

58°

Lab - AWS re/Start

Automatización de implementaciones con AWS CloudFormation



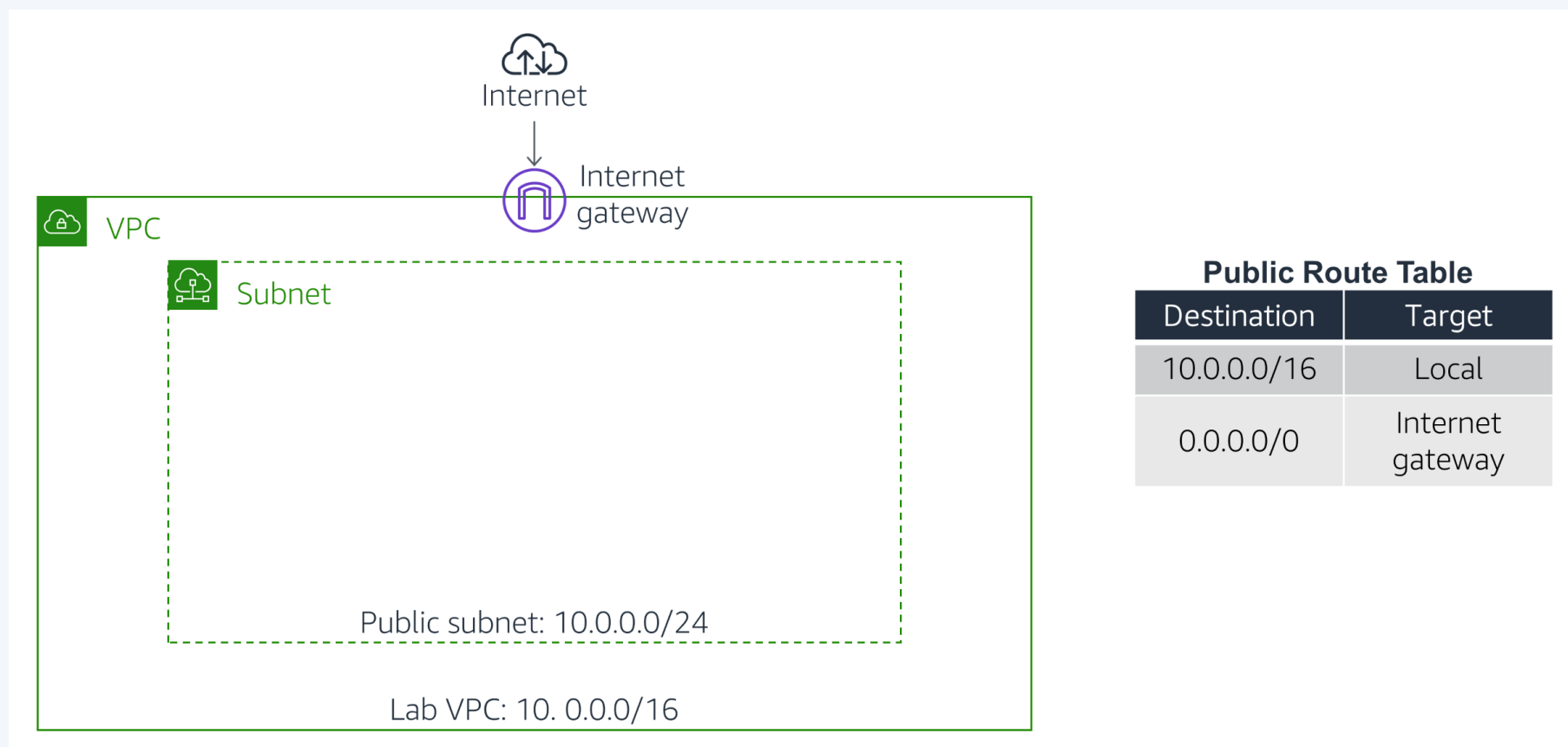
Tarea 01



Interactuando con AWS CloudWatch

Los objetivos son:

- Implementar una pila de AWS CloudFormation con una VPC y un grupo de seguridad definidos.
- Configurar una pila de AWS CloudFormation con recursos, como un bucket de S3 y EC2.
- Finalizar una pila de AWS CloudFormation y sus respectivos recursos.



Tarea 01



Empezamos implementando una pila de CloudFormation

Create stack

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready

Use a sample template

Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL

Provide an Amazon S3 URL to your template.

Upload a template file

Upload your template directly to the console.

Sync from Git - new

Sync a template from your Git repository.

Upload a template file

Choose file

task1.yaml

JSON or YAML formatted file

S3 URL: <https://s3.us-west-2.amazonaws.com/cf-templates-1e2oawpblbj4a-us-west-2/2024-03-08T211259.844Z4w7-task1.yaml>

View in Designer

Specify stack details

Provide a stack name

Stack name

Lab

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

LabVpcCidr

10.0.0.0/20

PublicSubnetCidr

10.0.0.0/24

task1.yaml

```
7
8   LabVpcCidr:
9     Type: String
10    Default: 10.0.0.0/20
11
12   PublicSubnetCidr:
13     Type: String
14     Default: 10.0.0.0/24
15
16
17   Resources:
18     #####
19     # VPC with Internet Gateway
20     #####
21
22   LabVPC:
23     Type: AWS::EC2::VPC
24     Properties:
25       CidrBlock: !Ref LabVpcCidr
26       EnableDnsSupport: true
27       EnableDnsHostnames: true
28       Tags:
29         - Key: Name
30           Value: Lab VPC
31
32   IGW:
33     Type: AWS::EC2::InternetGateway
34     Properties:
35       Tags:
36         - Key: Name
37           Value: Lab IGW
38
39   VPCtoIGWConnection:
40     Type: AWS::EC2::VPCGatewayAttachment
41     DependsOn:
42       - IGW
43       - LabVPC
44     Properties:
45       InternetGatewayId: !Ref IGW
46       VpcId: !Ref LabVPC
47
48   #####
49   # Public Route Table
50   #####
51
52   PublicRouteTable:
53     Type: AWS::EC2::RouteTable
54     DependsOn: LabVPC
55     Properties:
56       VpcId: !Ref LabVPC
57       Tags:
58         - Key: Name
59           Value: Public Route Table
60
61   PublicRoute:
62     Type: AWS::EC2::Route
63     DependsOn:
64       - PublicRouteTable
65       - IGW
66     Properties:
67       DestinationCidrBlock: 0.0.0.0/0
68       GatewayId: !Ref IGW
69       RouteTableId: !Ref PublicRouteTable
70
71   #####
72   # Public Subnet
73   #####
74
75   PublicSubnet:
76     Type: AWS::EC2::Subnet
77     DependsOn: LabVPC
78     Properties:
79       VpcId: !Ref LabVPC
80       MapPublicIpOnLaunch: true
81       CidrBlock: !Ref PublicSubnetCidr
82       AvailabilityZone: !Select
83         - 0
84         - !GetAZs
85           Ref: AWS::Region
86       Tags:
87         - Key: Name
88           Value: Public Subnet
89
90   PublicRouteTableAssociation:
91     Type: AWS::EC2::SubnetRouteTableAssociation
92     DependsOn:
93       - PublicRouteTable
94       - PublicSubnet
95
96   #####
```

Esta ha sido la infraestructura que se levantó

Lab

Delete

Update

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Git sync - new

Resources (8)

Search resources

< 1 >

Logical ID	Physical ID	Type	Status	Module
AppSecurityGroup	sg-0ba003c34ed259888	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
IGW	igw-044f12b679e07b1ed	AWS::EC2::InternetGateway	CREATE_COMPLETE	-
LabVPC	vpc-0e8c730289d829c95	AWS::EC2::VPC	CREATE_COMPLETE	-
PublicRoute	rtb-01eab2a4f7832759d 0.0.0.0/0	AWS::EC2::Route	CREATE_COMPLETE	-
PublicRouteTable	rtb-01eab2a4f7832759d	AWS::EC2::RouteTable	CREATE_COMPLETE	-
PublicRouteTableAssociation	rtbassoc-0990991d99c1bb8bf	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
PublicSubnet	subnet-01497aed485e34f39	AWS::EC2::Subnet	CREATE_COMPLETE	-
VPCtoIGWConnection	IGW/vpc-0e8c730289d829c95	AWS::EC2::VPCGatewayAttachment	CREATE_COMPLETE	-

Tarea 01



Ahora, procedemos a editar la plantilla con el fin de añadir un bucket dentro de la creación de infraestructura

Creating an Amazon S3 bucket with defaults

This example uses a `AWS::S3::Bucket` to create a bucket with default settings.

JSON

```
"myS3Bucket" : {
  "Type" : "AWS::S3::Bucket"
}
```

YAML

```
MyS3Bucket:
  Type: AWS::S3::Bucket
```

Resources:

MyS3Bucket:

Type: AWS::S3::Bucket

VPC with Internet Gateway

LabVPC:
 Type: AWS::EC2::VPC
 Properties:
 CidrBlock: !Ref LabVpcCidr
 EnableDnsSupport: true
 EnableDnsHostnames: true
 Tags:
 - Key: Name
 Value: Lab VPC

Update stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Use current template

Replace current template

Edit template in designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL

Upload a template file

Upload a template file

Choose file

task1.yaml

LabVPC	vpc-0e8c730289d829c95	AWS::EC2::VPC	CREATE_COMPLETE
MyS3Bucket	lab-mys3bucket-gl9ofz9ryvl	AWS::S3::Bucket	CREATE_IN_PROGRESS
PublicRoute	rtb-01eab2a4f7832759d 0.0.0.0/0	AWS::EC2::Route	CREATE_COMPLETE

También añadiremos una instancia de EC2

Parameters:

LabVpcCidr:

Type: String
Default: 10.0.0.0/20

PublicSubnetCidr:

Type: String
Default: 10.0.0.0/24

AmazonLinuxAMI ID:

Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

MyEC2Instance:

Type: AWS::EC2::Instance

Properties:

ImageID: !Ref AmazonLinuxAMIID
InstanceType: t3.micro
SecurityGroupIds:
 - !Ref AppSecurityGroup
SubnetId: !Ref PublicSubnet

Tags:
 - Key: Name
 Value: App Server

Y finalmente, eliminamos el stack (pila) de CloudFormation

Delete stack?

Delete stack Lab permanently? This action cannot be undone.

Deleting this stack will delete all stack resources. Resources will be deleted according to their DeletionPolicy. [Learn more](#)

Cancel

Delete

