1. **Write a cloud formation template to create a public and private subnet-understanding routing and subnet configuration(vpc)**

Step 1: Create a CloudFormation Template – Public & Private Subnet Setup

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: Create a vpc with a public and private subnet

Resources:
  MyVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: MyVPC

  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: MyInternetGateway

  AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId: !Ref MyVPC
      InternetGatewayId: !Ref InternetGateway

  PublicSubnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref MyVPC
      CidrBlock: 10.0.1.0/24
      AvailabilityZone: !Select [ 0, !GetAZs '' ]
      MapPublicIpOnLaunch: true
      Tags:
        - Key: Name
          Value: PublicSubnet
```

```yaml
PrivateSubnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref MyVPC
    CidrBlock: 10.0.2.0/24
    AvailabilityZone: !Select [ 1, !GetAZs '' ]
    MapPublicIpOnLaunch: false
    Tags:
      - Key: Name
        Value: PrivateSubnet

PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref MyVPC
    Tags:
      - Key: Name
        Value: PublicRouteTable

PublicRoute:
  Type: AWS::EC2::Route
  DependsOn: AttachGateway
  Properties:
    RouteTableId: !Ref PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

PublicSubnetRouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PublicSubnet
    RouteTableId: !Ref PublicRouteTable

PrivateRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref MyVPC
    Tags:
      - Key: Name
        Value: PrivateRouteTable

PrivateSubnetRouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PrivateSubnet
    RouteTableId: !Ref PrivateRouteTable
```

Step 2: Deploy the Template Using the AWS Management Console

1.Sign In to AWS Management Console

• Open the AWS Management Console and sign in with your credentials.

2. Navigate to CloudFormation

• In the AWS Services search bar, type CloudFormation and select it.

3. Create a New Stack

• Click on Create stack and then select With new resources.

4. Choose Template Source

• Choose Upload a template file and click Choose file to upload the CloudFormation template (vpc.yaml) that you have created.

5.Provide Stack Name:

• Enter a stack name (e.g., substack) and click Next.

6.Configure Stack Options:

• You can leave the options as default or configure them as needed. Click Next.

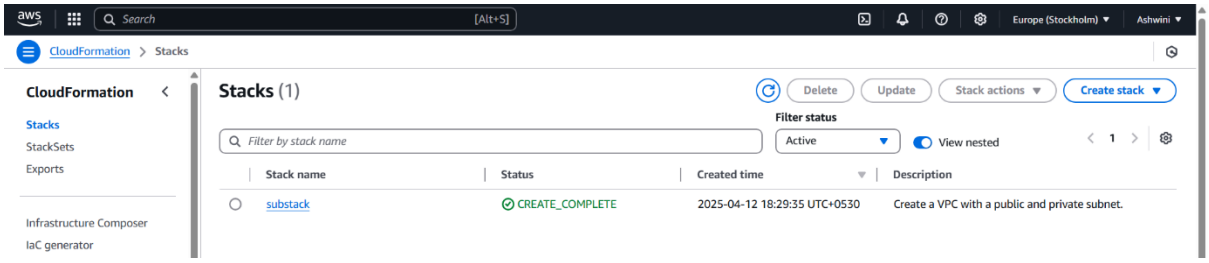
7.Review and Create:

• Review the configuration and click Create stack.

Step 3: Monitor Stack Creation:

• CloudFormation will start provisioning the resources. The process may take a few minutes.

• You can monitor the status in the CloudFormation Console. When the stack creation is complete.



2.Create an internet gateway and attach it to the vpc using cloud formation.

Step 1: Create a CloudFormation – Create and Attach Internet Gateway

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: Create Internet Gateway and attach to VPC

Resources:
  MyVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: MyVPC

  MyInternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: MyInternetGateway

  AttachGatewayToVPC:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId: !Ref MyVPC
      InternetGatewayId: !Ref MyInternetGateway
```

=Step 2: Deploy the Template Using the AWS Management Console

1.Sign In to AWS Management Console

- Open the AWS Management Console and sign in with your credentials.

2. Navigate to CloudFormation

- In the AWS Services search bar, type CloudFormation and select it.

3. Create a New Stack

- Click on Create stack and then select With new resources.

4. Choose Template Source

- Choose Upload a template file and click Choose file to upload the CloudFormation template (vpc-igw.yaml) that you have created.

5.Provide Stack Name:

• Enter a stack name (e.g., intgate) and click Next.

6.Configure Stack Options:

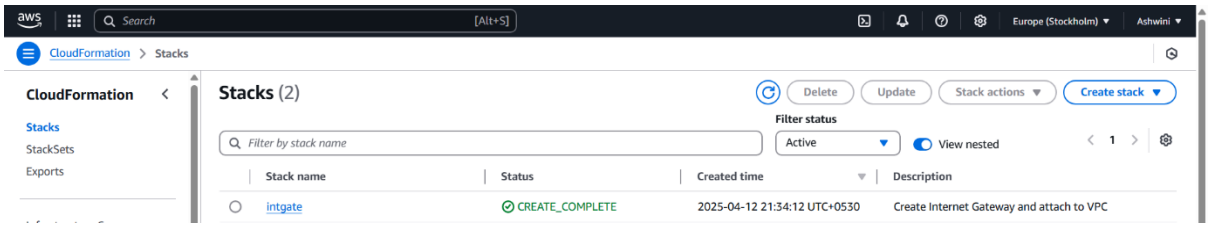
• You can leave the options as default or configure them as needed. Click Next.

7.Review and Create:

• Review the configuration and click Create stack.

Step 3: Monitor Stack Creation:

- CloudFormation will start provisioning the resources. The process may take a few minutes.
- You can monitor the status in the CloudFormation Console. When the stack creation is complete



3.Add a nat gateway to allow private subnets to access the internet.

Step 1: Create a CloudFormation Template: NAT Gateway Setup

```
AWSTemplateFormatVersion: '2010-09-09'
Description: vpc with nat gateway for private subnet internet access

Resources:
MyVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: MyVPC

  #InternetGateway:
MyInternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: MyInternetGateway

  AttachGatewayToVPC:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId: !Ref MyVPC
      InternetGatewayId: !Ref InternetGateway

  #PublicSubnet:
PublicSubnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref MyVPC
      CidrBlock: 10.0.1.0/24
      AvailabilityZone: !Select [ 0, !GetAZs '' ]
      MapPublicIpOnLaunch: true
      Tags:
        - Key: Name
          Value: PublicSubnet

  PrivateSubnet:
    Type: AWS::EC2::Subnet
```

```yaml
  Properties:
    VpcId: !Ref MyVPC
    CidrBlock: 10.0.2.0/24
    AvailabilityZone: !Select [ 1, !GetAZs '' ]
    MapPublicIpOnLaunch: false
    Tags:
      - Key: Name
        Value: PrivateSubnet

PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref MyVPC
    Tags:
      - Key: Name
        Value: PublicRouteTable

PublicRoute:
  Type: AWS::EC2::Route
  DependsOn: AttachGateway
  Properties:
    RouteTableId: !Ref PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

PublicSubnetRouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PublicSubnet
    RouteTableId: !Ref PublicRouteTable

PrivateRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref MyVPC
    Tags:
      - Key: Name
        Value: PrivateRouteTable

PrivateSubnetRouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PrivateSubnet
    RouteTableId: !Ref PrivateRouteTable

PrivateRoute:
  Type: AWS::EC2::Route
  Properties:
```

```
      RouteTableId: !Ref PrivateRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !Ref NatGateway

Outputs:
  VPCId:
    Value: !Ref MyVPC
    Export:
      Name: MyVPCId

  PublicSubnetId:
    Value: !Ref PublicSubnet

  PrivateSubnetId:
    Value: !Ref PrivateSubnet

  NatGatewayId:
    Value: !Ref NatGateway
```

Step 2: Deploy the Template Using the AWS Management Console

1.Sign In to AWS Management Console

• Open the AWS Management Console and sign in with your credentials.

2. Navigate to CloudFormation

• In the AWS Services search bar, type CloudFormation and select it.

3. Create a New Stack

• Click on Create stack and then select With new resources.

4. Choose Template Source

• Choose Upload a template file and click Choose file to upload the CloudFormation template (nat-gateway-stack.yaml) that you have created.

5.Provide Stack Name:

• Enter a stack name (e.g., natstack) and click Next.

1. Configure Stack Options:

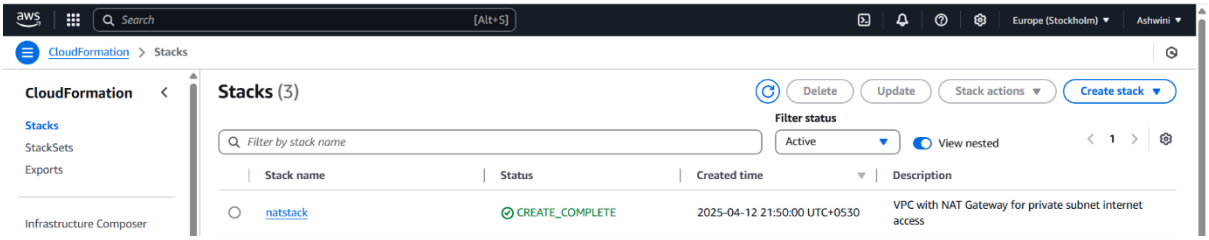
• You can leave the options as default or configure them as needed. Click Next.

7.Review and Create:

• Review the configuration and click Create stack.

Step 3: Monitor Stack Creation:

- CloudFormation will start provisioning the resources. The process may take a few minutes.
- You can monitor the status in the CloudFormation Console. When the stack creation is complete


aws
Search
[Alt+S]
Europe (Stockholm)
Ashwini
CloudFormation
Stacks
CloudFormation
Stacks
StackSets
Exports
Infrastructure Composer
Stacks (3)
Delete
Update
Stack actions
Create stack
Filter by stack name
Filter status
Active
View nested
1
Stack name
Status
Created time
Description
natstack
CREATE_COMPLETE
2025-04-12 21:50:00 UTC+0530
VPC with NAT Gateway for private subnet internet access