AWSTemplateFormatVersion: '2010-09-09'

Description: Multi-Tier Architecture with EC2 Instances and RDS

Parameters:

WebInstanceType:

Type: String

Default: t2.micro

Description: "EC2 Instance Type for Web Tier"

AppInstanceType:

Type: String

Default: t2.micro

Description: "EC2 Instance Type for Application Tier"

DBInstanceType:

Type: String

Default: db.t2.micro

Description: "RDS DB Instance Type"

DBUsername:

Type: String

Description: "Database master username"

DBPassword:

Type: String

NoEcho: true

Description: "Database master password"

HostedZoneName:

Type: String

Description: "Hosted Zone Name (e.g., example.com)"

Resources:

VPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: 10.0.0.0/16

EnableDnsSupport: true

EnableDnsHostnames: true

Tags:

- Key: Name

Value: "MyVPC"

PublicSubnet:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC

CidrBlock: 10.0.1.0/24

AvailabilityZone: !Select [0, !GetAZs '']

MapPublicIpOnLaunch: true

Tags:

- Key: Name

Value: "PublicSubnet"

PrivateSubnetApp:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC

CidrBlock: 10.0.2.0/24

AvailabilityZone: !Select [0, !GetAZs '']

Tags:

- Key: Name

Value: "PrivateSubnetApp"

PrivateSubnetDB:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC

CidrBlock: 10.0.3.0/24

AvailabilityZone: !Select [0, !GetAZs '']

Tags:

- Key: Name

Value: "PrivateSubnetDB"

InternetGateway:

Type: AWS::EC2::InternetGateway

AttachGateway:

Type: AWS::EC2::VPCGatewayAttachment

Properties:

VpcId: !Ref VPC

InternetGatewayId: !Ref InternetGateway

PublicRouteTable:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref VPC

Tags:

- Key: Name

Value: "PublicRouteTable"

PublicRoute:

Type: AWS::EC2::Route

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

WebInstance:

Type: AWS::EC2::Instance

Properties:

InstanceType: !Ref WebInstanceType

ImageId: ami-0c55b159cbfafe1fe # Replace with the latest Amazon Linux 2 AMI ID for your region

SubnetId: !Ref PublicSubnet

KeyName: your-key-pair # Replace with your EC2 key pair

SecurityGroupIds:

- !Ref WebInstanceSecurityGroup

WebInstanceSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: "Enable HTTP and SSH access"

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

AppInstance:

Type: AWS::EC2::Instance

Properties:

InstanceType: !Ref AppInstanceType

ImageId: ami-0c55b159cbfafe1fe # Replace with the latest Amazon Linux 2 AMI ID for your region

SubnetId: !Ref PrivateSubnetApp

KeyName: your-key-pair # Replace with your EC2 key pair

SecurityGroupIds:

- !Ref AppInstanceSecurityGroup

AppInstanceSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: "Allow SSH access from Web Tier"

VpcId: !Ref VPC

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 22

ToPort: 22

SourceSecurityGroupId: !Ref WebInstanceSecurityGroup

DBInstance:

Type: AWS::RDS::DBInstance

Properties:

DBInstanceClass: !Ref DBInstanceType

AllocatedStorage: 20

Engine: mysql

MasterUsername: !Ref DBUsername

MasterUserPassword: !Ref DBPassword

DBInstanceIdentifier: "MyDBInstance"

VPCSecurityGroups:

- !Ref DBSecurityGroup

DBSubnetGroupName: !Ref DBSubnetGroup

MultiAZ: false

DeletionPolicy: Snapshot # Ensure RDS is retained when stack is deleted

DBSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: "Allow MySQL access from Application Tier"

VpcId: !Ref VPC

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 3306

ToPort: 3306

SourceSecurityGroupId: !Ref AppInstanceSecurityGroup

DBSubnetGroup:

Type: AWS::RDS::DBSubnetGroup

Properties:

DBSubnetGroupDescription: "Subnet group for RDS"

SubnetIds:

- !Ref PrivateSubnetDB

DBSubnetGroupName: "MyDBSubnetGroup"

HostedZone:

Type: AWS::Route53::HostedZone

Properties:

Name: !Ref HostedZoneName

ARecord:

Type: AWS::Route53::RecordSet

Properties:

HostedZoneName: !Join [".", [!Ref HostedZoneName, ""]]

Name: !Join [".", [!Ref HostedZoneName, "www"]]

Type: A

AliasTarget:

DNSName: !GetAtt WebInstance.PublicDnsName

HostedZoneId: "Z3AQBSTGFYJSTF" # Change this value to the correct one for your region

Outputs:

WebInstancePublicDNS:

Description: "Public DNS of the Web Instance"

Value: !GetAtt WebInstance.PublicDnsName

DBInstanceEndpoint:

Description: "Endpoint of the RDS instance"

Value: !GetAtt DBInstance.Endpoint.Address



