International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 01, January-2022

Integration of Hyperledger Fabric Blockchain in Software Development

Sandip Bankar
Research Scholor, IT Department,
Mumbai University, India

Deven Shah
Research Guide, IT Department,
Mumbai University, India

***Abstract*— Software development methodologies are needed so that the software development life cycle can become systematic to complete application development within the right time frame. Nowadays, IT companies are using DevOps to increase the number of releases, but at the same time, the security of project documents is getting ignored. This paper focuses on an architecture that redefines the DevOps pipeline by turning the Version Control System (VCS) and the Integration Server (IS) into a Blockchain-based decentralized system as a separate and independent fabric channel. The advantage of having such a system is that it prevents the problems from occurring in security due to centralized architecture. This also allows the system to generate reports at each stage, hence improving the compliance and transparency of the architecture. This framework uses Hyperledger Fabric as a backbone to increase the security of the documents and prevent and track access to each of the software documents. After optimization, the performance of the proposed system is found to be better than similar systems in terms of throughput and latency.**

Keywords:- Hyperledger Fabric; Blockchain; decentralization; DevOps; Continuous Monitoring; Continuous Integration;

I. INTRODUCTION

The Software Development Life Cycle (SDLC) is a continuous process that starts when a decision is made to start the project and ends when the project is no longer used [1]. SDLC models are broken up into main groups, each with their own strengths and weaknesses. They came from the first and oldest SDLC "waterfall" method, and their range has grown a lot. SDLC models have a wide range of applications, from simple web apps to complicated applications. This means that the range of models is limited by the wide range of products. There was a problem with the "Agile methodology" when the organizations that took on client service started having trouble with collaboration with team members [2]. The development team usually gave the operations team a lot of work at once. All the teams didn't work well together, especially the teams that oversaw making the service (the development team) and the teams that oversaw putting the service in the customer's hands (the delivery team and operations team). In this case, the operations team didn't work at the same speed as the development team did. So, for some organizations, the Agile method wasn't enough. DevOps came into the picture after Agile methodology. Its software development culture includes a lot of different things, like how to organize teams, build systems, and even the structure of the systems you build [3,4]. When teams work together, it becomes difficult to manage teams, and coordination between them becomes challenging. There is a need to gain trust among the team members.

Blockchain is a technology that is used to create trust between untrusted parties [5]. It has many features that can be used to gain trust in heterogeneous teams. Hyperledger Fabric is a permissioned Blockchain application system and one of the Linux Foundation's partners Hyperledger projects. Hyperledger Fabric is meant to be a framework for making software or other things that can be put together in different ways. It has plug-and-play modules like consensus and membership services. Hyperledger Fabric uses containers to store "Chaincode" smart contracts that make up the system's application logic. These contracts are called "smart contracts" [6]. 3 Multiple stakeholders like managers, developers, testers, clients, and operations are involved in the overall software development life cycle (SDLC) throughout the traditional approach to application development and maintenance [7]. Software development approaches are often granulated in terms of features, cost, schedule, quality, performance, and other administrative overheads such as vendor integration by techniques, frameworks, processes, people, and tools that impact the final product. The SDLC operations and production engineers work together throughout the project lifecycle, from design through to the development process, to assist in the whole production. To avoid collaboration problems, it is important to bridge the gap between the engineering and operations teams by establishing a meeting point for two of Blockchain and SDLC's most promising technologies. A certain blockchain's strength lies in its decentralized nature [8], which means that each member of its peer-to-peer network can depend on the other. If some Blockchain businesses have systems that are optimally incorporated into the system and can produce high-quality software, but others cannot, the trust in the system begins to break down, and those within it. Implementing Blockchain with a DevOps approach is an IT philosophy that can prevent this issue. This consists of different practices and strategies that affect organizations in coordinating teams, building systems, and even affecting the programme structure they are building [9].

The rest of this paper is organized as follows. Section 2 provided a simple understanding of the Related work, proposed architecture is explained in Section 3.Section 4 explains the role of blockchain and Inplanetory File System(IPFS) in the proposed system.robust encryption algorithm prerequisites are given in Section 4. Section 5 and 6 describes implementation and performance analysis respectively. Finally, the paper is concluded with further recommendation.

II. RELATED WORK

Recent work has shifted away from traditional software development methods and toward DevOps. Due to the limitations of the traditional approach to development and operations management [10], DevOps was born. DevOps is a way of working that involves the use of several technologies

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 01, January-2022

to improve communication and cooperation between development and operations teams to minimize friction between the two groups and promote development efficiency and quality.This section will summarize the major issues [11,..,15] that prompted the development of the suggested technique.

- Smoothly delivering code from development to production is a challenge.
- A lack of trust between development and operations
- A communication and collaboration gap between development and operations
- When a problem occurs, people may assign blame between teams.
- Geographical distribution can present challenges.
- A lack of management structure between developers and operations
- DevOps has not been organized and managed in large-scale projects in a systematic manner.
- People are a challenge when implementing DevOps, as they are required to collaborate in ways they have never seen before.
- Smoothly delivering code from development to production is a challenge.

The reality of the Software Development Life Cycle is very complex and untrusted in enterprises with multiple lines of business. In an environment of rapid software development and deployment, it is necessary to strengthen DevOps by providing decentralized data storage of documents on Blockchain to increase trust, auditability, and traceability.

III. PROPOSED ARCHITECTURE

The proposed architecture is focused on utilizing a few blockchain advantages, namely, permissioned membership and channels. Permissioned Membership enables us to audit each transaction and the members associated with the transaction, and Channels assist in isolating and implementing a separate audit for a few critical parts of the pipeline by maintaining a separate copy of the ledger. Proposed system transaction consists of project file creation, its access, and updates. Every Blockchain-enabled framework provides a ledger which records each document with its version history on immutable storage so that every file transaction is tracked easily and available to each stakeholder for the project.

This framework also helps our product perform parallel monitoring throughout its lifecycle. The transactions that are present in our Chaincode/Smart Contract in this work are based on all the inputs and outputs of various tools and participants' documents. This proposed work consists of a decentralized architecture as shown in figure 1. The proposed solution redefines the DevOps pipeline by making the Version Control System (VCS) and Integration Server (IS) separate, isolated Hyperledger Fabric channels features. Isolation also allows us to make our pipeline have a pluggable environment. Above, Figure 1 shows the overall architecture which covers the different participants, including the Distributed Database (IPFS) and Permissioned Blockchain Network (Hyperledger Fabric). Each stakeholder creates a specific type of file, which is stored in a Blockchain-enabled decentralized environment.

This proposed system consists of a deployment pipeline that permits parallel monitoring service and storage in a distributed database [16]. This work focuses on preventing the problems caused by having a centralized architecture in the pipeline. Furthermore, keeping the two channels locally available will not disrupt the development team from committing their code and checking whether it fits with all the other modules coming from other teams built into one executable application. Figure 2 shows the proposed DevOps pipeline, which is flexible as it does not depend upon any specific tool that an organization would like to use in their current development or operations stages. The architecture currently plans on allowing the operations side to be hosted on the internet since it involves provisioning real-time virtual machines (VMs) and that has to be done physically. However, the scripts they write for provisioning, as seen in the case of tools like Chef or Puppet Configuration management, will be going under our VCS channel. The main features of the proposed system are the ability to perform continuous monitoring and the ability to generate reports at each stage of the pipeline. This is not only important to the people working on building and maintaining the product but also to the people who proposed the work on this product. Stakeholders have the right to see the status of their product and decide the compliance value of an organization. This will also add to the key point that the organization is being as transparent as they can to satisfy their customers' needs and make sure they are regularly keeping their clients updated about the status of the product they proposed [18].

IV. ROLE OF BLOCKCHAIN AND IPFS

This research work ensures pipeline transparency. Compliance professionals can retrieve a summary report containing all the pertinent data that can be added freely. Continuous monitoring is also supported throughout the pipeline, as the underlying blockchain records all transactions, such as data entering or being sent to a tool at each level, as well as the output from each tool. As a result, each transaction may be tracked back to get pertinent information about a particular level of the pipeline. The proposed system uses a backend file store, specifically the IPFS file store, which enables the storage of files on IPFS nodes connected to a network or in a cluster of nodes connected to a network and provides a mechanism for rapidly accessing these files without incurring significant latency caused by reliance on a remote server that hosts the file.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 01, January-2022

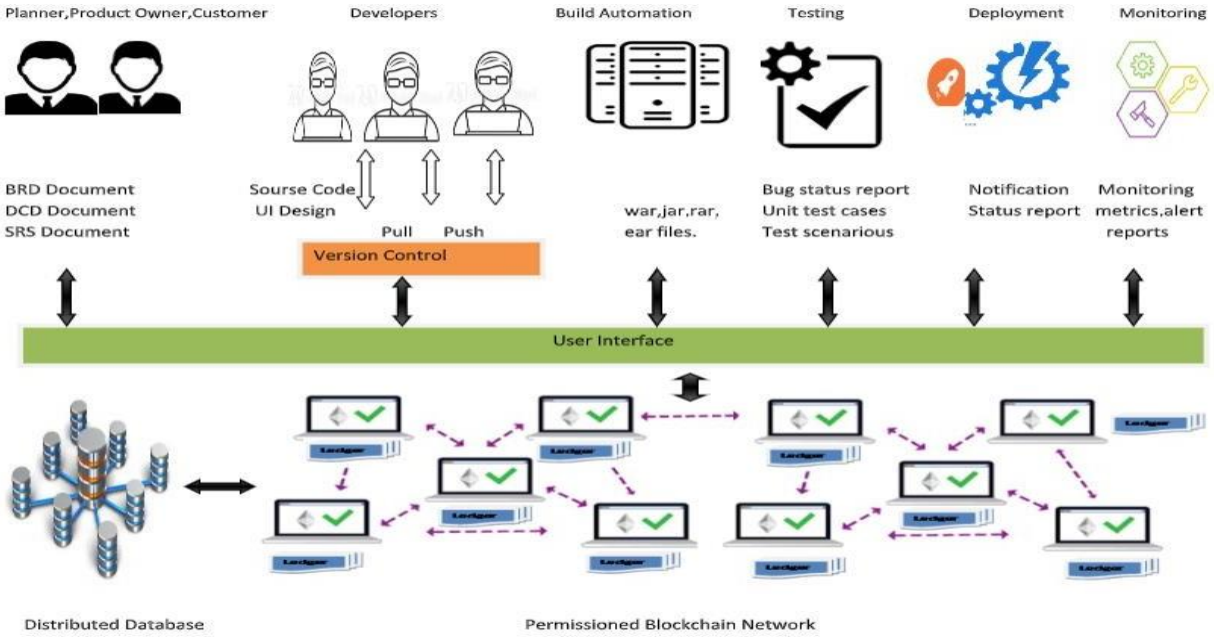

Fig.1. Proposed Architecture

Thus, even if a node or peer geographically adjacent to us has the file you're looking for, the network retrieves the file's contents from that node rather than relying on the central server. Nodes interact with one anotherand can store one another's files. They only store the files that the nodes request.

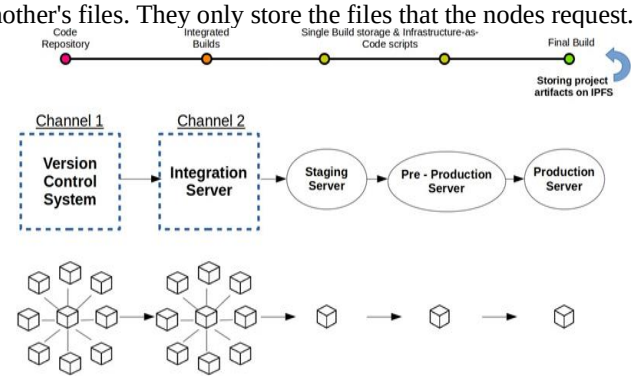

Fig. 2. Proposed DevOps pipeline

IPFS allows accessing files by their metadata hash. It is termed a "multihash". The purpose is to remove the dependency from referring to a physical location (via IP addresses) and refer to a content-based addressing mechanism to allow looking upon any peer that is in the network and fetching the desired file or folder faster as compared to a server's response time. Relying on the hash of a file or folder (collection of files) means that one can preserve the integrity of the file uploaded. Any change to the file will result in a new hash of the file and would require re-uploading the file. Thus, a new reference is generated to be able to access this modified file.

All the artefacts that are uploaded to or produced after a version control, build, or testing tool's operations will be stored on IPFS. IPFS utilizes content-based addressing and allows us to fetch files with the help of a file hash from a nearby IPFS node. This removes the unwanted overhead of relying on the URL-based addressing mechanisms if an organization were to go for cloud-based DevOps pipeline solutions. All the artefacts that are uploaded to or produced after a tool's operations will be stored on IPFS. IPFS utilizes content-based addressing and allows us to fetch files with the help of a file hash from a nearby IPFS node. This removes the unwanted overhead of relying on the URL-based addressing mechanisms if an organization were to go for cloud-based DevOps pipeline solutions.

Fig. 3. IPFS content-based addressing

As shown in Figure 3, the proposed work maintains hashes of several folders in a JSON file. It enables us to monitor several versions of a file, regardless of whether it is a single file or a batch of files, as each uploaded file is saved in a directory called "DocumentsDir". The JSON array 'dirHash' maintains several copies of the directory hash as files within the directory are updated or new files are added.The files, or artifacts, are stored in the form of chunks of 256 KB. After that, it utilises a linked list-based referencing mechanism to link chunks of a relatively larger file. This means that users need not worry about the confidentiality of their project artifacts.

V. IMPLEMENTATION

Version control is critical for files that are revised and redrafted frequently and is especially critical for electronic documents, which may be easily altered by several users. For example, knowing the version of a document you're working with is critical. For example, if you're attempting to determine which version of a policy is now valid or which version was in

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 01, January-2022

use at a particular time, Version control is also critical if stakeholders are working on a collaborative document that involves many participants and/or frequent modifications. As stakeholders create and execute documents such as standardized regulations or version tracking for project papers, they may trace the occurrence of a document over an extended period, for example, several draughts and final versions of a regulation. This enables traceability, the preservation of documents and details of changes made, typically because of contributions from multiple collaborators during the development of documents, the sequencing of changes, and the recording of those versions of documents considered final and/or approved by appropriate teams or individuals.

Metadata such as computer filename, author, and date are used to associate all documents. Additionally, several versions of documents must be distinguished, with the most recent version being shown. Additionally, continuous integration is the technique of merging feature branches into the primary branch of code that will be mechanically created and tested. Because this is frequently an automated flow, it can be performed several times daily or multiple times per week. Figure 4 shows the different versions of files that a developer handles. These versions are generated whenever a "Commit" operation is executed by clicking on the "Commit" button on the User Interface. Also, the user has to push their changes before committing since the idea is to only create versions of the file where you will see significant changes in each version you view.

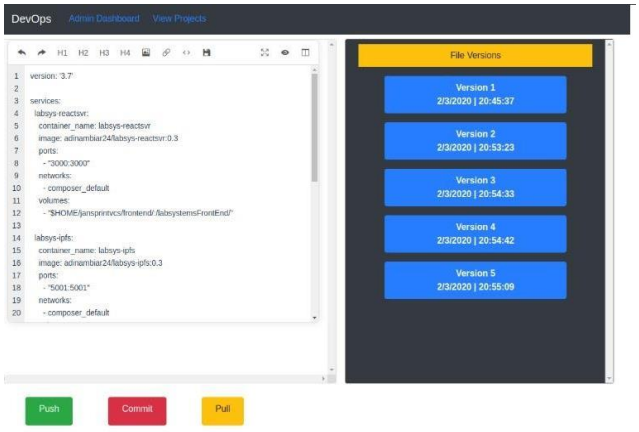

Fig. 4. File versions

The use of a git repository serves as a trigger for the build server to compile, run, and inspect the changed code. If code modifications fail in the automatic examination cases, an alert is triggered to inform the developers that their modification was unsuccessful. The developers may then either roll back their modifications to the previous productive version of the code or roll forward to incorporate a patch for the failing code. Continuous Integration/Continuous Deployment (CI/CD) is an agile development lifecycle that benefits big enterprises in comparison to the traditional waterfall development lifecycle, which requires weeks to months for the delivery phase after the integration and testing stages.

Figure 4 depicts the usage of the auditability and traceability of the batch of files a participant uploads. The architecture allows another participant (peer) to update a document of another peer since the fundamental feature of Blockchain is a distributed ledger that allows all participants to have access to it, perform transactions on it, and record on it.

VI. PERFORMANCE ANALYSIS

The proposed framework is implemented using Hyperledger Fabric, a Blockchain platform. Its performance is measured using the hyperledger calliper tool. Hyperledger Caliper is a blockchain benchmark tool that measures performance with a set of predefined uses. It measures performance with parameters like success rate, transaction throughput, transaction latency, and resource consumption. Throughput measures the flow rate of all transactions per second. Latency is the time taken for an issued transaction to complete. Resource consumption is the maximum and minimum memory, CPU resource consumption, as well as I/O traffic. They employed it to see the performance behavior of the proposed framework by giving a load of 100, 200, 300, and 400 transactions for open rounds and query transactions with three sub rounds with varied rate control.

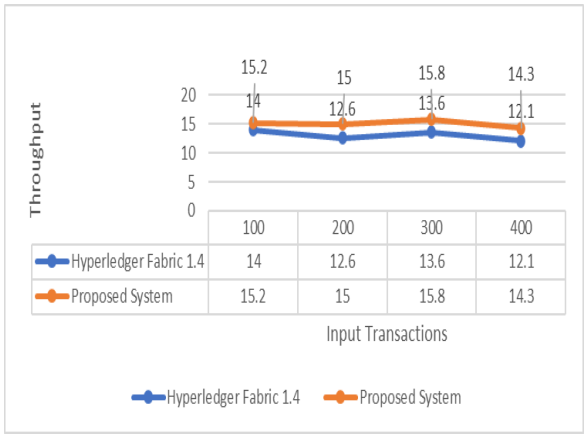

Fig. 5. Results of Proposed System in terms of Throughput

The performance metric used for comparison is Send Rate (TPS) Throughput (TPS) Latency (S). The Success Rate shows how many transactions out of all the submitted transactions have been successfully processed and written on the Blockchain. The graphical representation is shown in figures 5 and 6, in which the proposed system's performance is shown for throughput and latency, respectively. Then the proposed system performance is measured for latency and compared with Hyperledger Fabric 1.4 performance.

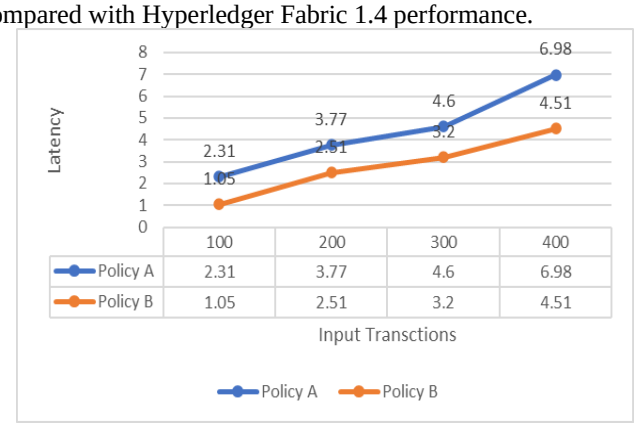

Fig. 6. Results of Proposed System in terms of Latency

The performance results indicate that the proposed system has better performance with adaptive block size and is

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 01, January-2022

optimized with endorsement policies when compared with Hyperledger Fabric 1.4. It has also been observed that the proposed framework performs well when tuned with respect to block size and arrival rate of transactions.

VII. CONCLUSION AND FUTURE WORK

The purpose of using Hyperledger Fabric in the proposed architecture was clearly to do justice to the corporate standards. This includes the organizational and disciplinary behaviors that are incorporated by the Hyperledger Fabric framework quite well. Its features enable one to utilize it as a base for development in the corporate environment. This also ensures that the organization can deal with emerging and trending technologies while remaining true to its core values. The practice of SDLC in an organization tends to help in deploying the outcome of a project faster by including the operations team with the development team, which is quite different than what is done traditionally in the real world. This also ensures the product quality and withstands most of the flaws and potential rectifications that must be made once the product passes various tests and eventually the customer acceptance test. The purpose of this architecture was to merely build the SDLC practices onto the Hyperledger Fabric platform such that organizations find it easier, better, and still maintain the same quality of the product as they would without any additional functionalities. The performance of the system is optimized at the validation and configuration levels and compared with the existing system. A comparison shows the proposed system is found to be efficient when modified with block size and endorsement policies.

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] Mohit Kumar Sharma , “A study of SDLC to develop well engineered software” , Inter- national Journal of Advanced Research in Computer Science, IEEE Trans. on Neural Networks, vol. 4, pp. 570-578, July 2017.
[2] Gaurav Kumar and Pradeep Kumar Bhatia,”Comparative Analysis of Software Engineering Models from Traditional to Modern Methodologies”, 2014,978-1-4799-4910-6/14 $31.00 IEEE pp 162-167
[3] Sheetal Sharma,Darothi Sarkar and Divya Gupta,”Agile Processes and Methodologies: A Conceptual Study” ,International Journal on Computer Science and Engineering, ISSN : 0975-3397. Vol. 4 No. 05 May 2012. 892.
[4] Kalpana Sureshchandra and Jagadish Shrinivas avadhani, “Moving from Waterfall to Agile”,2008, 97-101 ,978-0- 7695-3321-6 IEEE
[5] Bashir, I.“Mastering Blockchain”, 1st ed. Birmingham, UK.: PACKT Publishing, pp.10,12,16- 23,27-32,35-38,44,359,362-373.
[6] Hyperledger docs available at https://hyperledger.github.io/ composer/v0.19/ installing/ installing-index 2017
[7] Bass L, Weber I, Zhu L ,”DevOps: A Software Architect’s Perspective”, Addison-Wes- ley, Old Tappan, New Jersey,2015.
[8] Andreas M. Antonopoulos ,”Mining and Consensus. In: Mastering Bitcoin: Programming the Open Blockchain”, 1st edn, O'Reilly Media, Inc, California, United States. P 181,2014
[9] Logica Banica , Magdalena Radulescu , Doina Rosca and Alina Hagiu, “Is DevOps another Project Management Methodology?” , Informatica Economică vol. 21, no. 3/2017 39,2019
[10] Roche, J. “Adopting devops practices in quality assurance”. Communications of the ACM, 56, 11 , pp 38-43,2017.
[11] Z. Ahmed and S. C. Francis, "Integrating Security with DevSecOps: Techniques and Challenges," 2019 International Conference on Digitization (ICD), 2019, pp. 178-182.
[12] Dhillon, V., Metcalf, D., Hooper, M.,.”The hyperledger project. In: Blockchain Enabled Applications”. Springer, pp. 139-149,2017.
[13] M. Shahin, M. Ali Babar, and L. Zhu, “Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices”, IEEE Access.2019
[14] Lwakatare, Lucy Ellen , Kilamo, Terhi , Karvonen, Teemu ,Sauvola, Tanja , Heikkilä, Ville , Itkonen, Juha, Kuvaja, Pasi , Mikkonen, Tommi , Oivo, Markku , “DevOps in practice : A multiple case study of five companies”. Information and Software Technology. 2019 ; Vol. 114. pp. 217-230.
[15] N. Paez, “Versioning Strategy for DevOps Implementations” ,2018 Congreso Argentino de Ciencias de la Informática y Desarrollos de Investigación (CACIDI), 2018, pp. 1-6,
[16] M. B. Kamuto and J. J. Langerman, “Factors inhibiting the adoption of DevOps in large organisations”, South African context, 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017, pp. 48-51.
[17] Hyperledger Caliper: HyperledgerCaliper,[online-] https://hyperledger.github.io/caliper/v0.4.2/ getting-started/#architecture
[18] P. Agrawal and N. Rawat, Devops, “A New Approach To Cloud Development & Testing”, 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2019, pp. 1-4
[19] Rahman, Akond Ashfaque Ur and Laurie A. Williams,”Software Security in DevOps: Synthesizing Practitioners’ Perceptions and Practices”. 2016 IEEE/ACM International Workshop on Continuous Software Evolution and Delivery (CSED) (2016): 70-76.
[20] M. B. Kamuto and J. J. Langerman, "Factors inhibiting the adoption of DevOps in large organisations: South African context," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017, pp. 48-51