



AWS
re:Start
LAB

Automation with CloudFormation



WEEK 11





Overview

AWS CloudFormation simplifies provisioning and managing AWS infrastructure by defining resources in code. This automation allows you to deploy complex environments, such as VPCs, security groups, S3 buckets, and EC2 instances, consistently and efficiently. Using CloudFormation templates ensures that your infrastructure setups are repeatable, minimizing manual errors and adhering to best practices and compliance standards.

Furthermore, CloudFormation allows for efficient resource lifecycle management. When resources are no longer needed, terminating the CloudFormation stack automatically deletes all associated components. This automated cleanup helps avoid orphaned resources and unnecessary costs, enhancing resource utilization and cost management. CloudFormation's automation capabilities significantly improve operational efficiency and infrastructure management in the AWS environment.

Topics covered

- Deploy an AWS CloudFormation stack with a defined Virtual Private Cloud (VPC), and Security Group.
- Configure an AWS CloudFormation stack with resources, such as an Amazon Simple Storage Service (S3) bucket and Amazon Elastic Compute Cloud (EC2).
- Terminate an AWS CloudFormation and its respective resources.



Task 1

Deploy a CloudFormation Stack

Step 1: Review the CloudFormation template

Download and review the CloudFormation template [task1.yaml](#).

```
AWSTemplateFormatVersion: 2010-09-09
Description: Lab template

# Lab VPC with public subnet and Internet Gateway

Parameters:

  LabVpcCidr:
    Type: String
    Default: 10.0.0.0/20

  PublicSubnetCidr:
    Type: String
    Default: 10.0.0.0/24
```

```
Resources:

  #####
  # VPC with Internet Gateway
  #####

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

  IGW:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: Lab IGW

  VPCtoIGWConnection:
    Type: AWS::EC2::VPCGatewayAttachment
    DependsOn:
      - IGW
      - LabVPC
    Properties:
      InternetGatewayId: !Ref IGW
      VpcId: !Ref LabVPC
```

```
#####
# Public Route Table
#####

PublicRouteTable:
  Type: AWS::EC2::RouteTable
  DependsOn: LabVPC
  Properties:
    VpcId: !Ref LabVPC
    Tags:
      - Key: Name
        Value: Public Route Table

PublicRoute:
  Type: AWS::EC2::Route
  DependsOn:
    - PublicRouteTable
    - IGW
  Properties:
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref IGW
    RouteTableId: !Ref PublicRouteTable
```

```
#####
# Public Subnet
#####

PublicSubnet:
  Type: AWS::EC2::Subnet
  DependsOn: LabVPC
  Properties:
    VpcId: !Ref LabVPC
    MapPublicIpOnLaunch: true
    CidrBlock: !Ref PublicSubnetCidr
    AvailabilityZone: !Select
      - 0
      - !GetAZs
        Ref: AWS::Region
    Tags:
      - Key: Name
        Value: Public Subnet

PublicRouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  DependsOn:
    - PublicRouteTable
    - PublicSubnet
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnet

#####
# App Security Group
#####

AppSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  DependsOn: LabVPC
  Properties:
    GroupName: App
    GroupDescription: Enable access to App
    VpcId: !Ref LabVPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
    Tags:
      - Key: Name
        Value: App

#####
# Outputs
#####

Outputs:

  LabVPCDefaultSecurityGroup:
    Value: !Sub ${LabVPC.DefaultSecurityGroup}
```

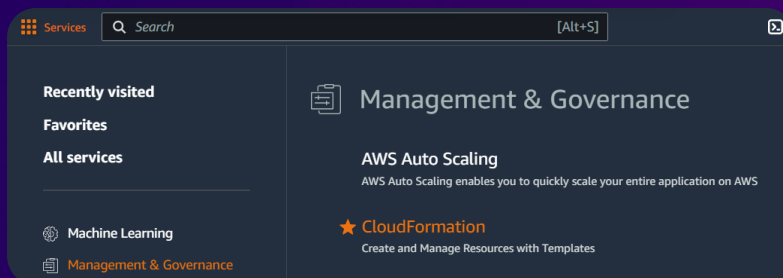


Task 1

Deploy a CloudFormation Stack

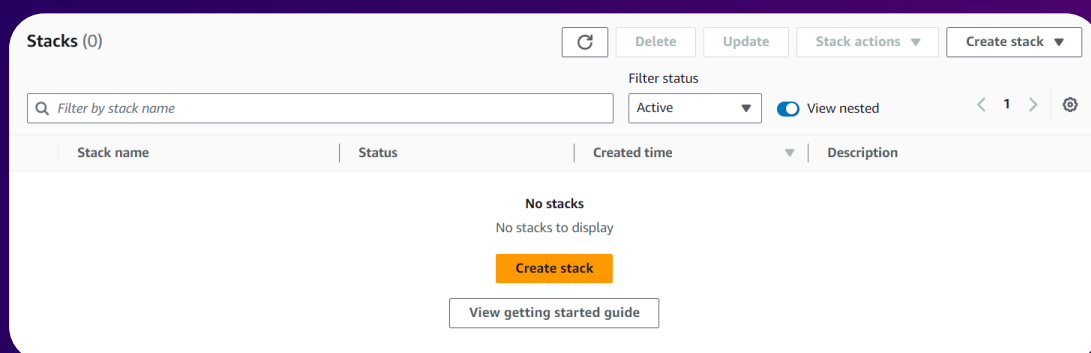
Step 2: Access the CloudFormation service

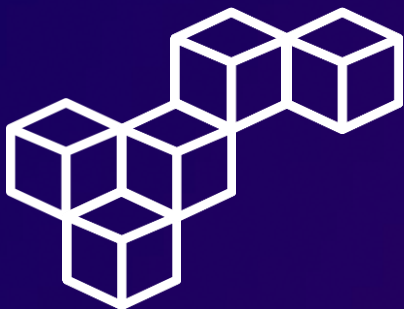
In the AWS Management Console, select CloudFormation.



Step 3: Create stack

In the **Stacks** section, select [Create stack](#).





Task 1

Deploy a CloudFormation Stack

Step 4: Stack Creation details

In the **Prepare template**, **Specify template**, and **Provide a stack name** sections, configure the following settings.

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.

Choose an existing template

Upload or choose an existing template.

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.

Upload a template file

Upload your template directly to the console.

Upload a template file

Choose file

task1.yaml

JSON or YAML formatted file

Provide a stack name


Stack name

Lab

Step 5: Review Events

Click the **Events** tab. The listing shows the activities performed by CloudFormation, such as starting to create a resource and then completing the resource creation.

Events (28)					Detect root cause	
Search events						
Timestamp	Logical ID	Status	Detailed status	Status reason		
2024-06-04 11:28:47 UTC-0500	Lab	CREATE_COMPLETE	-	-		
2024-06-04 11:28:46 UTC-0500	PublicRouteTableAssociation	CREATE_COMPLETE	-	-		
2024-06-04 11:28:45 UTC-0500	PublicRoute	CREATE_COMPLETE	-	-		

 re/start



Task 1

Deploy a CloudFormation Stack

Step 6: Review Resources

Click the **Resources** tab. The listing shows the resources that are being created. CloudFormation determines the optimal order for resources to be created, such as creating the VPC before the subnet.

Resources (8)

Q Search resources

< 1 >

⚙

Logical ID	Physical ID	Type	Status
AppSecurityGroup	sg-0a25191798ae5b739	AWS::EC2::SecurityGroup	CREATE_COMPLETE
IGW	igw-0e65c17866bf37dab	AWS::EC2::InternetGateway	CREATE_COMPLETE
LabVPC	vpc-09651790c02e86a64	AWS::EC2::VPC	CREATE_COMPLETE
PublicRoute	rtb-0f0a6eac0567f79b8 0.0.0.0/0	AWS::EC2::Route	CREATE_COMPLETE
PublicRouteTable	rtb-0f0a6eac0567f79b8	AWS::EC2::RouteTable	CREATE_COMPLETE
PublicRouteTableAssociation	rtbassoc-09f0bcd9b9e9d081	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE
PublicSubnet	subnet-01179903ebbf3ac77	AWS::EC2::Subnet	CREATE_COMPLETE
VPCtoIGWConnection	IGW vpc-09651790c02e86a64	AWS::EC2::VPCGatewayAttachment	CREATE_COMPLETE

Step 7: Review Stack Status

In the **Stacks** section, wait for the **Lab** stack Status to **CREATE_COMPLETE**.

Stacks (1)

Q Filter by stack name

Filter status

Active

View nested

< 1 >

⚙

Stack name	Status	Created time	Description
Lab	CREATE_COMPLETE	2024-06-04 11:28:16 UTC-0500	Lab template



Task 2

Add an Amazon S3 Bucket to the Stack

Step 1: Edit the .yaml file

Edit the .yaml file to add an Amazon S3 bucket to the template.

```
Resources:
  #####
  # S3 Bucket
  #####

  MyBucket:
    Type: AWS::S3::Bucket
```

Step 2: Update stack

In the **Stacks** section, select the **Lab** stack, and choose [Update](#).

Stacks (1)

Delete

Update

Stack actions ▾

Create stack ▾

Filter by stack name

Filter status

Active ▾

View nested

< 1 >

Stack name	Status	Created time ▾	Description
<div><div></div>Lab</div>	<div><div></div>CREATE_COMPLETE</div>	2024-06-04 11:28:16 UTC-0500	Lab template



Task 2

Add an Amazon S3 Bucket to the Stack

Step 3: Stack Update details

In the **Prepare template** and **Specify template** sections, configure the following settings.

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.

☒

Replace existing template

Replace your existing template with a new template.

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.

☒

Upload a template file

Upload a template file

Choose file

task2.yaml

X

JSON or YAML formatted file

Step 4: Review Changes

The **Changes** section indicates that CloudFormation will **Add** an Amazon S3 bucket. All other resources defined in the template will be unchanged. This demonstrates that it is fast and easy to add additional resources to an existing stack, since those resources do not need to be redeployed.

Changes (1)			
<div><div>Q Search changes</div><div>< 1 ></div></div>			
Action	Logical ID	Physical ID	Resource type
Add	MyBucket	-	AWS::S3::Bucket



Task 2

Add an Amazon S3 Bucket to the Stack

Step 5: Review Events

Click the **Events** tab. The listing shows the bucket creation activities performed by CloudFormation.

Events (34)			
<div>Detect root cause</div>			
<div>Search events</div>			
Timestamp	Logical ID	Status	Detailed status
2024-06-04 11:45:15 UTC-0500	Lab	UPDATE_COMPLETE	-
2024-06-04 11:45:14 UTC-0500	Lab	UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	-
2024-06-04 11:45:12 UTC-0500	MyBucket	CREATE_COMPLETE	-
2024-06-04 11:44:49 UTC-0500	MyBucket	CREATE_IN_PROGRESS	-

Step 6: Review Resources

Click the **Resources** tab. The bucket will now be displayed in the list of resources. CloudFormation will have assigned it a random name so that it does not conflict with any existing buckets.

Resources (1)			
<div>MyBucket</div>			
Logical ID	Physical ID	Type	Status
MyBucket	lab-mybucket-tbmzvwfmzswm	AWS::S3::Bucket	CREATE_COMPLETE



Task 3

Add an Amazon EC2 Instance to the Stack

Step 1: Edit the .yaml file

Edit the .yaml file to add a parameter and an Amazon EC2 instance to the template.

Parameters:

AmazonLinuxAMIID:

Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>

Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:

EC2 Instance
#####

MyInstance:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref AmazonLinuxAMIID

InstanceType: t3.micro

SubnetId: !Ref PublicSubnet

SecurityGroups:

- !Ref AppSecurityGroup

Tags:

- Key: Name

Value: App Server

Step 2: Update stack

In the **Stacks** section, select the **Lab** stack, and choose [Update](#).

Stacks (1)

Delete

Update

Stack actions ▾

Create stack ▾

🔍

Filter by stack name

Filter status

Active ▾

View nested

<

1

>

Stack name	Status	Created time ▾	Description
<div><div></div>Lab</div>	<div><div></div>UPDATE_COMPLETE</div>	2024-06-04 11:28:16 UTC-0500	Lab template



Task 3

Add an Amazon EC2 Instance to the Stack

Step 3: Stack Update details

In the **Prepare template** and **Specify template** sections, configure the following settings.

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.

☒ **Replace existing template**
Replace your existing template with a new template.

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.

☒ **Upload a template file**

Upload a template file

task3.yaml

JSON or YAML formatted file

Step 4: Review Changes

The **Changes** section indicates that CloudFormation will **Add** an Amazon EC2 instance.

Changes (1)			
<input type="text" value="Search changes"/>			
Action	Logical ID	Physical ID	Resource type
Add	MyInstance	-	AWS::EC2::Instance



Task 3

Add an Amazon EC2 Instance to the Stack

Step 5: Review Events

Click the **Events** tab. The listing shows the instance creation activities performed by CloudFormation.

Events (47)			
<div>Detect root cause</div>			
<div>Search events</div>			
Timestamp	Logical ID	Status	Detailed status
2024-06-04 12:09:12 UTC-0500	Lab	UPDATE_COMPLETE	-
2024-06-04 12:09:11 UTC-0500	Lab	UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	-
2024-06-04 12:09:09 UTC-0500	MyInstance	CREATE_COMPLETE	-
2024-06-04 12:08:58 UTC-0500	MyInstance	CREATE_IN_PROGRESS	-

Step 6: Review Resources

Click the **Resources** tab. The instance will now be displayed in the list of resources.

Resources (1)			
<div>MyInstance</div>			
Logical ID	Physical ID	Type	Status
MyInstance	i-088dd1634c47ca97e	AWS::EC2::Instance	CREATE_COMPLETE

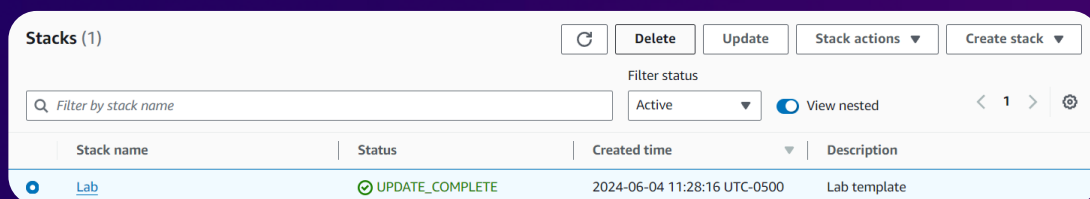


Task 4

Delete the Stack

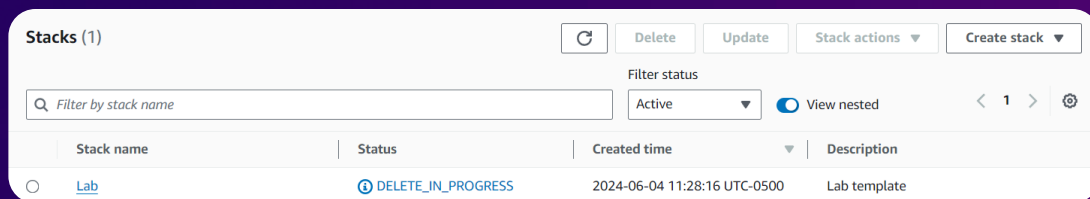
Step 1: Delete stack

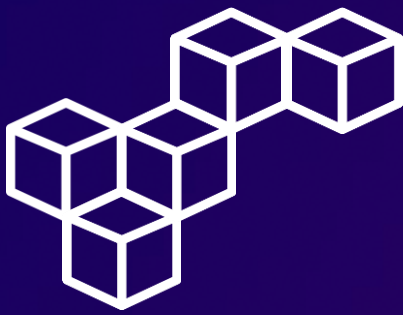
In the **Stacks** section, select the **Lab** stack, and choose **Delete**.



Step 2: Review stack deletion

The stack will show **DELETE_IN_PROGRESS** status. After a few minutes, the stack will disappear.





Conclusions

CloudFormation Stacks

CloudFormation Stacks enable the deployment and management of AWS resources as a single unit, ensuring consistency and efficiency across your infrastructure.

CloudFormation Templates

CloudFormation Templates define the structure and configuration of AWS resources, providing a blueprint for automated and repeatable deployments.

CloudFormation Template Parameters

CloudFormation Template Parameters allow customization of stack configurations, offering flexibility and reusability for different environments and use cases.

CloudFormation Template Resources

CloudFormation Template Resources specify the AWS resources to be created, ensuring precise and consistent provisioning as defined in the template.

CloudFormation Template Outputs

CloudFormation Template Outputs provide useful information about the resources created, such as resource IDs and endpoint URLs, facilitating easy access and integration.



Cristhian Becerra



[cristhian-becerra-espinoza](#)



+51 951 634 354



cristhianbecerra99@gmail.com



Lima, Peru

