# Terraform VPC Setup with Explanation

## Terraform Code with Line-by-Line Explanation

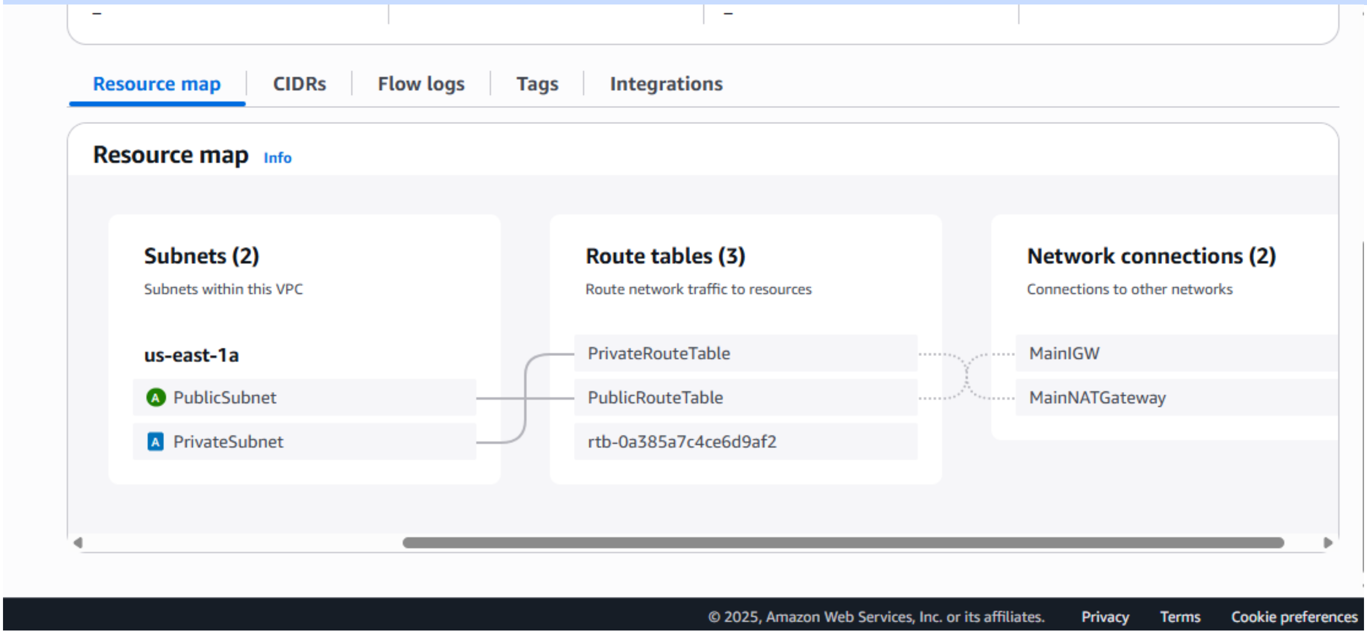
```
provider "aws" {
  region = "us-east-1"
}
# Specifies AWS as the provider and selects the us-east-1 region.

resource "aws_vpc" "main" {
  cidr_block           = "10.0.0.0/16"
  enable_dns_support   = true
  enable_dns_hostnames = true
  tags = {
    Name = "MainVPC"
  }
}
# Creates a VPC with DNS support and hostname resolution. CIDR block defines the address range.

resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.main.id
  tags = {
    Name = "MainIGW"
  }
}
# Attaches an Internet Gateway to the VPC for external connectivity.

resource "aws_subnet" "public_subnet" {
  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 = "PublicSubnet"
  }
}
# Creates a public subnet in AZ us-east-1a. It maps public IPs to instances launched.

resource "aws_subnet" "private_subnet" {
  vpc_id            = aws_vpc.main.id
  cidr_block        = "10.0.2.0/24"
  availability_zone = "us-east-1a"
  tags = {
    Name = "PrivateSubnet"
  }
}
# Creates a private subnet in AZ us-east-1a without public IP mapping.

resource "aws_eip" "nat_eip" {
  vpc = true
}
# Allocates an Elastic IP for the NAT Gateway to use.
```

# Terraform VPC Setup with Explanation

```
resource "aws_nat_gateway" "nat" {
  allocation_id = aws_eip.nat_eip.id
  subnet_id     = aws_subnet.public_subnet.id
  tags = {
    Name = "MainNATGateway"
  }
}
# Creates a NAT Gateway in the public subnet to allow private subnet instances outbound access.

resource "aws_route_table" "public" {
  vpc_id = aws_vpc.main.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.igw.id
  }
  tags = {
    Name = "PublicRouteTable"
  }
}
# Defines a public route table with default route to Internet Gateway.

resource "aws_route_table_association" "public" {
  subnet_id      = aws_subnet.public_subnet.id
  route_table_id = aws_route_table.public.id
}
# Associates the public route table with the public subnet.

resource "aws_route_table" "private" {
  vpc_id = aws_vpc.main.id
  route {
    cidr_block     = "0.0.0.0/0"
    nat_gateway_id = aws_nat_gateway.nat.id
  }
  tags = {
    Name = "PrivateRouteTable"
  }
}
# Defines a private route table with route to NAT Gateway for internet access.

resource "aws_route_table_association" "private" {
  subnet_id      = aws_subnet.private_subnet.id
  route_table_id = aws_route_table.private.id
}
# Associates the private route table with the private subnet.
```

# Terraform VPC Setup with Explanation

## Network Architecture: Subnet Overview



**Subnets (2)**

Subnets within this VPC

us-east-1a

- PublicSubnet
- PrivateSubnet

**Route tables (3)**

Route network traffic to resources

- PrivateRouteTable
- PublicRouteTable
- rtb-0a385a7c4ce6d9af2

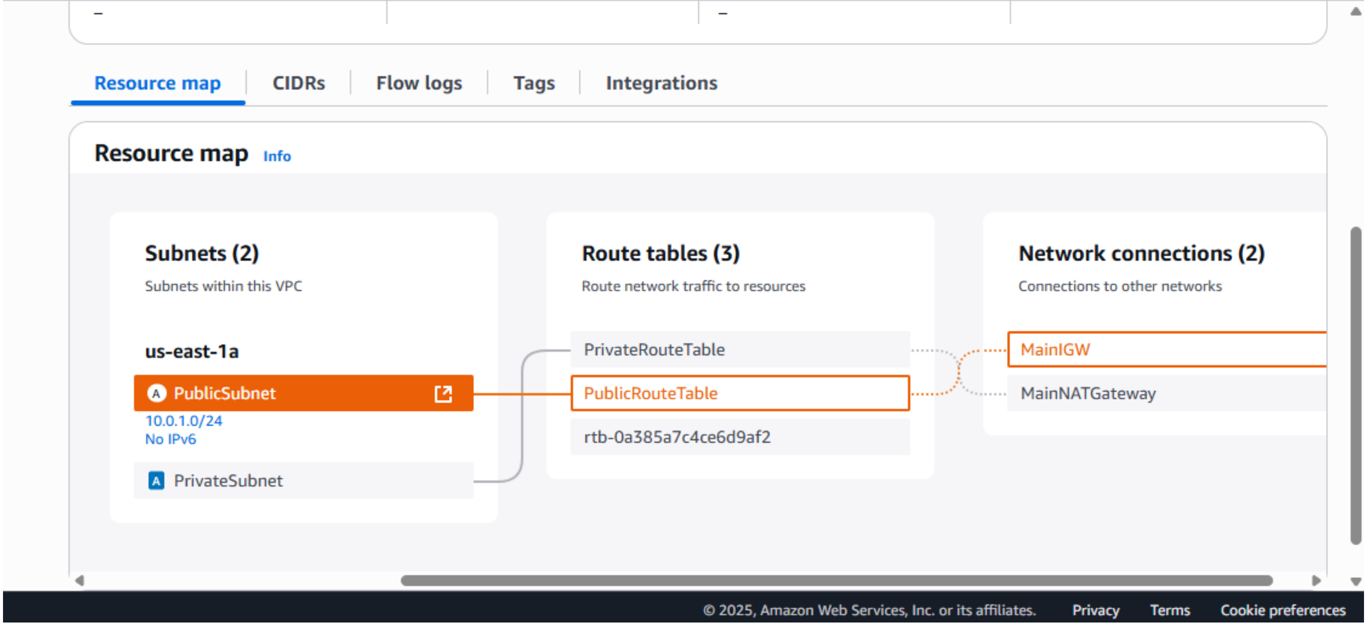
**Network connections (2)**

Connections to other networks

- MainIGW
- MainNATGateway

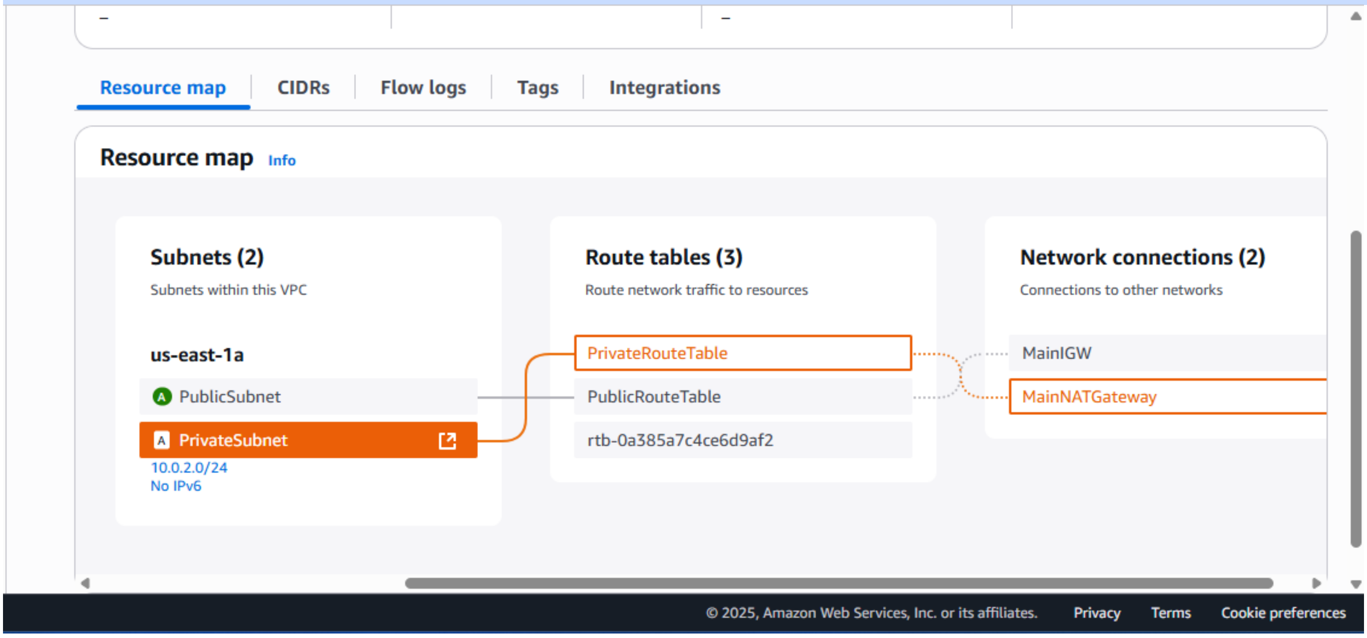
# Terraform VPC Setup with Explanation

## Public Subnet Routing



# Terraform VPC Setup with Explanation

## Private Subnet Routing



Resource map
CIDRs
Flow logs
Tags
Integrations
Resource map
Info
Subnets (2)
Subnets within this VPC
us-east-1a
PublicSubnet
PrivateSubnet
10.0.2.0/24
No IPv6
Route tables (3)
Route network traffic to resources
PrivateRouteTable
PublicRouteTable
rtb-0a385a7c4ce6d9af2
Network connections (2)
Connections to other networks
MainIGW
MainNATGateway
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences