VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA

A TECHNICAL SEMINAR REPORT

ON

AUTOMATION PIPELINE AND BUILD INFRASTRUCTURE USING DEVOPS

A report submitted in the partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Information Science & Engineering

Submitted by

DEEPAK K(1SG17IS027)

Under the guidance of

Dr. Asha P.N.
Associate Professor,
Dept. of I.S.E., S.C.E.

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

SAPTHAGIRI COLLEGE OF ENGINEERING

Bengaluru-57

2020-21

SAPTHAGIRI COLLEGE OF ENGINEERING

14/5, Chikkasandra, Hesaraghatta Main Road, Bengaluru-560057

[*AFFILIATED TO VIVSVESWARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI*]

Department of Information Science & Engineering

CERTIFICATE

Certified that the Technical Seminar work entitled **AUTOMATION PIPELINE AND BUILD INFRASTRUCTURE USING DEVOPS** carried out by **DEEPAK K (1SG17IS027)** bonafide student of 8th semester, Department of **Information Science & Engineering, Sapthagiri College of Engineering,** Bengaluru in partial fulfillment of the award of **Bachelor of Engineering** in **Information Science & Engineering** of the **Visvesvaraya Technological University,** Belagavi during the year 2020-21. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Technical Seminar report has been approved as it satisfies the academic requirements in respect of Technical Seminar prescribed for the said Degree.

Signature of the Guide **Dr. Asha P N** **Associate Professor,** **Dept. of ISE**	**Signature of the Coordinator** **Prof. Sushmitha M S** **Assistant Professor,** **Dept. of ISE**	**Signature of the HOD** **Dr. H R Ranganatha** **Professor & Head,** **Dept. of ISE**

Name of the Examiner	**Signature of the Examiner with date**
1. ……	1. ……..
2. ……	2. ……

ABSTRACT

A complex project involves complex development effort and it involves more manual work and more time. It is very difficult to reduce the time involved. In the development cycle the use of the concurrent engineering concept only increases the problem by increasing the number of people and department involved at any point in the development effort. DevOps techniques can be used in these projects to control the use of project elements. Using DevOps Technique many companies involved in complex projects are able to reduce both development time and manual work involved in projects. This project solves issues that arise due to manual work on a project that involve more than one member in a team. The usual way of building a project in which code are distributed among team members leads to many issues like integration and compatible problems. The existing version control system gives us integration but that need to be tuned for more effective way of using that tool. This project builds an automation system that handles all the integration and deployment phrases automatically. This also reduces the time of compilation of huge codes. Manager need not wait for the results from team members since the results are stored in Git. Infrastructure as code (IaC) is a set of methods which make use of "code (in lieu of than manual operations) for getting setup with (virtual) machines as well as networks, installing dependencies, and configuring the development and production environment for the tool or software at hand. The infrastructure controlled by this code includes both the physical machines ("bare metal") and virtualized machines, docker containers, software-defined virtual networks. This code should be developed and managed using the same version control system as any other repository, for illustration, it should be constructed, tested, and warehoused in a version controlled repository.

ACKNOWLEDGEMENT

Any achievement doesn't depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities have helped us. I would like to take this opportunity to thank all of them.

I would like to express our sincere gratitude to **Dr. H Ramakrishna,** Principal, S. C. E., Bengaluru, for his help and inspiration during the tenure of the course.

I extend warm thanks to **Dr. H R Ranganatha**, H.O.D., Dept. of I.S.E., S.C.E., Bengaluru, for his constant encouragement, motivation and guidance.

I also extend my thanks to Technical Seminar Coordinator **Prof. Sushmitha M S**, Assistant Professor, Dept. of I.S.E., S.C.E., Bengaluru, for their valuable coordination and support.

I would like to thank my guide **Dr.Asha P N,** Associate Professor, Dept. of I.S.E., S.C.E., Bengaluru, for her timely advice, constructive suggestions and regular assistance in the Technical Seminar work.

Wholeheartedly I would like to thank the faculty members and staff of the Department of I.S.E., S.C.E., for their valuable time and expertise.

We would like to extend our heartfelt gratitude to our parents and to all our friends for their co-operation and motivation.

DEEPAK K

(1SG17IS027)

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In the technology driven world, building well defined applications are becoming a core approach for solving complex real-world problems, transforming industries, and delivering value in respective domains. Therefore, there are many software developer teams in the technical and open source communities trying to improve the overall business value using descriptive models. As a consequence, software engineer teams study how to apply DevOps principles to their complex systems under study.

DevOps is a set of practices and tools based on software and systems engineering. Software engineering can be defined as a discipline dedicated to developing tools and techniques which allow creation and use of sophisticated software systems. Agile, on the other hand, refers to an iterative approach which focuses on collaboration, customer feedback, and small and rapid releases. DevOps and Agile are two pillars to support in achieving business strategy and overlap the traditional operational and developmental teams to create an environment that is continually improving operations through a cross-functional team of developers and operators The main purpose of this project is to automate the development process of a project that includes the phrases of building the code, testing and deployment of the project which need more time when doing these tasks manually. Eight builds are involved which would take a huge time if done manually. The existing version control system does the three phrases separately which takes more time and work. It is a lengthy process that needs more space and time. The continuous integration and deployment were required that will be presented by this project. This project uses DevOps which helps in Continuous Integration and Continuous Deployment of the distributed code that are required at the single phrase for building, testing and deploying the project. The main of this project is designing of an application automation system using the technologies DevOps and CICD pipeline through GitLab. The CICD means continuous integration and continuous deployment. This method may likely produce unexpected results due to the dependency on data collection, preparation and preprocessing, model training, validation, and testing. Moreover, this method led to the conclusion that no apparent advantage exists in utilizing our manual approach for projects. Also, in terms of quality results, this manual pipeline method produces high operational costs and delays, which directly or indirectly affect the revenue or quality reputation of the business.

1.1 Overview

The entire model will be integrated with the important two devops principals the continuous Integration and continuous delivery practices. Building complex system model means that we advocate for automation and monitoring at all steps of the system construction, including integration, testing, releasing, deployment, and infrastructure management.

The functionality of CI is not limited only for testing, and validating code and components, but also testing and validating data, data schemas, and models. CD is no longer about a single package or service, but a pipeline that should automatically deploy another module service. IaC code can be used common throughout development, integration, and production environments. This improved environment parity and can remove scenarios where software works in one developer's environment but not for another developer, or scenarios where software works in development but not in the integration or production environment. The infrastructure code used for IaC ought to be stored in a version-controlled repository. This enables vigorous versioning of a deployed infrastructure: Any adaptation of the infrastructure can be produced using the IaC code corresponding to the desired edition. Together, automation and versioning deliver the potential to recreate a composition efficiently and consistently. This can be used to roll back a switch made during development, integration, or just as production and to support trouble-ticket regeneration and debugging..

1.2 Organization of Report

The report is organized into 5 chapters, starting with the chapter 1 provides a brief introduction to the proposed system and an overview of the proposed system. Chapter 2 mainly discusses about the literature survey which includes details about the papers and journals referred. It also discusses about the drawbacks and limitations in the existing system and advantages of using proposed system. Chapter 3 provides the working principle of the proposed system, It provides information about the different components used and their working in the system to yield the required. Chapter 4 provides merits and demerits of the proposed system. Chapter 5 provides information about the different applications of the proposed system.

CHAPTER 2

LITERATURE SURVEY

2.1 Related works

2.1.1 Description

A systematic learning about the mapping of existing IaC pillar technologies which had already helped rapidly deliver software and service to the consumer, talks about the IT giants such as GitHub, Facebook, Mozilla Firefox, Google and Netflix premier who have implemented IaC and collection a set of complex artifact information about DevOps and its insights. talks about various frameworks/tools for infrastructure as code, adoption of IaC, empirical investigations which are related to infrastructure as code. According to analysis, 50% of the studies publication reports propose to choose a framework or tool which implements the practice of IaC or to some extend provide the capabilities of some previously existing IaC tool. findings lead us to believe that framework or tools is a well premeditated topic and research about the faults and security flaws can have major significances for the production positioning and development environments involved in DevOps, it is well observed that the need for studies and research in this area is important and much needed. talks about how IaC is recent approach that and intend to improve collaboration between the development team and IT operation teams and create a channel for efficient communication and understanding. The studies involved are for the current frameworks which support such dimensions of operations. The approaches are compared and the tools like Chef and Puppet, are evaluation for coherency with existing CI/CD pipelines. The lessons learned in are those which allows use to create a set of concrete practices that would assist in transitioning from existing traditional approaches to an automation process of continuous software delivery. Works in are primarily focused in proposing abstract frameworks, which are designed to create a consensus to the ownership of DevOps characterization and their features. Some components such a collaboration as a philosophy and monitoring have surfaced as part of the study and are well discussed to and event approach.

2.1.2 Drawbacks

- One of the main obstacles for system module being deployed in production is the level of disruption that may occur with inclusion of other modules in frontend technologies.
- Operational mode, e.g., how new units are distributed, prioritized, and scan targets selected in a scalable manner or what a reliable strategy is to re-schedule scans to derive meaningful flow on trends.
- One of the major technical challenge is regarding the lack of coordination and improper handoffs between developers and operation teams which can lead to delays and errors.
- No proper tools and technology to take the integrated system to testing and deployment impact in increase of manual intervention in project.

CHAPTER 3

PRINCIPLE/WORKING

3.1 DevOps Methodologies

DevOps is a combination of the terms development and operations, meant to represent a collaborative or shared approach to the tasks performed by a company's application development and IT operations teams.

3.1.1 Concurrent versions system

Concurrent VERSIONS SYSTEM also known as Computer Vision Syndrome is a version control system, a vital part of Source Configuration Management (SCM). Utilizing it, developer can record the historical backdrop of sources documents, and reports. It fills a comparable job to the free programming RCS, PRCS, and Aegis bundles. CVS is a generation quality framework in wide use the world over, including many free programming tasks. It additionally lets to share various adaptations of records in a typical vault between group of engineers. CVS monitors numerous duplicates of source code documents and furthermore keep up a solitary duplicate and all the progressions are recorded.

Every designer's work independently in a different working index and CVS monitors all engineer's work. Crafted by a group of designers can be converged in a typical vault when wanted. Submit order is utilized to blend the changes. In cross-platform development, It is very easy to mess up the management of binary files (e.g., graphics such as icons). It is a bit difficult to learn about the myriad of configuration files that control CVS's behavior.

3.1.2 Client-Server Engineering

1) Client/worker CVS empowers engineers dispersed by geology or moderate modems to work as a solitary team.The adaptation history is put away on a solitary focal worker and the customer machines have a duplicate of the apparent multitude of documents that the designers are working on. Therefore, the organization between the customer and the worker must be up to perform CVS tasks, (for example, registration or updates) however need not be up to alter or control the current renditions of the files. Clients can play out in no way different activities which are accessible locally.

2) In situations where a few engineers or groups need to each keep up their own form of the records, on account of topography and additionally strategy, CVS's seller branches can import a

variant from another group (regardless of whether they don't utilize CVS), and afterward CVS can merge the changes from the merchant branch with the most recent records.

3) Open checkouts, permitting more than one engineer to chip away at similar records at a similar time. CVS gives an adaptable modules information base that gives an emblematic planning of names to segments of a bigger programming dispersion. It applies names to accumulations of indexes and documents. A solitary order can control the whole collection.

3.2 Working

3.2.1 Proposed automate pipeline with CI/CD

In request to beat downsides, for example, time, manual work and space gives needs to change to the advanced most recent philosophy like DevOps and Agile in computerization measure so as to get a proficient framework. 9DevOps is a product

improvement approach which includes ceaseless turn of events, constant testing, persistent mix, consistent organization, and nonstop checking of the product all through its advancement lifecycle. A persistent incorporation and sending pipeline (CD/CI) are such a significant part of a product venture. It spares a huge load of

manual, blunder inclined arrangement work. It results in higher quality software for continuous integration, automated tests, and code metrics. Auto DevOps plans to improve the arrangement and execution of a mature & modern programming advancement lifecycle. Highlight rich: Git archive the board, code audits, issue following, action takes care of and wikis. Relapse testing is the way toward testing

changes to PC projects to ensure that the more seasoned programming actually works with the new changes. Regression testing is re-running practical and non-utilitarian tests to guarantee that recently created and tried programming actually performs after a change. To lessen the exertion needed to finish the test robotization measure.

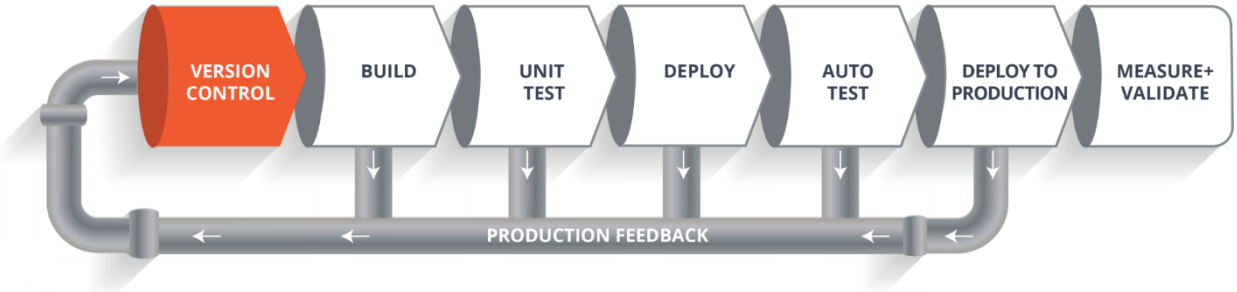

Figure 3.2.1.1: Continuous Integration/Continuous Delivery Pipeline

The system that is proposed has following steps for automation:

1) Configure the yml in GitLab according to developers need.

2) Make the GitLab for storing code and make it a centralized repository.

3) Construct the CICD pipeline for building, testing and deployment of the code across the developers.

4) Push the modified code to the pipeline.

5) Build the test suite for regression testing.

6) Let's initiate the system to do different phrases.

7) As per the report do the improvement process

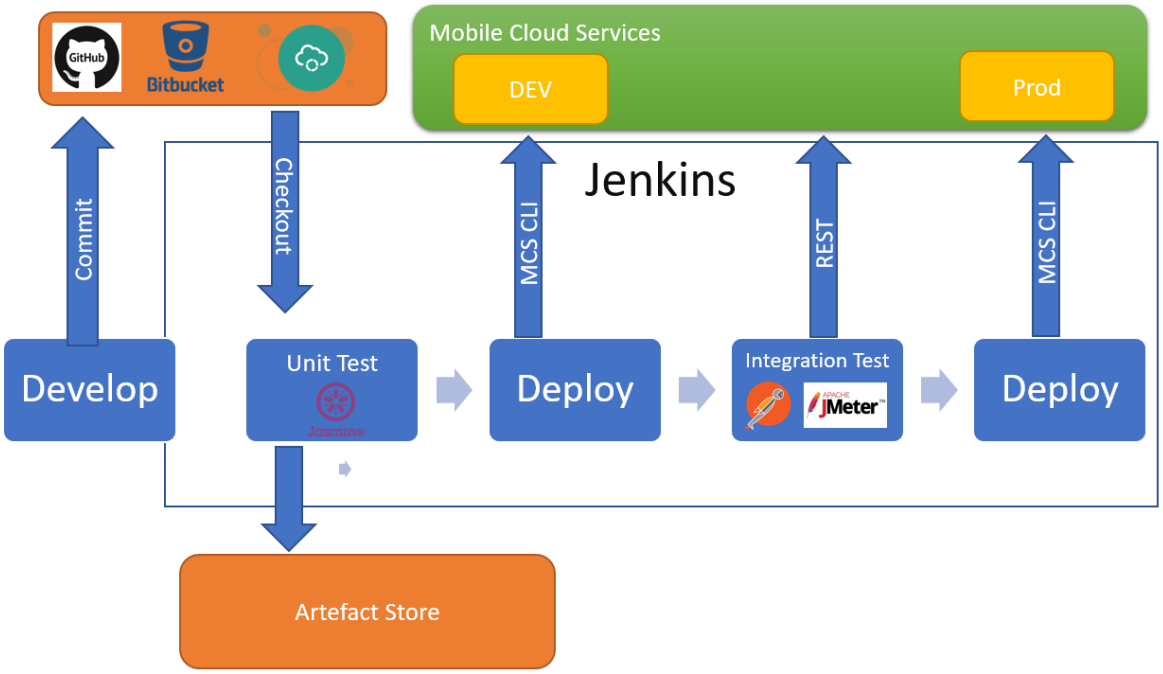

Figure 3.2.1.2 DevOps Architecture

CHAPTER 4

MERITS AND DEMERITS

4.1 Merits

The DevOps approach is very autonomous compared to conventional approach and very helpful because of the following reasons:

- Rapid innovation through robust machine learning lifecycle management.
- Creation of reproducible workflows and models.
- Effective management of entire development
- Easy Deployment of high precision application in any location.

4.2 Demerits

- Since the codes are build and deployed automatically, We should be careful while writing script to run the complete pipeline. Any semantic error may lead to build of invalid product or may conflict in production
- Business is about the evaluation consideration of model risks when actualizing a system.
- Major technical challenge is regarding the lack of coordination and improper handoffs between the developers and operation teams which can lead to delays and errors.

CHAPTER 5

APPLICATIONS

- Deployment, testing and rapid designing became ten times faster. It became effortless for the telco service provider to add patches of security every day, which used to be done only every three months.
- Pipeline definition defined and jobs of the pipeline limited only to the action of context required.
- Mindful resource management by the pipeline, this means that resource be destroyed as soon as their scope of use is closed.

CONCLUSION

In this paper, we extensively discussed automate pipeline with CI/CD principles. The method is based on DevOps practices, which are responsible for the integration and delivery of various models. The base motivation for the project was the limitation and enhancements identify by the existing testing strategy, which involved manual effort and crucial developer time. This limitation was then translated to the requirement for the project "automation provisioning of virtualized resources", to overcome the limitation faced by the existing developer team. For teams which aim at using Terraform as a key player to their changing management and deployment pipeline, it becomes crucial to identify a orchestration terraform module which can also deal with some category of automation in order to guarantee consistency flanked by runs, and deliver interesting insights about such deployments. The project was successful in bring about this automation and should be considered while implementing any future endeavors of the same nature In the existing system, the developer need to manually do the process of building, testing and deployment which consumes more time and even if user use scripts to do these process user can't stop while execution but if user use these automation system, the developer do these process in a pipeline manner and get the report periodically even if any phase got error the system don't move to next phase. It helps the developer to track the issues and fix it easier. In future this work can be further improved with the help of different phrases to be added to the existing system that can further improve the time and space issues. This work can be further extended for list of projects also. Along with this project some special phrases like system testing also can be added.

GLOSSARY

DevOps

DevOps is a set of practices that combines software development and IT operations. It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.

GIT

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

ACRONYMS

CI Continuous Integration

CD Continuous Delivery

IAC Infrastructure as Code

BIBLIOGRAPHY

[1] [1] Vidroha Debroy, Senecca Miller, “Overcoming Challenges with Continuous Integration and Deployment Pipelines When Moving from Monolithic Apps to

[2] Microservices”, IEEE Software, IEEE, ISSN: 2169-3536, Vol. 37, Issue: 3, Feb 2020, pp.21-29.

[3] [2] Keheliya Gallaba, “Improving the Robustness and Efficiency of Continuous Integration and Deployment”, 2019 IEEE International Conference on Software

[4] Maintenance and Evolution (ICSME), IEEE, ISSN: 2576-3148, Vol. 10, Dec 2019, pp.619 - 623.

[5] [3] Ana Filipa Nogueira, Jos´e C. B. Ribeiro, M´ario A. Zenha-Rela, Antoine Craske, “Improving La Redoute’s CI/CD pipeline and DevOps processes by applying

[6] Machine Learning techniques”, 2018 International Conference on the Quality of Information and Communications Technology, IEEE, ISBN: 978-1-5386-

[7] 5841-3, Vol. 96, Dec 2018, pp.282 - 286.

[8] [4] Mohammed Shamsul Arefeen, Michael Schiller, “Continuous Integration Using Gitlab”, URNCST Journal, Vol. 3, Issue: 8, Nov 2019, pp.40-45.

[9] [5] Alexander Poth, Mark Werner, and Xinyan Lei, “How to Deliver Faster with CI/CD IntegratedTesting Services?”, European Conference on Software Process

[10] Improvement, Springer, ISBN: 978-3-319-97924-0, Vol. 896, Aug 2018, pp.401 – 409. Jenkins. Build Great Things at any Scale..