



**SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

## **“Infrastructure as a code-Cloud Formation-AWS”**

Under the guidance of  
**Dr. J Cruz Antony**

By

**EARLADINNE SAI DHEERAJ**  
(41110359)

# ABSTRACT

- Infrastructure as Code is a key DevOps concept that is essential in the Data Science world when we're building and defining production level workloads. Infrastructure as a code allows developers to manage a project's infrastructure as software. This enables developers to easily maintain and configure changes within a project's resources and architecture. While similar to traditional scripting, Infrastructure as a code allows for developers to use declarative language to provision resources. Without IaC, costs and time for manual deployment of different infrastructures can skyrocket, by maintaining your infrastructure as software you're able to easily and quickly test various deployments from a central source. Increasing the number of production and delivery cycles using Infrastructure as Code activities has changed how software engineers design, test, and release apps
- KEYWORDS: Cloud Computing, Stack, Ec2 Instance,S3

# INTRODUCTION

- Infrastructure as Code is a practice and a set of tools that use software development practices to manage infrastructure.
- Companies like Amazon, Facebook, and Netflix have led the adoption and influenced the development of these techniques, driven by their global scale, and the need to make changes more frequently to their software and environments than would be possible without a high degree of automation.
- At the same time, these companies need the highest levels of quality, reliability, and security for their systems, so their growing businesses can continue to succeed.
- Infrastructure as Code is a powerful tool for managing infrastructure. It automates many tasks that your team is doing today, by addressing your infrastructure as if it is software. It is applicable whether you are working with servers in your own office or using multiple cloud service providers to manage a large global footprint of resources for your applications.

# INTRODUCTION

- Working with cloud infrastructure from the command line or the console is already easier than working with hardware and software in a physical data center environment. Although some might argue the point, it is probably even easier than operating virtualization software, because the interfaces were created more recently, are more flexible, and the approach to working is different; you get a more global view of your infrastructure.
- Infrastructure as a code takes this ease of use to the next level, and lets you address your infrastructure as if it is software. This means that instead of clicking through an interface with a list of 50 or 100 servers to find the machine or settings you wish to change, you write a program in a specialized language, execute it, and it makes the changes for you.

# LITERATURE REVIEWS

1. **Testing idempotence for infrastructure as code**, Waldemar Hummer, Florian Rosenberg, Fábio Oliveira, Tamar Eilam **Middleware 2013: ACM/IFIP/USENIX 14th International Middleware Conference, Beijing, China, December 9-13, 2013, Proceedings 14, 368-388, 2013:**

Due to the competitiveness of the computing industry, software developers are pressured to quickly deliver new code releases. At the same time, operators are expected to update and keep production systems stable at all times. To overcome the development–operations barrier, organizations have started to adopt Infrastructure as Code (IaC) tools to efficiently deploy middleware and applications using automation scripts

2. **Infrastructure as code: managing servers in the cloud** Kief Morris" O'Reilly Media, Inc.", 2016:

Virtualization, cloud, containers, server automation, and software-defined networking are meant to simplify IT operations. But many organizations adopting these technologies have found that it only leads to a faster-growing sprawl of unmanageable systems.

3. **A systematic mapping study of infrastructure as code research** Akond Rahman, Rezvan Mahdavi Laurie Williams **Information and Software Technology 108, 65-77, 2019:**

Infrastructure as code (IaC) is the practice to automatically configure system dependencies and to provision local and remote instances. Practitioners consider IaC as a fundamental pillar to implement DevOps practices, which helps them to rapidly deliver software and services to end-users

# LITERATURE REVIEWS

- 4. Adoption, support, and challenges of infrastructure-as-code: Insights from industry**Michele Guerriero, Martin Garriga, Damian A Tamburri, Fabio Palomba2019 IEEE international conference on software maintenance and evolution (ICSME), 580-589, 2019:  
Infrastructure-as-code (IaC) is the DevOps tactic of managing and provisioning infrastructure through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools
- 5. OpenIaC: open infrastructure as code-the network is my computer**Chunming Rong, Jiahui Geng, Thomas J Hacker, Haakon Bryhni, Martin G JaatunJournal of Cloud Computing 11 (1), 1-13, 2022:  
Modern information systems are built from a complex composition of networks, infrastructure, devices, services, and applications, interconnected by data flows that are often private and financially sensitive. The 5G networks, which can create hyperlocalized services, have highlighted many of the deficiencies of current practices in use today to create and operate information systems.

# OBJECTIVES

- **Consistency and reliability:** IAC enables you to define your infrastructure as code. This consistency ensures that your infrastructure is deployed in a reliable and predictable manner across different environments, reducing the risk of configuration errors or inconsistencies.
- **Efficiency and scalability:** With IAC and CloudFormation, you can automate the provisioning and management of your infrastructure. This automation eliminates manual intervention and reduces the time and effort required to deploy or update resources.
- **Simplified resource management:** CloudFormation provides a single, centralized way to manage and orchestrate your AWS resources. This simplifies resource management and ensures your infrastructure is correctly configured and maintained.
- **Stability:** Things go wrong with servers. Servers have a lot of moving parts. For me, I've had such a great experience working with AWS over the years, that I always assume the problem is me, not them.
- **Self documentation:** It is the fact that you are writing this program to configure infrastructure, rather clicking on choices in a web user interface or typing commands at the command line, that creates a self-documenting process.
- **Risk management:** Increasingly, companies are prioritizing compliance and governance as ways to improve quality and decrease risk.

# SOFTWARE AND HAEDWARE REQUIREMENTS

## ❖ **Hardware :**

- Operating system : Windows 10 or 11
- RAM : 8 GB
- Hard disc or SSD : More than 500 GB• Processor : Intel 3rd generation or high or Ryzen with 8 GB Ram

## ❖ **Software :**

- Software's used : Amazon web services



# EXISTING METHOD

- With IaC, configuration files are created that contain your infrastructure specifications, which makes it easier to edit and distribute configurations. It also ensures that you provision the same environment every time. By codifying and documenting your configuration specifications, IaC aids configuration management and helps you to avoid undocumented, ad-hoc configuration changes.
- Version control is an important part of IaC, and your configuration files should be under source control just like any other software source code file. Deploying your infrastructure as code also means that you can divide your infrastructure into modular components that can then be combined in different ways through automation.

1. **DISADVANTAGES:**
2. Complexity
3. Logic,
4. Conventions
5. Lack of Skills

# PROPOSED METHOD

- **Step 1** - Code your Infrastructure from scratch with the help of CloudFormation template language, in either YAML or JSON format, or start from many available sample templates.
- **Step 2** - Check your template code locally or upload your template code into the S3 bucket.
- **Step 3** - Use AWS CloudFormation from the browser console; then, use command line tools or APIs to create a stack based on your template code.
- **Step 4** - After this, AWS CloudFormation provisions and configure the stack and resources you specified on your template.

# INFRASTRUCTURE DIAGRAM

## CloudFormation Overview



Code in YAML or JSON  
directly or use sample  
templates

Upload local files or  
from an S3 bucket

Create stack  
using API via AWS  
CloudFormation

Stacks and resources are  
provisioned as a running  
environment

# STRUCTURE OF CLOUDFORMATION JSON TEMPLATE

- **Format version:** It defines the version of a template.
- **Description:** Any extra description or comments about your template are written in the description of the template.
- **Metadata:** It can be used to provide further information using JSON objects.
- **Parameters:** Parameters are used when you want to provide custom or dynamic values to the stack during runtime. Therefore, we can customize templates using parameters.
- **Mappings:** Mapping in the JSON template helps you to map keys to a corresponding named value that you specify in a conditional parameter.
- **Conditions:** Conditions are used to define if certain resources are created or when the resource's properties are assigned to a value when the stack is created.
- **Transform:** Transform helps in reusing the template components by building a simple declarative language for AWS CloudFormation.
- **Resources:** In this, you can specify the properties of AWS resources (AWS EC2 instance, S3 bucket, AWS lambda ) you want in your stack.
- **Output:** The output defines the value which is generated as an output when you view your cloud formation stack properties.

# EC2 INSTANCE CODE

- Resource : AWS::EC2::Instance
- Key Properties: Image Id, Instance Type, Key Name
- ❖ **CODE:**

AWSTemplateFormatVersion: 2010-09-09

Resources:

S3Bucket:

Type: AWS::S3::Bucket

Description: Create Amazon S3 bucket using CloudFormation

Properties:

BucketName: gddyfy-jvh-ghv-uu5


Outputs:

S3Bucket:

Description: S3 bucket created from a CloudFormation template

Value: !Ref S3Bucket

# EC2INSTANCE CREATION

 Services  [Alt+S]

≡

Step 1  
Create stack

Step 2  
Specify stack details

Step 3  
Configure stack options

Step 4  
Review

## Create stack

### Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready

☐ Use a sample template

☐ Create template in Designer

### Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

#### Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

☐ Amazon S3 URL

☒ Upload a template file

#### Upload a template file

EC2Instance.json

JSON or YAML formatted file

S3 URL: <https://s3.eu-north-1.amazonaws.com/cf-templates-ztmnhwn5jlz3-eu-north-1/2023-09-23T133332.169Z6i4-EC2Instance.json>

View in Designer

# EC2CREATION COMPLETE

The screenshot displays the AWS Management Console interface for the **EC2INSTANCESTACK**. The left sidebar shows the navigation menu with **CloudFormation** > **Stacks** > **EC2INSTANCESTACK**. The main content area is divided into two panels. The left panel, titled **Stacks (1)**, shows a list of stacks with the following details:

Stacks
<b>EC2INSTANCESTACK</b>
2023-09-23 19:06:57 UTC+0530
<b>CREATE_IN_PROGRESS</b>

The right panel, titled **EC2INSTANCESTACK**, shows the **Events** tab. It includes a search bar and a table of events:

Timestamp	Logical ID	Status	Status reason
2023-09-23 19:07:22 UTC+0530	EC2INSTANCESTACK	<b>CREATE_COMPLETE</b>	-
2023-09-23 19:07:21 UTC+0530	S3Bucket	<b>CREATE_COMPLETE</b>	-
2023-09-23 19:07:00 UTC+0530	S3Bucket	<b>CREATE_IN_PROGRESS</b>	Resource creation Initiated
2023-09-23 19:06:59 UTC+0530	S3Bucket	<b>CREATE_IN_PROGRESS</b>	-
2023-09-23 19:06:57 UTC+0530	EC2INSTANCESTACK	<b>CREATE_IN_PROGRESS</b>	User Initiated

# OUTPUT

The screenshot displays the AWS CloudFormation console. On the left, the 'Stacks' list shows 'EC2INSTANSTACK' with a status of 'CREATE\_IN\_PROGRESS'. The main panel shows the 'Resources' tab for this stack, listing one resource: 'S3Bucket' with a status of 'CREATE\_COMPLETE'.

**Stacks (1)**

Filter status:  Active ☒ View nested

Stacks

Stack	Created	Status
EC2INSTANSTACK	2023-09-23 19:06:57 UTC+0530	CREATE_IN_PROGRESS

**EC2INSTANSTACK**

Stack actions: Delete, Update, Stack actions, Create stack

Stack info | Events | **Resources** | Outputs | Parameters | Template | Change sets

**Resources (1)**

Search resources

Logical ID	Physical ID	Type	Status	More
S3Bucket	gdddyfy-jvh-ghv-uu5	AWS::S3::Bucket	CREATE_COMPLETE	-



# IN AMAZON S3 BUCKET

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

▼ Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight 7

► AWS Marketplace for S3

Amazon S3 > Buckets

► Account snapshot

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

View Storage Lens dashboard

Buckets (3) Info

Refresh

Copy ARN

Empty

Delete

Create bucket

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

< 1 > ⚙

Name	AWS Region	Access	Creation date
<a href="#">cf-templates-ztmnhwn5jiz3-eu-north-1</a>	Europe (Stockholm) eu-north-1	<a href="#">Bucket and objects not public</a>	September 23, 2023, 17:43:21 (UTC+05:30)
<a href="#">cf-templates-ztmnhwn5jiz3-us-east-1</a>	US East (N. Virginia) us-east-1	<a href="#">Bucket and objects not public</a>	September 22, 2023, 17:49:04 (UTC+05:30)
<a href="#">gddyfy-jvh-ghv-uus</a>	Europe (Stockholm) eu-north-1	<a href="#">Bucket and objects not public</a>	September 23, 2023, 19:07:01 (UTC+05:30)

School of Computing

17

# BUCKET VERSIONING DISABLED

The screenshot displays the AWS Management Console interface for an Amazon S3 bucket named 'gddyfy-jvh-ghv-uu5'. The left sidebar shows the 'Amazon S3' navigation menu with options like Buckets, Access Points, and Storage Lens. The main content area is divided into tabs: Objects, Properties (selected), Permissions, Metrics, Management, and Access Points. Under the 'Properties' tab, the 'Bucket overview' section shows the AWS Region as 'Europe (Stockholm) eu-north-1', the Amazon Resource Name (ARN) as 'arn:aws:s3::gddyfy-jvh-ghv-uu5', and the Creation date as 'September 23, 2023, 19:07:01 (UTC+05:30)'. Below this, the 'Bucket Versioning' section indicates that versioning is 'Disabled'. It also shows 'Multi-factor authentication (MFA) delete' is 'Disabled'. An 'Edit' button is visible next to the 'Bucket Versioning' status.

**Amazon S3**

**Buckets**

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

▼ Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight 7

► AWS Marketplace for S3

**gddyfy-jvh-ghv-uu5** Info

Objects Properties Permissions Metrics Management Access Points

**Bucket overview**

AWS Region  
Europe (Stockholm) eu-north-1

Amazon Resource Name (ARN)  
arn:aws:s3::gddyfy-jvh-ghv-uu5

Creation date  
September 23, 2023, 19:07:01 (UTC+05:30)

**Bucket Versioning**

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Edit

Bucket Versioning  
Disabled

Multi-factor authentication (MFA) delete  
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)

Disabled

# TO ENABLE BUCKET VERSIONING

## ➤ CODE:

AWSTemplateFormatVersion: 2010-09-09

Resources:

S3Bucket:

Type: AWS::S3::Bucket

Description: Create Amazon S3 bucket using CloudFormation

Properties:

BucketName: gddyfy-jvh-ghv-uu5

BucketEncryption:

ServerSideEncryptionConfiguration:

- ServerSideEncryptionByDefault:

SSEAlgorithm: AES256

## ➤ Outputs:

S3Bucket:

Description: S3 bucket created from a CloudFormation template

Value: !Ref S3Bucket

# REPLACE CURRENT TEMPLATE

The screenshot shows the AWS CloudFormation console interface for creating a change set. The top navigation bar includes the AWS logo, 'Services', a search bar, and user information for 'Stockholm' and 'Siva'. A left-hand sidebar lists the steps of the wizard: Step 1 (Create change set for EC2INSTANSTACK), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review EC2INSTANSTACK).

The main content area is titled 'Create change set for EC2INSTANSTACK'. It is divided into two sections: 'Prerequisite - Prepare template' and 'Specify template'.

In the 'Prerequisite - Prepare template' section, there is a description: 'Prepare template. Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.' Below this, three radio buttons are present: 'Use current template', 'Replace current template' (which is selected), and 'Edit template in designer'.

The 'Specify template' section has a description: 'A template is a JSON or YAML file that describes your stack's resources and properties.' Under 'Template source', it states 'Selecting a template generates an Amazon S3 URL, where it will be stored.' There are two radio buttons: 'Amazon S3 URL' and 'Upload a template file' (which is selected).

Under 'Upload a template file', there is a 'Choose file' button. Below it, a text box shows the filename 'EC2Instance.json' with a close button (X) on the right. Below the text box, it says 'JSON or YAML formatted file'.

At the bottom of the 'Specify template' section, an 'S3 URL' is displayed: 'https://s3.eu-north-1.amazonaws.com/cf-templates-ztmnhwn5jtz3-eu-north-1/2023-09-23T135726.371Ze3m-EC2Instance.json'. To the right of the URL is a 'View in Designer' button.

# UPDATE COMPLETED

The screenshot displays the AWS CloudFormation console interface. At the top, a blue notification banner states: "arn:aws:cloudformation:eu-north-1:818198869983:changeSet/EC2INSTANC-1n6c481x0e-gs9azcvmw7s/ee4d3b0d-5d80-44f9-a4eb-9ac4fc42cf06 has been executed." Below this, the left sidebar shows the "Stacks (1)" section with a list of stacks. The "EC2INSTANCSTACK" is highlighted, showing its creation time as "2023-09-23 19:06:57 UTC+0530" and its status as "CREATE\_COMPLETE".

The main panel displays the details for the "EC2INSTANCSTACK". It includes tabs for "Stack info", "Events", "Resources", "Outputs", "Parameters", "Template", and "Change sets". The "Events" tab is selected, showing a list of 10 events. The events table has columns for "Timestamp", "Logical ID", "Status", and "Status reason".

Timestamp	Logical ID	Status	Status reason
2023-09-23 19:29:27 UTC+0530	EC2INSTANCSTACK	UPDATE_COMPLETE	-
2023-09-23 19:29:27 UTC+0530	EC2INSTANCSTACK	UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	-
2023-09-23 19:29:25 UTC+0530	S3Bucket	UPDATE_COMPLETE	-
2023-09-23 19:29:05 UTC+0530	S3Bucket	UPDATE_IN_PROGRESS	-
2023-09-23 19:29:02 UTC+0530	EC2INSTANCSTACK	UPDATE_IN_PROGRESS	User Initiated

# CONCLUSION

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

Access analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

Earn an AWS Learning Badge to showcase your knowledge of S3.

Start now

Tags (3)

Track storage cost or other criteria by tagging your bucket. [Learn more](#)

Key	Value
aws:cloudformation:stack-id	arn:aws:cloudformation:us-east-1:21711655051:stack/create-bucket-cloudformation/f2bfa600-d867-11ec-8859-120a680fe489
aws:cloudformation:stack-name	create-bucket-cloudformation
aws:cloudformation:logical-id	S3Bucket

Edit

Default encryption

Automatically encrypt new objects stored in this bucket. [Learn more](#)

Default encryption

Enabled

Server-side encryption

Amazon S3-managed keys (SSE-S3)

Edit

Intelligent-Tiering Archive configurations (0)

Enable objects stored in the Intelligent-Tiering storage class to tier-down to the Archive Access tier or the Deep Archive Access tier which are optimized for objects that will be rarely accessed for long periods of time. [Learn more](#)

View details

Edit

Delete

Create configuration

Find Intelligent-Tiering Archive configurations

Name	Status	Scope	Days until transition to Archive Access tier	Days until transition to Deep Archive Access tier
No archive configurations				
No configurations to display.				

# REFERENCES

1. Wittig, Andreas; Wittig, Michael (2016). Amazon Web Services in Action. Manning Press. p. 93. ISBN 978-1-61729-288-0.
2. Bower, Joseph L.; Christensen, Clayton M. "Disruptive Technologies: Catching the Wave". Harvard Business Review.
3. Jump up to: Fletcher, Colin; Cosgrove, Terrence (26 August 2015). Innovation Insight for Continuous Configuration Automation Tools. Gartner (Report).
4. Riley, Chris (12 November 2015). "Version Your Infrastructure".
5. DevOps.com.
6. Phillips, Andrew (14 May 2015). "Moving from Infrastructure Automation to True DevOps" DevOps.com.
7. "Declarative v. Imperative Models for Configuration Management: Which Is Really Better?". Scriptrock.com. Archived from the original on 31 March 2015. Retrieved 14 December 2015.

# REFERENCES

8. Loschwitz, Martin (14 November 2014). "Choosing between the leading open source configuration managers". Admin Network & Security. Lawrence, KS USA: Linux New Media USA LLC.
9. Venezia, Paul (21 November 2013). "Puppet vs. Chef vs. Ansible vs. Salt". networkworld.com. Network World. Archived from the original on 18 July 2018. Retrieved 14 December 2015.
10. Garner Market Trends: DevOps – Not a Market, but Tool-Centric Philosophy That supports a Continuous Delivery Value Chain (Report). Gartner. 18 February 2015.
11. Jump up to:Chaganti, Ravikanth (5 January 2016). "DevOps, Infrastructure as Code, and PowerShell DSC: The Introduction". PowerShell Magazine. PowerShell Magazine. Retrieved 11 January 2016.
12. "Introducing AWS CloudFormation".
13. Sturgeon, Phil (28 October 2012). "Puppet or Chef?".
14. Ramos, Martin (4 November 2015). "Continuous Integration: Infrastructure as Code in DevOps". easydynamics.com. Archived from the original on 6 February 2016. Retrieved 29 January 2016.



# REFERENCES

15. Infrastructure As Code: Fueling the Fire for Faster Application Delivery (Report). Forrester. March 2015.
16. Wurster, Laurie F.; Colville, Ronni J.; Height, Cameron; Tripathi, Somendra; Rastogi, Aditi. Emerging Technology Analysis: DevOps a Culture Shift, Not a Technology (Report). Gartner.
17. "Cloud Threat Report Shows Need for Consistent DevSecOps", InformationWeek. 13 February 2020. Retrieved 24 February 2020.

# THANK YOU

We thank our guide and panel and all technical and non technical staff helped us in achieving this.