Asian Journal of Engineering and Applied Technology
ISSN: 2249-068X (P) Vol.12 No.1, 2023, pp.23-27

DOI: https://doi.org/10.51983/ajeat-2023.12.1.3598

Application Deployment Automation by Streamlining CI/CD Pipelines

Upasana Ameta[1] and Ruchi Vyas[2]
[1]Research Scholar, [2]Assistant Professor,
[1&2]Department of Computer Science and Engineering, Geetanjali Institute of Technical Studies, Udaipur, Rajasthan, India
E-mail: ametaupasana@gmail.com, ruchi.vyas@gits.ac.in
(Received 20 March 2023; Revised 8 April 2023; Accepted 20 April 2023; Available online 30 April 2023)

***Abstract* - Conceptualize a scenario where the development, actions, quality affirmation, and information surveillance teams collaborate with the product owners to guarantee that the organisation achieves its goals for profitability, security, and cost-cutting. Together, they achieve world-class stability, reliability, availability, and security while conducting multiple number of code deployments each day. This enables the quick progress of planned work into creation. The testing of apps and InfoSec operations only take place at the conclusion of a project when it is too late to make any corrections. Development and IT Operations are adversaries in our environment. If any problems are found, and nearly every important task requires excessive amounts of backbreaking labour and handoffs, keeping us waiting all the while. The work quality, particularly the deployment of product, is challenging and disordered as a result, which has an adverse effect on customers and the business. This not only adds to the extraordinarily long lead times for getting anything done. As a result, there is lack of goals, and the whole company is dissatisfied with developers' performance, which leads to decrement in budget and frustrated workers feel incapable to modify the scenario and its results. The major objective is to build a platform for developers that can compile, test and run the application with the least specifications and configurations and also conceptualize the piece of deployment with a more relaxed architecture and a small learning period so that developers can take full advantage of multiple platforms available as a cloud service without any inconvenience.**
Keywords: **Tektone, Continuous Integration (CI), Kubernetes, Continuous Deployment (CD)**

I. INTRODUCTION

The goal of the proposed research project is to build a platform for the developers with a low learning curve for backend deployment that just needs a few configurations to operate the app. The proposed platform encourages features like Automatic Builds, CI/CD Pipelines, Resource Finiteness, and Default Scale-Up as per availability and needs, and also the application will incorporate all the advantages of the cloud, i.e., scalability, availability, crisis handling and recovery, security factors, and budget decrement. CI/CD is the most crucial part of DevOps the work of CI/CD platform is to maximize the productivity of any organization. The primary goals are to,

1. Configure Automatic Builds for various programming languages.
2. Project deployment resource limits.
3. Automatic scaling based on thresholds according to needs is item number.
4. Pipelines for continuous integration and deployment.
5. Based on resource usage, per-second billing.

The primary aim is to build an application platform for developers that, in addition to compiling and running the application with the least configurations possible, also hides the deployment related details.

II. BACKGROUND OF THE STUDY

A. Existing System

After examining the techniques and deployment strategies used by various IT organisations, we analysed that around 90% of them only release fresh updates on weekends without the permission of a senior developer. Second, virtually every other feature being implemented is closely connected to the entire technology stack.

B. Drawbacks

Whether the code a developer has created will be successfully deployed or not is something that always nags them at the production level. Second, even if it works in practise, it is extremely time-consuming to build up various OS versions, SDKs, Application Environments, and managing firewalls each time a new dependency or update/feature is made in order to test applications across many platforms.

III. LITERATURE REVIEW

Research related to the given problem has already been performed by various researchers and scientist.

A. Related work

Following research has been undertaken on the given problem statement.

1. "Applying DevOps Practices of Continuous Automation for Machine Learning", by Karamitsos, I.; Albarhami, S.; Apostolopoulos, C., published in Information 2020, proposed the integration of DevOps practices and tools with Machine learning Application

and development and operation. The paper suggests integrating CI/CD pipelines and tools to reduce wastage, improve operation, delivery, and maintenance of Machine learning applications.

2. "An Investigation of Automating Software Deployment Using Continuous Delivery Tools", by Yousif Touma, published at Linköping University draw comparison between automated deployments using CD and manually with respect to cost.
3. "Software development using DevOps tools and CD pipelines, A case study", by Oskari Jokinen published at University of Helsinki, studies the impact of using DevOps and automation pipelines for development team in Software development process.

IV. PROPOSED SYSTEM

Based on the shortcomings, a platform is required for the developers that only asks to upload the source code and general setup instructions to run the platform with very less configurations, while still having all the cloud services like scalability, serviceability, reliability, efficiency, decrement in deployment costs and improved performance.

V. METHODOLOGY

A software development and operations technique is known as DevOps [1]. Making team communication simpler will enable teams to produce, test, and distribute software more quickly and effectively. The main objective of DevOps is to ensure secure and reliable deployment of a software, with minimum manual arbitration of testing and advocating actions. The DevOps process is shown in Fig. 1.

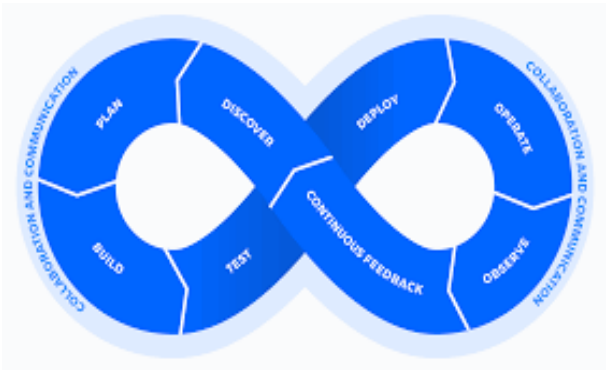

Fig. 1 DevOps Methodology

The core idea behind DevOps is to manage the whole development procedure between the development and executive teams through collaboration. There is a need of a thorough knowledge of software release and deployment in both the teams, software/hardware requirements, and deployment implications, which can be seen in Fig. 2.

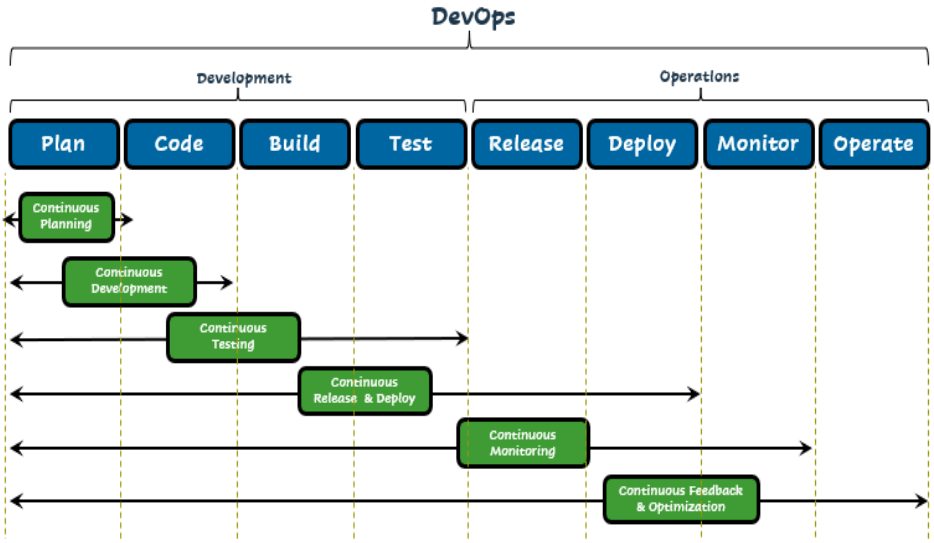

Fig. 2 DevOps Work Plan

VI. SCOPE OF THE STUDY

Many Fortune 500 companies have experienced world-class performance improvements since the advent of DevOps practises and methodologies, resulting in higher revenue, faster release, and minimum hassle at time of deployment process [2]. The development rate of products has improved, and failure rates have reduced after using DevOps. DevOps has managed the pipeline process such that high-quality software may be produced with a minimum of work.

In order to reduce security risks that can happen after the product reach the production servers, continuous feedback is required on the deployment methodology at time of development, testing, and release stages for any application platform. By employing this project, developers will be able to gain significant reliability, security, regular integration, constant delivery, and major cost reductions, but they can also release their applications at greater scales across several clouds and managed services [3].

The future scope of the proposed system is as follows:

A. AI/ML Pipelines

DevOps has revolutionised the environment for software development [4]. But with the technology of artificial intelligence and machine learning, it is possible to create more controlled automated applications. Pipelines when integrated with AI and ML can assist in execution and automation in an effective and controlled manner, as well as providing a practical way to automate a complete pipeline.

There are several applications for the burgeoning field of platform as a service (PaaS). The need for developers to worry about creating a whole application architecture is long gone. With PaaS, teams can increase their development capacity without hiring more personnel, potentially cutting costs for engineering. The underlying cloud infrastructure, including the network, servers, operating systems, and storage, are out of the developers' hands, but they do have control over the deployed programmes and perhaps even the configuration options for the application hosting environment. Fig. 3 shows the difference between IaaS and PaaS. Fig. 3 displays the activity diagram for CI/CD that uses Tektone Pipeline.

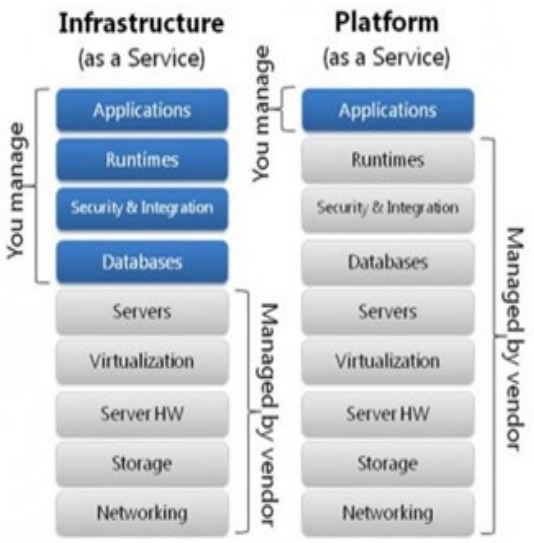

Fig. 3 IaaS vs. PaaS

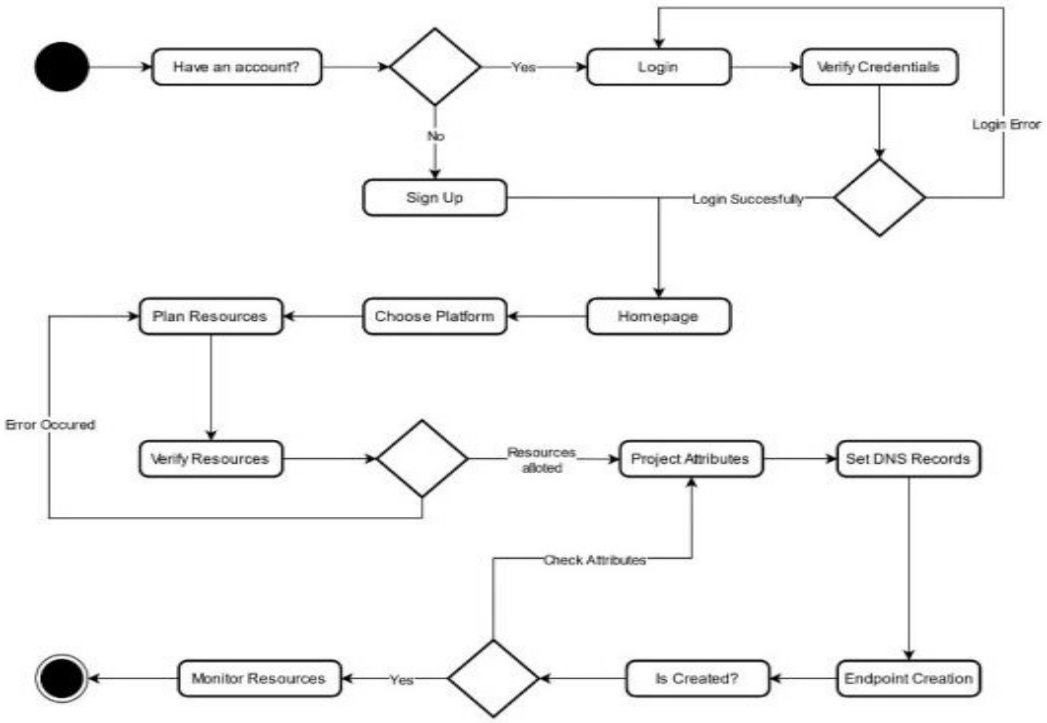

Fig. 4 CI/CD using Tektone Pipeline

B. Configuration Management of Overridden Containers

Container orchestration platforms are upending the technological landscape. The container orchestration techniques might take the place of many technologies like Ansible, Chef, and Puppet. Kubernetes is the most popular and widely employed container orchestration system [5], but there exists many more on the vista. When properly developed, container orchestration solutions may make infrastructure provisioning and much of the associated complexity simpler.

VII. WORKING

A. Distributed Workloads

In modern systems APIs (Application Programming Interface) provide many services over the network.

Distribution systems are used to spread and distribute them, which function across numerous servers and configurations located in various locations. Communication is enabled among these systems to facilitate coordination between their respective operations.

B. Load Balancers for Zero Downtime

Earlier, an application was deployed on a single VM, and a DNS(Domain Name Sever) was directed to it. However, now, by creating distributed systems like this, load balancing and horizontal scalability are achieved.

C. Restful API's

The publicly accessible APIs are used around the globe daily, that makes these systems need very dependable and available, which means failure is not allowed and shouldn't experience any downtime. All of the mentioned services are used around the world, so they need to be scalable without requiring an entire redesign of the existing infrastructure. Kubernetes offers the right services for your containerized application to perform all of these tasks [6].

D. Using Tektone Pipelines for CI/CD

Continuous Integration (CI) enables the continuous integration of code into a single, accessible, and shared repository [7]. We can regularly push code from the repository to production with the aid of continuous delivery (CD). By creating a quick and efficient procedure, CI/CD enables us to release new features and problem fixes considerably faster than the competition [8].

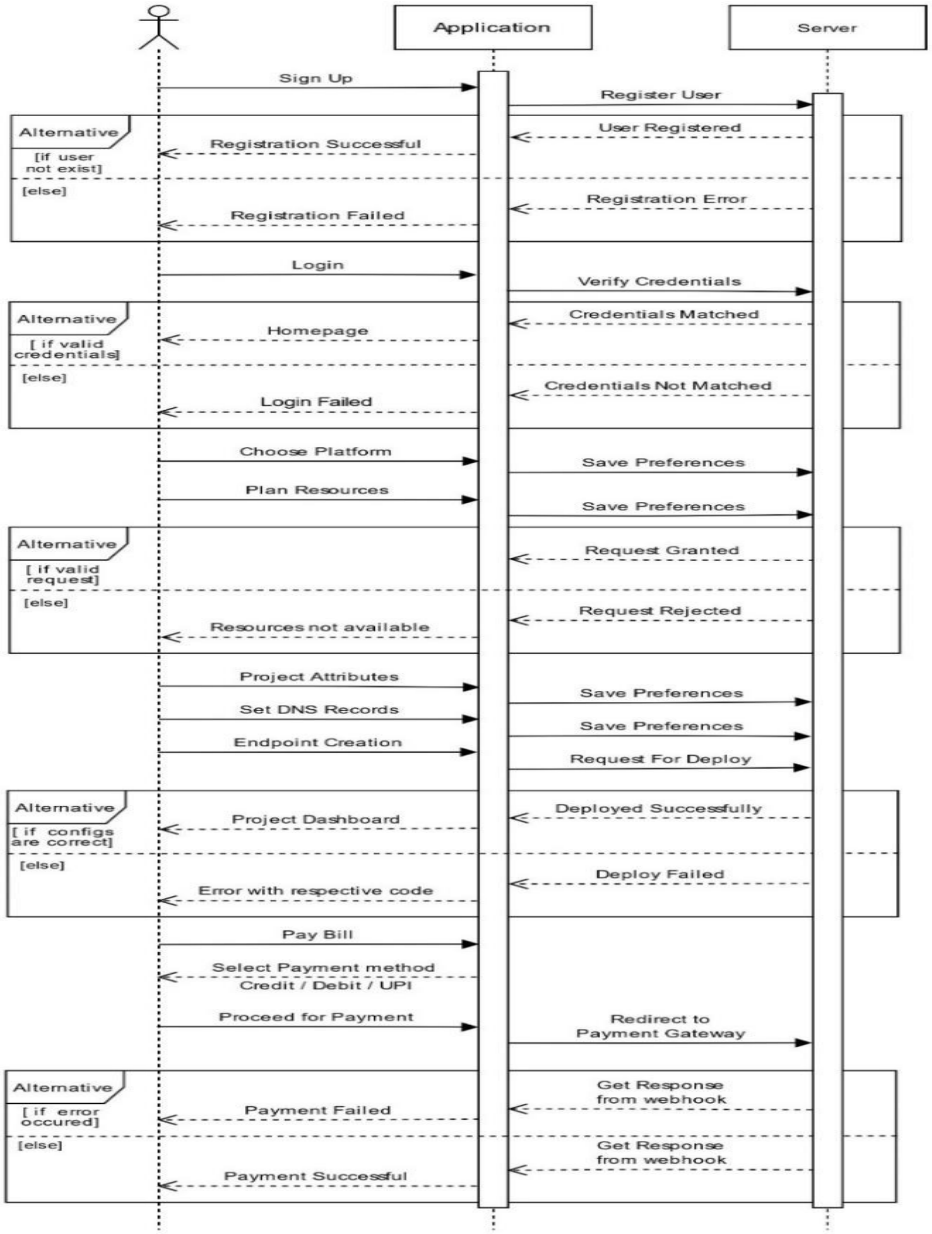

Fig. 5 Sequence Diagram for Proposed System

VIII. CONCLUSION

According to the report, numerous IT companies have demonstrated that releases and deployments shouldn't be high-risk, tightly connected, or necessitate the deployment of hundreds or even thousands of engineers. Instead, it ought to be carried out in a way that becomes commonplace and a necessary component of daily operations. By doing this, we may further reduce the lead times in seconds and limit the impact of botched deployments and pandemonium at the production level. Our project makes this possible by reducing the time and labour that is required to develop, test, and compile the application at Testing and Production environment. Numerous deployment strategies like A/B Testing, Blue-Green, Shadow, Ramped etc. are assisted by providing automated development for multiple programming languages. In order to reduce the release time, Continuous Integration and Continuous Deployment Pipelines are developed and executed. The scalability of the application is vast according to user requirement requirements and traffic while integrating Kubernetes and multi-cloud platforms. This not only encourages the isolation between different applications but also makes it possible for per-second billing approaches to earn revenue from the resources based on usage per second.

IX. FUTURE SCOPE

For future enhancement, the effect of using CI/CD can be analysed in various contexts towards finding a generalized result. The proposed solution can be extended to be made compatible to work differently on different types of systems and platforms. The system can also be improved in order to reduce downtime and analysed to find best cloud platform for it.

REFERENCES

[1] Floris Erich, Chintan Amrit and Maya Daneva, "A Qualitative Study of DevOps Usage in Practice," *Journal of Software: Evolution and Process,* Vol. 29, No. 6, pp. 1-20, 2017.

[2] L. E. Lwakatare, P. Kuvaja and M. Oivo, "Dimensions of DevOps," In: Lassenius, C., Dingsøyr, T., Paasivaara, M. (eds) Agile Processes in Software Engineering and Extreme Programming. XP 2015. *Lecture Notes in Business Information Processing, Springer, Cham,* Vol. 212, 2015. https://doi.org/10.1007/978-3-319-18612-2_19.

[3] G. Blinowski, A. Ojdowska and A. Przybyłek, "Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation," in *IEEE Access,* Vol. 10, pp. 20357-20374, 2022, DOI: 10.1109/ACCESS.2022.3152803.

[4] Valentin Adamescu, *Analysing monolithic and microservices software architecture for SME web services/applications,* 2020.

[5] Tesliuk, S. Bobkov, V. Ilyin, A. Novikov, A. Poyda and V. Velikhov, "Kubernetes Container Orchestration as a Framework for Flexible and Effective Scientific Data Analysis," *Ivannikov Ispras Open Conference (ISPRAS),* Moscow, Russia, pp. 67-71, 2019, DOI: 10.1109/ISPRAS47671.2019.00016.

[6] S. K. Mondal, R. Pan, H. M. D. Kabir, *et al.,* "Kubernetes in IT administration and serverless computing: An empirical study and research challenges," *J Supercomput,* Vol. 78, pp. 2937-2987, 2022. DOI: https://doi.org/10.1007/s11227-021-03982-3.

[7] Mojtaba Shahin, Ali Babar, Muhammad, Liming Zhu, "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices," *IEEE Access,* Vol. 5, pp. 3909-3943, 2017.

[8] S. A. I. B. S. Arachchi and I. Perera, "Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management," *Moratuwa Engineering Research Conference (MERCon),* Moratuwa, Sri Lanka, pp. 156-161, 2018, DOI:10.1109/MERCon.2018.8421965.