

## CLOUD FORMATION

If you need to deploy more than just a handful of services or applications on AWS, you will quickly discover that it is tedious and time-consuming to set each one up manually.

To configure each AWS resource by hand or manually which leaves you at higher risk of making errors or introducing inconsistency.

### AWS CloudFormation:

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 allows you to quickly and easily deploy your infrastructure resources and applications on AWS.
- AWS CloudFormation allows you to use programming languages or a simple text file to model and provision, in an automated and secure manner, all the resources needed for your applications across all regions and accounts. This gives you a single source of truth for your AWS and third party resources.
- AWS CloudFormation is an AWS service that uses template files to automate the setup of AWS resources.
- It can also be described as infrastructure automation or Infrastructure-as-Code (IaC) tool and a cloud automation solution because it can automate the setup and deployment of various Infrastructure-as-a-Service (IaaS) offerings on the AWS CloudFormation supports virtually every service that runs in AWS.
- You can use CloudFormation to automate the configuration of workloads that run on the most popular AWS services, like the EC2 compute service, the S3 storage service, and the IAM service for configuring access control.

- In general, if a service runs on AWS, it is a safe bet that you can use CloudFormation to automate its configuration and deployment.
- It is worth noting that CloudFormation is not the only way to configure and deploy services on AWS. You can handle these processes manually using the AWS command-line interface, API, or Web console. Manual provisioning is the approach that teams typically take when they are just getting started with AWS and learning how to deploy services. However, as they scale their environments up in size, many teams quickly realize that they need a solution like CloudFormation to make the deployment process faster and more consistent.

## **CloudFormation Stack**

A Stack is a collection of resources which are created and managed as a single unit by CloudFormation.

A Stack can contain any number of resources.

### **Scenario:**

Developers request to create an environment, infrastructure to work on project with DevOpsAdmin. We might think that, this can be done by normal admin, why we require the DevOpsAdmin. But, DevOps Admin only can understand the developers language like libraryfiles, environment files, RuntimeEnv. Containers etc.

-Usually when developers complete their work they send request to the Devops team to deploy.

-Everyweek developers require VPC,Subnets,RouteTable,Security Group, EC2 , S3 setup.

-Developers test the application with the help of the above setup and after testing the application setup has to remove. Again on Monday, AWS Solution Architect has to setup the similar one again.

- Developers require the above setup. Everyweek they need the similar kind of setup.

It's typical to setup the exact or identical setup every week to the developers who are working in the different regions. So, CloudFormation Template will be useful in this kind of scenarios.

- So with the above scenarios, if DevOps Admin create an CloudFormation Template which is useful for him to create the identical setup every week.

It's important to keep in mind that CloudFormation checks if each stack resource is properly created and configured. If a resource cannot be created, CloudFormation rolls back and destroys all created resources.

- With AWS CloudFormation, we can describe our IT infrastructure and deploy it any time we require it, creating different environments, for example testing, staging, production, etc in a few minutes.

**CloudFormation = Everything as a code in a file.**

it is like IAAC (Infrastructure as a code)

Cloud Formation is of two types:

- JSON

- YAML

- Install Microsoft Visual Studio Code or Sublime or any 3rd party tool to read the code.

download visual studio code from the following url

<https://code.visualstudio.com/download>

IDE(integrated Development Environment) Tools: EditPlus,Eclipse

Step1:

====

First Need to understand the code.

- Open the file "AWS-Infra"file using the sublime/visual studio code.
- Make some changes in the json script

```
"VPCNAME": {  
  
    "Default": "AUG25-AWS-VPC",  
  
    "Description": "VPC Name for Dev Team",  
  
    "Type": "String"  
  
}
```

- To add the labels

add under the "parameter labels"

```
"VPCNAME": {  
  
    "default": "Enter VPC Name Here:"  
  
}
```

Resources(\*\*\*)Important)

1. Make some changes in the VPC

```
"Type": "AWS::EC2::VPC",
```

to

```
"Type": "AWS::EC2::VPCC",
```

-save the file and again upload the code. we get the error here.because Resource doesn't recognize the

"VPCC". It will understand only VPC only.

=====

Do not scare by seeing code, we have n number of aws templates available.

search for 'aws cloudformation template github'

<https://github.com/aws-labs/aws-cloudformation-templates/tree/master/aws/services>

-Whenever you read the templates provided by the AWS, under the 'Resources' they were not mention any VPC name

that means they referred to the 'Default-VPC'

Note: Whenever VPC is not mentioned in the json/yaml script, that means they are intended to mention the 'Default -VPC'.

If you delete the default vpc, can we recreate it. yes we can recreate the default vpc.

-Actions ---> create default-vpc

-CloudFormation templates we used to keep it either in our laptop or code repository(github)

=====

Deploy the template

Services - CloudFormation

1. create stack - upload the AWS\_Basic\_Infra.json file

Stack Name = AWS-ONLINECLASS-STACK

Enter CIDR of new VPC =172.31.0.0/16

Enter CIDR of the public subnet =172.31.1.0/24

Enter VPC Name Here: =AWS-ONLINECLASS-CF-VPC

Click <Next>

2. Tab <Advanced>

Termination Protection =Enabled (this will protect your stack to delete anyone accidentally)

Rollback on failure =No (This option is selected because , to know the errors when it is rollback)

Click <Next>

3. if we can observe it will upload the S3 and from there it is accessing this file.

Click <create>

## CloudFormation Pricing

There is no additional charge for AWS CloudFormation.

You pay for AWS resources (such as Amazon EC2 instances, Elastic Load Balancers etc.)

Created using AWS CloudFormation in the same manner as if you created them manually.

You only pay for what you use, as you use it;

There are no minimum fees and no required upfront commitments.

Modern IT infrastructures can be quite complex and are usually built from many distinct components: computing instances, load balancers, storage volumes, CDNs, caching nodes, one or more database clusters, etc. You need to set up each component specifically for the environment that you are using (like production, staging, development, etc.), to keep them updated and to remove or swap components due to software changes. CloudFormation is a great tool that helps system administrators by managing these details automatically.

AWS CloudFormation gives you an easy way to create a set of cloud resources, such as Amazon EC2 instances, CloudFront distributions, RDS database instances, and other services

CloudFormation also helps me maintain control over my monthly AWS bill. If I have AWS resources that I'm not actively using or experimenting with, I can use CloudFormation to temporarily tear down a stack when I don't need it. When I need it again, CloudFormation can easily rebuild it.