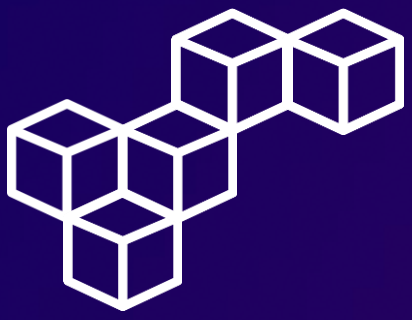
AWS re:Start LAB

Automation with CloudFormation

WEEK 11

aws re/start



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.

```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
```

```yaml
##########
# 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}
```

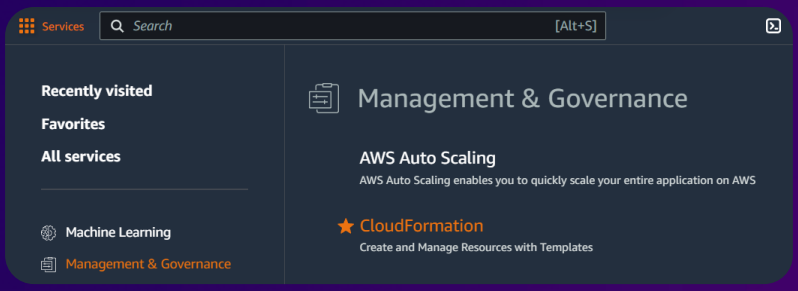
aws re/start

Task 1

Deploy a CloudFormation Stack

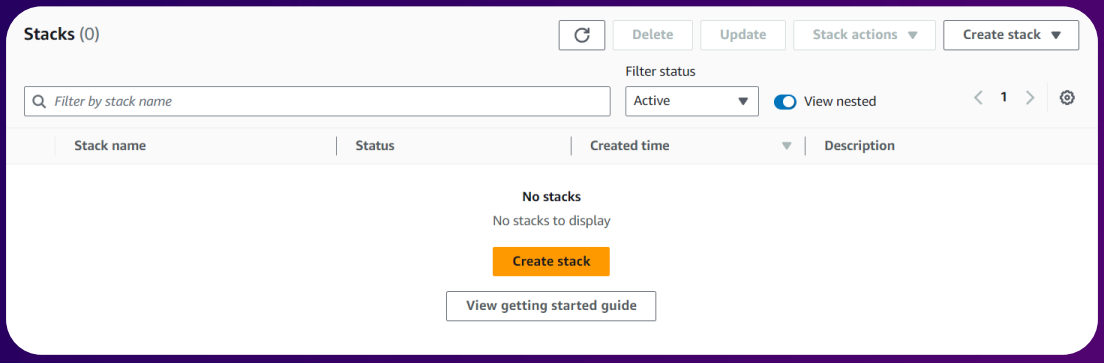
Step 2: Access the CloudFormation service

In the AWS Management Console, select CloudFormation.



Step 3: Create stack

In the **Stacks** section, select Create stack.



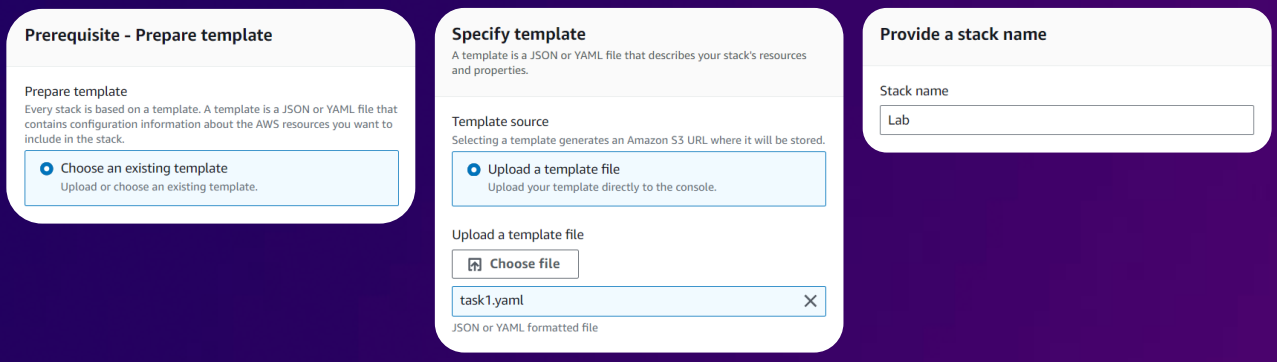
aws re/start

Task 1

Deploy a CloudFormation Stack

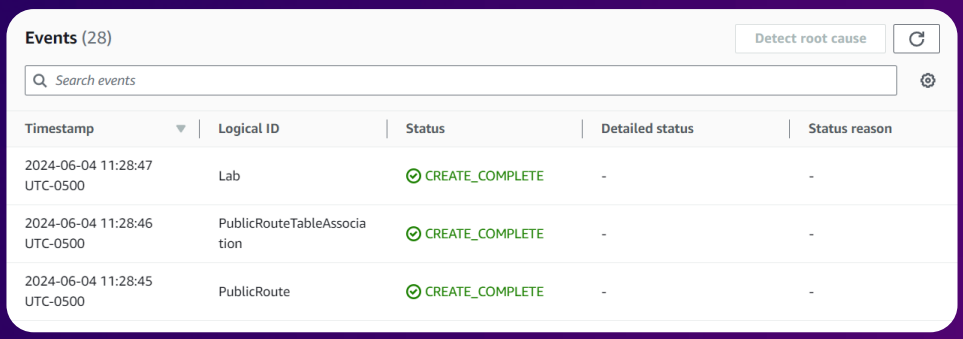
Step 4: Stack Creation details

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



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.



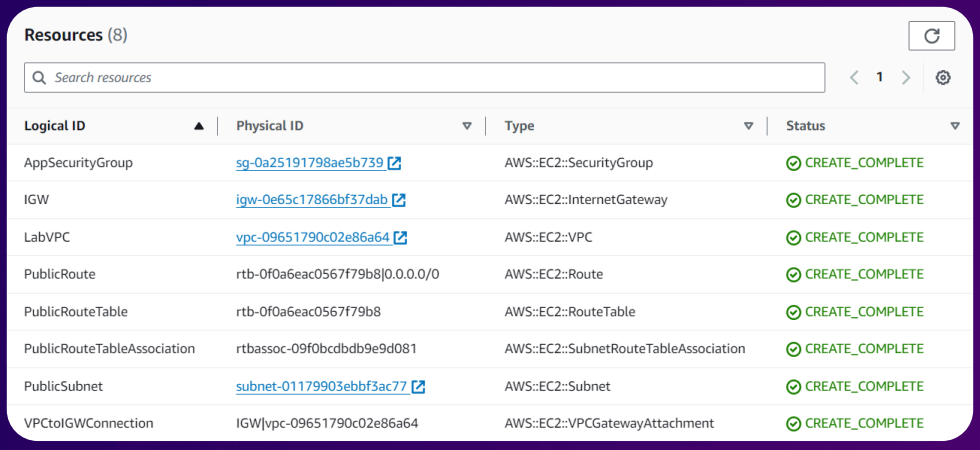
aws re/start

Task 1

Deploy a CloudFormation Stack

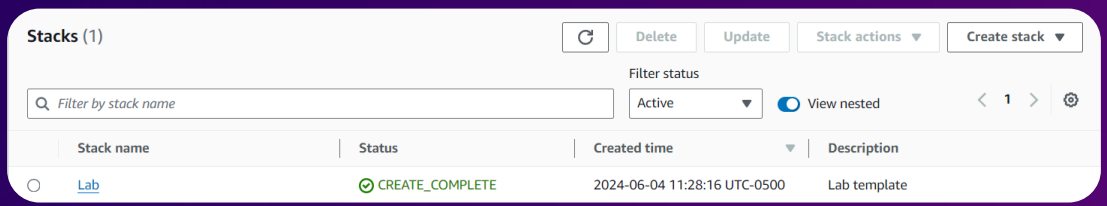
Step 6: Review Resources

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.



Step 7: Review Stack Status

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



aws re/start

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.

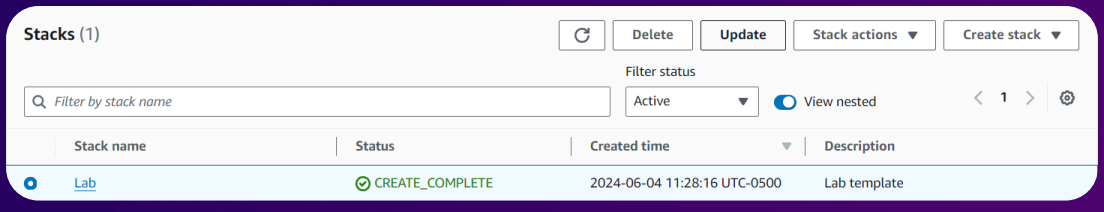
```
Resources:

##########
# S3 Bucket
##########

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

Step 2: Update stack

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



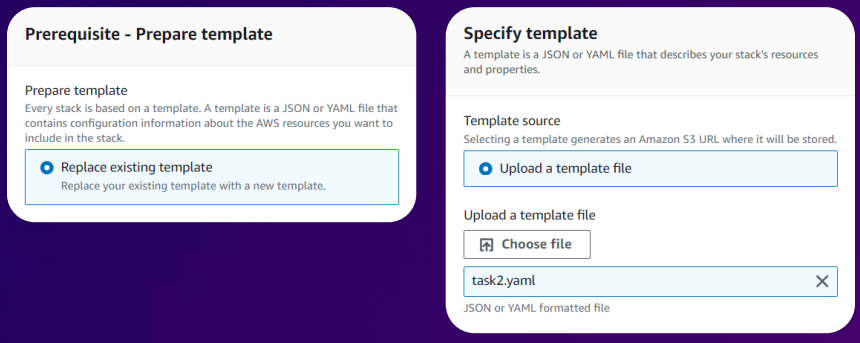
aws re/start

Task 2

Add an Amazon S3 Bucket to the Stack

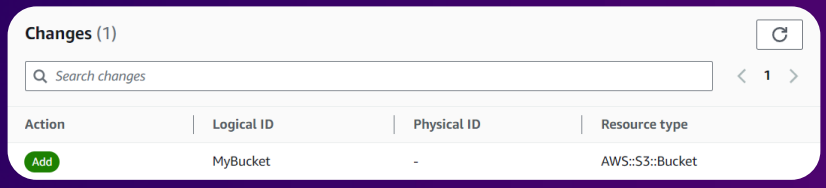
Step 3: Stack Update details

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



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.



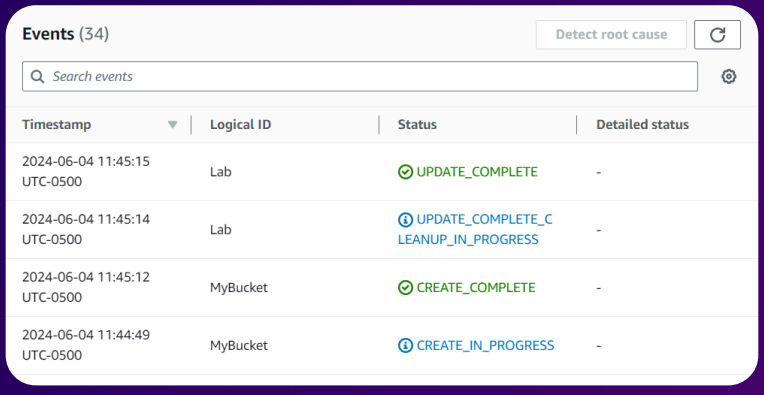
aws re/start

Task 2

Add an Amazon S3 Bucket to the Stack

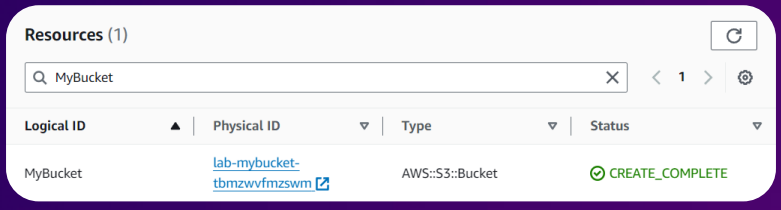
Step 5: Review Events

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



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.



aws re/start

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
```

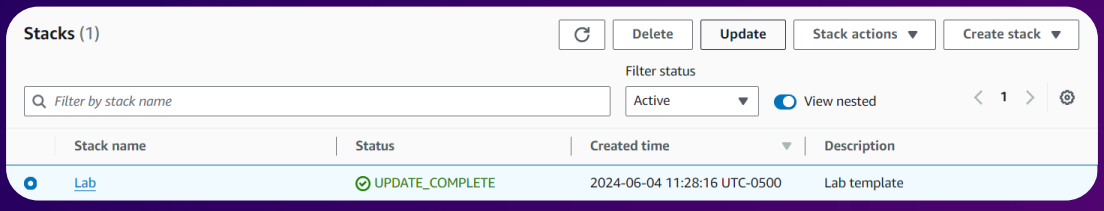
```
Resources:

##########
# EC2 Instance
##########

  MyInstance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref AmazonLinuxAMIID
      InstanceType: t3.micro
      SubnetId: !Ref PublicSubnet
      SecurityGroupsIds:
        - !Ref AppSecurityGroup
      Tags:
        - Key: Name
          Value: App Server
```

Step 2: Update stack

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



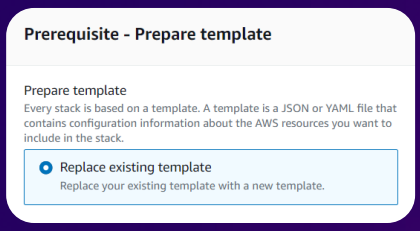
aws re/start

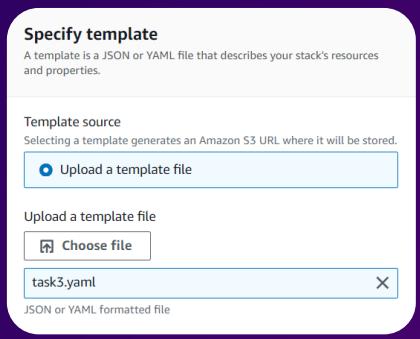
Task 3

Add an Amazon EC2 Instance to the Stack

Step 3: Stack Update details

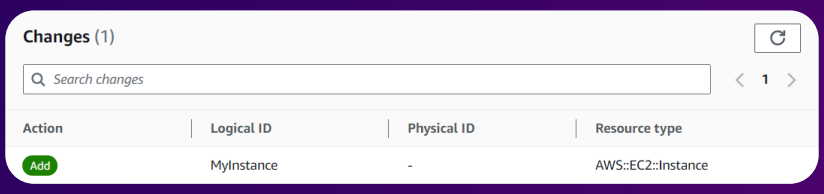
In the **Prepare template** and **Specify template** sections, configure the following settings.





Step 4: Review Changes

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



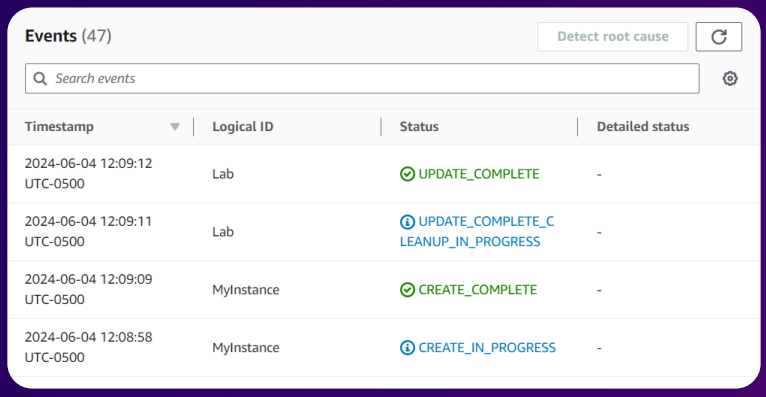
aws re/start

Task 3

Add an Amazon EC2 Instance to the Stack

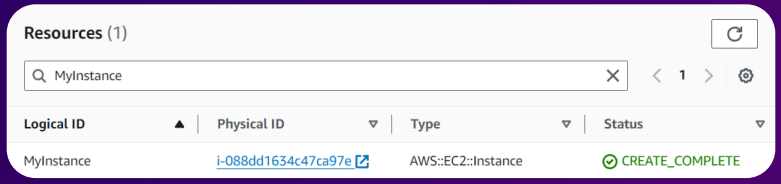
Step 5: Review Events

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



Step 6: Review Resources

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



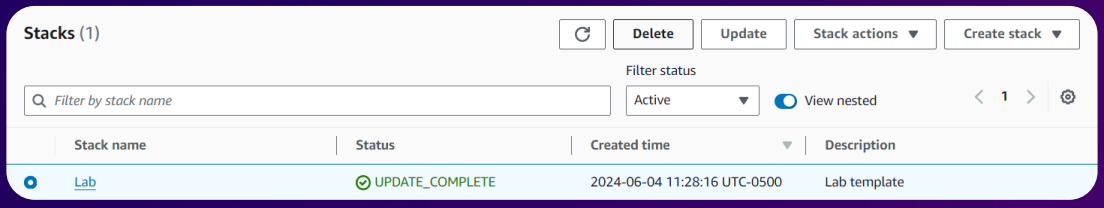
aws re/start

Task 4

Delete the Stack

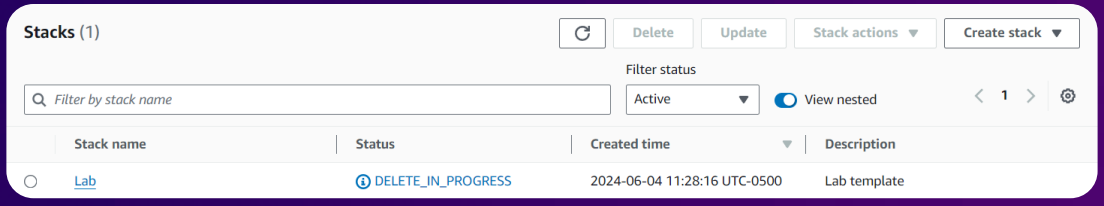
Step 1: Delete stack

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

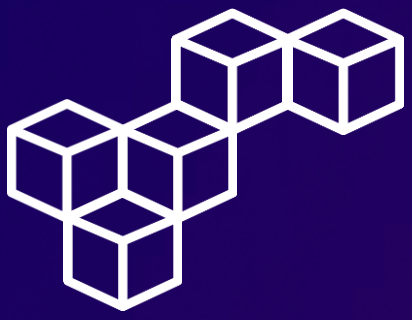


Step 2: Review stack deletion

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



aws re/start



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.

aws re/start

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru