AWS Cloud Formation

By: Tanner Brown, Khanh Nguyen, Ali Nemati

What is AWS CloudFormation?

Pic ref:https://n2ws.com/wp-content/uploads/2015/02/cloudformation.jpg

AWS CloudFormation

Why?

Create and manage infrastructure

an application stack in controlled and predictable way.

Initial release: February 25, 2011

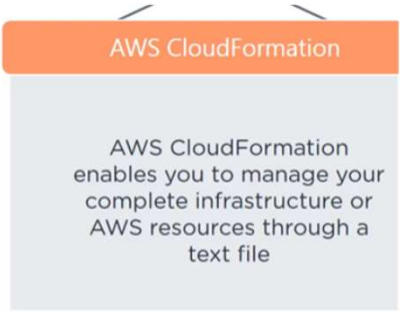
Ref:
https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/ReleaseHistory.html

AWS CloudFormation benefits:

Easy Declarative & Flexible
Transparent and Open
No Extra Charge
Integration Ready
Customized via Parameters

Ref:
https://aws.amazon.com/cloudformation/

Example :



Note: Using stack, AWS resources can be created or updated

Did you know?

All the resource required by a user in an application can be deployed easily using templates

Also, you can reuse your templates to replicate your infrastructure in multiple environments

To make templates reusable, use the parameters, mappings and conditions sections in the template so that you can customize your stacks when you create them

Use Cases

Use Case: Expedia

- Leading online travel booking company worldwide
- Began using AWS in 2010
- Using AWS services such as EMR, EC2, IAM, S3, VPC etc. in multi region & AZ
- Use CloudFormation (w/ Chef) to deploy entire front end and backend to its VPC on AWS

Use case: Coinbase

- Bitcoin wallet and exchange service company in San Francisco
- Supports 3 million global users, facilitates transactions in 190 countries
- Using AWS services such as RDS, EC2, IAM, Kinesis, S3, Lambda, VPC etc.
- Uses CloudFormation templates to design, built, and maintain its networks
- CF helps version-controlling its network, network duplication for on-demain development and staging environments.

Advantages

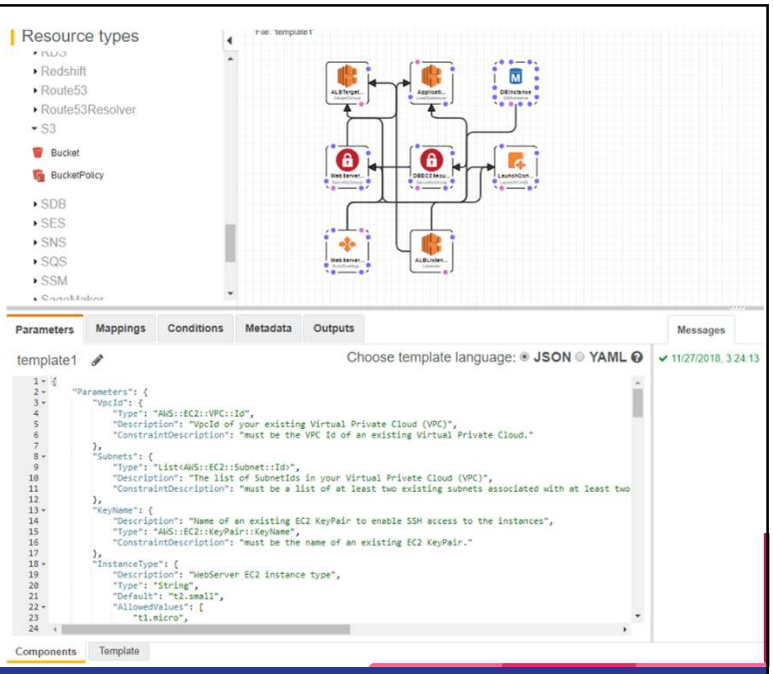
- **Stable** - Create and manage infrastructure/application stack in controlled and predictable environment.
- **Scalable** - Can be used for everything from hosting a single EC2 instance, to hosting an entire multi-tier, multi-region, multi-service application.
- **Flexible** - Can configure/map parameters that can be resolved at a later time (don't have to hardcode everything into your blueprint).
- Minimizes human error
- Uses infrastructure as Code
- Supports virtually all AWS resources

Disadvantages

- **Tightly Coupled** - Essentially unusable outside of the AWS platform.
  - Similar cross-platform services exist *(e.g. Terraform by HashiCorp)*
- **Difficult to use with existing resources** - CFT doesn't offer easy solutions to access, manage or integrate existing resources.

Usability

- Drag and drop functionality
- Save templates as JSON or YAML
- Store template locally or in S3 bucket
- Load or Build upon existing Templates



Cost

- No cost, only pay for resources.
- Several non-free alternatives exist
- Exists as a tool to help implement AWS services

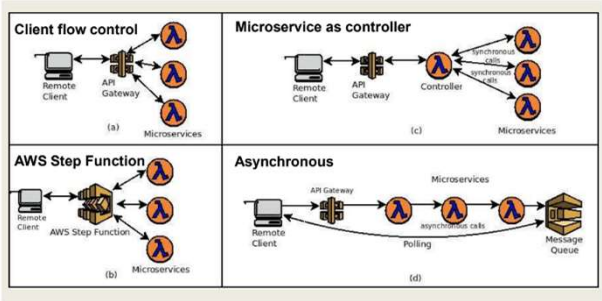
Conclusions

- Free service
- Infrastructure as a code,
- Flexible
- Supports entire AWS platform
- Allows for simple automation
- Templates can be saved and reused.

Demonstration: Deploy ETL Project

- Create a AWS CF Stack using CF template, written in yaml

- Deploy one of the models in the Application Flow Control case study.



References

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-whatis-howdoesitwork.html

https://aws.amazon.com/cloudformation/

https://node.university/courses/aws-intermediate/lectures/3151096

https://searchaws.techtarget.com/definition/AWS-CloudFormation-Amazon-Web-Services-CloudFormation

https://www.youtube.com/watch?v=kyGR5WAqx6E

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/ReleaseHistory.html

https://n2ws.com/wp-content/uploads/2015/02/cloudformation.jpg

https://aws.amazon.com/cloudformation/features/

Questions?