AWS Solutions Architect Certification Training
support@intellipaat.com - +91-7022374614 - US: 1-800-216-8930 (Toll Free)

Module 7 - Assignment IaaC App Cloud Formation COMPLETED by Nagesha. K.S.
Please check the following screenshots for each question.

Problem Statement:
You work for XYZ Corporation. Your company 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 System Admins to build a web server to test the code, but the System Admins are not available.
You are asked to create three-tier architecture and perform following tasks in each tier:

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

You are also asked to propose a solution so that:

1. The development team can test the 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. When the development team deletes the stack,the RDS DB instance should not be deleted

I referred the following link to better understand AWS cloud YAML
https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-formats.html

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/format-version-structure.html
Template reference > Resource and property reference > Amazon EC2 > AWS::EC2::Instance

However, I keep encountering compilation errors despite numerous attempts—over 20 in the past 2 to 3 days.

I would greatly appreciate your help in resolving this assignment.

Thank you and regards,
Nagesha. K.S

```
AWSTemplateFormatVersion: '2010-09-09'

Resources:
  # Web Tier
  WebInstance:
    Type: 'AWS::EC2::Instance'
    Properties:
      ImageId: ami-03972092c42e8c0ca # Amazon Linux 2 AMI (HVM)
      InstanceType: 't2.micro'
      SubnetId: !Ref WebSubnet
      SecurityGroups:
        - !Ref WebSecurityGroup

  WebSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: 'Web Security Group'
      VpcId: !Ref VPC
      SecurityGroupIngress:
        - IpProtocol: 'tcp'
          FromPort: '80'
          ToPort: '80'
          CidrIp: '0.0.0.0/0'
        - IpProtocol: 'tcp'
          FromPort: '22'
          ToPort: '22'
          CidrIp: '0.0.0.0/0'

  WebSubnet:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref VPC
      CidrBlock: '10.0.1.0/24'
      AvailabilityZone: 'us-east-1a'

  # Application Tier
  AppInstance:
    Type: 'AWS::EC2::Instance'
    Properties:
      ImageId: ami-03972092c42e8c0ca # Amazon Linux 2 AMI (HVM)
      InstanceType: 't2.micro'
      SubnetId: !Ref AppSubnet
      SecurityGroups:
        - !Ref AppSecurityGroup

  AppSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: 'App Security Group'
      VpcId: !Ref VPC
      SecurityGroupIngress:
        - IpProtocol: 'tcp'
          FromPort: '22'
          ToPort: '22'
          SourceSecurityGroupId: !GetAtt WebSecurityGroup.GroupId
```

```
AppSubnet:
  Type: 'AWS::EC2::Subnet'
  Properties:
    VpcId: !Ref VPC
    CidrBlock: '10.0.2.0/24'
    AvailabilityZone: 'us-east-1a'
    # AvailabilityZone: !GetAtt EC2Instance.AvailabilityZone

# DB Tier
DBInstance:
  Type: 'AWS::RDS::DBInstance'
  Properties:
    Engine: 'MySQL'
    DBInstanceClass: 'db.t3.micro'
    SubnetGroup:
      - !Ref DBSubnetGroup
    VPCSecurityGroups:
      - !GetAtt DBSecurityGroup.GroupId

DBSecurityGroup:
  Type: 'AWS::EC2::SecurityGroup'
  Properties:
    GroupDescription: 'DB Security Group'
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: 'tcp'
        FromPort: '3306'
        ToPort: '3306'
        SourceSecurityGroupId: !GetAtt AppSecurityGroup.GroupId

DBSubnetGroup:
  Type: 'AWS::RDS::DBSubnetGroup'
  Properties:
    DBSubnetGroupDescription: "Subnet group for RDS instances"
    SubnetIds:
      - !Ref DBSubnet

DBSubnet:
  Type: 'AWS::EC2::Subnet'
  Properties:
    VpcId: !Ref VPC
    CidrBlock: '10.0.3.0/24'
    AvailabilityZone: 'us-east-1a'

# Route 53
HostedZone:
  Type: 'AWS::Route53::HostedZone'
  Properties:
    Name: 'nagcorp.com'

RecordSet:
  Type: 'AWS::Route53::RecordSet'
  Properties:
    HostedZoneId: !Ref HostedZone
    Name: '(link unavailable)'
    Type: 'A'
```

```
      AliasTarget:
        DNSName: !GetAtt WebInstance.PublicDnsName
        HostedZoneId: !GetAtt WebInstance.VpcId

  # VPC
  VPC:
    Type: 'AWS::EC2::VPC'
    Properties:
      CidrBlock: '10.0.0.0/16'
```

error messages

The following resource(s) failed to create: [DBSubnetGroup, AppSecurityGroup, WebInstance, HostedZone]. Rollback requested by user.

Resource handler returned message: "The DB subnet group doesn't meet Availability Zone (AZ) coverage requirement. Current AZ coverage: us-east-1a. Add subnets to cover at least 2 AZs. (Service: Rds, Status Code: 400, Request ID: 7890a74c-d70b-4a89-9be9-d9b536aae558)" (RequestToken: ac37ebf2-ad87-19de-a830-28fc190956c1, HandlerErrorCode: InvalidRequest)

Resource handler returned message: "The parameter groupName cannot be used with the parameter subnet (Service: Ec2, Status Code: 400, Request ID: fa4ffb7c-f210-492f-9c75-fceff09037ad)" (RequestToken: 871fd218-3b53-5f85-20cd-f44503d4e9f9, HandlerErrorCode: InvalidRequest)