

Siddhesh Ghangale

Creating 2-Tier Architecture using Cloudformation

Code:

```
AWSTemplateFormatVersion: 2010-09-09 Description: Creates a VPC with public and private subnets and EC2 instances. Resources: # Creating A VPC
```

```
MyVPC:
```

```
  Type: 'AWS::EC2::VPC'
```

```
  Properties:
```

```
    CidrBlock: 10.0.0.0/16
```

```
    EnableDnsSupport: true
```

```
    EnableDnsHostnames:
```

```
    true Tags:
```

```
      - Key: Name
```

```
        Value: MyVPC
```

```
# Creating Public and Private Subnets
```

```
PublicSubnet:
```

```
  Type: 'AWS::EC2::Subnet'
```

```
  Properties:
```

```
    VpcId: !Ref MyVPC
```

```
    CidrBlock: 10.0.1.0/24
```

```
    MapPublicIpOnLaunch: true
```

```
    Tags:
```

```
      - Key: Name
```

```
        Value: PublicSubnet
```

```
PrivateSubnet:
```

```
  Type: 'AWS::EC2::Subnet'
```

```
  Properties:
```

```
    VpcId: !Ref MyVPC
```

```
    CidrBlock: 10.0.2.0/24
```

```
    Tags:
```

```
      - Key: Name
```

```
        Value: PrivateSubnet
```

```
# Creating an Internet Gateway
```

```
InternetGateway:
```

```
  Type: 'AWS::EC2::InternetGateway'
```

```
  Properties:
```

```

    Tags:
      - Key: Name
        Value: InternetGateway
# Creating an Attach Gateway
AttachGateway:
  Type: 'AWS::EC2::VPCGatewayAttachment'
  Properties:
    VpcId: !Ref MyVPC
    InternetGatewayId: !Ref InternetGateway
# Creating Route Table
PublicRouteTable:
  Type: 'AWS::EC2::RouteTable'
  Properties:
    VpcId: !Ref MyVPC
    Tags:
      - Key: Name
        Value: PublicRouteTable
# Creating Route for public subnets
PublicRoute:
  Type: 'AWS::EC2::Route'
  DependsOn: AttachGateway
  Properties:
    RouteTableId: !Ref PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway
# Creating Subnet Route Table Association
AssociatePublicSubnetRouteTable:
  Type: 'AWS::EC2::SubnetRouteTableAssociation'
  Properties:
    SubnetId: !Ref PublicSubnet
    RouteTableId: !Ref PublicRouteTable

# Creating a NAT Gateway
NATGateway:
  Type: 'AWS::EC2::NatGateway'
  DependsOn: AttachGateway
  Properties:
    AllocationId: !GetAtt ElasticIP.AllocationId
    SubnetId: !Ref PublicSubnet    # attaching NAT Gateway to public
subnet

```

```

    Tags:
      - Key: Name
        Value: NATGatewayForPrivateSubnet
ElasticIP:
  Type: 'AWS::EC2::EIP'
  Properties:
    Domain: vpc
    Tags:
      - Key: Name
        Value: ElasticIPFORNATGateway
# Creating a Route Table for private subnets
PrivateRouteTable:
  Type: 'AWS::EC2::RouteTable'
  Properties:
    VpcId: !Ref MyVPC
    Tags:
      - Key: Name
        Value: PrivateRouteTable
# Creating a Route for private subnets
PrivateRoute:
  Type: 'AWS::EC2::Route'
  Properties:
    RouteTableId: !Ref PrivateRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NATGateway # Use NatGatewayId for private route

# Associating private subnet with route table
AssociatePrivateSubnetRouteTable:
  Type: 'AWS::EC2::SubnetRouteTableAssociation'
  Properties:
    SubnetId: !Ref PrivateSubnet
    RouteTableId: !Ref PrivateRouteTable
# creating a Security Group in VPC
MySecurity:
  Type: 'AWS::EC2::SecurityGroup'
  Properties:
    GroupDescription: 'Allow SSH and HTTP traffic'
    VpcId: !Ref MyVPC
    SecurityGroupIngress:
      - IpProtocol: tcp

```

```

        FromPort: '22'
        ToPort: '22'
        CidrIp: 0.0.0.0/0
    - IpProtocol: tcp
      FromPort: '80'
      ToPort: '80'
      CidrIp: 0.0.0.0/0
    - IpProtocol: tcp
      FromPort: '8080'
      ToPort: '8080'
      CidrIp: 0.0.0.0/0
    - IpProtocol: tcp
      FromPort: '3306'
      ToPort: '3306'
      CidrIp: 0.0.0.0/0
  Tags:
    - Key: Name
      Value: MySecurityGroup
# Creating a Public Instance Name NginxInstance
PublicInstance:
  Type: 'AWS::EC2::Instance'
  Properties:
    InstanceType: t2.micro
    ImageId: 'ami-0762cf9da6ee99049'
    SubnetId: !Ref PublicSubnet
    KeyName : 'Ohio'
    SecurityGroupIds:
      - !Ref MySecurity
  Tags:
    - Key: Name
      Value: NginxInstance
# Creating a Private Instance Name ApacheInstance
PrivateInstance:
  Type: 'AWS::EC2::Instance'
  Properties:
    InstanceType: t2.micro
    ImageId: 'ami-03da541f7751c76b8'
    SubnetId: !Ref PrivateSubnet
    SecurityGroupIds:
      - !Ref MySecurity

```

```
KeyName : 'Ohio'
UserData:
  Fn::Base64: |
    #!/bin/bash -xe
    sudo -i
    cd apache-tomcat-8.5.97/bin/
    ./catlina.sh start
Tags:
  - Key: Name
    Value: ApacheInstance
```

```
Outputs:
  VPCId:
    Value: !Ref MyVPC
  PublicSubnetId:
    Value: !Ref PublicSubnet
  PrivateSubnetId:
    Value: !Ref PrivateSubnet
  PublicInstanceId:
    Value: !Ref PublicInstance
  PrivateInstanceId:
    Value: !Ref PrivateInstance
```

Overview:

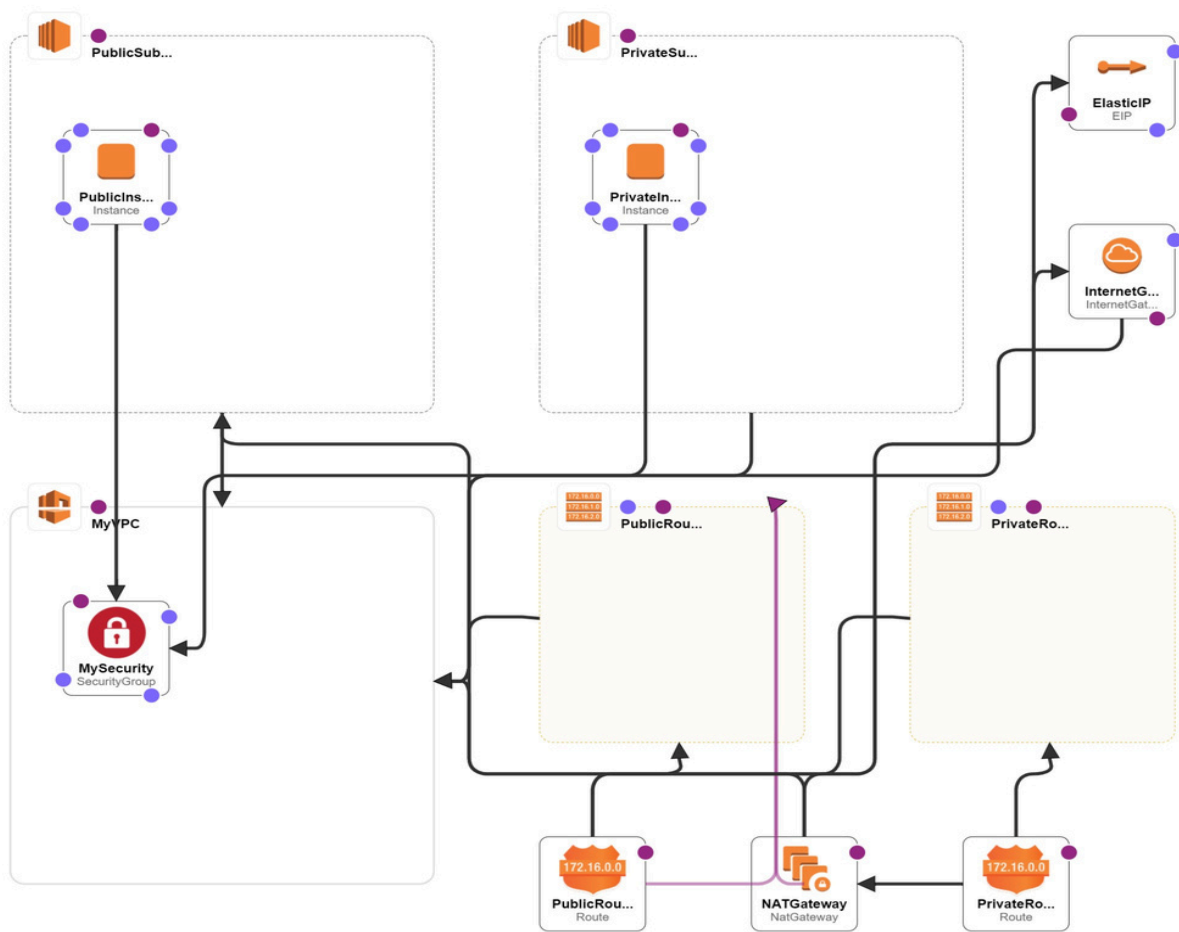
This CloudFormation template creates a Virtual Private Cloud (VPC) in AWS with public and private subnets, Internet Gateway, NAT Gateway, EC2 instances, and a Security Group. Here's a breakdown of the resources:

1. VPC(MyVPC):
 - Creates a VPC with the CIDR block 10.0.0.0/16.
 - Enables DNS support and hostnames.
2. Public Subnet(PublicSubnet):
 - Creates a public subnet within the VPC with the CIDR block 10.0.1.0/24.
 - Enables auto-assignment of public IP addresses.
3. Private Subnet (PrivateSubnet):
 - Creates a private subnet within the VPC with the CIDR block 10.0.2.0/24.
4. Internet Gateway (InternetGateway):
 - Creates an Internet Gateway and attaches it to the VPC.
5. Attach Gateway (AttachGateway):
 - Attaches the Internet Gateway to the VPC.
6. Public Route Table (PublicRouteTable):

- Creates a route table for the public subnet.
- 7. Public Route (PublicRoute):
 - Adds a route to the Internet Gateway in the public route table.
- 8. Subnet Route Table Association (AssociatePublicSubnetRouteTable):
 - Associates the public subnet with the public route table.
- 9. NAT Gateway (NATGateway):
 - Creates a NAT Gateway in the public subnet.
 - Depends on the Internet Gateway attachment.
 - Requires an Elastic IP (ElasticIP) for the NAT Gateway.
- 10. Elastic IP (ElasticIP):
 - Allocates an Elastic IP for the NAT Gateway.
- 11. Private Route Table (PrivateRouteTable):
 - Creates a route table for the private subnet.
- 12. Private Route (PrivateRoute):
 - Adds a route to the NAT Gateway in the private route table.
- 13. Subnet Route Table Association (AssociatePrivateSubnetRouteTable):
 - Associates the private subnet with the private route table.
- 14. Security Group (MySecurity):
 - Creates a security group allowing SSH (port 22), HTTP (port 80), Tomcat (port 8080), and MySQL (port 3306) traffic.
- 15. Public Instance (PublicInstance):
 - Launches a public EC2 instance in the public subnet.
- 17. Private Instance (PrivateInstance):
 - Launches a private EC2 instance in the private subnet.
 - Uses user data to start Apache Tomcat upon instance creation.

Outputs:

- VPCId: Outputs the ID of the created VPC.
- PublicSubnetId: Outputs the ID of the public subnet.
- PrivateSubnetId: Outputs the ID of the private subnet.
- PublicInstanceId: Outputs the ID of the public EC2 instance.
- PrivateInstanceId: Outputs the ID of the private EC2 instance.



CloudFormation > Stacks > three-ten

Stacks (1)

Filter by stack name

Filter status: Active View nested

three-ten
2024-02-17 13:54:20 UTC+0530
CREATE_IN_PROGRESS

Stack info Events Resources Outputs Parameters Template Change sets Git sync - new

Resources (16)

Logical ID	Physical ID	Type	Status	Module
AssociatePrivateSubnetRouteTable	rtbassoc-090bce48a264fe758	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
AssociatePublicSubnetRouteTable	rtbassoc-00902f1f63f34174	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
AttachGateway	igwvpc-05a76e42f9ea3121d	AWS::EC2::VPCEGatewayAttachment	CREATE_COMPLETE	-
ElasticIP	eip-3.19.157.206	AWS::EC2::EIP	CREATE_COMPLETE	-
InternetGateway	igw-0713f151866612a3f	AWS::EC2::InternetGateway	CREATE_COMPLETE	-
MySecurity	sg-0f778835caf1b668f	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
MyVPC	vpc-05a76e42f9ea3121d	AWS::EC2::VPC	CREATE_COMPLETE	-
NATGateway	nat-0b81c145427f65f49	AWS::EC2::NatGateway	CREATE_COMPLETE	-
PrivateInstance	i-07d85578842329884	AWS::EC2::Instance	CREATE_COMPLETE	-