

<https://www.youtube.com/watch?v=tp--_hipB0U&list=PLg5SS_4L6LYsxrZ_4xE_U95AtGsIB96k9&index=31>

Создание полной инфраструктуры из кода, пример создания VPC сети:

# --------------------------------------------------------------------------------------------------

# CloudFormation Network Layer: VPC, Subnets + RouteTables, Internet + NAT Gateways

# Author: Denis Astahov

#

# Version Date Name Info

# 1.0 26-Nov-2017 Denis Astahov Initial Version

#

# --------------------------------------------------------------------------------------------------

AWSTemplateFormatVersion: 2010-09-09

Description: "Network: VPC, Subnets + RouteTables, Internet + NAT Gateways"

Metadata:

AWS::CloudFormation::Interface:

ParameterGroups:

-

Label:

default: "Network Configuration"

Parameters:

- Environment

- VPCBlock

- VPCBlock2

-

Label:

default: "Subnets CIDR Blocks"

Parameters:

- PublicSubnetACIDR

- PublicSubnetBCIDR

- PrivateSubnetACIDR

- PrivateSubnetBCIDR

- DatabaseSubnetACIDR

- DatabaseSubnetBCIDR

Parameters:

Environment:

Type: String

Default: "KGB-Prod"

VPCBlock:

Type: String

Default: '10.0.0.0/16'

VPCBlock2:

Type: String

Default: '192.0.0.0/16'

PublicSubnetACIDR:

Type: String

Default: '10.0.10.0/24'

Description: "Public Subnet-A CIDR"

PublicSubnetBCIDR:

Type: String

Default: '10.0.20.0/24'

Description: "Public Subnet-B CIDR"

PrivateSubnetACIDR:

Type: String

Default: '10.0.11.0/24'

Description: "Private Subnet-A CIDR"

PrivateSubnetBCIDR:

Type: String

Default: '10.0.21.0/24'

Description: "Public Subnet-B CIDR"

DatabaseSubnetACIDR:

Type: String

Default: '192.0.30.0/24'

Description: "Database Subnet-A CIDR"

DatabaseSubnetBCIDR:

Type: String

Default: '192.0.31.0/24'

Description: "Database Subnet-B CIDR"

Resources:

#============= VPC ==================

VPC:

Type: AWS::EC2::VPC

Properties: # Attach Primary CIDR Block

CidrBlock: !Ref VPCBlock

EnableDnsSupport: true

EnableDnsHostnames: true

Tags:

- Key: Name

Value: !Ref Environment

VPCCidrBlock2: # Attach Secondary CIDR Block

Type: "AWS::EC2::VPCCidrBlock"

DependsOn: VPC

DeletionPolicy: Delete

Properties:

CidrBlock: !Ref VPCBlock2

VpcId: !Ref VPC

#====== Internet Gateway =======

GatewayInternet:

Type: "AWS::EC2::InternetGateway"

Properties:

Tags:

- Key: Name

Value: !Ref Environment

GatewayAttachmentInternet: # Attachment IGW to VPC

Type: "AWS::EC2::VPCGatewayAttachment"

Properties:

VpcId: !Ref VPC

InternetGatewayId: !Ref GatewayInternet

#====== Public RouteTables =========

RouteTableForPublicSubnet: # Creation of Empty Route Table

Type: "AWS::EC2::RouteTable"

Properties:

VpcId: !Ref VPC

Tags:

- Key: Name

Value: !Join [ " ", [ !Ref Environment, "PublicRouteTable" ] ]

RoutesForPublicRouteTable: # Creation and Attachment of Routes for Route Table

Type: "AWS::EC2::Route"

DependsOn: GatewayAttachmentInternet

Properties:

DestinationCidrBlock: 0.0.0.0/0

GatewayId: !Ref GatewayInternet

RouteTableId: !Ref RouteTableForPublicSubnet

#====== Private RouteTables ===========

RouteTableForPrivateSubnetA:

Type: "AWS::EC2::RouteTable"

Properties:

VpcId: !Ref VPC

Tags:

- Key: Name

Value: !Join [ " ", [ !Ref Environment, "PrivateRouteTableA" ] ]

RouteTableForPrivateSubnetB:

Type: "AWS::EC2::RouteTable"

Properties:

VpcId: !Ref VPC

Tags:

- Key: Name

Value: !Join [ " ", [ !Ref Environment, "PrivateRouteTableB" ] ]

#======= Routes for Private Subnet RouteTables using NAT

RoutesForPrivateRouteTableA:

Type: AWS::EC2::Route

DependsOn: NATGatewayA

Properties:

DestinationCidrBlock: 0.0.0.0/0

RouteTableId: !Ref RouteTableForPrivateSubnetA

NatGatewayId: !Ref NATGatewayA

RoutesForPrivateRouteTableB:

Type: AWS::EC2::Route

DependsOn: NATGatewayB

Properties:

DestinationCidrBlock: 0.0.0.0/0

RouteTableId: !Ref RouteTableForPrivateSubnetB

NatGatewayId: !Ref NATGatewayB

#====== Associate Public Route for Public Subnets

RouteAssociationPublicA:

Type: "AWS::EC2::SubnetRouteTableAssociation"

Properties:

RouteTableId: !Ref RouteTableForPublicSubnet

SubnetId: !Ref PublicSubnetA

RouteAssociationPublicB:

Type: "AWS::EC2::SubnetRouteTableAssociation"

Properties:

RouteTableId: !Ref RouteTableForPublicSubnet

SubnetId: !Ref PublicSubnetB

#===== Associate Private Route for Private Subnets

RouteAssociationPrivateA:

Type: "AWS::EC2::SubnetRouteTableAssociation"

Properties:

RouteTableId: !Ref RouteTableForPrivateSubnetA

SubnetId: !Ref PrivateSubnetA

RouteAssociationPrivateB:

Type: "AWS::EC2::SubnetRouteTableAssociation"

Properties:

RouteTableId: !Ref RouteTableForPrivateSubnetB

SubnetId: !Ref PrivateSubnetB

#==== Route Table and Route Association for Internal only - Databases

RouteTableForDatabase:

Type: "AWS::EC2::RouteTable"

Properties:

VpcId: !Ref VPC

Tags:

- Key: Name

Value: !Join [ " ", [ !Ref Environment, "DatabaseRouteTable" ] ]

RouteAssociationDatabaseA:

Type: "AWS::EC2::SubnetRouteTableAssociation"

Properties:

RouteTableId: !Ref RouteTableForDatabase

SubnetId: !Ref DatabaseSubnetA

RouteAssociationDatabaseB:

Type: "AWS::EC2::SubnetRouteTableAssociation"

Properties:

RouteTableId: !Ref RouteTableForDatabase

SubnetId: !Ref DatabaseSubnetB

#======= ElasticIP for NAT Gateways ======

EIP1:

Type: "AWS::EC2::EIP"

DependsOn : GatewayAttachmentInternet

Properties:

Domain: !Ref VPC

EIP2:

Type: "AWS::EC2::EIP"

DependsOn : GatewayAttachmentInternet

Properties:

Domain: !Ref VPC

#========= NAT Gateways ===============

NATGatewayA:

DependsOn: GatewayAttachmentInternet

Type: AWS::EC2::NatGateway

Properties:

SubnetId: !Ref PublicSubnetA

AllocationId: !GetAtt EIP1.AllocationId

Tags:

- Key: Name

Value: NATGateway-A

NATGatewayB:

DependsOn: GatewayAttachmentInternet

Type: AWS::EC2::NatGateway

Properties:

SubnetId: !Ref PublicSubnetB

AllocationId: !GetAtt EIP2.AllocationId

Tags:

- Key: Name

Value: NATGateway-B

#============ ALL Subnets ======================================================

PublicSubnetA:

Type: "AWS::EC2::Subnet"

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [ 0, "Fn::GetAZs": { Ref: "AWS::Region" } ]

CidrBlock: !Ref "PublicSubnetACIDR"

MapPublicIpOnLaunch: true

Tags:

- Key: Name

Value: !Join [ "", [ !Ref Environment, "-Public-A" ] ]

PublicSubnetB:

Type: "AWS::EC2::Subnet"

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [ 1, "Fn::GetAZs": { Ref: "AWS::Region" } ]

CidrBlock: !Ref "PublicSubnetBCIDR"

MapPublicIpOnLaunch: true

Tags:

- Key: Name

Value: !Join [ "", [ !Ref Environment, "-Public-B" ] ]

PrivateSubnetA:

Type: "AWS::EC2::Subnet"

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [ 0, "Fn::GetAZs": { Ref: "AWS::Region" } ]

CidrBlock: !Ref "PrivateSubnetACIDR"

MapPublicIpOnLaunch: false

Tags:

- Key: Name

Value: !Join [ "", [ !Ref Environment, "-Private-A" ] ]

PrivateSubnetB:

Type: "AWS::EC2::Subnet"

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [ 1, "Fn::GetAZs": { Ref: "AWS::Region" } ]

CidrBlock: !Ref "PrivateSubnetBCIDR"

MapPublicIpOnLaunch: false

Tags:

- Key: Name

Value: !Join [ "", [ !Ref Environment, "-Private-B" ] ]

DatabaseSubnetA:

Type: "AWS::EC2::Subnet"

DependsOn: VPCCidrBlock2

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [ 0, "Fn::GetAZs": { Ref: "AWS::Region" } ]

CidrBlock: !Ref "DatabaseSubnetACIDR"

MapPublicIpOnLaunch: false

Tags:

- Key: Name

Value: !Join [ "", [ !Ref Environment, "-Database-A" ] ]

DatabaseSubnetB:

Type: "AWS::EC2::Subnet"

DependsOn: VPCCidrBlock2

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [ 1, "Fn::GetAZs": { Ref: "AWS::Region" } ]

CidrBlock: !Ref "DatabaseSubnetBCIDR"

MapPublicIpOnLaunch: false

Tags:

- Key: Name

Value: !Join [ "", [ !Ref Environment, "-Database-B" ] ]

#=================== OUTPUTS ===========================

Outputs:

VPC:

Description: ID for the VPC

Value: !Ref VPC

Export:

Name: !Join [ "-", [ !Ref "Environment", "VPC" ] ]

VPCBlock1:

Description: Primary CIDR block for the VPC

Value: !GetAtt VPC.CidrBlock

Export:

Name: !Join [ "-", [ !Ref "Environment", "CIDR1" ] ]

VPCBlock2:

Description: Secondary CIDR block for the VPC

Value: !Ref VPCBlock2

Export:

Name: !Join [ "-", [ !Ref "Environment", "CIDR2" ] ]

PublicA:

Description: ID for Public Subnet A

Value: !Ref PublicSubnetA

Export:

Name: !Join [ "-", [ !Ref "Environment", "PublicSubnetA" ] ]

PublicB:

Description: ID for Public Subnet B

Value: !Ref PublicSubnetB

Export:

Name: !Join [ "-", [ !Ref "Environment", "PublicSubnetB" ] ]

PrivateA:

Description: ID for Private Subnet A

Value: !Ref PrivateSubnetA

Export:

Name: !Join [ "-", [ !Ref "Environment", "PrivateSubnetA" ] ]

PrivateB:

Description: ID for Private Subnet B

Value: !Ref PrivateSubnetB

Export:

Name: !Join [ "-", [ !Ref "Environment", "PrivateSubnetB" ] ]

DatabaseA:

Description: ID for Database Subnet A

Value: !Ref DatabaseSubnetA

Export:

Name: !Join [ "-", [ !Ref "Environment", "DatabaseSubnetA" ] ]

DatabaseB:

Description: ID for Database Subnet B

Value: !Ref DatabaseSubnetB

Export:

Name: !Join [ "-", [ !Ref "Environment", "DatabaseSubnetB" ] ]