CloudFormation

TECHNICAL

What is CloudFormation

AWS CloudFormation simplifies the task of repeatedly and predictably creating groups of related resources that power your applications.

What is cloudformation?

- Cloud formation is a declarative way of outlining your AWS Infrastructure, for any resources (Most of them are supported)
- Gives you an easy way to model a collection of related AWS and third-party resources, provision them quickly and consistently, and manage them throughout their lifecycles, by treating infrastructure as code
- Describes your desired resources and their dependencies so you can launch and configure them together as a stack
- You can use a template to create, update, and delete an entire stack as a single unit, as often as you need to, instead of managing resources individually
- You can manage and provision stacks across multiple AWS accounts and AWS Regions

What is cloudformation?

For example, within a CloudFormation template:

- Need a security group
- Need two EC2 Instances using the above security group
- Need two elastic IPs for these EC2 Instances
- Need an S3 bucket
- Need a load balancer (ELB) in front of these instances

CloudFormation template creates above resources for you, in right order, with the exact configuration that you specify.

After the AWS resources are deployed, you can modify and update them in a controlled and predictable way

What is CloudFormation

CLOUDFORMATION IS ALL ABOUT AUTOMATING RESOURCE PROVISONINING

What is CloudFormation

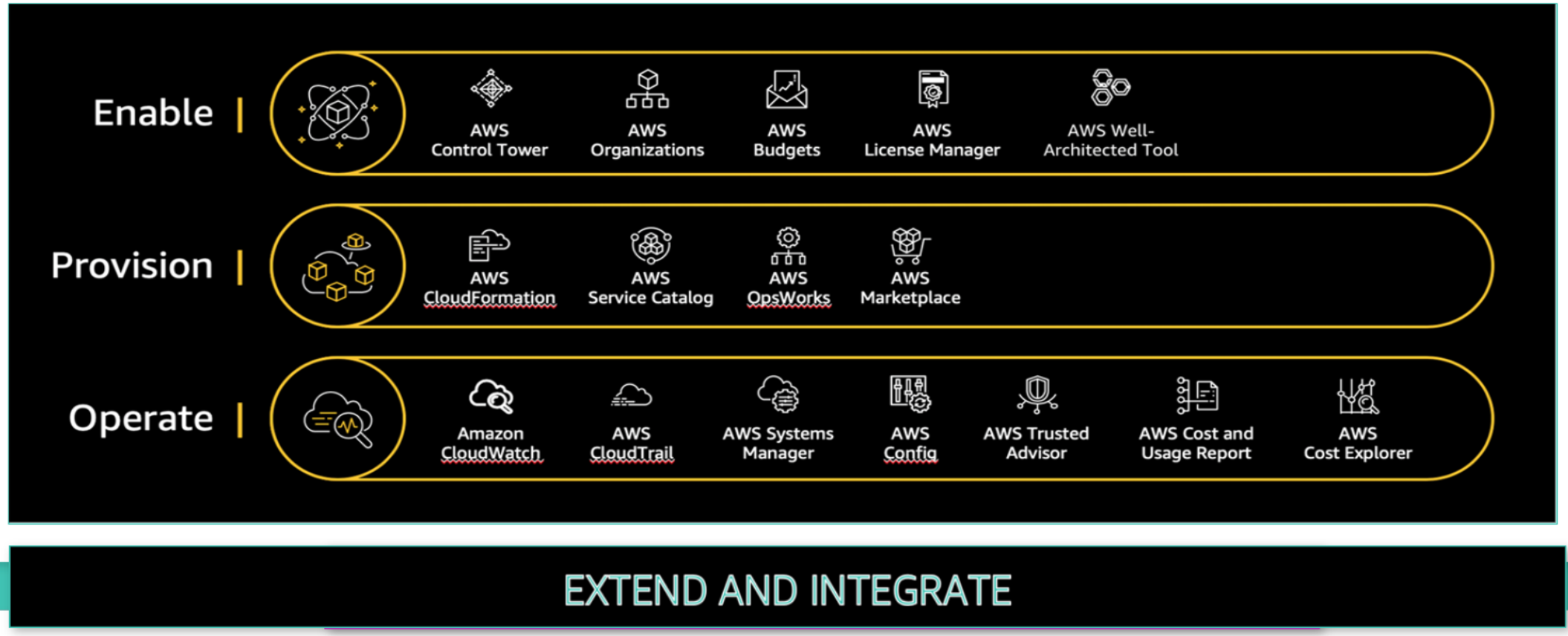
- CloudFormation is a declarative way of outlining your AWS Infrastructure, for any resources (most of them are supported).
- For example, within a CloudFormation template, you say:
  - I want a security group
  - I want two EC2 machines using this security group
  - I want two Elastic IPs for these EC2 machines
  - I want an S3 bucket
  - I want a load balancer (ELB) in front of these machines
- Then CloudFormation creates those for you, in the **right order**, with the **exact configuration** that you specify

CloudFormation Service

AWS CloudFormation

- Fully-managed service
- Create, update and delete resources in stacks

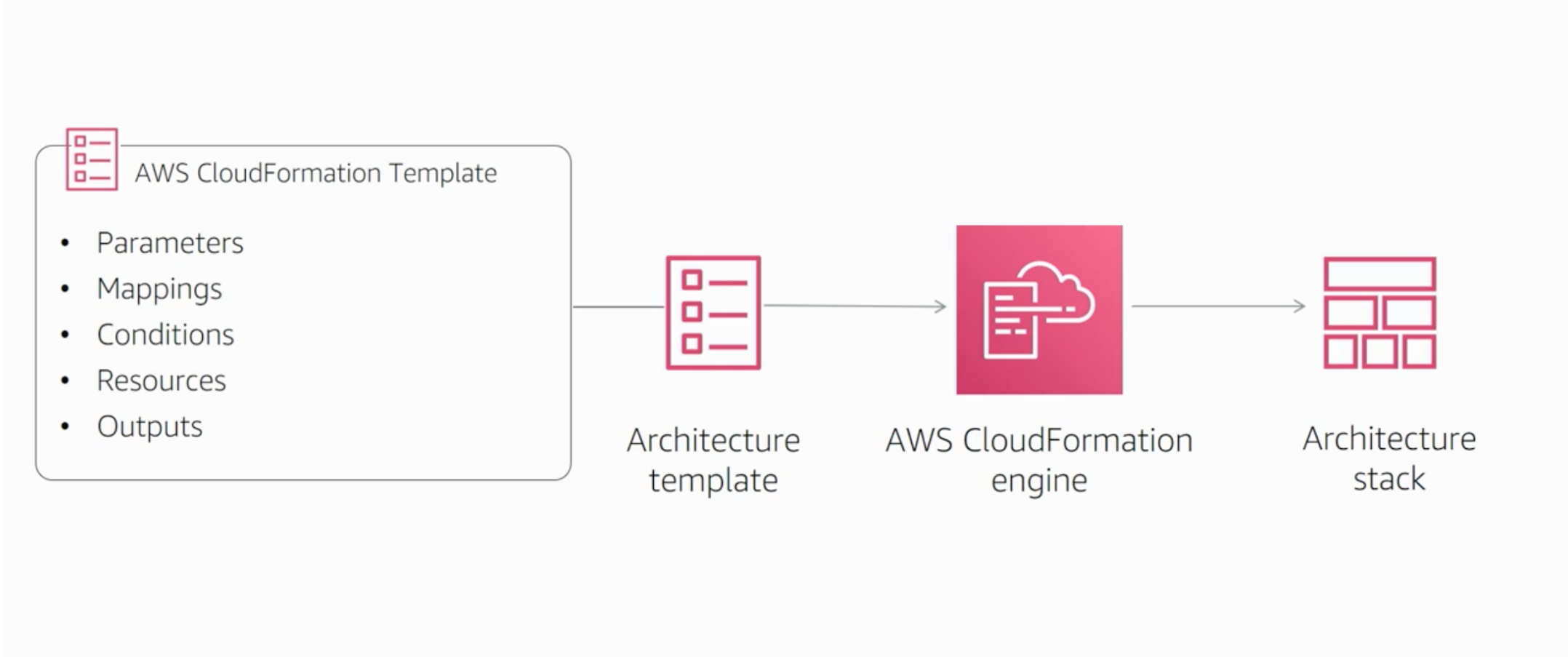
AWS Management and governance services



AWS
CloudFormation

- Infrastructure as Code
- Version control/replicate/update templates like code
- Integrates with development, CI/CD, management tools
- Run automated testing for CI/CD environments

AWS CloudFormation: Components



Template Section

- Templates can include several major sections
  - AWSTemplateFormatversion
  - Description
  - Parameters
  - Mappings
  - Conditions
  - Resources → Mandatory
  - Metadata
  - Outputs

Elements of a template

- Mandatory Elements:
  - List of AWS Resources and their associated configuration values
- Optional Elements:
  - Template's file format and version
  - Template parameters
    - **The input values are supplied at the stack creation time**
  - Output Values
    - **The output values required once a stack creation finished building (such as public ip, ELB address etc..)**

Parameters

Use the
optional Parameters section to
customize your templates.

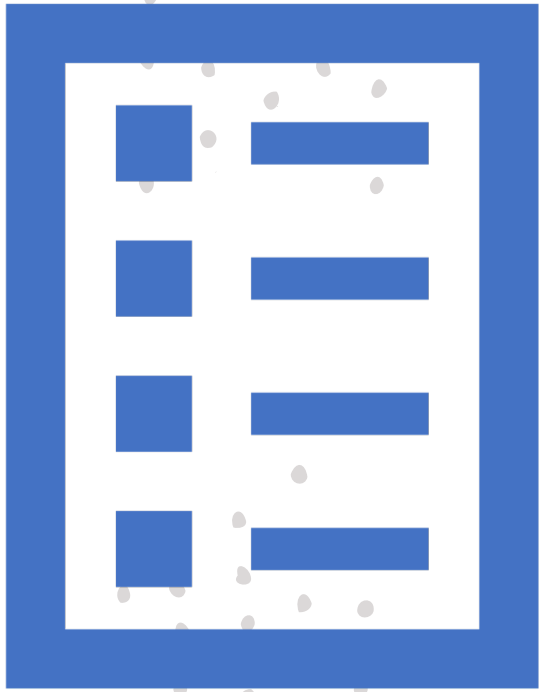
Parameters enable you to input
custom values to your template
each time you create or update a
stack.

Mapping

The optional Mappings section matches a key to a corresponding set of named values.

For example, if you want to set values based on a region, you can create a mapping that uses the region name as a key and contains the values you want to specify for each specific region.

You use the ***Fn::FindInMap*** intrinsic function to retrieve values in a map.



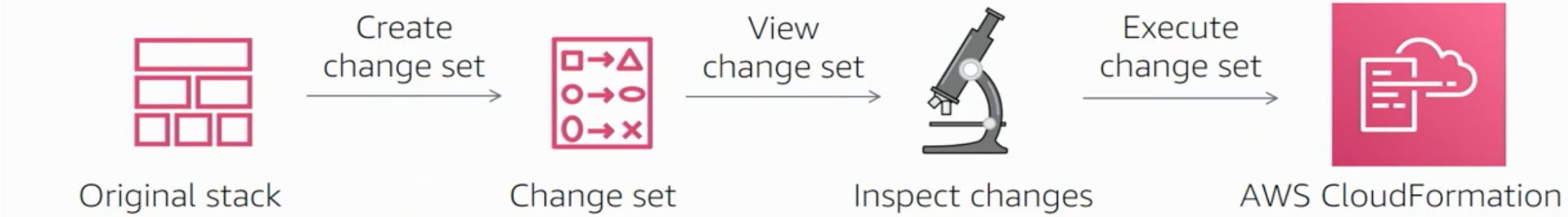
Outputs

- The optional Outputs section declares output values that you can import into other stacks, return in response (to describe stack calls), or view on the AWS CloudFormation Console. For example, you can output the S3 bucket name for a stack to make the bucket easier to find.

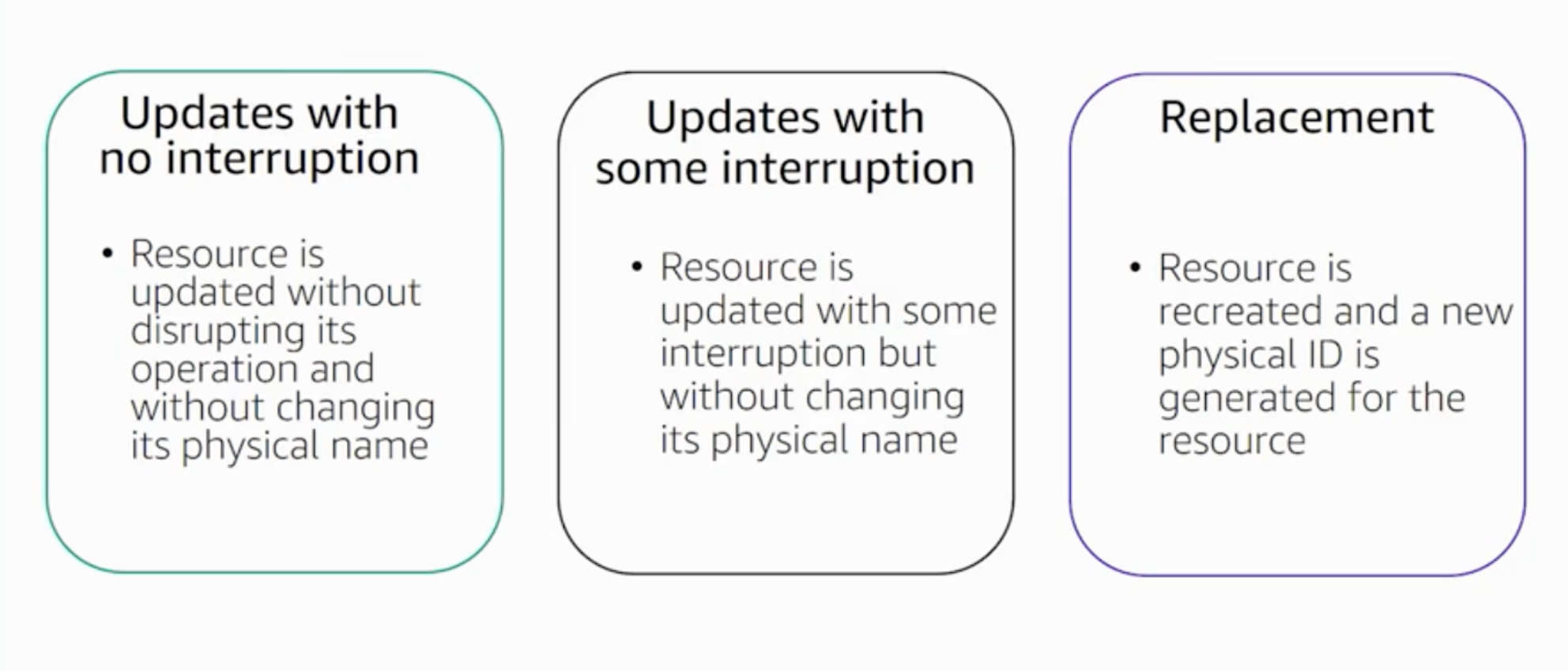
Resources

- The required Resources section declares the AWS resources that you want to include in the stack, such as an Amazon EC2 instance or an Amazon S3 bucket.

AWS CloudFormation: updating stacks

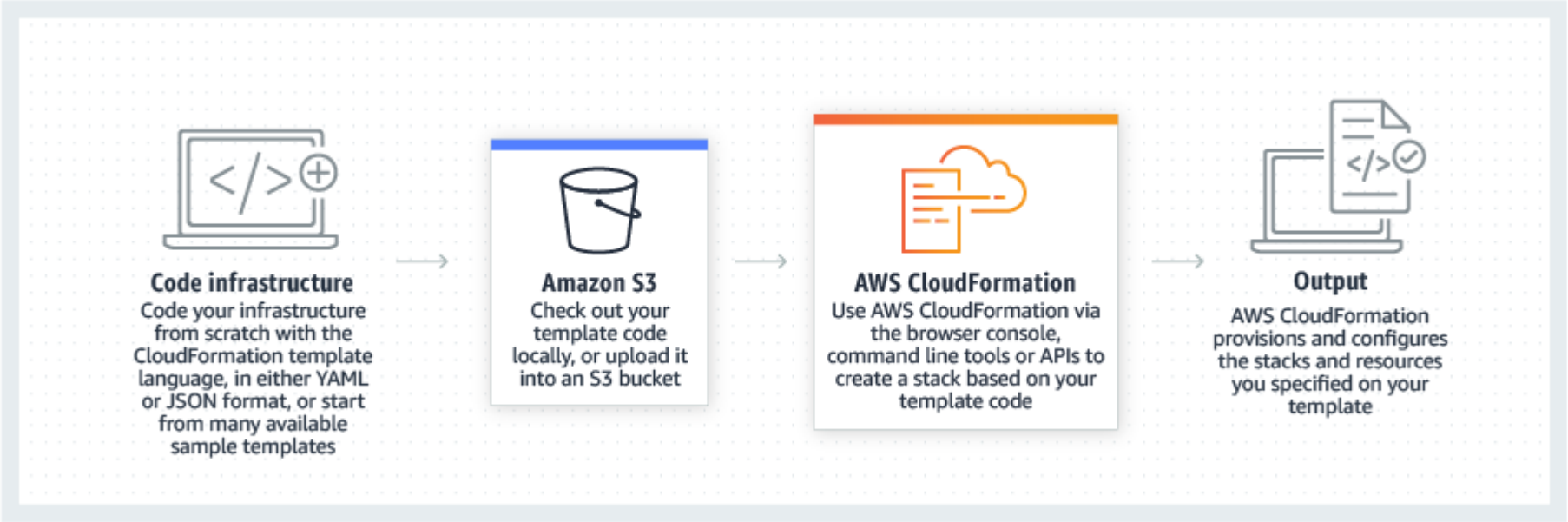


AWS CloudFormation: updating stacks



Cloudfromation Template vs Stack

- A CloudFormation template is essentially an architectural diagram and a CloudFormation Stack is the end result of that diagram (after provisioning)
- CloudFormation templates are in the JSON format or YAML



Benefits of cloudformation!

- Infrastructure as a code (IAAS)
- No resources are manually created, which is great for control
- Code can be version controlled, example using GIT
- Ability to destroy & re-create an infrastructure on the cloud on the fly
- Automated generation of Diagram for templates
- By default, the “Automatic rollback on error” feature is enabled
- CloudFormation is FREE
- Stacks can wait for applications to be provisioned using the “WaitCondition”
- You can use Fn:GetAtt to output data

Cloudformation parameters

- Parameters are the way to provide inputs to the template
- Helps in reuse the them across the template
- Some inputs cannot be determined ahead of time
- Parameters are extremely powerful & controlled

Cloudformation mappings

- Mapping are the fixed variables within your CloudFormation template
- They're very handy to differentiate between different environments (dev vs prod), regions, AMI types etc..
- All values are hardcoded within the template
- Example:

```
Mappings:
  Mapping01:
    Key01:
      Name: Value01
    Key02:
      Name: Value02
    Key03:
      Name: Value03
```

```
RegionMap:
  us-east-1:
    "32": "ami-6411e20d"
    "64": "ami-7a11e213"
  us-west-1:
    "32": "ami-c9c7978c"
    "64": "ami-cfc7978a"
  eu-west-1:
    "32": "ami-37c2f643"
    "64": "ami-31c2f645"
```

Cloudformation Conditions

- Conditions are used to control the creation of resources or outputs based on a condition
- Conditions can be whatever you want them to be example:
  - Environment (dev/test/prod)
  - AWS Region
  - Any parameter value
- Each condition can reference another condition, parameter value or mapping

Cloudformation Conditions

```
Conditions:
  Logical ID:
    Intrinsic function
```

- We need to choose logical id, That is how we name condition
- The intrinsic function (logical) can be any of the following
  - Fn::And
  - Fn::Equals
  - Fn::If
  - Fn::Not
  - Fn::Or

Cloudformation Resources

Resources are the core of the template

Resources represent the different AWS Components that will be created and configured

Resources are declared and can reference each other

AWS will take care of creation, updates & deletes of resources for us

There are over 224 types of resources

Resource types identifiers are the form :

AWS::aws-product-name::data-type-name

Cloudformation Metadata

```
Metadata:
  Instances:
    Description: "Information about the instances"
  Databases:
    Description: "Information about the databases"
```

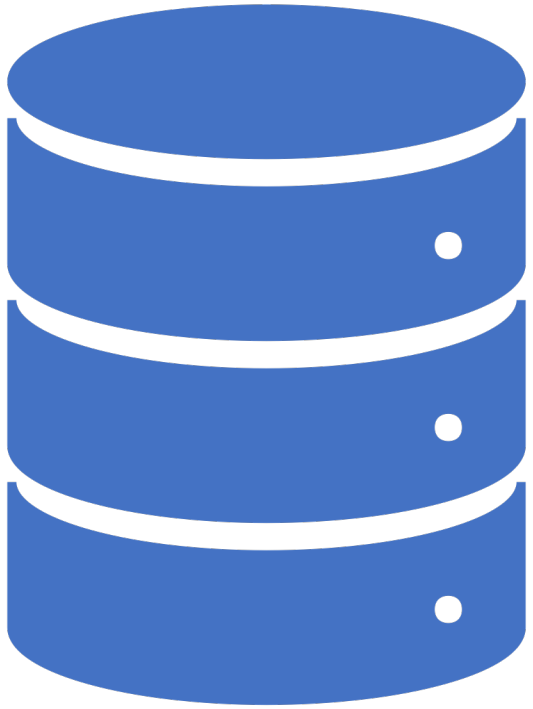
- You can use the optional metadata section to include arbitaryYAML that provide details about the template or resource
- Example:

Cloudformation outputs

Outputs section declares optional output values that we can use import into other stacks

You can also view the outputs in the AWS Console or in using the CLI

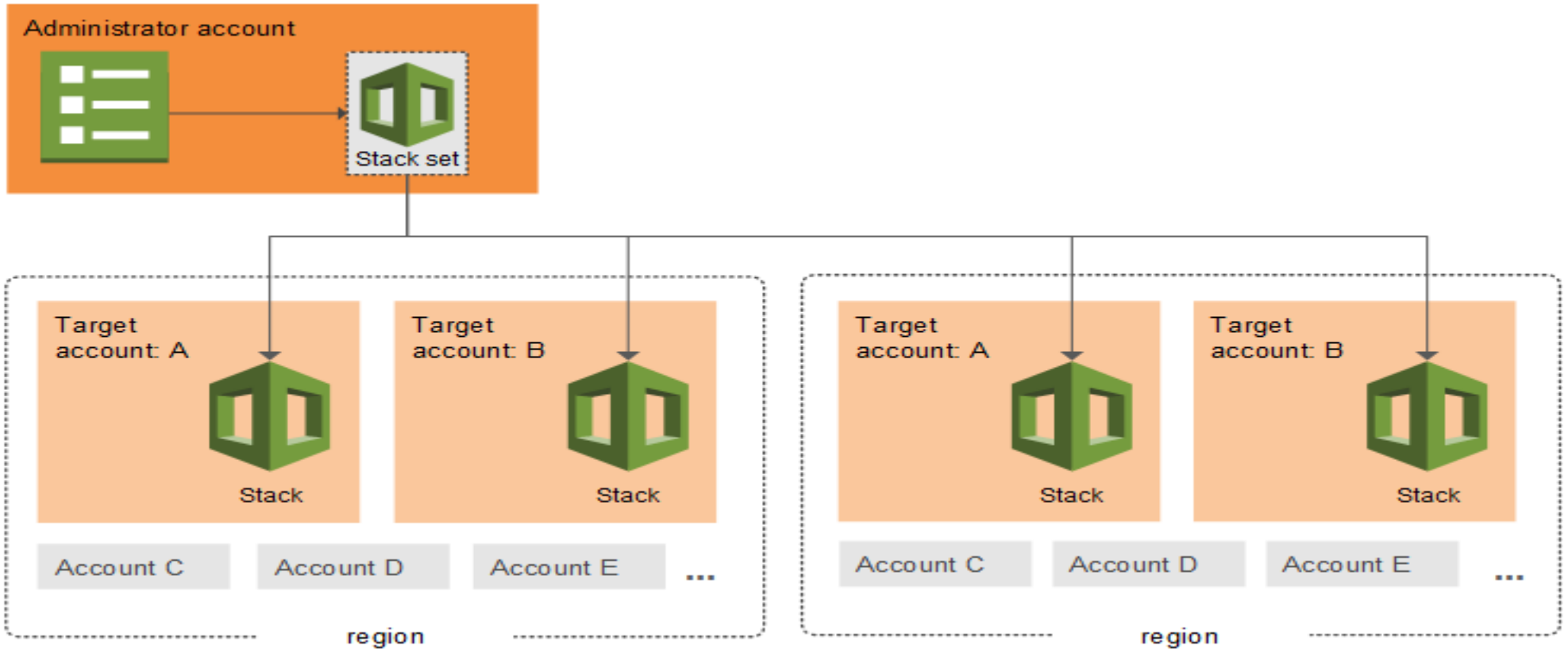
Outputs are very useful for example If you create a VPC and Subnets, output variable such as VPC ID and your subnet ids



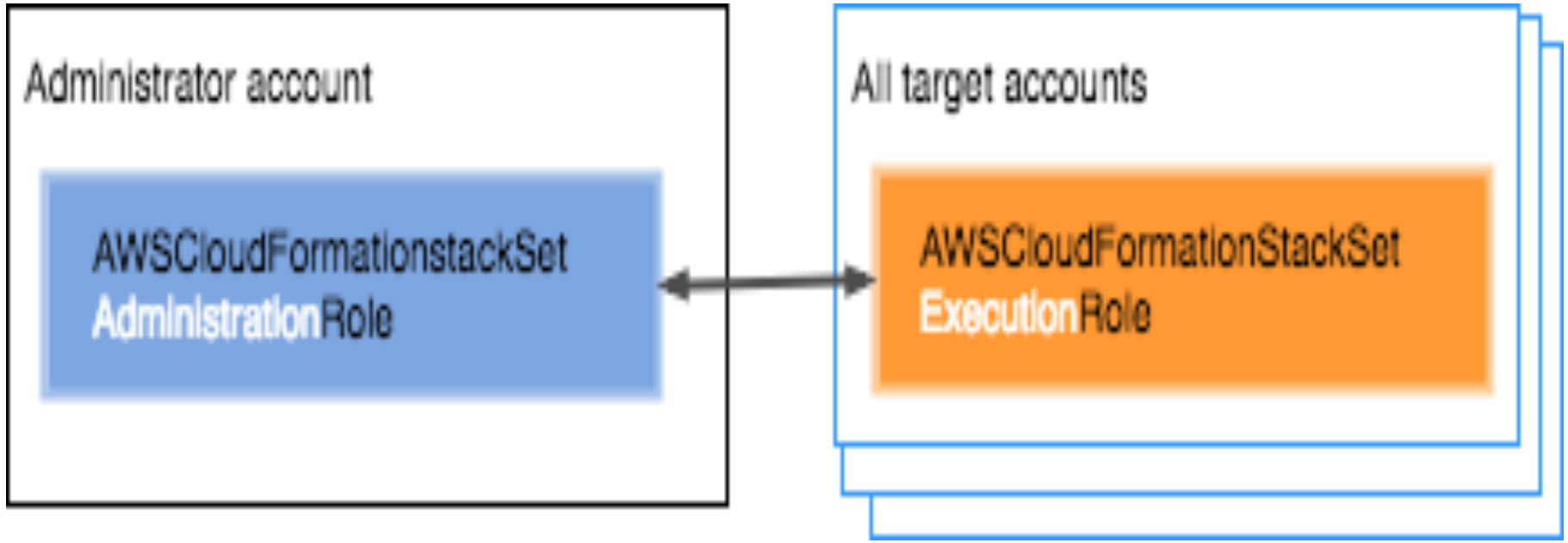
Working with AWS CloudFormation StackSets

- AWS CloudFormation StackSets extends the functionality of stacks by enabling you to create, update, or delete stacks across multiple accounts and regions with a single operation.

Working with AWS CloudFormation StackSets



Grant self-managed permissions



AWS CloudFormation StackSets and AWS Organizations

- Service-linked roles created when you enable integration

Troubleshooting: AWS CloudFormation Stacks

Obtaining Information about Your Stack

How can I obtain information about my stack?

A stack implements and manages the group of resources outlined in your template so the state and dependencies of those resources can be managed together

What information is available for my stack?

If CloudFormation fails to create, update, or delete your stack, you can view error messages or logs to help you learn more about the issue.

The following screenshots describe the types of information available for troubleshooting a CloudFormation issue.

Stack info

- The *Stack info* tab shows basic information about each stack: stack ID, status, description, and when it was last created, deleted, or updated

Stacks (2)

Filter by stack name

Active View nested

1

Q1
2021-01-20 10:17:22 UTC-0500
CREATE_COMPLETE

Q2
2021-01-20 10:11:26 UTC-0500
ROLLBACK_COMPLETE

Q1

Delete Update Stack actions Create stack

Stack info | Events | Resources | Outputs | Parameters | Template | Change sets

Overview

Stack ID
arn:aws:cloudformation:us-east-1:3[redacted]40:stack/Q1/9[redacted]0-5b32-11eb-90af-0a39c55408a9

Description
Simple SQS example

Status
CREATE_COMPLETE

Status reason
-

Root stack
-

Parent stack
-

Created time
2021-01-20 10:17:22 UTC-0500

Deleted time
-

Updated time
-

Drift status
NOT_CHECKED

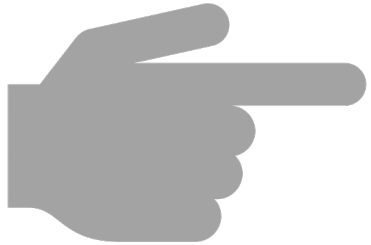
Last drift check time
-

Termination protection

IAM role

Resources

The *Resources* tab shows information about all the resources grouped by this stack.



You can view many of the resources through links to the AWS Management Console.

Stacks (2)

Filter by stack name

Active View nested

1

Q1
2021-01-20 10:17:22 UTC-0500
CREATE_COMPLETE

Q2
2021-01-20 10:11:26 UTC-0500
ROLLBACK_COMPLETE

Q1

Delete Update Stack actions Create stack

Stack info | Events | Resources | Outputs | Parameters | Template | Change sets

Resources (2)

Search resources

Logical ID	Physical ID	Type	Status	Status reason
TheQueue	https://sqs.us-east-1.amazonaws.com/353[redacted]0/MyFirstQueue	AWS::SQS::Queue	CREATE_COMPLETE	-
TheQueueUpdaterRole	Q1-TheQueueUpdaterRole-1GE4AQN56LGW	AWS::IAM::Role	CREATE_COMPLETE	-

Events

The *Events* tab shows all the events that have been generated for the stack, with their timestamp.

This is very useful when trying to figure out what exactly is happening if something goes wrong.

Stacks (2)

Filter by stack name

Active View nested

1

Q1
2021-01-20 11:26:09 UTC-0500
CREATE_COMPLETE

Q2
2021-01-20 10:11:26 UTC-0500
ROLLBACK_COMPLETE

Q1

Delete Update Stack actions Create stack

Stack info | Events | Resources | Outputs | Parameters | Template | Change sets

Events (8)

Search events

Timestamp	Logical ID	Status	Status reason
2021-01-20 11:26:33 UTC-0500	Q1	CREATE_COMPLETE	-
2021-01-20 11:26:31 UTC-0500	TheQueueUpdaterRole	CREATE_COMPLETE	-
2021-01-20 11:26:18 UTC-0500	TheQueueUpdaterRole	CREATE_IN_PROGRESS	Resource creation Initiated
2021-01-20 11:26:17 UTC-0500	TheQueueUpdaterRole	CREATE_IN_PROGRESS	-
2021-01-20 11:26:16 UTC-0500	TheQueue	CREATE_COMPLETE	-

AWS CLI

- **With the AWS Command Line Interface (AWS CLI), you can create, monitor, update, and delete stacks from your system's terminal. You can also use the AWS CLI to automate actions through scripts**

How can I obtain information about my stack using the AWS CLI?

- The AWS CLI can show you all the stack information. The following commands are useful and take a stack name as argument (--stack-name "...").

aws cloudformation describe-stacks

This AWS CLI command shows the basic information about the stack, similar to the *Stack info* tab in the AWS Management Console.

karamo@ATL11-FV656A8: ~

```
>
>aws cloudformation describe-stacks --stack-name Q1
{
    "Stacks": [
        {
            "StackId": "arn:aws:cloudformation:us-east-1:35[redacted]0:stack/Q1/343d4c90-5b3c-11eb-bfcd-0a[redacted]l",
            "StackName": "Q1",
            "Description": "Simple SQS example",
            "Parameters": [
                {
                    "ParameterKey": "QueueName",
                    "ParameterValue": "TheQueue1"
                }
            ],
            "CreationTime": "2021-01-20T16:26:09.818Z",
            "LastUpdatedTime": "2021-01-20T22:43:49.004Z",
            "RollbackConfiguration": {},
            "StackStatus": "UPDATE_COMPLETE",
            "DisableRollback": false,
            "NotificationARNs": [],
            "Capabilities": [
                "CAPABILITY_IAM"
            ],
            "Outputs": [
                {
                    "OutputKey": "QueueARN",
                    "OutputValue": "arn:aws:sqs:us-east-1:35[redacted]0:TheQueue1"
                },
                {
                    "OutputKey": "QueueUrl",
                    "OutputValue": "https://sqs.us-east-1.amazonaws.com/35[redacted]0/TheQueue1"
                }
            ],
            "Tags": [],
            "EnableTerminationProtection": false,
```

aws cloudformation describe-stack-resources

- This command shows information about the resources belonging to a stack, similar to the *Resources* tab in the AWS Management Console.

```
karamo@ATL11-FV656A8: ~
>aws cloudformation describe-stack-resources --stack-name Q1
{
    "StackResources": [
        {
            "StackName": "Q1",
            "StackId": "arn:aws:cloudformation:us-east-1:3[redacted]0:stack/Q1/343d4c90-5b3c-11eb-bf[redacted]:1",
            "LogicalResourceId": "TheQueue",
            "PhysicalResourceId": "https://sqs.us-east-1.amazonaws.com/3[redacted]0/TheQueue1",
            "ResourceType": "AWS::SQS::Queue",
            "Timestamp": "2021-01-20T22:43:58.186Z",
            "ResourceStatus": "UPDATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "NOT_CHECKED"
            }
        },
        {
            "StackName": "Q1",
            "StackId": "arn:aws:cloudformation:us-east-1:3[redacted]0:stack/Q1/343d4c90-5b3c-11eb-b[redacted]:1",
            "LogicalResourceId": "TheQueueUpdaterRole",
            "PhysicalResourceId": "Q1-TheQueueUpdaterRole-18IOEQKJTKH6Q",
            "ResourceType": "AWS::IAM::Role",
            "Timestamp": "2021-01-20T22:44:13.117Z",
            "ResourceStatus": "UPDATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "NOT_CHECKED"
            }
        }
    ]
}
>
>
>
>
>
>
>_
```

aws cloudformation describe-stack-events

- This command shows information about all the events generated for a stack, in reverse chronological order, similar to the *Events* tab in the AWS Management Console.

```
>aws cloudformation describe-stack-events --stack-name Q1
{
    "StackEvents": [
        {
            "StackId": "arn:aws:cloudformation:us-east-1:3[redacted]0:stack/Q1/343d4c90-5b3c-11eb-bfcd-0a[redacted]1",
            "EventId": "4295c8d0-5b3c-11eb-bb1a-0adbfd4180bb",
            "StackName": "Q1",
            "LogicalResourceId": "Q1",
            "PhysicalResourceId": "arn:aws:cloudformation:us-east-1:3[redacted]0:stack/Q1/343d4c90-5b3c-11eb-bfcd-0a[redacted]1",
            "ResourceType": "AWS::CloudFormation::Stack",
            "Timestamp": "2021-01-20T16:26:33.811Z",
            "ResourceStatus": "CREATE_COMPLETE"
        },
        {
            "StackId": "arn:aws:cloudformation:us-east-1:3[redacted]0:stack/Q1/343d4c90-5b3c-11eb-bfcd-0a[redacted]1",
            "EventId": "TheQueueUpdaterRole-CREATE_COMPLETE-2021-01-20T16:26:31.892Z",
            "StackName": "Q1",
            "LogicalResourceId": "TheQueueUpdaterRole",
            "PhysicalResourceId": "Q1-TheQueueUpdaterRole-18IOEQKJTKH6Q",
            "ResourceType": "AWS::IAM::Role",
            "Timestamp": "2021-01-20T16:26:31.892Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "ResourceProperties": "{\"Policies\":[{\"PolicyName\":\"QueueUpdater\",\"PolicyDocument\":{\"Version\":\"2012-10-17\",\"Sta
tement\":[{\"Action\":[\"sqs:Get*\",\"sqs:List*\",\"sqs:SendMessage*\",\"sqs:ReceiveMessage\"],\"Resource\":\"arn:aws:sqs:us-east-1:3[redacted]0:TheQueue\",\"Effect\":\"Allow\"}]}}],\"AssumeRolePolicyDocument\":{\"Version\":\"2012-10-17\",\"Statement\":[{\"Action\":[\"s
ts:AssumeRole\"],\"Effect\":\"Allow\",\"Principal\":{\"Service\":[\"ec2.amazonaws.com\"]}}]}}"
        },
        {
            "StackId": "arn:aws:cloudformation:us-east-1:3[redacted]0:stack/Q1/343d4c90-5b3c-11eb-bfcd-0a[redacted]1",
            "EventId": "TheQueueUpdaterRole-CREATE_IN_PROGRESS-2021-01-20T16:26:18.181Z",
            "StackName": "Q1",
            "LogicalResourceId": "TheQueueUpdaterRole",
            "PhysicalResourceId": "Q1-TheQueueUpdaterRole-18IOEQKJTKH6Q",
            "ResourceType": "AWS::IAM::Role",
            "Timestamp": "2021-01-20T16:26:18.181Z",
```

What potential issues might I encounter?

- The following slides describe the four most common issues that you might encounter with your CloudFormation stack.

Your template doesn't pass validation.

- There is likely a syntax problem with your template, a typo, or you may be missing some information needed to create your resources.

Your stacks are not creating as expected.

- The template is valid, but there is a problem creating one or more resources.

Your stacks are not deleting as expected.

AWS will not delete some resources (usually those containing data, like Amazon S3 buckets) unless they are empty.

There may also be permission or other issues that prevent you from deleting some resources.

Your stacks are *not* updating as expected

This issue shares root causes in common with previous *not creating* and *not deleting* issues.

Some resource updates require the resource to be destroyed and a new one created.

- Now that you have examined potential root causes for each of the following issues, let's investigate troubleshooting methods that could resolve the issue.

What can I do if my stack won't create as expected?

If your template passes validation but your stack won't create as expected, it will still be listed in the console (or AWS CLI). From there, you can view information about which events were created for your stack and their status. You will get detailed information about which resources were not created and what the errors were.

If your stack doesn't create, you can take the following actions:

Verify events

Check the *Events* tab for your stack in the AWS Management Console to see which resources failed to create and the reason for failure.

For resources that fail to create, the *Status reason* field will provide more detailed information.

Check the template

- Go through the stack template, located in the *Template* tab in the AWS Management Console. See if the resource(s) have been defined correctly

Stacks (2)

Filter by stack name

Active View nested

1

Q1
2021-01-20 11:26:09 UTC-0500
CREATE_COMPLETE

Q2
2021-01-20 10:11:26 UTC-0500
ROLLBACK_COMPLETE

Q2

Delete Update Stack actions Create stack

Stack info | Events | Resources | Outputs | Parameters | Template | Change sets

Template

View in Designer

```
---
AWSTemplateFormatVersion: "2010-09-09"
Description: "Simple SQS example"
Parameters:
  QueueName:
    Type: String
    Default: TheQueue
    Description: Please enter the name of the Queue.

Resources:
  TheQueue:
    Type: AWS::SQS::Queue
    Properties:
      QueueName: !Ref QueueName

  TheQueueUpdaterRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          -
            Principal:
              Service:
              - ec2.amazonaws.com
            Action:
              - 'sts:AssumeRole'
      Policies:
        -
          PolicyName: QueueUpdater
```

Examine resources in the console

- Use the AWS Management Console or AWS CLI to verify that the resource exists. Your template may be trying to create a resource with the same name as one that already exists.

Try manual creation

Try creating the resource with the same configuration outside of CloudFormation. Use the AWS Management Console, AWS CLI, or AWS SDK.

Be aware that the AWS Management Console sometimes automatically creates dependent resources. An example might be AWS Identity and Access Management (IAM) roles, which you would need to include in your CloudFormation template.

What can I do if my stacks won't update?

View the **Events** tab to see the sequence of events and the resources that were being updated. The resource that failed to update should have UPDATE_FAILED marked next to it and also the reason for the failure.

Choose the **Properties** section to view the configuration properties that were being applied to the resource. Determine if the change being applied is valid. This can be done using the documentation for the resource and reviewing the property types.