

AWS CloudFormation

Infrastructure as Code

Infrastructure as Code

- Infrastructure as Code (IaC) is a way to provision and manage your infrastructure through code instead of through manual processes
- With IaC, you can create configuration files that contain your infrastructure specifications, which makes it easier to edit and distribute configurations.
- You provision the same environment every time in other account or region
- That code would be deployed and create/update/delete our infrastructure
- Version control is an important part of IaC, and your configuration files should be under source control just like any other software source code file.

What is CloudFormation

- You can use AWS CloudFormation to create AWS resources in an orderly and predictable fashion.
- Resources are written in text files using **JSON** or **YAML** format.
- The templates require a specific syntax and structure that depends on the types of resources being created and managed.
- A CloudFormation template is deployed into the AWS environment as a stack
- For example, within a CloudFormation template, you can create a set of:
 - One security group
 - Two EC2 instance using this security group
 - One S3 bucket
 - One load balancer (ELB) in front of these EC2

CloudFormation Concept

- When you use AWS CloudFormation, you work with templates and stacks
- Topics
 - Templates
 - Stacks
 - Change sets
- **Templates**
 - A CloudFormation template is a JSON or YAML formatted text file
 - CloudFormation uses these templates as blueprints for building your AWS resources

YAML

```
AWS::CloudFormation::Template
```

```
AWSTemplateFormatVersion: "2010-09-09"
```



```
Description: A sample template
```

```
Resources:
```

```
  MyEC2Instance:
```

```
    Type: "AWS::EC2::Instance"
```

```
    Properties:
```

```
      ImageId: "ami-0ff8a91507f77f867"
```

```
      InstanceType: t2.micro
```

```
      KeyName: testkey
```

```
      BlockDeviceMappings:
```

```
        -
```

```
          DeviceName: /dev/sdm
```

```
          Ebs:
```

```
            VolumeType: io1
```

```
            Iops: 200
```

```
            DeleteOnTermination: false
```

```
            VolumeSize: 20
```

CloudFormation Concept

- **Stacks**

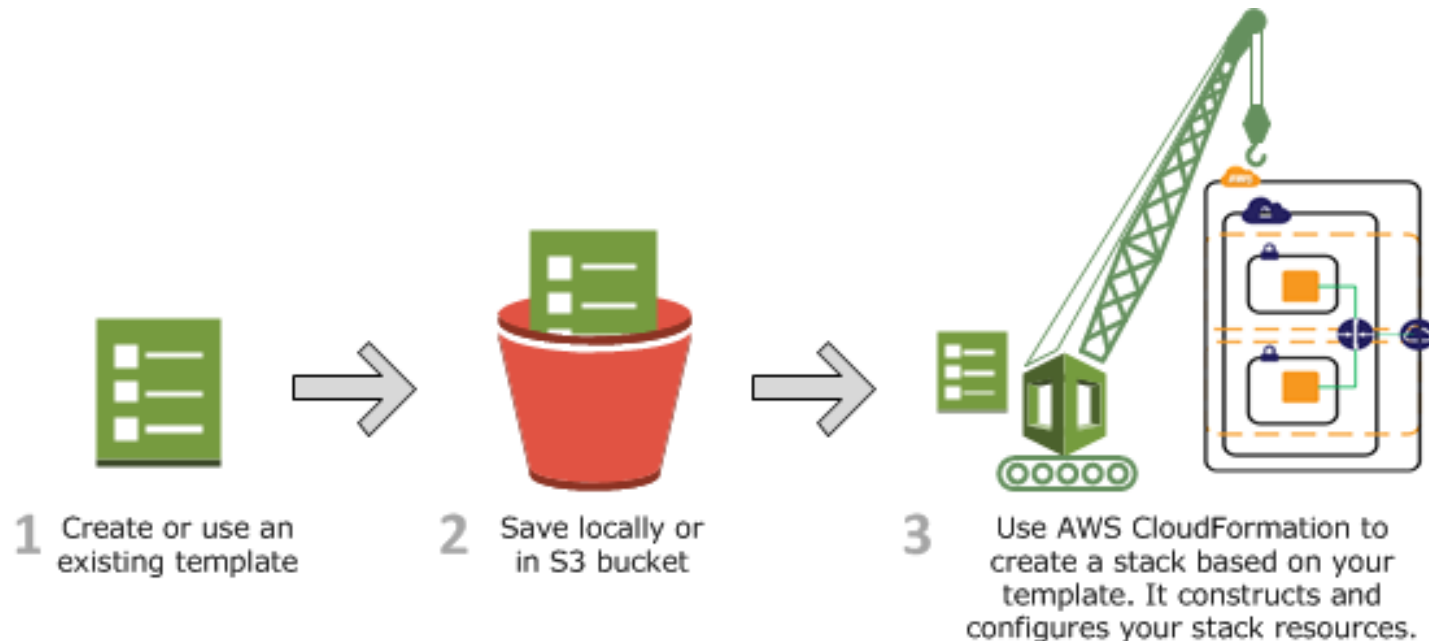
- When you use CloudFormation, you manage related resources as a single unit called a stack.
- You create, update, and delete a collection of resources by creating, updating, and deleting stacks.

- **Change sets**

- If you need to make changes to the running resources in a stack, you update the stack.
- Before making changes to your resources, you can generate a change set, which is a summary of your proposed changes.
- Change sets allow you to see how your changes might impact your running resources, especially for critical resources, before implementing them.

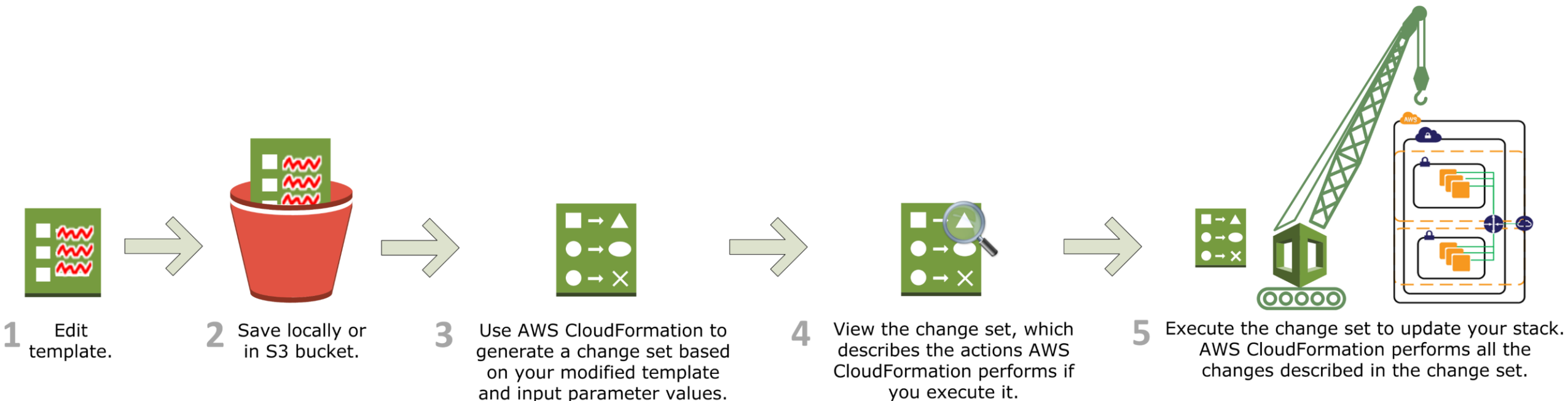
How AWS CloudFormation work

- When creating a stack, AWS CloudFormation calls to AWS to provision and configure your resources.
- CloudFormation can only perform actions that you have IAM permission to do.
- The calls that CloudFormation makes are all declared by your template.



Updating a stack with change sets

- You can modify the stack's template.
- Create a change set by submitting a modified version of the original template
- CloudFormation compares the modified template with the original template and generates a change set.
- The change set lists the proposed changes.



Deploying CloudFormation templates

- Manual way:
 - Editing templates in the CloudFormation Designer
 - Using the console to input parameters, etc
- Automated way:
 - Editing templates in a YAML file
 - Using the AWS CLI (Command Line Interface) to deploy the templates
 - Recommended way when you fully want to automate your flow
- DEMO

CloudFormation Template Component

- **Resources:** AWS resources declared in the template (MANDATORY)
- Parameters: Dynamic inputs for your template
- Mappings: Static variables for your template
- Outputs: References to what has been created
- Conditionals: List of conditions to perform resource creation
- Metadata
- Helper
 - References
 - Functions

AWS Resources

- Resources are the core of your CloudFormation template
- They represent the different AWS Components that will be created and configured
- Resources are declared and can reference each other
- CloudFormation service take cares of creation, updates and deletes of resources.
- Resource types identifiers are of the form:
- `AWS::aws-product-name::data-type-name`
 - Ex: `AWS::EC2::Instance`