

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

"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 EilamMiddleware 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

- Infrastructure as code: managing servers in the cloudKief 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.
- A systematic mapping study of infrastructure as code research Akond Rahman, Rezvan Mahdavi Laurie WilliamsInformation 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 industryMichele 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

OpenIaC: open infrastructure as code-the network is my computerChunming Rong, Jiahui Geng, Thomas J Hacker, Haakon Bryhni, Martin G JaatunJournal of Cloud Computing 11 (1), 1-13, 2022:

Modern information systems are built fron 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

ॐ _H

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

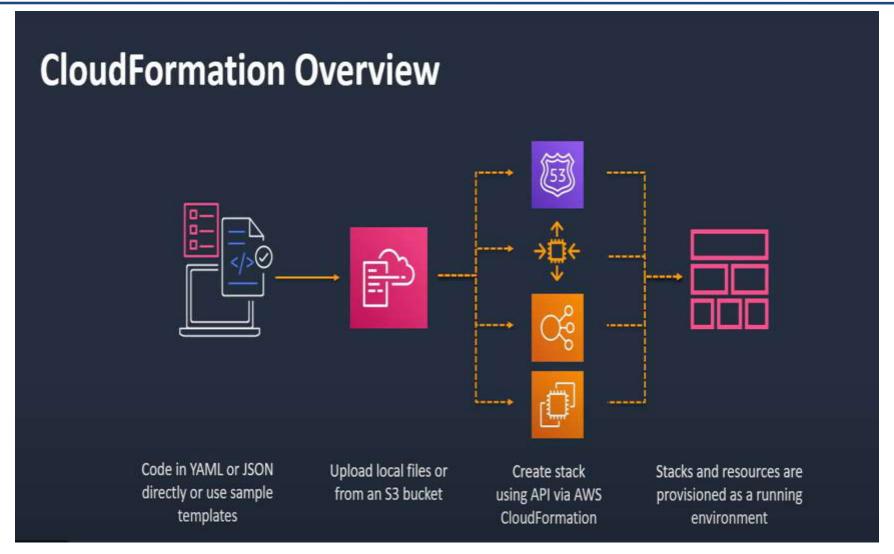
EXISTING METHOD

- 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:
- Complexity
- 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



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

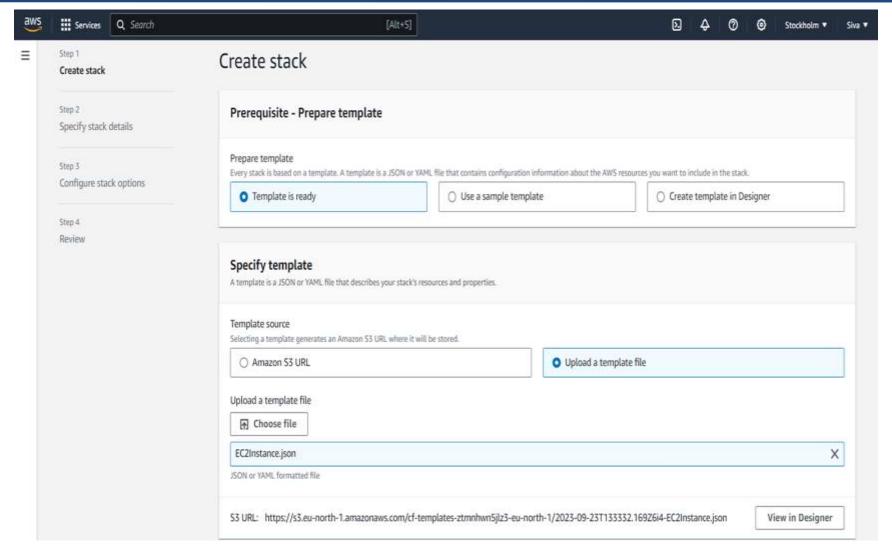
Outputs:

S3Bucket:

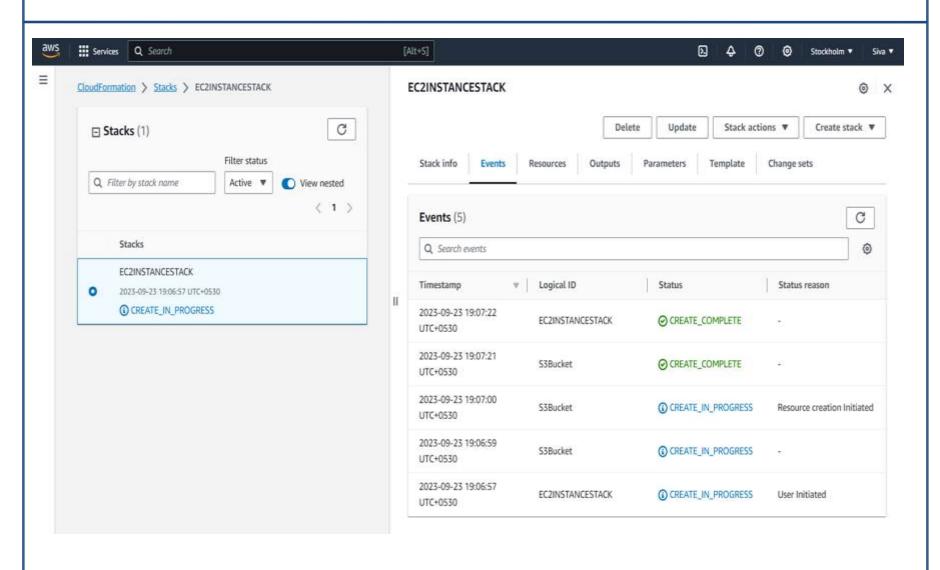
Description: S3 bucket created from a CloudFormation template

Value: !Ref S3Bucket

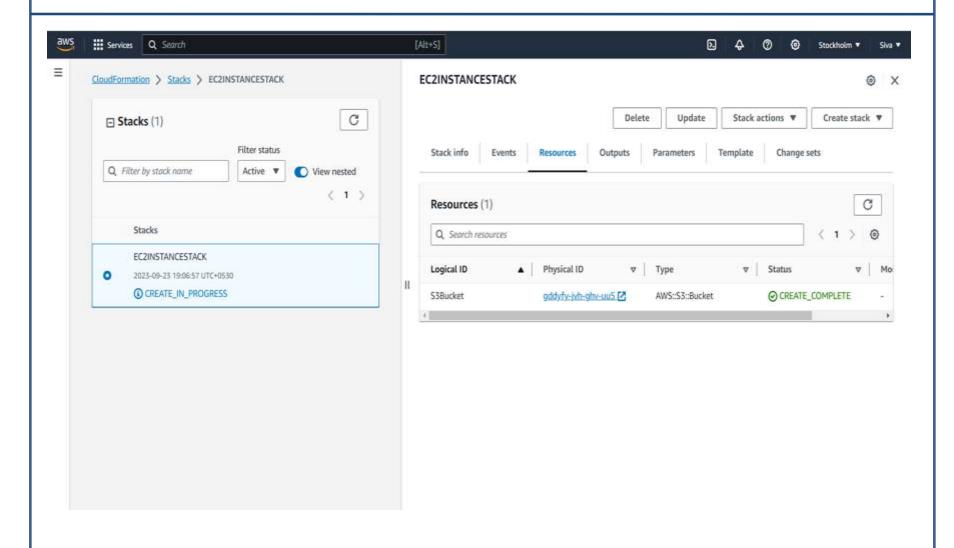
EC2INSTANCE CREATION



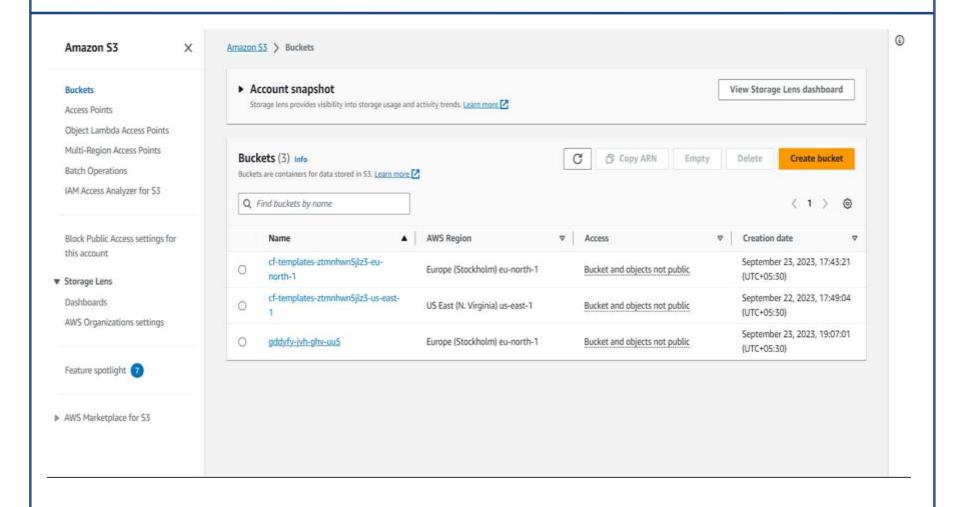
EC2CREATION COMPLETE



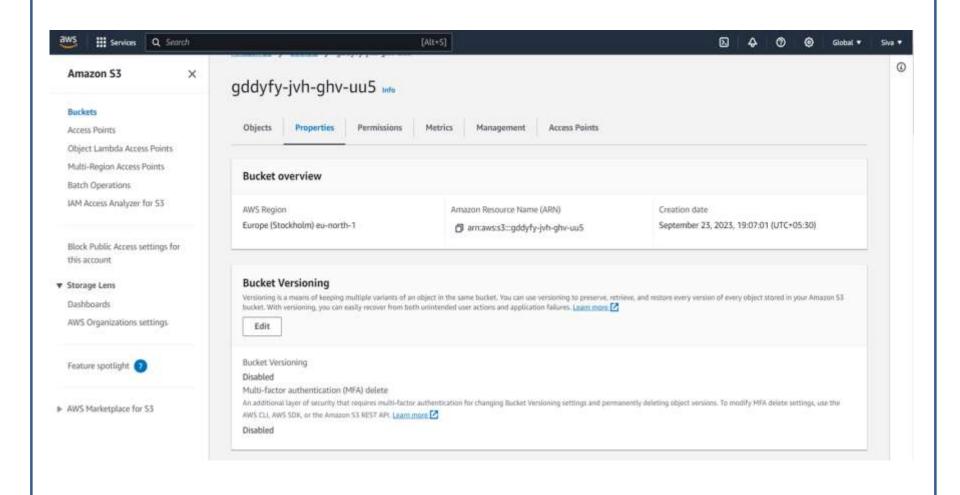
OUTPUT



IN AMAZON S3 BUCKET



BUCKET VERSIONING 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

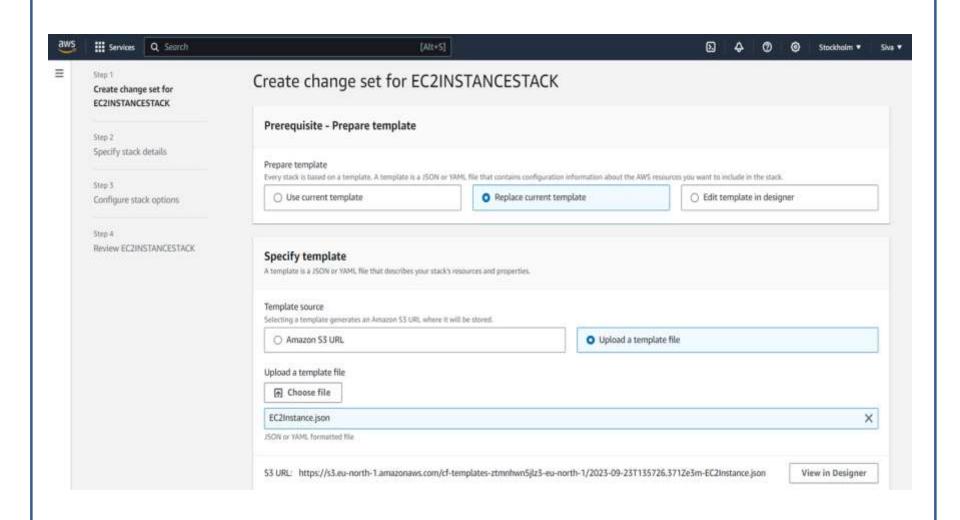
Outputs:

S3Bucket:

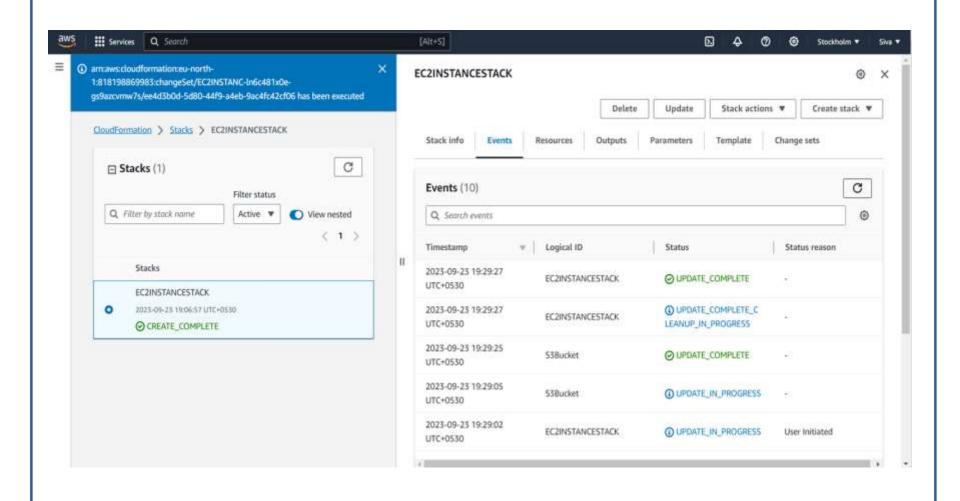
Description: S3 bucket created from a CloudFormation template

Value: !Ref S3Bucket

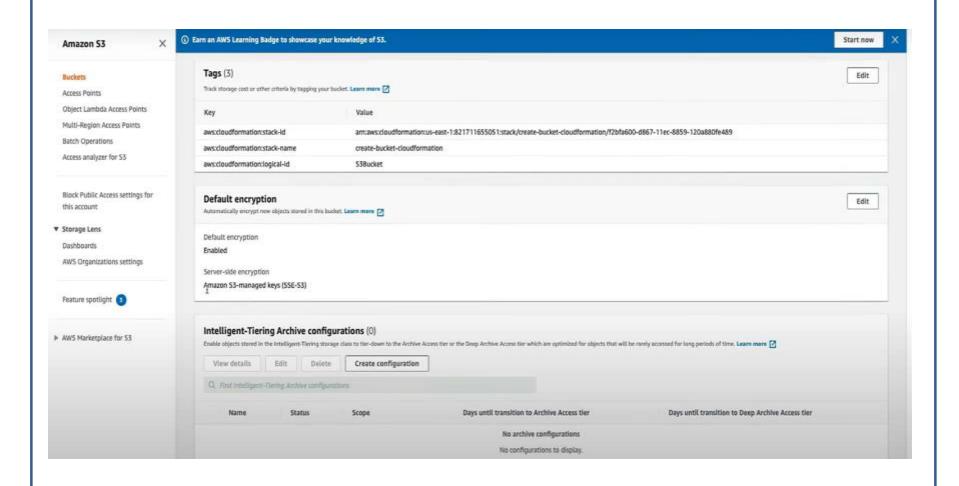
REPLACE CURRENT TEMPLATE



UPDATE COMPLETED



CONCLUSION



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. 5. Riley, Chris (12 November 2015). "Version Your Infrastructure".
- DevOps.com.
- 6. Phillips, Andrew (14 May 2015). "Moving from Infrastructure Automation to True DevOps" DevOps.com.
- "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 originalon 18 July 2018. Retrieved 14 December 2015.
- Garner Market Trends: DevOps Not a Market, but Tool-Centric Philosophy That supports a Continuous Delivery Value Chain (Report). Gartner. 18 February 2015.
- 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.
- "Introducing AWS CloudFormation".
- 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.