Cloud Lab 3 - CloudFormation

Cloud Systems

Lara Alferez

2021

Table of Contents

[CloudFormation Script 2](#_Toc89718901)

[Snippet of Components 14](#_Toc89718902)

[a) Two VPCs 15](#_Toc89718903)

[b) Two subnets per VPC 15](#_Toc89718904)

[c) Route tables 15](#_Toc89718905)

[d) Security Groups 15](#_Toc89718906)

[e) 2 Internet Gateways 15](#_Toc89718907)

[Proof Of Successful Stack 15](#_Toc89718908)

[VPC-A Webserver Proof 16](#_Toc89718909)

[VPC-B Webserver Proof 16](#_Toc89718910)

# CloudFormation Script

# Lab 3 Script

# Lara Alferez 991540084

#############################################################

Parameters:

# parameter VPC CIDR block or IP address space

VPC1Cidr:

Type: String

Default: 10.0.0.0/16

Description: this is the VPC IP address space or CIDR for VPC-A

VPC2Cidr:

Type: String

Default: 192.168.0.0/16

Description: this is the VPC IP address space or CIDR for VPC-B

AZ1:

Type: AWS::EC2::AvailabilityZone::Name

Default: us-east-1a

Description: Choose an availability zone from a drop menu or get the default

AZ2:

Type: AWS::EC2::AvailabilityZone::Name

Default: us-east-1c

Description: Choose an availability zone from a drop menu or get the default

SN1cidr:

Type: String

Default: 10.0.1.0/24

Description: this is the Public subnet CIDR for VPC-A

SN2cidr:

Type: String

Default: 10.0.2.0/24

Description: this is the Private subnet CIDR for VPC-A

SN3cidr:

Type: String

Default: 192.168.1.0/24

Description: this is the Public subnet CIDR for VPC-B

SN4cidr:

Type: String

Default: 192.168.2.0/24

Description: this is the Private subnet CIDR for VPC-B

################################################################

Resources:

# Create VPC-A (10.0.0.0/16) with the following

VPC1:

Type: AWS::EC2::VPC

Properties:

CidrBlock: !Ref VPC1Cidr

EnableDnsSupport: true

EnableDnsHostnames: true

Tags:

- Key: Name

Value: VPC-A

# In Availability Zone US-East-1a

# EC2 webserver-01

# Create an EC2 Apache Webserver with a security group and User Data Bootstrap

EC2Instance:

Type: AWS::EC2::Instance

Properties:

AvailabilityZone: us-east-1a

ImageId: ami-02e136e904f3da870

InstanceType: t2.micro

KeyName: Key-demo-01

NetworkInterfaces:

- AssociatePublicIpAddress: "true"

DeviceIndex: "0"

GroupSet:

- Ref: PublicSG

SubnetId:

Ref: PublicSN

Tags:

- Key: Name

Value: EC2-A

UserData:

Fn::Base64: !Sub |

#!/bin/bash

yum update -y

yum install -y httpd.x86\_64

systemctl start httpd.service

systemctl enable httpd.service

echo "<html><body><h1>Hello VPC-A! Lara Alferez 991540084 $(hostname -f)</h1></body></html>" > /var/www/html/index.html

# Security group for Public Subnet

PublicSG:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Allow SSH, HTTP and VPC address

VpcId: !Ref VPC1

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

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 10.0.0.0/16

# Create Internet Gateway

IGW:

Type: AWS::EC2::InternetGateway

DependsOn: VPC1

# Attach Internet gateway and the VPC

IGWattachVPC:

Type: AWS::EC2::VPCGatewayAttachment

Properties:

VpcId: !Ref VPC1

InternetGatewayId: !Ref IGW

# Public Subnet-1 (10.0.1.0/24)

PublicSN:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC1

CidrBlock: !Ref SN1cidr

AvailabilityZone: !Ref AZ1

Tags:

- Key: Name

Value: PublicSN1-a

# Create a public route table

PublicRT:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref VPC1

Tags:

- Key: Name

Value: PublicRT

# Add a route for the public route table

RoutePublic:

Type: AWS::EC2::Route

DependsOn: IGWattachVPC

Properties:

RouteTableId: !Ref PublicRT

DestinationCidrBlock: 0.0.0.0/0

GatewayId: !Ref IGW

# Associate the the routes with the route table

PublicRouteAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

SubnetId: !Ref PublicSN

RouteTableId: !Ref PublicRT

# In Availability Zone US-East-1c Create

# Private Subnet-1 (10.0.2.0/24)

PrivateSN:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC1

CidrBlock: !Ref SN2cidr

AvailabilityZone: !Ref AZ2

Tags:

- Key: Name

Value: PrivateSN1-a

# Create Elastic IP address

EIP:

Type: AWS::EC2::EIP

Properties:

Domain: VPC1

# Create NAT gateway

NAT:

Type: AWS::EC2::NatGateway

Properties:

AllocationId: !GetAtt EIP.AllocationId

SubnetId: !Ref PublicSN

Tags:

- Key: Name

Value: natgw

# Create Private Route Table

PrivateRT:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref VPC1

Tags:

- Key: Name

Value: PrivateRT

# Create route for the private route table

PrivateRoute:

Type: AWS::EC2::Route

Properties:

RouteTableId: !Ref PrivateRT

DestinationCidrBlock: 0.0.0.0/0

NatGatewayId: !Ref NAT

# Associate the private subnet and the private route table

PrivateRouteAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

SubnetId: !Ref PrivateSN

RouteTableId: !Ref PrivateRT

# Security group for Private Subnet

PrivateSG:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Allow only VPC address for Private

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 10.0.0.0/16

#########################################################################

# Create VPC-B (192.168.0.0/16) with the following

VPC2:

Type: AWS::EC2::VPC

Properties:

CidrBlock: !Ref VPC2Cidr

EnableDnsSupport: true

EnableDnsHostnames: true

Tags:

- Key: Name

Value: VPC-B

# In Availability Zone US-East-1a Create

# EC2 webserver-02

# Create an EC2 Apache Webserver with a security group and User Data Bootstrap

EC2Instance2:

Type: AWS::EC2::Instance

Properties:

AvailabilityZone: us-east-1a

ImageId: ami-02e136e904f3da870

InstanceType: t2.micro

KeyName: Key-demo-01

NetworkInterfaces:

- AssociatePublicIpAddress: "true"

DeviceIndex: "0"

GroupSet:

- Ref: PublicSG2

SubnetId:

Ref: PublicSN2

Tags:

- Key: Name

Value: EC2-B

UserData:

Fn::Base64: !Sub |

#!/bin/bash

yum update -y

yum install -y httpd.x86\_64

systemctl start httpd.service

systemctl enable httpd.service

echo "<html><body><h1>Hello VPC-B! Lara Alferez 991540084 $(hostname -f)</h1></body></html>" > /var/www/html/index.html

# Security group for Public Subnet for VPC-B

PublicSG2:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Allow SSH, HTTP and VPC address

VpcId: !Ref VPC2

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

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 192.168.0.0/16

# Create Internet Gateway

IGW2:

Type: AWS::EC2::InternetGateway

DependsOn: VPC2

# Attach Internet gateway and the VPC

IGWattachVPC2:

Type: AWS::EC2::VPCGatewayAttachment

Properties:

VpcId: !Ref VPC2

InternetGatewayId: !Ref IGW2

# Public Subnet-2 (192.168.1.0/24)

PublicSN2:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC2

CidrBlock: !Ref SN3cidr

AvailabilityZone: !Ref AZ1

Tags:

- Key: Name

Value: PublicSN2-b

# Create a public route table

PublicRT2:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref VPC2

Tags:

- Key: Name

Value: PublicRT2

# Add a route for the public route table

RoutePublic2:

Type: AWS::EC2::Route

DependsOn: IGWattachVPC2

Properties:

RouteTableId: !Ref PublicRT2

DestinationCidrBlock: 0.0.0.0/0

GatewayId: !Ref IGW2

# Associate the the routes with the route table

PublicRouteAssociation2:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

SubnetId: !Ref PublicSN2

RouteTableId: !Ref PublicRT2

# In Availability Zone US-East-1c Create

# Private Subnet-2 (192.168.2.0/24)

PrivateSN2:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC2

CidrBlock: !Ref SN4cidr

AvailabilityZone: !Ref AZ2

Tags:

- Key: Name

Value: PrivateSN2-b

# Create Elastic IP address

EIP2:

Type: AWS::EC2::EIP

Properties:

Domain: VPC2

# Create NAT gateway

NAT2:

Type: AWS::EC2::NatGateway

Properties:

AllocationId: !GetAtt EIP2.AllocationId

SubnetId: !Ref PublicSN2

Tags:

- Key: Name

Value: natgw2

# Create Private Route Table

PrivateRT2:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref VPC2

Tags:

- Key: Name

Value: PrivateRT2

# Create route for the private route table

PrivateRoute2:

Type: AWS::EC2::Route

Properties:

RouteTableId: !Ref PrivateRT2

DestinationCidrBlock: 0.0.0.0/0

NatGatewayId: !Ref NAT2

# Associate the private subnet and the private route table

PrivateRouteAssociation2:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

SubnetId: !Ref PrivateSN2

RouteTableId: !Ref PrivateRT2

# Security group for Private Subnet

PrivateSG2:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Allow only VPC address for Private

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 192.168.0.0/16

# Snippet of Components

\*\*Error Loading Template to view in Designer. The template would never load so I was unable to View in Designer\*\*

Graphical user interface, text, application, email

Description automatically generated

## a) Two VPCs



## b) Two subnets per VPC

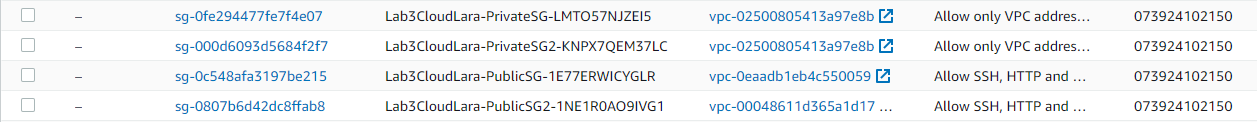
A picture containing graphical user interface

Description automatically generated

## c) Route tables



## d) Security Groups



## e) 2 Internet Gateways



## Proof Of Successful Stack

Graphical user interface, text, application, email

Description automatically generated

# VPC-A Webserver Proof

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, website

Description automatically generated

# VPC-B Webserver Proof

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, website

Description automatically generated