

AWS CloudFormation Tool

Sanjay Vuppugandla

* Presenting Remotely, >300 miles from campus

Sai Prateek Atluri

Ankit Kadian*

Introduction

- > Ever find yourself drowning in the sea of details when creating an EC2 VM on AWS? What if I told you there's a simpler way, a shortcut to bypass the tedious manual process of filling out every detail in the AWS console?

AWS CloudFormation



History: Who

- > **AWS CloudFormation was developed by Amazon Web Services (AWS), and as such, the credit for inventing this technology goes to AWS. The service was officially launched on February 25, 2011.**
- > **Competing/Similar Alternatives:**
 - **Azure Resource Manager (ARM)**
 - **Google Cloud Deployment Manager**
 - **Terraform**

History: Why

AWS CloudFormation was developed to address the challenges associated with manually provisioning and managing infrastructure in the cloud. Prior to the introduction of Infrastructure as Code (IaC) tools like CloudFormation, the process of setting up and maintaining cloud resources was often manual, time-consuming, error-prone, and lacked scalability.

History: How

Since its launch in 2011, AWS CloudFormation has evolved significantly, incorporating new features and enhancements to address the changing needs and demands of users.

Features Summary

- > CloudFormation allows you to model your infrastructure in a text file. You can use JSON or YAML to describe what AWS resources you want to create, configure, and update.
- > Easily control and track changes to your infrastructure
- > It supports 220+ services.

Example Use Cases

- > **Multi-region deployments**
- > **Serverless applications**
- > **Application Stacks**
- > **Monitoring and logging applications**
- > **IAM roles and policies**

Cloudformation Advantages

- > **Multi-region deployments**
- > **Easy to provision resources using the files rather than console or CLI.**
- > **Zero cost for the cloudformation tool - it only charges for the resources created.**
- > **Reusable templates -makes it easy to replicate**
- > **rollback facility.**

Cloudformation Disadvantages

- > It is hard to write YAML or JSON files (error prone) and might get verbose for heavy apps.
 - To further simplify, can use the CDK (Cloud Development Kit) an open-source programmatic way to provision resources.
 - AWS CloudFormation Designer interface is a visual tool that provides a graphical interface for creating, viewing, and modifying CloudFormation templates.

CloudFormation Designer interface overview

Cost Discussion With Example

- > It is free of cost and you will be charged for only the provisioned resources. It simplifies the developer experience and ease of provisioning.
- > Say you create an EC2 instance by using a CloudFormation template, you will be charged for only the instance running time and associated charges.

Usability

- > We need to know the Domain-Specific Language.
- > The initial learning curve is steep for beginners as they need to know all the properties and parameters.
- > Easy to update the resources and also deletion.

Conclusions

- > To reiterate, CloudFormation eases the process of provisioning resources through using templates in YAML or JSON format.
- > It enables easy scaling of resources based on demand and supports multi-region deployment.
- > It is particularly beneficial for managing application stacks, offering a structured approach to defining and deploying infrastructure configurations.

Anatomy of a template

- > **Yaml or json formats are supported.**
- > **There are several sections you can include in the template.**
- > **Resources section is the required and rest are optional**
- > **Resources section defines what resources need to be provisioned or updated.**

AWS::TemplateFormatVersion: 'version date' (optional) # version of the CloudFormation template. Only accepted value is '2010-09-09'

Description: 'string' (optional) # a text description of the CloudFormation template

Metadata: 'template metadata' (optional) # objects that provide additional information about the template

Parameters: 'set of parameters' (optional) # a set of inputs used to customize the template

Rules: 'set of rules' (optional) # a set of rules to validate the parameters provided at deployment/update

Mappings: 'set of mappings' (optional) # a mapping of keys and associated values

Conditions: 'set of conditions' (optional) # conditions that control whether certain resources are created

Transform: 'set of transforms' (optional) # for serverless applications

Resources: 'set of resources' (required) # a components of your infrastructure

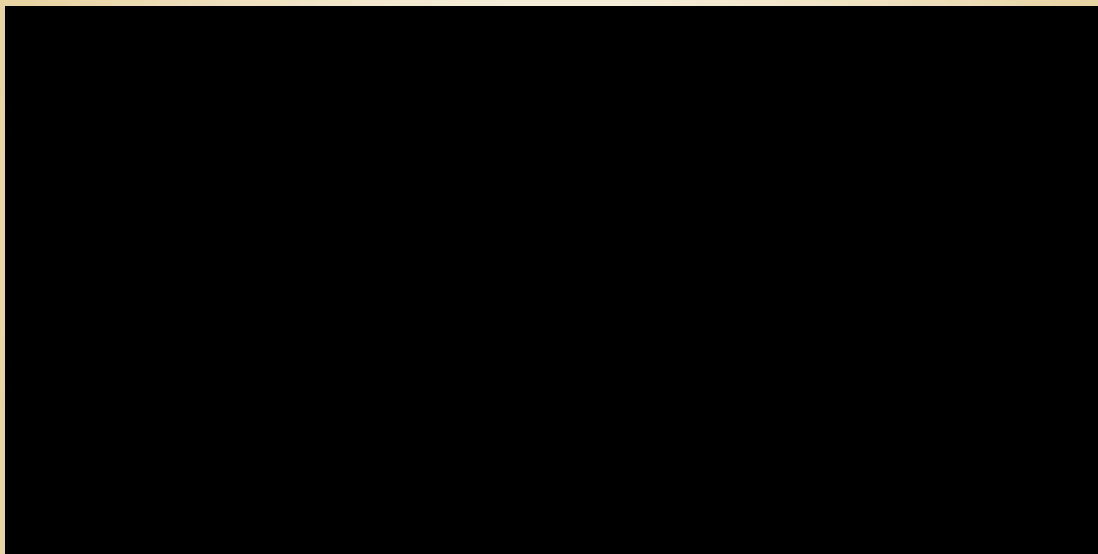
Hooks: 'set of hooks' (optional) # Used for ECS Blue/Green Deployments

Outputs: 'set of outputs' (optional) # values that are returned whenever you view your stack's properties

Cloud9 IDE

- > A cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser.
- > We are using the Cloud9 IDE to show the demo. Just go to the console and create an environment (In background it runs on a EC2 instance)

[Tour the AWS Cloud9 IDE](#)



Questions?