**Knowledge Transfer( Creating Cloud Formation Stack )**

**Dated : 08/08/2023**

**\*\*Challenge :\*\***

Hey geeks, after working for so long with AWS and doing some task repetitive like creating VPC,EC2 cluster or hosting a word-press website and wasting time in that I am tired of that, So today’s take is to ease this work and save our precious time…

==============================================================================

**\*\*Task :\*\***

Ticket: Get a solution for saving time and energy of doing repetitive work

==============================================================================

**\*\*Solution :\*\***

Finally I have guys the solution for our problem, while searching I have found out a service named CLOUD FORMATION(CF), this service do the task in less than 1 min by building the infrastructure of content using YAML scripts…

==============================================================================

**\*\*Pre-requisites:\*\***

• Have a good knowledge about YAML Script

• Know the all services of AWS

==============================================================================

**\*\*Objective:\*\***

The objective is to ease the work by creating infrastructure of service which you want using CF and Yaml scripts and save time for more Important things and complications

.==============================================================================

**\*\*Description:\*\***

CloudFormation is an essential service provided by Amazon Web Services (AWS) that allows users to define and provision cloud infrastructure as code. It enables the automated creation and management of AWS resources using templates, which are defined in JSON or YAML format. YAML (YAML Ain't Markup Language) is a human-readable data serialization format that is commonly used for creating CloudFormation templates due to its simplicity and readability.

With CloudFormation and YAML, users can define their entire infrastructure stack, including computing resources, networking configurations, security settings, and more, in a declarative manner.

This approach offers several benefits:

1. **Infrastructure as Code (IaC):** CloudFormation templates represent infrastructure as code, making it easier to version, collaborate on, and automate deployments. This reduces the risk of manual errors and streamlines the development and operational processes.
2. **Consistency and Reproducibility:** By describing infrastructure in templates, you ensure consistency across different environments (e.g., development, testing, production) and can reliably reproduce these environments whenever needed.
3. **Automation and Efficiency:** CloudFormation automates the deployment process, eliminating the need for manual intervention during resource provisioning. This increases operational efficiency and reduces the time required to set up complex infrastructure.
4. **Scalability:** CloudFormation allows users to define scalable architectures using features like auto-scaling groups and load balancers, ensuring that applications can handle varying workloads.
5. **Easy Updates and Rollbacks:** Modifications to infrastructure can be made by updating the CloudFormation template. If an update fails, CloudFormation can perform a rollback, reverting to the previous state.
6. **Dependency Management:** CloudFormation handles resource dependencies intelligently, ensuring resources are provisioned in the correct order.

==============================================================================

**\*\*Steps:\*\***

# Creating a CloudFormation stack involves several steps to define the resources you want to provision and then execute the template to create those resources.

# Here's a step-by-step guide on how to create a Cloud-Formation stack

**Step 1 ( Creating Template )**

# First you have to create a YAML or JSON template that defines the AWS resources you want to provision.

# Include necessary sections in yaml script like **Resources, Parameters, Mappings, Outputs, and others**, as required by your infrastructure design.

# There are pre-built scripts on Google you can browse it and copy from there also…

# Down I have provided one sample YAML script for creating EC2 instance with NGINX in it…

**------------------------------------------------------------------------------------------------------------------------------------------**

AWSTemplateFormatVersion: '2010-09-09'

Description: AWS CloudFormation Template to create EC2 instance with NGINX.

Resources:

MyInstance:

Type: AWS::EC2::Instance

Properties:

InstanceType: t2.micro

ImageId: ami-0041b98fa770e38cd

# Replace with your desired Amazon Linux AMI

KeyName: optimus # Replace with your desired KEY-PAIR

SecurityGroupIds:

- !Ref MySecurityGroup

UserData:

Fn::Base64: !Sub |

#!/bin/bash

yum update -y

sudo amazon-linux-extras install nginx1

sudo systemctl start nginx1

sudo systemctl enable nginx1

MySecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Security Group for EC2 Instance

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 0.0.0.0/0

- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp: 0.0.0.0/0

**# To use this script first save it as Yaml file and that will be done by .yml extension…**

**# In this script replace the ImageID = Instance AMI ID and Keypair to your current one…**

# AMI ID will be in EC2 launch template AMI were we select Linux AMI…

**Note :- As CF is a region wise service so if you want the stack to build in another region then mention the region and if not mentioned then the stack will create in the region were you are operating CF**

**------------------------------------------------------------------------------------------------------------------------------------------**

**Step 2 ( Login AWS )**

# Login to your AWS Account and search Cloud-Formation Service and come to it’s Dashboard…

**Step 3 ( Creating Stack )**

# At left have side of dashboard you will see stacks option click on it and follow the steps…

* First click create stack
* Now in Prerequisite section select Template is ready
* Next in specify template select upload template file
* After uploading click next
* Now give the stack name and click next
* Again click next
* Lastly click submit

# We are not going to give any value except uploading the file…

# Now wait for the stack status to change from "CREATE\_IN\_PROGRESS" to "CREATE\_COMPLETE" if successful.

**# After completing the work DELETE the stack and all the things will be deleted with the single click…**

# After creation is complete go to EC2 and see whether the instance is created or not…

**# By this way you can upload other task scripts also and reuse it whenever you want and save your time…**

**#### THE END ####**

==============================================================================

**\*\*Explanation:\*\***

Creating a CloudFormation stack involves a step-by-step process to efficiently deploy AWS resources using infrastructure as code. To begin, you need to prepare a CloudFormation template, a YAML or JSON file that outlines the desired AWS resources and their configurations. This template encapsulates your infrastructure design, making it manageable and repeatable.

Access the AWS Management Console using your credentials and navigate to "CloudFormation" under "Management & Governance." Click on the "Create stack" button to initiate the stack creation process. You'll then choose how to specify the template – either by uploading a file or providing a URL.

Next, provide essential stack details. Give your stack a unique name that identifies its purpose within your AWS environment. If your template has parameters, input values for them here. Parameters enable customization of the stack without modifying the template directly.

Further configure your stack by adding tags for organizational purposes, setting permissions using IAM roles, and adjusting advanced settings if required. Once all settings are defined, review them to ensure accuracy.

With settings reviewed, click "Create stack" to start provisioning resources. The "Stacks" dashboard will display the stack's progress, initially showing "CREATE\_IN\_PROGRESS." You can monitor the advancement here.

If your template includes output values, access the "Outputs" tab to retrieve useful information about the created resources, such as URLs or IDs.

As your infrastructure needs evolve, you can update your stack by modifying the template and using the "Update stack" option. Alternatively, if resources are no longer needed, you can choose "Delete stack" to remove all associated components. Confirm deletion to prevent accidental data loss.