AWS SysOps Administrator – Associate Level

TECHNOLOGY

Automation

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```

Learning Objectives

By the end of this lesson, you will be able to:

- Create CloudFormation stacks
- Deploy an application using Elastic Beanstalk
- Create templates for resources to be used in an application

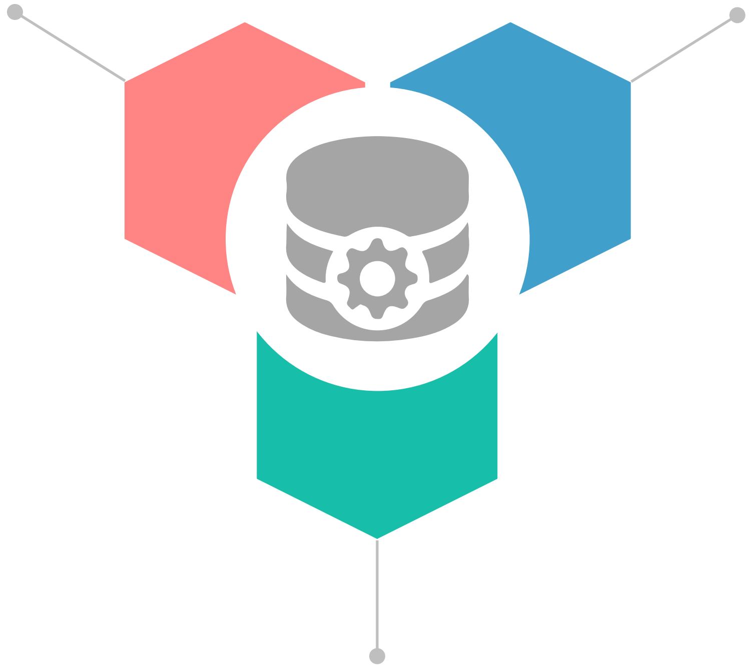
AWS CloudFormation

What Is CloudFormation?

AWS CloudFormation is a service that helps an end user automate AWS services by creating templates.

- A user can create a template describing a resource, and AWS CloudFormation handles provisioning and configuration for it.
- AWS CloudFormation eliminates the process of individual configuration of AWS resources and their dependencies.
- CloudFormation interprets a template, and makes API calls to create resources defined by the user.
- Apart from the UI, CloudFormation supports YAML and JSON formats for defining resources.

Benefits of CloudFormation



Managing the infrastructure

A single template can contain information of services, and other configurations are done by AWS.

Replicating the infrastructure

A user can replicate the same infrastructure for multiple regions for dealing with service failures using rollback operations.

Tracking changes in the infrastructure

AWS describes what changes happened in resources that got provisioned, thus helping in easy logging of activities.

CloudFormation Template

- CloudFormation template is either a JSON or a YAML text file.
- AWS CloudFormation uses these templates as reference for building resources.
- Example: A template contains information about an EC2 instance, such as the instance type, AMI ID, block device mappings, and key-pair name.
- A user can also upload a template to CloudFormation using S3.
- The resource created using the template is called a **stack.**

Template: Components

1. **Data tables:** Static configuration values, such as AMI names

1. **Outputs:** URL to a web application

1. **Parameters:** Input values provided while creating stacks

1. **Resources:** Names and configurations of services to be added

1. **Conditions:** Actions to be taken based on the conditional statements, such as equals

1. **Mappings:** Set values based on a region, such as AMIs

1. **Transform:** Includes snippets from outside the main template

Template: Example

The sample template given below shows how to create an EC2 instance:

```
AWSTemplateFormatVersion: "2010-09-09"
Description: A sample template
Resources:
  MyEC2Instance:
    Type: "AWS::EC2::Instance"
    Properties:
      ImageId: "ami-0ff8a91507f77f867"
      InstanceType: t2.micro
      KeyName: testkey
      BlockDeviceMappings:
        -
          DeviceName: /dev/sdm
          Ebs:
            VolumeType: io1
            Iops: 200
            DeleteOnTermination: false
            VolumeSize: 20
```

CloudFormation

Duration: 10 Min.

Problem Statement:

Create a stack with S3 bucket using AWS CloudFormation.

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to create a stack with S3 bucket using AWS CloudFormation:

1. Login to your AWS lab

1. Select **CloudFormation** from **Services**

1. Design a template with S3

1. Render the template

Elastic Beanstalk

What Is Elastic Beanstalk?

AWS Elastic Beanstalk is a service used to deploy and scale web applications developed in various languages, such as Java, .NET, PHP, Python, and Ruby on servers, such as Apache.

- The infrastructure supporting the application is managed by AWS.
- There is no additional installation required, such as Java or .NET.
- On uploading the code, Elastic Beanstalk handles the deployment, load balancing, and auto scaling it.
- The user has full control over the application and the resources used to run it.

Benefits of Elastic Beanstalk

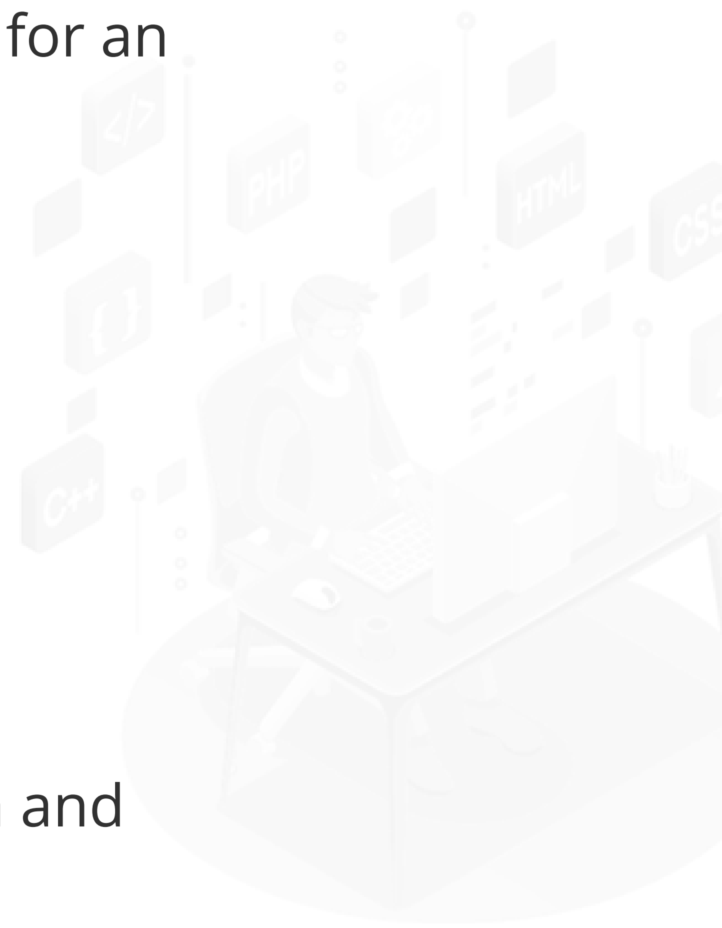
- It is the fastest and simplest way to deploy an application on AWS.
- Auto scaling is done automatically for an application.
- Users can choose the most suitable EC2 instance for an application.
- Users have the flexibility to handle all resources, or they can also let Elastic Beanstalk take care of them.

Benefits of Elastic Beanstalk

Automatic platform updates are performed on the OS for the tools used for an application, such as Java.

It provides a dashboard to manage the application health.

It is integrated with CloudWatch and X-Ray for logging performance data and metrics.



Elastic Beanstalk

Duration: 10 Min.

Problem Statement:

Create a web application using Elastic Beanstalk.

Assisted Practice: Guidelines

Steps to create a web application using Elastic Beanstalk:

1. Login to your AWS lab

1. Select **Elastic Beanstalk** from **Services**

1. Upload an HTML page for rendering

1. Select the platform and language

TECHNOLOGY

APS OpsWorks

AWS OpsWorks

AWS OpsWorks is a service that allows users to automate server configuration on Chef or Puppet.

- It allows you to use code to automate configurations of your servers.
- It uses managed instances of Chef or Puppet.
- It allows you to manage configurations of operating systems and applications.
- OpsWorks maintains a server by automatically patching, updating, and backing it up.

Key Takeaways

- AWS CloudFormation is a service that helps an end user automate AWS services by creating templates.
- CloudFormation template is a JSON or YAML text file.
- Resources is the most important section of a CloudFormation template.
- On uploading the code, Elastic Beanstalk handles the deployment, load balancing, and auto scaling of an application.
- OpsWorks maintains a server by automatically patching, updating, and backing it up.

Deploy Elastic Beanstalk Using CLI

Project Objective:

Create and configure an Amazon Elastic Beanstalk, add a user with the attached policy, and deploy the app.

Background of the problem statement:

Your manager has assigned you a task where you have to deploy Elastic Beanstalk (EB) using CLI to add a policy of EB with full access to the user.

LESSON-END PROJECT