# Module 8: Case Study - 1

## Problem Statement:

You work for XYZ Corporation. Your corporation wants to launch a new web-based application. The development team has prepared the code but it is not tested yet. The development team needs the system admins to build a web server to test the code but the system admins are not available.

## Tasks To Be Performed:

1. Web tier: Launch an instance in a public subnet and that instance should allow HTTP and SSH from the internet.
2. Application tier: Launch an instance in a private subnet of the web tier and it should allow only SSH from the public subnet of Web Tier-3.
3. DB tier: Launch an RDS MYSQL instance in a private subnet and it should allow connection on port 3306 only from the private subnet of Application Tier-4.
4. Setup a Route 53 hosted zone and direct traffic to the EC2 instance.

You have been also asked to propose a solution so that:

1. Development team can test their code without having to involve the system admins and can invest their time in testing the code rather than provisioning, configuring and updating the resources needed to test the code.
2. Make sure when the development team deletes the stack, RDS DB instances should not be deleted.

```
Parameters:
  EnvironmentName:
    Description: An environment name that is prefixed to resource names
    Type: String

  VpcCIDR:
    Description: Please enter the IP range (CIDR notation) for this VPC
    Type: String
    Default: 10.192.0.0/16

  PublicSubnetCIDR:
    Description: Please enter the IP range (CIDR notation) for the public subnet in
the first Availability Zone
    Type: String
    Default: 10.192.10.0/24

  PrivateSubnetCIDR:
    Description: Please enter the IP range (CIDR notation) for the private subnet
in the first Availability Zone
    Type: String
    Default: 10.192.20.0/24
  KeyName:
    Description: 'Optional key pair of the ec2-user to establish a SSH connection
to the EC2 instance.'
    Type: AWS::EC2::KeyPair::KeyName
  InstanceType:
    Description: 'The instance type for the EC2 instance.'
    Type: String
    Default: 't2.small'
    AllowedValues:
      - t1.micro
      - t2.nano
      - t2.micro
      - t2.small
      - t2.medium
      - t2.large
      - m1.small
      - m1.medium
      - m1.large
      - m1.xlarge
      - m2.xlarge
      - m2.2xlarge
      - m2.4xlarge
      - m3.medium
      - m3.large
      - m3.xlarge
      - m3.2xlarge
      - m4.large
      - m4.xlarge
      - m4.2xlarge
      - m4.4xlarge
```

```
      - m4.10xlarge
      - c1.medium
      - c1.xlarge
      - c3.large
      - c3.xlarge
      - c3.2xlarge
      - c3.4xlarge
      - c3.8xlarge
      - c4.large
      - c4.xlarge
      - c4.2xlarge
      - c4.4xlarge
      - c4.8xlarge
      - g2.2xlarge
      - g2.8xlarge
      - r3.large
      - r3.xlarge
      - r3.2xlarge
      - r3.4xlarge
      - r3.8xlarge
      - i2.xlarge
      - i2.2xlarge
      - i2.4xlarge
      - i2.8xlarge
      - d2.xlarge
      - d2.2xlarge
      - d2.4xlarge
      - d2.8xlarge
      - hi1.4xlarge
      - hs1.8xlarge
      - cr1.8xlarge
      - cc2.8xlarge
      - cg1.4xlarge
  Name:
    Description: 'The name for the EC2 instance.'
    Type: String
    Default: 'test'
  RootVolumeSize:
    Description: 'The root volume size, in Gibibytes (GiB) (if RestoreImageId is
set, value must be >= snapshot of AMI).'
    Type: Number
    Default: 8
    ConstraintDescription: 'Must be in the range [8-1024]'
    MinValue: 8
    MaxValue: 1024
  DBName:
    Default: MyDatabase
    Description: The database name
    Type: String
    MinLength: '1'
    MaxLength: '64'
```

```
    AllowedPattern: '[a-zA-Z][a-zA-Z0-9]*'
    ConstraintDescription: must begin with a letter and contain only alphanumeric
      characters.
  DBUser:
    NoEcho: 'true'
    Description: The database admin account username
    Type: String
    MinLength: '1'
    MaxLength: '16'
    AllowedPattern: '[a-zA-Z][a-zA-Z0-9]*'
    ConstraintDescription: must begin with a letter and contain only alphanumeric
      characters.
  DBPassword:
    NoEcho: 'true'
    Description: The database admin account password
    Type: String
    MinLength: '8'
    MaxLength: '41'
    AllowedPattern: '[a-zA-Z0-9]*'
    ConstraintDescription: must contain only alphanumeric characters
  DNSName:
    Description: Enter the DNS Name
    Type: String
Mappings:
  AWSInstanceType2Arch:
    c1.medium:
      Arch: HVM64
    c1.xlarge:
      Arch: HVM64
    c3.2xlarge:
      Arch: HVM64
    c3.4xlarge:
      Arch: HVM64
    c3.8xlarge:
      Arch: HVM64
    c3.large:
      Arch: HVM64
    c3.xlarge:
      Arch: HVM64
    c4.2xlarge:
      Arch: HVM64
    c4.4xlarge:
      Arch: HVM64
    c4.8xlarge:
      Arch: HVM64
    c4.large:
      Arch: HVM64
    c4.xlarge:
      Arch: HVM64
    cc2.8xlarge:
      Arch: HVM64
```

```
    cr1.8xlarge:
      Arch: HVM64
    d2.2xlarge:
      Arch: HVM64
    d2.4xlarge:
      Arch: HVM64
    d2.8xlarge:
      Arch: HVM64
    d2.xlarge:
      Arch: HVM64
    g2.2xlarge:
      Arch: HVMG2
    g2.8xlarge:
      Arch: HVMG2
    hi1.4xlarge:
      Arch: HVM64
    hs1.8xlarge:
      Arch: HVM64
    i2.2xlarge:
      Arch: HVM64
    i2.4xlarge:
      Arch: HVM64
    i2.8xlarge:
      Arch: HVM64
    i2.xlarge:
      Arch: HVM64
    m1.large:
      Arch: HVM64
    m1.medium:
      Arch: HVM64
    m1.small:
      Arch: HVM64
    m1.xlarge:
      Arch: HVM64
    m2.2xlarge:
      Arch: HVM64
    m2.4xlarge:
      Arch: HVM64
    m2.xlarge:
      Arch: HVM64
    m3.2xlarge:
      Arch: HVM64
    m3.large:
      Arch: HVM64
    m3.medium:
      Arch: HVM64
    m3.xlarge:
      Arch: HVM64
    m4.10xlarge:
      Arch: HVM64
    m4.2xlarge:
```

```
    Arch: HVM64
  m4.4xlarge:
    Arch: HVM64
  m4.large:
    Arch: HVM64
  m4.xlarge:
    Arch: HVM64
  r3.2xlarge:
    Arch: HVM64
  r3.4xlarge:
    Arch: HVM64
  r3.8xlarge:
    Arch: HVM64
  r3.large:
    Arch: HVM64
  r3.xlarge:
    Arch: HVM64
  t1.micro:
    Arch: HVM64
  t2.large:
    Arch: HVM64
  t2.medium:
    Arch: HVM64
  t2.micro:
    Arch: HVM64
  t2.nano:
    Arch: HVM64
  t2.small:
    Arch: HVM64
AWSRegionArch2AMI:
  af-south-1:
    HVM64: ami-0412806bd0f2cf75f
    HVMG2: NOT_SUPPORTED
  ap-east-1:
    HVM64: ami-0900a8f768a21540a
    HVMG2: NOT_SUPPORTED
  ap-northeast-1:
    HVM64: ami-0c3e3e7af817ad732
    HVMG2: NOT_SUPPORTED
  ap-northeast-2:
    HVM64: ami-0f8dbbf156e3a5cc6
    HVMG2: NOT_SUPPORTED
  ap-northeast-3:
    HVM64: ami-02a371c41f08cc499
    HVMG2: NOT_SUPPORTED
  ap-south-1:
    HVM64: ami-0f4ab3c8db917e421
    HVMG2: NOT_SUPPORTED
  ap-south-2:
    HVM64: ami-008b9c53bb1dcd29c
    HVMG2: NOT_SUPPORTED
```

```
    ap-southeast-1:
      HVM64: ami-0c3189395e5b39df7
      HVMG2: NOT_SUPPORTED
    ap-southeast-2:
      HVM64: ami-040d698318c0b1575
      HVMG2: NOT_SUPPORTED
    ap-southeast-3:
      HVM64: ami-065dcca47dde26602
      HVMG2: NOT_SUPPORTED
    ap-southeast-4:
      HVM64: ami-043e25432cf94e107
      HVMG2: NOT_SUPPORTED
    il-central-1:
      HVM64: ami-0054be7d7d9d65a1d
      HVMG2: NOT_SUPPORTED
    ca-central-1:
      HVM64: ami-05f40104305a2cdf7
      HVMG2: NOT_SUPPORTED
    cn-north-1:
      HVM64: ami-03f1e08d409b1e5fd
      HVMG2: NOT_SUPPORTED
    cn-northwest-1:
      HVM64: ami-00093746b9a0e272a
      HVMG2: NOT_SUPPORTED
    eu-central-1:
      HVM64: ami-0f454ec961da9a046
      HVMG2: NOT_SUPPORTED
    eu-north-1:
      HVM64: ami-0e78cd18c67fcf512
      HVMG2: NOT_SUPPORTED
    eu-south-1:
      HVM64: ami-07d048788725b9602
      HVMG2: NOT_SUPPORTED
    eu-west-1:
      HVM64: ami-0db5ca3e5748fb7e2
      HVMG2: NOT_SUPPORTED
    eu-west-2:
      HVM64: ami-07baf6b15b7387f24
      HVMG2: NOT_SUPPORTED
    eu-west-3:
      HVM64: ami-05a13fbd8aa57eedc
      HVMG2: NOT_SUPPORTED
    me-south-1:
      HVM64: ami-0007de3fdcaba7e44
      HVMG2: NOT_SUPPORTED
    me-central-1:
      HVM64: ami-06ce88defa3fc74ed
      HVMG2: NOT_SUPPORTED
    eu-south-2:
      HVM64: ami-051306f4e885d6de4
      HVMG2: NOT_SUPPORTED
```

```
    eu-central-2:
      HVM64: ami-0fcd532574732cb0f
      HVMG2: NOT_SUPPORTED
    sa-east-1:
      HVM64: ami-07f6e9fce0e888425
      HVMG2: NOT_SUPPORTED
    us-east-1:
      HVM64: ami-01bc990364452ab3e
      HVMG2: NOT_SUPPORTED
    us-east-2:
      HVM64: ami-0de69dde1945155da
      HVMG2: NOT_SUPPORTED
    us-west-1:
      HVM64: ami-08fe20a82dcaa1c92
      HVMG2: NOT_SUPPORTED
    us-west-2:
      HVM64: ami-05848d23360f5edfe
      HVMG2: NOT_SUPPORTED
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VpcCIDR
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: !Ref EnvironmentName

  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: !Ref EnvironmentName

  InternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref VPC

  PublicSubnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 0, !GetAZs '' ]
      CidrBlock: !Ref PublicSubnetCIDR
      MapPublicIpOnLaunch: true
      Tags:
        - Key: Name
```

```
        Value: !Sub ${EnvironmentName} Public Subnet

PrivateSubnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 1, !GetAZs  '' ]
    CidrBlock: !Ref PrivateSubnetCIDR
    MapPublicIpOnLaunch: false
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Subnet

PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Public Routes

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

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

PrivateRouteTable1:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Routes (AZ1)

PrivateSubnetRouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable1
    SubnetId: !Ref PrivateSubnet

WebtierSG:
  Type: AWS::EC2::SecurityGroup
```

```
  Properties:
    GroupName: "WebtierSG"
    GroupDescription: Allows ssh and HTTP connection from internet
    SecurityGroupIngress:
              -
                  CidrIp: 0.0.0.0/0
                  FromPort: '22'
                  IpProtocol: tcp
                  ToPort: '22'
              -
                  CidrIp: 0.0.0.0/0
                  FromPort: '80'
                  IpProtocol: tcp
                  ToPort: '80'
    VpcId: !Ref VPC
ApplicationtierSG:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "ApplicationtierSG"
    GroupDescription: Allows ssh connection from publicsubnet
    SecurityGroupIngress:
              -
                  CidrIp: !Ref  PublicSubnetCIDR
                  FromPort: '22'
                  IpProtocol: tcp
                  ToPort: '22'
    VpcId: !Ref VPC
DBtierSG:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "DBtierSG"
    GroupDescription: Allows ssh connection  from priviatesubnet
    SecurityGroupIngress:
              -
                  CidrIp: !Ref  PrivateSubnetCIDR
                  FromPort: '22'
                  IpProtocol: tcp
                  ToPort: '22'
    VpcId: !Ref VPC
Webinstance:
  Type: AWS::EC2::Instance
  Properties:
    InstanceType: !Ref InstanceType
    ImageId: !FindInMap
      - AWSRegionArch2AMI
      - !Ref 'AWS::Region'
      - !FindInMap
        - AWSInstanceType2Arch
        - !Ref InstanceType
        - Arch
    KeyName: !Ref KeyName
```

```
    UserData:
      Fn::Base64:
        !Sub |
          #!/bin/bash
          yum update -y
          yum install -y httpd
          systemctl start httpd.service
          systemctl enable httpd.service
          echo ?Hello World from $(hostname -f)? > /var/www/html/index.html
    SecurityGroupIds:
     - !GetAtt WebtierSG.GroupId
    SubnetId:
     !GetAtt PublicSubnet.SubnetId
    Tags:
      - Key: Name
        Value: Webinstance
Appinstance:
  Type: AWS::EC2::Instance
  Properties:
    InstanceType: !Ref InstanceType
    ImageId: !FindInMap
      - AWSRegionArch2AMI
      - !Ref 'AWS::Region'
      - !FindInMap
        - AWSInstanceType2Arch
        - !Ref InstanceType
        - Arch
    KeyName: !Ref KeyName
    SecurityGroupIds:
      - !GetAtt ApplicationtierSG.GroupId
    SubnetId:
     !GetAtt PrivateSubnet.SubnetId
    Tags:
      - Key: Name
        Value: Appinstance
MyDBSubnetGroup:
  Type: AWS::RDS::DBSubnetGroup
  Properties:
    DBSubnetGroupDescription: My DBSubnetGroup for RDS
    SubnetIds:
      - !GetAtt PrivateSubnet.SubnetId
      - !GetAtt PublicSubnet.SubnetId
MyDB:
  Type: AWS::RDS::DBInstance
  DeletionPolicy: Retain
  Properties:
    DBName: !Ref 'DBName'
    AllocatedStorage: '5'
    DBInstanceClass: db.t2.small
    Engine: MySQL
    EngineVersion: 5.7.37
```

```
    MasterUsername: !Ref 'DBUser'
    MasterUserPassword: !Ref 'DBPassword'
    Port: '3306'
    VPCSecurityGroups:
     - !Ref DBtierSG
    DBSubnetGroupName: !Ref MyDBSubnetGroup
Hostedzone:
  Type: "AWS::Route53::HostedZone"
  Properties:
   HostedZoneConfig:
    Comment: 'My hosted zone for example.com'
   Name: !Ref DNSName
   VPCs:
    -
      VPCId: !Ref VPC
      VPCRegion: 'us-east-1'
   HostedZoneTags:
    -
      Key: Name
      Value: DS Design
myDNSRecord:
  Type: AWS::Route53::RecordSet
  Properties:
    HostedZoneId : !Ref Hostedzone
    Name: !Ref DNSName
    ResourceRecords:
    - !GetAtt Webinstance.PublicIp
    TTL: 900
    Type: A
```