

AWS CloudFormation

Topics : [AWS](#)

Written on [December 08, 2023](#)

AWS CloudFormation is a service provided by Amazon Web Services (AWS) that allows you to define and provision AWS infrastructure as code. Instead of manually creating and configuring resources, CloudFormation allows you to use a template to describe the architecture and resources needed for your application. Here are key points about AWS CloudFormation:

1. Infrastructure as Code (IaC):

- **Description:** CloudFormation enables the practice of Infrastructure as Code, allowing you to define and manage AWS infrastructure in a declarative manner using JSON or YAML templates.
- **Benefits:** IaC enhances reproducibility, consistency, and version control of your infrastructure.

2. Templates:

- **JSON/YAML Format:** CloudFormation templates are written in either JSON or YAML format, providing a human-readable and version-controllable representation of your infrastructure.
- **Declarative:** Templates declare the desired state of your infrastructure, and CloudFormation handles the provisioning and updating.

3. Resources:

- **Definition:** Resources represent the AWS components that make up your infrastructure (e.g., EC2 instances, S3 buckets, RDS databases).
- **Configuration:** Each resource in a CloudFormation template has its configuration defined, including properties, dependencies, and other settings.

4. Stacks:

- **Definition:** A stack is a set of resources created from a CloudFormation template. It represents a single deployable unit.
- **Lifecycle:** Stacks can be created, updated, and deleted. Updating a stack allows you to modify existing resources or add new ones.

5. Change Sets:

- **Description:** Before making changes to a stack, CloudFormation allows you to preview the changes using a Change Set.

- **Review and Execute:** Change Sets provide a detailed preview of the modifications, helping you understand the impact before applying changes.

6. Nested Stacks:

- **Description:** CloudFormation supports the creation of nested stacks, allowing you to modularize and reuse templates.
- **Modularity:** Nested stacks improve template organization, promote reusability, and simplify management.

7. Outputs:

- **Description:** Outputs in CloudFormation allow you to expose certain values (e.g., IP addresses, resource names) for use by other stacks or applications.
- **Inter-Stack Communication:** Outputs facilitate communication between different parts of your infrastructure.

8. Parameters:

- **Description:** CloudFormation parameters allow you to customize template values at runtime.
- **Flexibility:** Parameters make templates more flexible and adaptable to different environments without modifying the template itself.

9. Mappings and Conditions:

- **Mappings:** Mappings enable the definition of key-value pairs for use in templates (e.g., AMI mappings based on region).
- **Conditions:** Conditions allow you to control the creation of resources based on logical expressions.

10. Rollback on Failure:

- **Feature:** CloudFormation provides the option to automatically roll back a stack to its previous state in case of a failure during stack creation or update.
- **Safety Mechanism:** Rollback on failure helps maintain a consistent state and avoid partially deployed configurations.

11. Drift Detection:

- **Description:** Drift detection allows you to identify and compare differences between the desired stack configuration and the actual deployed resources.
- **Monitoring and Compliance:** Drift detection helps ensure that the infrastructure remains compliant with the desired state.

12. Integration with AWS Organizations:

- **Description:** CloudFormation integrates with AWS Organizations, allowing you to centrally manage and apply templates across multiple accounts.
- **Consistency and Governance:** Centralized management improves consistency and governance in multi-account AWS environments.

ARYATECHNO