# 1. Create VPC
resource "aws_vpc" "main_vpc" {
  cidr_block      = "10.0.0.0/16"
  enable_dns_support   = true
  enable_dns_hostnames = true
  tags = {
          Name = "main_vpc"
  }
}

# 2. Create Subnets (2 Public, 2 Private)
resource "aws_subnet" "public_subnet_1" {
  vpc_id          = aws_vpc.main_vpc.id
  cidr_block      = "10.0.1.0/24"
  availability_zone = "us-east-1a"
  map_public_ip_on_launch = true
  tags = {
          Name = "public_subnet_1"
  }
}

resource "aws_subnet" "public_subnet_2" {
  vpc_id          = aws_vpc.main_vpc.id
  cidr_block      = "10.0.2.0/24"
  availability_zone = "us-east-1b"
  map_public_ip_on_launch = true
  tags = {
          Name = "public_subnet_2"
  }
}

resource "aws_subnet" "private_subnet_1" {
  vpc_id          = aws_vpc.main_vpc.id
  cidr_block      = "10.0.3.0/24"
  availability_zone = "us-east-1a"
  tags = {
          Name = "private_subnet_1"
  }
}

resource "aws_subnet" "private_subnet_2" {
  vpc_id          = aws_vpc.main_vpc.id
  cidr_block      = "10.0.4.0/24"
  availability_zone = "us-east-1b"
```

```
  tags = {
        Name = "private_subnet_2"
  }
}

# 3. Create Internet Gateway and Attach to VPC
resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.main_vpc.id
  tags = {
        Name = "internet_gateway"
  }
}

# 4. Create NAT Gateway
resource "aws_eip" "nat_eip" {
  vpc = true
}

resource "aws_nat_gateway" "nat_gateway" {
  allocation_id = aws_eip.nat_eip.id
  subnet_id     = aws_subnet.public_subnet_1.id
  tags = {
        Name = "nat_gateway"
  }
}

# 5. Create Route Tables and Associate with Subnets

# Public route table
resource "aws_route_table" "public_route_table" {
  vpc_id = aws_vpc.main_vpc.id
  route {
        cidr_block = "0.0.0.0/0"
        gateway_id = aws_internet_gateway.igw.id
  }
  tags = {
        Name = "public_route_table"
  }
}

# Associate public route table with public subnets
resource "aws_route_table_association" "public_route_table_assoc_1" {
  subnet_id     = aws_subnet.public_subnet_1.id
  route_table_id = aws_route_table.public_route_table.id
}

resource "aws_route_table_association" "public_route_table_assoc_2" {
  subnet_id     = aws_subnet.public_subnet_2.id
```

```
  route_table_id = aws_route_table.public_route_table.id
}

# Private route table
resource "aws_route_table" "private_route_table" {
  vpc_id = aws_vpc.main_vpc.id
  route {
        cidr_block = "0.0.0.0/0"
        nat_gateway_id = aws_nat_gateway.nat_gateway.id
  }
  tags = {
        Name = "private_route_table"
  }
}

# Associate private route table with private subnets
resource "aws_route_table_association" "private_route_table_assoc_1" {
  subnet_id      = aws_subnet.private_subnet_1.id
  route_table_id = aws_route_table.private_route_table.id
}

resource "aws_route_table_association" "private_route_table_assoc_2" {
  subnet_id      = aws_subnet.private_subnet_2.id
  route_table_id = aws_route_table.private_route_table.id
}

# 6. Create Security Group for SSH Access
resource "aws_security_group" "ssh_security_group" {
  vpc_id = aws_vpc.main_vpc.id
  name   = "ssh_security_group"

  ingress {
        from_port   = 22
        to_port           = 22
        protocol          = "tcp"
        cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
        from_port   = 0
        to_port           = 0
        protocol          = "-1"
        cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
        Name = "ssh_security_group"
  }
```

```
}

# 7. Launch EC2 Instances (Public and Private)

# Public EC2 instance
resource "aws_instance" "public_ec2" {
  ami                         = "ami-0c55b159cbfafe1f0"  # Replace with your preferred AMI ID
  instance_type               = "t2.micro"
  subnet_id                   = aws_subnet.public_subnet_1.id
  security_groups             = [aws_security_group.ssh_security_group.name]
  associate_public_ip_address = true

  tags = {
        Name = "public_ec2"
  }
}

# Private EC2 instance
resource "aws_instance" "private_ec2" {
  ami                         = "ami-0c55b159cbfafe1f0"  # Replace with your preferred AMI ID
  instance_type               = "t2.micro"
  subnet_id                   = aws_subnet.private_subnet_1.id
  security_groups             = [aws_security_group.ssh_security_group.name]
  associate_public_ip_address = false

  tags = {
        Name = "private_ec2"
  }
}

# 8. Set up SSH Key for EC2 Access
resource "aws_key_pair" "key_pair" {
  key_name   = "my_key"
  public_key = file("~/.ssh/id_rsa.pub")
}
```