Cloud Formation

- **AWS CloudFormation** is a service that helps you ***model and set up*** your Amazon Web Services resources so that you can spend less time managing those resources and more time focusing on your applications that run in **AWS**.
- AWS CloudFormation gives developers and system administrator an easy way to ***create and manage a collection of related AWS resources***, provisioning and updating them in an orderly and predictable fashion.

Templates:

- You can create **templates** for the service or application architectures you want and have AWS **CloudFormation** use those **templates** for quick and reliable provisioning of the services or applications (called “stacks”).

Stack

- A **stack** is a collection of AWS resources that you can manage as a single unit. In other words, you can create, update, or delete a collection of resources by creating, updating, or deleting **stacks**. All the resources in a **stack** are defined by the **stack's**AWS **CloudFormation** template.

- You can use AWS CloudFormation sample templates or create your own templates to describe the AWS resources, and any associated dependencies or runtime parameters, required to run your application
- After the AWS resources are deployed, you can modify and update them in a controlled and predictable way, in effect applying version control to your AWS infrastructure the same way you do with your software. (It's the biggest advantage as the whole infra converts into code)
- You can deploy and update a template and its associated collection of resources (called stack) by using AWS management Console, AWS CLI or AWS API.
- There is no additional cost for using this service but you have to pay for the AWS services which is running in this stack

Benefits of Cloud Formation

- Infrastructure as code
- Ease of replication
- Easily control and track changes to your infrastructure
- Supports a wide range of AWS resources

Templates

- CloudFormation Template is a JSON or YAML formatted text file
- We can save this file with any extension like .json, .yaml, .template or .txt
- AWS CloudFormation uses this templates as blueprint for building your AWS resources
- For Example: In a template, you can describe an Amazon EC2 instance, such as the instance type, the AMI ID, block device mappings, and its Amazon EC2 key pair name.
- Whenever we create a stack, we also specify a template that AWS CloudFormation uses to create whatever you described in the template.

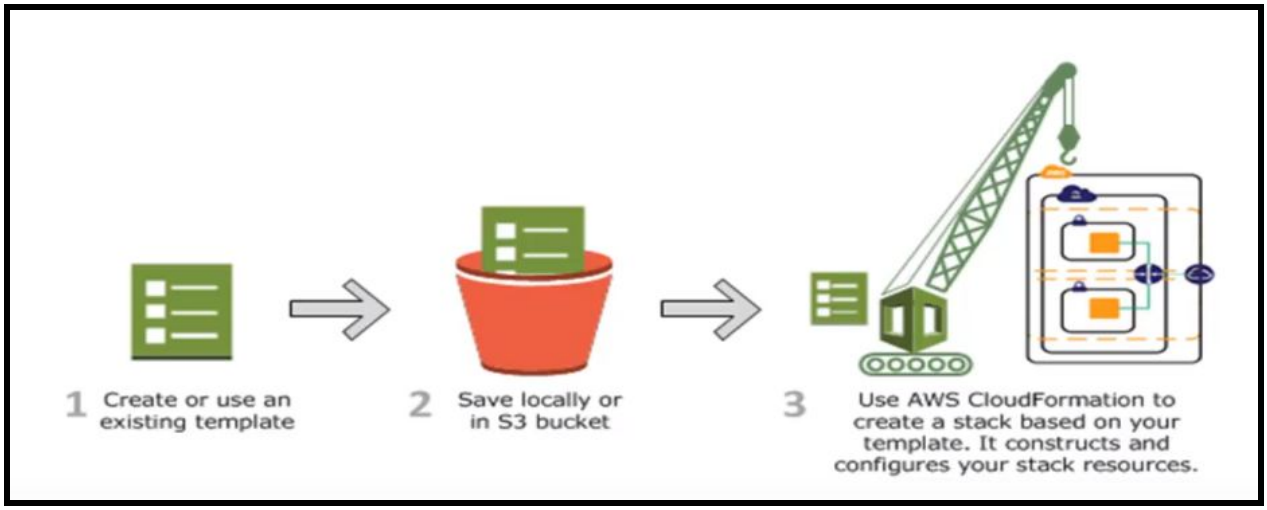
Stack

- When we use AWS CloudFormation, you manage related resources as a single unit called a stack
- We create, update and delete a collection of resources by creating, updating, and deleting stacks.
- All the resources in a stack are defined by the stack's AWS CloudFormation template.
- For Example: we created a template that includes an AutoScaling group, Elastic Load Balancer and RDS database instance. To create those resources, you create a stack by submitting the template that you created, and AWS CloudFormation provisions all those resources for you.
- You can work with stacks by using AWS CloudFormation Console, API, or CLI

Change Set

- Suppose 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 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.
- For Example, if you change the name of an Amazon RDS database instance, AWS CLoudFormation will create a new database and delete the old one. You will lose the data in the old database unless you've already backed it up. If you generate a change set, you will see that your change will cause your database to be replaced, and you will be able to plan accordingly before you update your stack.

Below is the diagrammatic representation



If you are making changes over your json file (CloudFormation template)