Assignment: **Auto Deployment using Cloudformation Template**

* Cloud formation template in Yaml

1. Open the delta-vpc-cfn.yaml attached in the assignment
2. Go through the file and understanding the code
3. You need to create a couple of resources in the yaml file for creating security group and instance of Type: AWS: :EC2::SecurityGroup & Type : AWS::EC2::instance
4. You need to use Nginx ami image id for instance creation.
5. You need to allow port 80 as inbound and outbound, port 22 as inbound
6. Once you added these resources you can save the file
7. Navigate to Cloud formation page
8. Click on create stack
9. In the Template source select upload a template file
10. Choose the modified codebuild-vpc-cfn.yaml click on next
11. Provide the stack name
12. Click next
13. You can add tags, leave other as default and click Next
14. Review your stack details once and you can click on create stack
15. You can see the status of the stack when it started creating

A screenshot of a computer

Description automatically generated

1. You can see the status create \_complete if you don’t have any errors in yaml file.
2. Once your stack is created you can go a head and verify all the resources created and instance is in running state.
3. Now you can access the nginx server using the public Ip of the instance.

Graphical user interface, text, application, email

Description automatically generated

1. Once you delete the stack all the resources created using this yaml file will be deleted.

Yaml script used for provision of Nginx server (instance) and security groups inside the vpc

Description:  This template deploys a VPC, with a pair of public and private subnets spread

  across two Availability Zones. It deploys an internet gateway, with a default

  route on the public subnets. It deploys a pair of NAT gateways (one in each AZ),

  and default routes for them in the private subnets.

Parameters:

  EnvironmentName:

    Description: An environment name that is prefixed to resource names

    Type: String

    Default: Shiva-CFN

  VpcCIDR:

    Description: Please enter the IP range (CIDR notation) for this VPC

    Type: String

    Default: 10.10.0.0/16

  PublicSubnet1CIDR:

    Description: Please enter the IP range (CIDR notation) for the public subnet in the first Availability Zone

    Type: String

    Default: 10.10.1.0/24

  PublicSubnet2CIDR:

    Description: Please enter the IP range (CIDR notation) for the public subnet in the second Availability Zone

    Type: String

    Default: 10.10.2.0/24

  PrivateSubnet1CIDR:

    Description: Please enter the IP range (CIDR notation) for the private subnet in the first Availability Zone

    Type: String

    Default: 10.10.101.0/24

  PrivateSubnet2CIDR:

    Description: Please enter the IP range (CIDR notation) for the private subnet in the second Availability Zone

    Type: String

    Default: 10.10.201.0/24

  ImageId:

    Description: Nginx-AMI

    Type: String

    Default: ami-0e53fc7bfc076dd21

  InstanceType:

    Description: specify the type of instance

    Type: String

    Default: t3.micro

  KeyName:

    Description: Name of an existing EC2 KeyPair to enable SSH access to the instance

    Type: 'AWS::EC2::KeyPair::KeyName'

    ConstraintDescription: must be the name of an existing EC2 KeyPair.

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

  PublicSubnet1:

    Type: AWS::EC2::Subnet

    Properties:

      VpcId: !Ref VPC

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

      CidrBlock: !Ref PublicSubnet1CIDR

      MapPublicIpOnLaunch: true

      Tags:

        - Key: Name

          Value: !Sub ${EnvironmentName} Public Subnet (AZ1)

  PublicSubnet2:

    Type: AWS::EC2::Subnet

    Properties:

      VpcId: !Ref VPC

      AvailabilityZone: !Select [ 1, !GetAZs  '' ]

      CidrBlock: !Ref PublicSubnet2CIDR

      MapPublicIpOnLaunch: true

      Tags:

        - Key: Name

          Value: !Sub ${EnvironmentName} Public Subnet (AZ2)

  PrivateSubnet1:

    Type: AWS::EC2::Subnet

    Properties:

      VpcId: !Ref VPC

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

      CidrBlock: !Ref PrivateSubnet1CIDR

      MapPublicIpOnLaunch: false

      Tags:

        - Key: Name

          Value: !Sub ${EnvironmentName} Private Subnet (AZ1)

  PrivateSubnet2:

    Type: AWS::EC2::Subnet

    Properties:

      VpcId: !Ref VPC

      AvailabilityZone: !Select [ 1, !GetAZs  '' ]

      CidrBlock: !Ref PrivateSubnet2CIDR

      MapPublicIpOnLaunch: false

      Tags:

        - Key: Name

          Value: !Sub ${EnvironmentName} Private Subnet (AZ2)

  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

  PublicSubnet1RouteTableAssociation:

    Type: AWS::EC2::SubnetRouteTableAssociation

    Properties:

      RouteTableId: !Ref PublicRouteTable

      SubnetId: !Ref PublicSubnet1

  PublicSubnet2RouteTableAssociation:

    Type: AWS::EC2::SubnetRouteTableAssociation

    Properties:

      RouteTableId: !Ref PublicRouteTable

      SubnetId: !Ref PublicSubnet2

  EC2SecurityGroup:

    Type: AWS::EC2::SecurityGroup

    Properties:

      GroupName: CNF-SG

      GroupDescription: Creating SG using CNF

      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

      SecurityGroupEgress:

            - IpProtocol: tcp

              FromPort: 80

              ToPort: 80

              CidrIp: 0.0.0.0/0

      Tags:

            - Key: Name

              Value: CFN\_SG

  EC2Instance:

    Type: 'AWS::EC2::Instance'

    Properties:

      InstanceType: !Ref InstanceType

      SecurityGroupIds:

          - !GetAtt "EC2SecurityGroup.GroupId"

      SubnetId: !Ref PublicSubnet1

      KeyName: !Ref KeyName

      ImageId: !Ref ImageId

Outputs:

  InstanceId:

    Description: InstanceId of the newly created EC2 instance

    Value: !Ref EC2Instance

  AZ:

    Description: Availability Zone of the newly created EC2 instance

    Value: !GetAtt

        - EC2Instance

        - AvailabilityZone

  PublicDNS:

    Description: Public DNSName of the newly created EC2 instance

    Value: !GetAtt

        - EC2Instance

        - PublicDnsName

  PublicIP:

    Description: Public IP address of the newly created EC2 instance

    Value: !GetAtt

        - EC2Instance

        - PublicIp

  VPC:

    Description: A reference to the created VPC

    Value: !Ref VPC

  PublicSubnets:

    Description: A list of the public subnets

    Value: !Join [ ",", [ !Ref PublicSubnet1, !Ref PublicSubnet2 ]]

  PrivateSubnets:

    Description: A list of the private subnets

    Value: !Join [ ",", [ !Ref PrivateSubnet1, !Ref PrivateSubnet2 ]]

  PublicSubnet1:

    Description: A reference to the public subnet in the 1st Availability Zone

    Value: !Ref PublicSubnet1

  PublicSubnet2:

    Description: A reference to the public subnet in the 2nd Availability Zone

    Value: !Ref PublicSubnet2

  PrivateSubnet1:

    Description: A reference to private subnet in the 1st Availability Zone

    Value: !Ref PrivateSubnet1

  PrivateSubnet2:

    Description: A reference to private subnet in the 2nd Availability Zone

    Value: !Ref PrivateSubnet2