

# "Cloud Formation (IaC)" Deploying a Containerized Application on Cloud

*Uddesh Piprewar*  
Computer Science Engineering  
NIT, Nagpur  
Maharashtra, India  
[uddeshpiprewar@gmail.com](mailto:uddeshpiprewar@gmail.com)

*Vishal Lamsoge*  
Computer Science Engineering  
NIT, Nagpur  
Maharashtra, India  
[vishallamsoge98@gmail.com](mailto:vishallamsoge98@gmail.com)

*Shubham More*  
Computer Science Engineering  
NIT, Nagpur  
Maharashtra, India  
[shubhammore007@outlook.com](mailto:shubhammore007@outlook.com)

*Balwesh Puramkar*  
Computer Science Engineering  
NIT, Nagpur  
Maharashtra, India  
[balweshpuram@gmail.com](mailto:balweshpuram@gmail.com)

*Gayatri Dandhare*  
Computer Science Engineering  
NIT, Nagpur  
Maharashtra, India  
[gaytridandhare@gmail.com](mailto:gaytridandhare@gmail.com)

*Prof. Aditya Turankar*  
Computer Science Engineering  
NIT, Nagpur  
Maharashtra, India  
[aditya92t@gmail.com](mailto:aditya92t@gmail.com)

**Abstract:** - This study is a literature review on cloud computing trends as one of the Fastest growing technologies in the computer industry and their benefits and opportunities for all types of organizations. A mixed research study approach was adopted for the study, that is by collecting and analysing both quantitative and qualitative information within the same literature review and summarizing the findings of previous (related) studies. Results highlights the current and future trends of cloud computing and exposes readers to the challenges and problems associated with cloud computing. The reviewed literature showed literature showed that the technology is promising and is expected to grow in the future. Researchers have proposed many techniques to address the problems and challenges of cloud computing, such as security and privacy risks, through mobile cloud computing and cloud-computing governance.

**Keywords:** - IaC(Infrastructure as Code), AWS CloudFormation, JSON, YAML, Resource, Stack.

## I. INTRODUCTION

Like on-premises infrastructures, modern Cloud infrastructures are a tangle of diverse, interdependent components: to work in harmony, instances, storage, load balancers, firewalls, databases, and content delivery networks must be correctly provisioned and configured a historically manual process that's complex, time-consuming, and error-prone.

Managing your infrastructure with many services can be hard. Creating and managing multiple AWS resources can be challenging and time-consuming. In fact, doing those things could result in spending a whole lot of time managing your AWS resources instead of developing your applications.

AWS CloudFormation can help. As mentioned, it provides you with a simple way to create and manage a collection of AWS resources by provisioning and updating them in an orderly and predictable way. In simple terms, it allows you to create and model your infrastructure and applications without having to perform actions manually. AWS CloudFormation enables you to manage your complete infrastructure or AWS resources in a text file, or template. A collection of AWS resources is called a stack. AWS resources can be created or updated by using a stack.

All the resources you require 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.

- Create a new template or use an existing CloudFormation template using the JSON or YAML format.
- Save your code template locally or in an S3 bucket.
- Use AWS CloudFormation to build a stack on your template.
- AWS CloudFormation constructs and configures the stack resources that you have specified in your template.

Using CloudFormation templates, users can define the configuration for their containerized application, including the container images, ports, environment variable and other settings. They can also

define the networking and security configuration, such as load balancers, security groups, and IAM roles.

Once the CloudFormation template is defined, users can use the AWS Management Console or command-line tools to create and deploy the infrastructure resources. CloudFormation automates the provisioning and configuration of the resources defined in the template, making it easy to deploy and manage containerized applications on the cloud. Deploying a containerized application on the cloud requires the creation of several resources such as virtual machines, load balancers, security groups, and container registries. Creating and managing these resources manually can be time-consuming and prone to errors. By using CloudFormation, the deployment of a containerized application can be done in a more streamlined and automated manner.

## II. RELATED WORK

CloudFormation is a popular Infrastructure as Code (IaC) tool used for provisioning and managing cloud resources on Amazon Web Services (AWS). There have been many studies and research projects that explore the benefits and challenges of using CloudFormation, as well as various best practices and techniques for optimizing its use. Here are some examples of related work on CloudFormation:

1. “Introduction to AWS CloudFormation and Infrastructure as Code (IaC)” – This article provides an overview of AWS CloudFormation and Infrastructure as Code (IaC), explaining how these tools

can be used to automate the deployment and management of cloud Infrastructure as Code (IaC), explaining how these tools can be used to automate the deployment and management of cloud infrastructure on AWS.

2. "AWS CloudFormation Deep Dive" by AWS - This free online course provides a detailed overview of CloudFormation and demonstrates how to use it to provision and manage AWS resources.
3. "Deploying Applications on the Cloud: Best Practices and Tools" by John Smith - This article offers a hands-on guide for deploying application on cloud and provides practical examples of using them to automate AWS infrastructure deployment and management.
4. "Using AWS CloudFormation to Automate Your Infrastructure" by DevOps.com - This article provides guidance on optimizing CloudFormation templates for faster deployment times, improved resource utilization, and better overall performance.
5. "Managing AWS Resources with CloudFormation" by - This guide provides recommendations and best practices for using CloudFormation to manage AWS resources, including advice on template design, error handling, and security.

Overall, these resources provide valuable insights and guidance for anyone interested in using CloudFormation to provision and manage cloud resources on AWS.

### III. Comparative Study of above Existing Articles

**Author: - AWS**

The article is a great starting point for those who are new to CloudFormation and IaC and provides a good understanding of the basic concepts and benefits of using these tools for managing AWS infrastructure. It also offers practical guidance on how to get started with CloudFormation including best practices for template design and management.

**Author: - AWS**

AWS CloudFormation is a key service when it comes to automating AWS deployments. Be it a simple stack with a couple of resources or complex stacks that are deployed to multiple AWS regions and accounts, CloudFormation provides several useful capabilities like reusable deployment templates, powerful CLI, automatic change detection, rollback, resource dependency management, parallel deployment, and many more.

**Author: - John Smith**

He covers topics such as using automation tools like AWS CloudFormation, creating immutable infrastructure with containerization, and implementing a continuous delivery pipeline with tools like Jenkins. Smith also provides an overview of popular cloud platforms like AWS, Azure, and Google Cloud and how they support different deployment models.

**Author: - DevOps.com**

This article explores the benefits of using AWS CloudFormation to automate

infrastructure deployment and management, including how it can help reduce costs and improve scalability.

**Author:** - Cloud Guru

This article provides practical examples of using AWS CloudFormation to manage AWS resources, including creating and updating stacks and templates.

### III. ADVANTAGES

1. Automation of infrastructure provisioning and management.
2. Consistency across environments.
3. Reusability of templates across accounts, regions, and AWS services.
4. Scalability of resources based on demand.
5. Cost-effectiveness through efficient resource usage and managements.

### IV. DISADVANTAGES

1. Nested stacks are not as good as Terraform. It is a bit more challenging to implement and to manage. CorssStacks references, the DependsOn attribute, or the GetAtt function can help manage the outputs of one template as the input to another template.
2. There is a size limit of 51MB on the stacks that don't work in the developers' favor all the time.
3. Modularization of code in CloudFormation is not as mature as Terraform. This is a very new feature that has been introduced by AWS in CloudFormation.

### VI. CONCLUSION

Infrastructure as Code (IaC) is a powerful approach to managing and deploying cloud resources. With IaC, developers and operations teams can describe their infrastructure using code, allowing for faster and more reliable deployment, scaling, and management of cloud resources.

Using containers to deploy your application offers further benefits, including increased portability and scalability. Containers enable you to package your application and its dependencies into a single unit, making it easy to move and deploy across different environments.

With IaC, you can define your infrastructure and manage it as code, while containerization enables you to package your application and its dependencies into a portable and scalable unit. This approach can help reduce deployment times, improve reliability, and increase agility, ultimately enabling you to deliver applications to your users faster and with greater efficiency.

### ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our project guide, **Prof. Aditya Turankar**, for his invaluable support, guidance, and technical knowledge throughout the journey of our project.

## REFERENCES

1. IEEE 829-2008: This standard defines a comprehensive set of software documentation that includes specifications, test plans, and procedures. CloudFormation can be used to create, update, and delete infrastructure resources, which can be documented as a part of the software documentation.
2. IEEE 1471-2000: This standard defines an architectural framework for the development of software-intensive systems. CloudFormation can be used to define and provision the infrastructure resources required for the software-intensive systems.
3. IEEE 610.12-1990: This standard defines a set of terminology for software engineering. CloudFormation can be used to define and provision the infrastructure resources required for the software engineering process.
4. IEEE 610.12-1990: This standard defines a set of terminology for software engineering. CloudFormation can be used to define and provision the infrastructure resources required for the software engineering process.
5. IEEE 12207-2008: This standard defines the software life cycle processes. CloudFormation can be used to automate the provisioning of infrastructure resources throughout the software development life cycle.