Certainly! To create the required AWS resources within an existing VPC using an AWS CloudFormation script (YAML format), you'll need to define the following components:

1. **Public Subnets**: Two public subnets.
2. **Private Subnets**: Two private subnets.
3. **Route Tables**: Route tables associated with the subnets.
4. **NAT Gateway**: To enable outbound internet access for the private subnets.

Below is a sample CloudFormation YAML script to achieve this:

yaml

Code kopieren

AWSTemplateFormatVersion: '2010-09-09'

Resources:

# Define the NAT Gateway Elastic IP

NATGatewayEIP:

Type: 'AWS::EC2::EIP'

# Define the NAT Gateway

NATGateway:

Type: 'AWS::EC2::NatGateway'

Properties:

AllocationId: !GetAtt NATGatewayEIP.AllocationId

SubnetId: !Ref PublicSubnet1

# Define Route Table for Public Subnets

PublicRouteTable:

Type: 'AWS::EC2::RouteTable'

Properties:

VpcId: !Ref VPCId

# Define Route Table for Private Subnets

PrivateRouteTable:

Type: 'AWS::EC2::RouteTable'

Properties:

VpcId: !Ref VPCId

# Public Route Table Route to Internet Gateway

PublicRoute:

Type: 'AWS::EC2::Route'

DependsOn: 'InternetGatewayAttachment'

Properties:

RouteTableId: !Ref PublicRouteTable

DestinationCidrBlock: '0.0.0.0/0'

GatewayId: !Ref InternetGateway

# Private Route Table Route to NAT Gateway

PrivateRoute:

Type: 'AWS::EC2::Route'

Properties:

RouteTableId: !Ref PrivateRouteTable

DestinationCidrBlock: '0.0.0.0/0'

NatGatewayId: !Ref NATGateway

# Define Public Subnets

PublicSubnet1:

Type: 'AWS::EC2::Subnet'

Properties:

VpcId: !Ref VPCId

CidrBlock: '10.0.1.0/24'

AvailabilityZone: 'us-east-1a'

MapPublicIpOnLaunch: true

PublicSubnet2:

Type: 'AWS::EC2::Subnet'

Properties:

VpcId: !Ref VPCId

CidrBlock: '10.0.2.0/24'

AvailabilityZone: 'us-east-1b'

MapPublicIpOnLaunch: true

# Define Private Subnets

PrivateSubnet1:

Type: 'AWS::EC2::Subnet'

Properties:

VpcId: !Ref VPCId

CidrBlock: '10.0.3.0/24'

AvailabilityZone: 'us-east-1a'

PrivateSubnet2:

Type: 'AWS::EC2::Subnet'

Properties:

VpcId: !Ref VPCId

CidrBlock: '10.0.4.0/24'

AvailabilityZone: 'us-east-1b'

# Associate Route Tables with Subnets

PublicSubnet1RouteTableAssociation:

Type: 'AWS::EC2::SubnetRouteTableAssociation'

Properties:

SubnetId: !Ref PublicSubnet1

RouteTableId: !Ref PublicRouteTable

PublicSubnet2RouteTableAssociation:

Type: 'AWS::EC2::SubnetRouteTableAssociation'

Properties:

SubnetId: !Ref PublicSubnet2

RouteTableId: !Ref PublicRouteTable

PrivateSubnet1RouteTableAssociation:

Type: 'AWS::EC2::SubnetRouteTableAssociation'

Properties:

SubnetId: !Ref PrivateSubnet1

RouteTableId: !Ref PrivateRouteTable

PrivateSubnet2RouteTableAssociation:

Type: 'AWS::EC2::SubnetRouteTableAssociation'

Properties:

SubnetId: !Ref PrivateSubnet2

RouteTableId: !Ref PrivateRouteTable

# Define Internet Gateway

InternetGateway:

Type: 'AWS::EC2::InternetGateway'

# Attach Internet Gateway to VPC

InternetGatewayAttachment:

Type: 'AWS::EC2::VPCGatewayAttachment'

Properties:

VpcId: !Ref VPCId

InternetGatewayId: !Ref InternetGateway

Parameters:

VPCId:

Description: 'The ID of the VPC where the subnets will be created'

Type: 'AWS::EC2::VPC::Id'

Default: 'vpc-12345678' # Replace with your VPC ID

**Explanation:**

1. **NAT Gateway EIP and NAT Gateway**: The NAT Gateway requires an Elastic IP address, so it's defined first and then used in the NAT Gateway resource.
2. **Route Tables and Routes**:
   * The public route table is associated with an Internet Gateway.
   * The private route table is associated with the NAT Gateway for outbound internet access.
3. **Subnets**:
   * Two public and two private subnets are created in different availability zones.
   * Public subnets have MapPublicIpOnLaunch set to true.
4. **Associations**:
   * Each subnet is associated with its respective route table.

Make sure to replace placeholder values such as VPC IDs and availability zones with your specific values. This template assumes you are familiar with AWS CloudFormation and the resources mentioned. You can deploy this CloudFormation stack via the AWS Management Console, AWS CLI, or other methods supported by AWS.