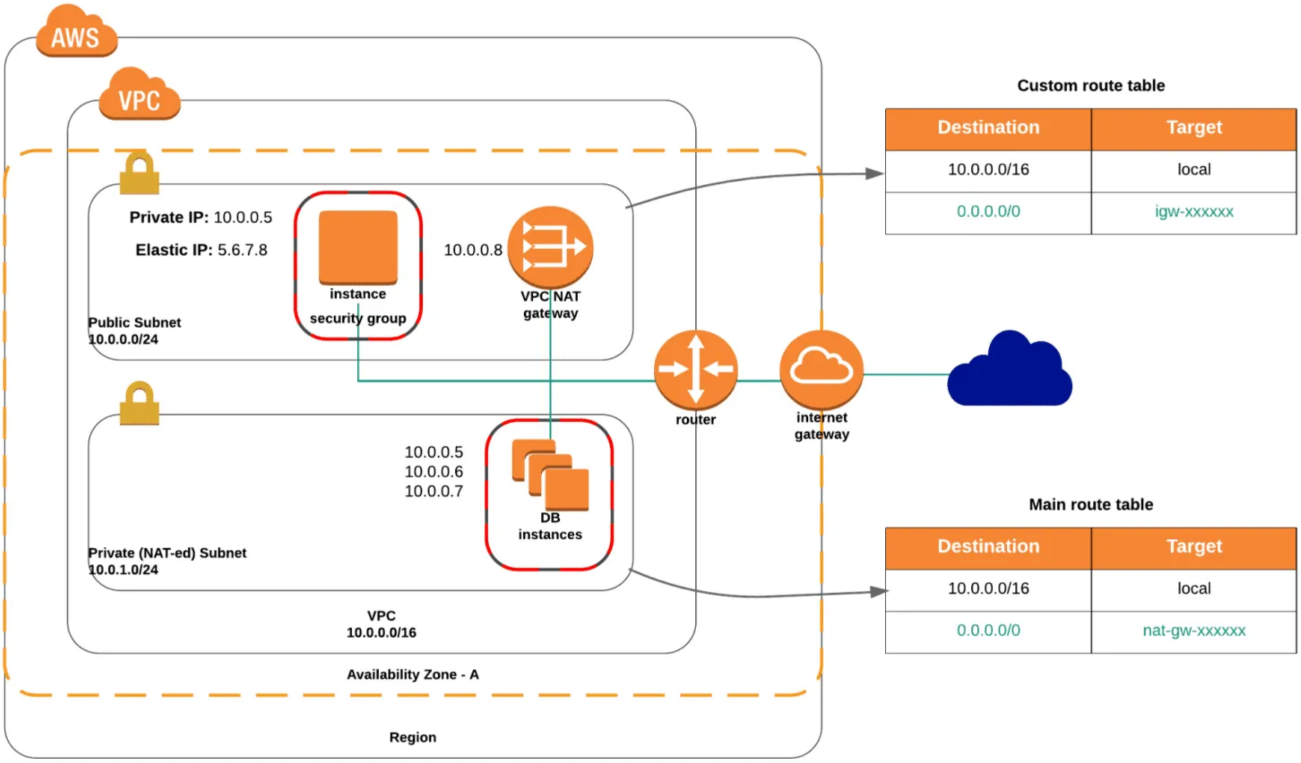
Terraform – Managing VPC With Public and Private Subnetes

Table of contents

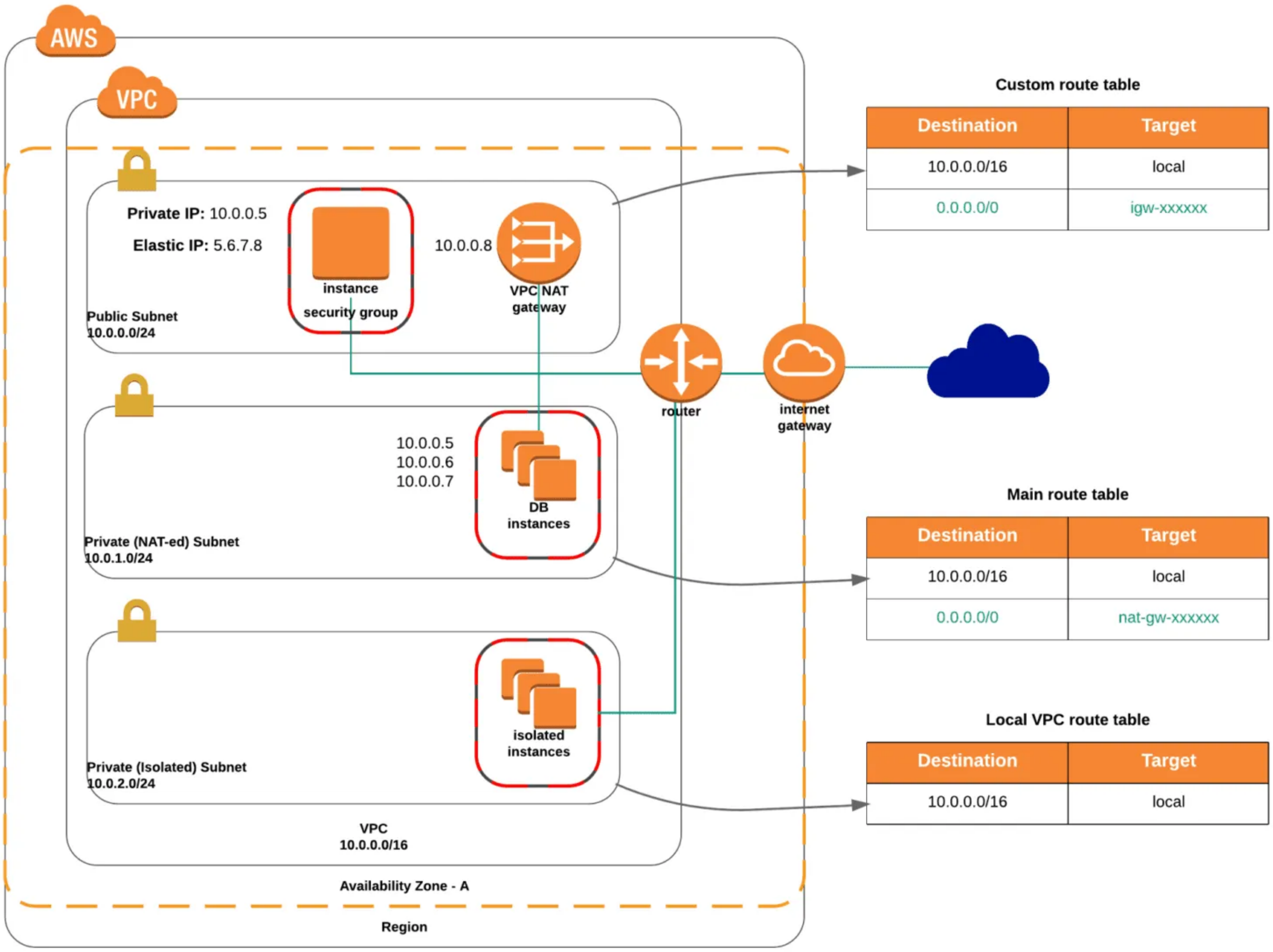


Custom route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-xxxxxx

Main route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat-gw-xxxxxx



Custom route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-xxxxxx

Main route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat-gw-xxxxxx

Local VPC route table

Destination	Target
10.0.0.0/16	local

```
resource "aws_vpc" "my_vpc" {
  cidr_block       = "10.0.0.0/16"
  enable_dns_hostnames = true

  tags = {
    Name = "My VPC"
  }
}

resource "aws_subnet" "public" {
  vpc_id     = aws_vpc.my_vpc.id
  cidr_block = "10.0.0.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "Public Subnet"
  }
}

resource "aws_internet_gateway" "my_vpc_igw" {
  vpc_id = aws_vpc.my_vpc.id

  tags = {
    Name = "My VPC - Internet Gateway"
  }
}

resource "aws_route_table" "my_vpc_us_east_1a_public" {
    vpc_id = aws_vpc.my_vpc.id

    route {
        cidr_block = "0.0.0.0/0"
        gateway_id = aws_internet_gateway.my_vpc_igw.id
    }

    tags = {
        Name = "Public Subnet Route Table."
    }
}

resource "aws_route_table_association" "my_vpc_us_east_1a_public" {
    subnet_id = aws_subnet.public.id
    route_table_id = aws_route_table.my_vpc_us_east_1a_public.id
}

resource "aws_security_group" "allow_ssh" {
  name        = "allow_ssh_sg"
  description = "Allow SSH inbound connections"
  vpc_id = aws_vpc.my_vpc.id

  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 = "allow_ssh_sg"
  }
}

resource "aws_instance" "my_instance" {
  ami           = "ami-0ac019f4fcb7cb7e6"
  instance_type = "t2.micro"
  key_name = "Lenovo T410"
  vpc_security_group_ids = [ aws_security_group.allow_ssh.id ]
  subnet_id = aws_subnet.public.id
  associate_public_ip_address = true

  tags = {
    Name = "My Instance"
  }
}
```

```
resource "aws_subnet" "nated" {
  vpc_id     = aws_vpc.my_vpc.id
  cidr_block = "10.0.1.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "NAT-ed Subnet"
  }
}
```

```
resource "aws_eip" "nat_gw_eip" {
  vpc = true
}

resource "aws_nat_gateway" "gw" {
  allocation_id = aws_eip.nat_gw_eip.id
  subnet_id     = aws_subnet.public.id
}
```

```
resource "aws_route_table" "my_vpc_us_east_1a_nated" {
    vpc_id = aws_vpc.my_vpc.id

    route {
        cidr_block = "0.0.0.0/0"
        nat_gateway_id = aws_nat_gateway.gw.id
    }

    tags = {
        Name = "Main Route Table for NAT-ed subnet"
    }
}

resource "aws_route_table_association" "my_vpc_us_east_1a_nated" {
    subnet_id = aws_subnet.nated.id
    route_table_id = aws_route_table.my_vpc_us_east_1a_nated.id
}
```

```
resource "aws_subnet" "private" {
  vpc_id     = aws_vpc.my_vpc.id
  cidr_block = "10.0.2.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "Isolated Private Subnet"
  }
}
```

```
resource "aws_route_table" "my_vpc_us_east_1a_private" {
    vpc_id = aws_vpc.my_vpc.id

    tags = {
        Name = "Local Route Table for Isolated Private Subnet"
    }
}

resource "aws_route_table_association" "my_vpc_us_east_1a_private" {
    subnet_id = aws_subnet.private.id
    route_table_id = aws_route_table.my_vpc_us_east_1a_private.id
}
```

```
# declare a VPC
resource "aws_vpc" "my_vpc" {
  cidr_block       = "10.0.0.0/16"
  enable_dns_hostnames = true

  tags = {
    Name = "My VPC"
  }
}

resource "aws_subnet" "public" {
  vpc_id     = aws_vpc.my_vpc.id
  cidr_block = "10.0.0.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "Public Subnet"
  }
}

resource "aws_internet_gateway" "my_vpc_igw" {
  vpc_id = aws_vpc.my_vpc.id

  tags = {
    Name = "My VPC - Internet Gateway"
  }
}

resource "aws_route_table" "my_vpc_us_east_1a_public" {
    vpc_id = aws_vpc.my_vpc.id

    route {
        cidr_block = "0.0.0.0/0"
        gateway_id = aws_internet_gateway.my_vpc_igw.id
    }

    tags = {
        Name = "Public Subnet Route Table"
    }
}

resource "aws_route_table_association" "my_vpc_us_east_1a_public" {
    subnet_id = aws_subnet.public.id
    route_table_id = aws_route_table.my_vpc_us_east_1a_public.id
}

resource "aws_security_group" "allow_ssh" {
  name        = "allow_ssh_sg"
  description = "Allow SSH inbound connections"
  vpc_id = aws_vpc.my_vpc.id
  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 = "allow_ssh_sg"
  }
}

resource "aws_instance" "my_instance" {
  ami           = "ami-0ac019f4fcb7cb7e6"
  instance_type = "t2.micro"
  key_name = "Lenovo T410"
  vpc_security_group_ids = [ aws_security_group.allow_ssh.id ]
  subnet_id = aws_subnet.public.id
  associate_public_ip_address = true

  tags = {
    Name = "My Instance"
  }
}

resource "aws_subnet" "nated" {
  vpc_id     = aws_vpc.my_vpc.id
  cidr_block = "10.0.1.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "NAT-ed Subnet"
  }
}

resource "aws_eip" "nat_gw_eip" {
  vpc = true
}

resource "aws_nat_gateway" "gw" {
  allocation_id = aws_eip.nat_gw_eip.id
  subnet_id     = aws_subnet.public.id
```

```
}

resource "aws_route_table" "my_vpc_us_east_1a_nated" {
    vpc_id = aws_vpc.my_vpc.id

    route {
        cidr_block = "0.0.0.0/0"
        nat_gateway_id = aws_nat_gateway.gw.id
    }

    tags = {
        Name = "Main Route Table for NAT-ed subnet"
    }
}

resource "aws_route_table_association" "my_vpc_us_east_1a_nated" {
    subnet_id = aws_subnet.nated.id
    route_table_id = aws_route_table.my_vpc_us_east_1a_nated.id
}

resource "aws_subnet" "private" {
  vpc_id     = aws_vpc.my_vpc.id
  cidr_block = "10.0.2.0/24"
  availability_zone = "us-east-1a"

  tags = {
    Name = "Isolated Private Subnet"
  }
}

resource "aws_route_table" "my_vpc_us_east_1a_private" {
    vpc_id = aws_vpc.my_vpc.id

    tags = {
        Name = "Local Route Table for Isolated Private Subnet"
    }
}

resource "aws_route_table_association" "my_vpc_us_east_1a_private" {
    subnet_id = aws_subnet.private.id
    route_table_id = aws_route_table.my_vpc_us_east_1a_private.id
}

output "instance_public_ip" {
  value = aws_instance.my_instance.public_ip
}
```