AWS CLOUDFORMATION TEMPLATE

What is CloudFormation Template?

An AWS CloudFormation template is a JSON or YAML-formatted text file that describes the set of resources and properties needed to launch and configure an AWS infrastructure.

CloudFormation allows you to use a template to define, provision, and manage AWS resources in a predictable and repeatable way.

What is stack in CloudFormation template?

In AWS CloudFormation, a stack is a collection of AWS resources that you can manage as a single unit. A stack is created from a CloudFormation template, which defines the set of resources and their configurations. When you create a stack, AWS CloudFormation provisions the specified resources and handles the dependencies between them.

What is VPC?

A Virtual Private Cloud (VPC) is a virtual network dedicated to your AWS account. It provides a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. Within a VPC, you have control over your network environment, including IP address ranges, subnets, routing tables, network gateways, and security settings.

What is Network Security group?

In AWS, a security group is a fundamental component of the network security for your Amazon Elastic Compute Cloud (EC2) instances and other resources. It acts as a virtual firewall for your instances to control inbound and outbound traffic.

What is Internet Gateway?

An Internet Gateway (IGW) is a key component of Amazon Web Services (AWS) networking that enables communication between resources in your Virtual Private Cloud (VPC) and the internet. It serves as a horizontally scalable, redundant, and highly available connection point for your VPC.

What is VPCGatewayAttachment?

It is a resource in AWS CloudFormation is used to attach an Amazon Virtual Private Cloud (VPC) to an internet gateway or a virtual private gateway. It establishes the connectivity between your VPC and the outside network.

What is Route table?

A route table is a fundamental component of a VPC and plays a crucial role in determining how network traffic is routed within the VPC.

What is SubnetRouteAssociation?

The AWS::EC2:SubnetRouteTableAssociation resource in AWS CloudFormation is used to associate a subnet with a specific route table within an Amazon Virtual Private Cloud (VPC). This association determines which route table is used for routing traffic within the associated subnet.

What is InternetGatewayRoute?

InternetGatewayRoute appears to be a custom logical name for a route that directs traffic to the internet. However, it's not a standard AWS CloudFormation resource type. It seems to be a label or tag used in the template to describe a specific route configuration associated with an Internet Gateway.

What is Active Directory Domain Services?

AD DS is a Windows Server role responsible for providing directory services, including authentication and authorization.

What is Domain Controller?

The primary role of a domain controller is to host and provide access to a centralized directory service, and this service is typically provided by Microsoft's Active Directory (AD).

What is ADDS forest?

An AD DS Forest serves as a logical boundary for the organization of resources in a Windows network. It is the top-level container that contains one or more domain trees, which, in turn, consist of individual domains.

TASK

Develop CloudFormation templates to create Two windows servers with one server acting as an AD server and another server acting as a member server of the AD in the AWS Cloud.

```
AWSTemplateFormatVersion: "2010-09-09"

Description: Create an EC2 instance for Active Directory


Parameters:

  VpcCIDR:

    Description: The IP address range for the VPC

    Type: String

    Default: 10.0.0.0/16


  VpcName:

    Description: Name for the VPC

    Type: String

    Default: VPC_AD11


  InstanceTypeParameter:

    Description: Instance type

    Type: String

    Default: t3.micro

    AllowedValues:

      - t3.micro

      - m1.small

      - t3.xlarge


  HostServerInstanceName:

    Description: Name of the host server EC2 instance

    Type: String
```

```yaml
  Default: ActiveDirectoryHostServer11
MemberServerInstanceName:
  Description: Name of the member server EC2 instance
  Type: String
  Default: ActiveDirectoryMemberServer11


ImageId:
  Description: Enter the image ID for the EC2 instance.
  Type: AWS::EC2::Image::Id
  Default: ami-009b52c0f357dd769

KeyPairName:
  Description: Name of the EC2 Key Pair
  Type: AWS::EC2::KeyPair::KeyName
  Default: keypair1

SubnetCIDR:
  Description: The IP address range (CIDR notation) for the subnets
  Type: String
  Default: 10.0.0.0/24

SubnetName:
  Description: Name for the Subnet
  Type: String
  Default: Subnet_AD11

AdminPassword:
  Type: String
  NoEcho: true
  Description: Windows Administrator Password
```

```
    MinLength: 8

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VpcCIDR
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: !Ref VpcName

  ActiveDirectorySecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: AD server Security Group11
      GroupDescription: Enable AD Server traffic
      VpcId: !Ref VPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 3389
          ToPort: 3389
          CidrIp: 0.0.0.0/0 # Allow RDP access from anywhere
        - IpProtocol: tcp
          FromPort: 389
          ToPort: 389
          CidrIp: 0.0.0.0/0
        - IpProtocol: -1
          CidrIp: 0.0.0.0/0
```

```yaml
Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    CidrBlock: !Ref SubnetCIDR
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Ref SubnetName

InternetGateway:
  Type: AWS::EC2::InternetGateway

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

RouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC

SubnetRouteAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref Subnet
    RouteTableId: !Ref RouteTable

InternetGatewayRoute:
    Type: AWS::EC2::Route
    DependsOn: InternetGateway
    Properties:
      RouteTableId: !Ref RouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref InternetGateway

  ActiveDirectoryHostServer:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: !Ref InstanceTypeParameter
      SecurityGroupIds:
        - !GetAtt ActiveDirectorySecurityGroup.GroupId
      KeyName: !Ref KeyPairName
      ImageId: ami-009b52c0f357dd769
      SubnetId: !Ref Subnet
      PrivateIpAddress: 10.0.0.18
      UserData:
        Fn::Base64: !Sub |
          <powershell>

          $AdminPassword = "${AdminPassword}"
          net user Administrator "${AdminPassword}"

          # Install AD DS
          Install-WindowsFeature -Name AD-Domain-Services -
IncludeManagementTools

          Install-ADDSDomainController -DomainName "awsdevops.com" -
SafeModeAdministratorPassword (ConvertTo-SecureString -AsPlainText
"Welcome@123" -Force) -Force -NoReboot
          # Configure AD DS
          Install-ADDSForest -DomainName "awsdevops.com" -
SafeModeAdministratorPassword (ConvertTo-SecureString -AsPlainText
"Welcome@123" -Force) -Force -NoReboot
          # Allow RDP and DNS inbound traffic


          New-NetFirewallRule -DisplayName "Allow RDP" -Direction Inbound -
Protocol TCP -LocalPort 3389 -Action Allow


          New-NetFirewallRule -DisplayName "Allow DNS" -Direction Inbound -
Protocol UDP -LocalPort 53 -Action Allow
          # Reboot to complete the AD DS setup


          Restart-Computer -Force


          </powershell>


      Tags:
        - Key: Name
          Value: !Ref HostServerInstanceName



  ActiveDirectoryMemberServer:
    Type: AWS::EC2::Instance
    DependsOn: ActiveDirectoryHostServer
    Properties:
      InstanceType: !Ref InstanceTypeParameter
      SecurityGroupIds:
        - !GetAtt ActiveDirectorySecurityGroup.GroupId
      KeyName: !Ref KeyPairName
      ImageId: ami-009b52c0f357dd769
      SubnetId: !Ref Subnet
      UserData:
        Fn::Base64: !Sub |
          <powershell>
          # Parameters
          $DomainName = "awsdevops.com"  # Your AD Domain Name
          $DomainAdminUser = "AWSDEVOPS\Administrator"  # The AD Admin User
          $AdminPassword = "NewPassword@1234"  # The AD Admin User Password

          # Fetch the interface index
          $InterfaceIndex = (Get-NetAdapter | Where-Object { $_.Name -eq
"Ethernet 3" }).InterfaceIndex

          # Set DNS to point to the AD Server (replace with the actual IP of
your AD Server)
          $DnsIpAddress = "10.0.0.18"
          Set-DnsClientServerAddress -InterfaceIndex $InterfaceIndex -
ServerAddresses $DnsIpAddress
          New-NetFirewallRule -DisplayName "Allow RDP" -Direction Inbound -
Protocol TCP -LocalPort 3389 -Action Allow

          New-NetFirewallRule -DisplayName "Allow DNS" -Direction Inbound -
Protocol UDP -LocalPort 53 -Action Allow
           Start-Sleep -Seconds 900

          # Join the domain
          Add-Computer -DomainName $DomainName -Credential (New-Object
PSCredential "$DomainAdminUser", (ConvertTo-SecureString $AdminPassword -
AsPlainText -Force)) -Restart

          </powershell>
      Tags:
        - Key: Name
          Value: !Ref MemberServerInstanceName
```