



# Automated Word App Using DevOps Technology

Mrs. K. Preethi Kumar<sup>1\*</sup>, Mahendran E<sup>2</sup>, Prakash S<sup>3</sup>, Poongkannan P<sup>4</sup>

<sup>1</sup>Assistant Professor, Department of Computer Science and Engineering, Government college of Engineering Srirangam, India.

<sup>2,3,4</sup>Student, Department of Computer Science and Engineering, Government college of Engineering Srirangam, India.

\*Corresponding author

DoI: <https://doi.org/10.5281/zenodo.7883225>

## Abstract

The Word App is a web application, which is publicly accessed by the people who have created accounts on it. The application gives twenty different types of relations for the single word input like definition, synonyms, frequency, and so on. The project's motto is to automate the deployment process using DevOps technology by wrapping up all the dependencies and runtime into a single container using Docker. Then, the application containers and database containers are deployed into the Kubernetes cluster specifically for managing applications and orchestrating containers. Also, Prometheus was deployed inside the Kubernetes cluster in order to get the metrics from the cluster and to monitor and set alerts for any malicious and unpredictable behaviour. Finally, the entire manual task was automated using Jenkins by fetching the source code from the GitHub repository. Our aim is to create a web application with achievements of automation, elasticity, reduced downtime and instant recovery using modern DevOps tools.

**Keywords:** DevOps, Version control, Git, GitHub, Docker, Kubernetes, Jenkins, Continuous Integration, Continuous Deployment, Prometheus, Alert manager, Grafana, DevSecOps

---

## **1. Introduction**

The blend of social methods of reasoning, practices, techniques, and instruments known as DevOps works on an association's ability to convey applications and administrations at high speed: Organizations that employ conventional software development and infrastructure management processes are slower to adapt and improve their products. This speed enables businesses to compete in more profitable market and provide superior customer service.

Improvement and task groups are no longer "siloed" in a DevOps architecture. These two teams can occasionally be united into a single unit where engineers work on all stages of the application lifecycle, from development and testing to deployment and operations, and get a broad range of abilities that aren't specific to any one pitch.

Quality assurance and security teams may also interact more closely with development and operations throughout the lifecycle of an application under various DevOps models. DevSecOps is a term used to describe a DevOps team where security is everyone's top priority. Formerly manual and slow processes are now automated by any of these teams using best practices. They employ a technological stack and infrastructure that enable them to operate and evolve apps dependably and swiftly. As a result, a team's velocity is increased.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 briefly explains the application development and deployment architecture together. Section 4 presents the proposed work of the application development and deployment. Section 5 presents the conclusion of the system. Finally, section 6 suggests the future enhancement of the current system.

---

## 2. Related Works

### 2.1 SDLC

The term “Software Development Life Cycle” (SDLC) refers to a conceptual model that is used to explain the process of planning, designing, testing, and delivering software. To handle SDLC, the industry uses a variety of approaches. The Agile methodology was developed in reaction to the shortcomings of rigid blueprint process models (such as Waterfall, RUP, etc.) and other resistant techniques. The Agile technique is a project management structure that is most commonly employed in software development.

There were a few businesses that used specific agile methods to reduce bottlenecks in the existing process, despite the fact that the majority of businesses adopted agile practices. Continuous Deployment was first developed by businesses like IBM, Facebook, and Microsoft.

### 2.2 DevOps Culture

“DevOps is a collection of design principles designed to enhance communication between development and operations. Along with addressing shared procedures and tools, Shared objectives and incentives may not always be feasible due to the inherent tensions between various organizations. Yet, they ought to be in close proximity to one another. The primary objective of DevOps is to locate and close gaps between the Development team and the Operations team. Professionals who work in software development frequently approach their work with the mindset that they are expected to bring about product transformation.

The following are the primary objectives of DevOps: 1. Provide quantifiable company value through ongoing, high-quality service provision. 2. Place a strong emphasis on simplicity and

agility across all domains, including technology, workflow, and human factors. 3. Remove obstacles between development and operations by fostering shared ownership and trust, fostering innovation, and promoting cooperation. 4. Control dynamic compliance because access and sharing laws are evolving.

### **2.3 CAMS Framework**

John Willis and Damon Edward, two leaders in DevOps, developed the CAMS Model. Its acronym stands for sharing, measurement, automation, and culture. Many DevOps professionals now utilize the CAMS model as their preferred model set of principles. These are the four core, mutually reinforcing values that should be incorporated into a DevOps deployment.

Culture is the most important entity, which needed to be set up prior to implementing DevOps in any environment, where teams work together and share the responsibility for the end users of their application. It not only encourages the adoption of agile practices in operations work but also allows developers to learn from real-world Ops experience and starts a mutual knowledge exchange that breaks down the walls between teams.

Automation is a key enabler for DevOps adoption. The right people, processes, and tools are needed to create an automation framework for DevOps. Providing automated mechanisms to monitor performance in all environments, from CI and test environments to the actual production deployment, allows the shared language of performance to be spoken.

Measurements such as business KPIs, system metrics, and application behavior have to be open, transparent, accessible, consumable meaningful, and able to be visualized in an ad hoc

manner by all constituents of the DevOps model. With performance aspects being covered in earlier testing stages, performance engineers on testing teams have time to focus on large-scale load tests in production-like environments.

Every organization must share its tools, findings, and lessons with others in order to succeed. Collaboration is based on the sharing of issues and solutions, just as DevOps is. Users should anticipate that openness and transparency are highly valued in the DevOps environment.

## **2.4 Software Quality**

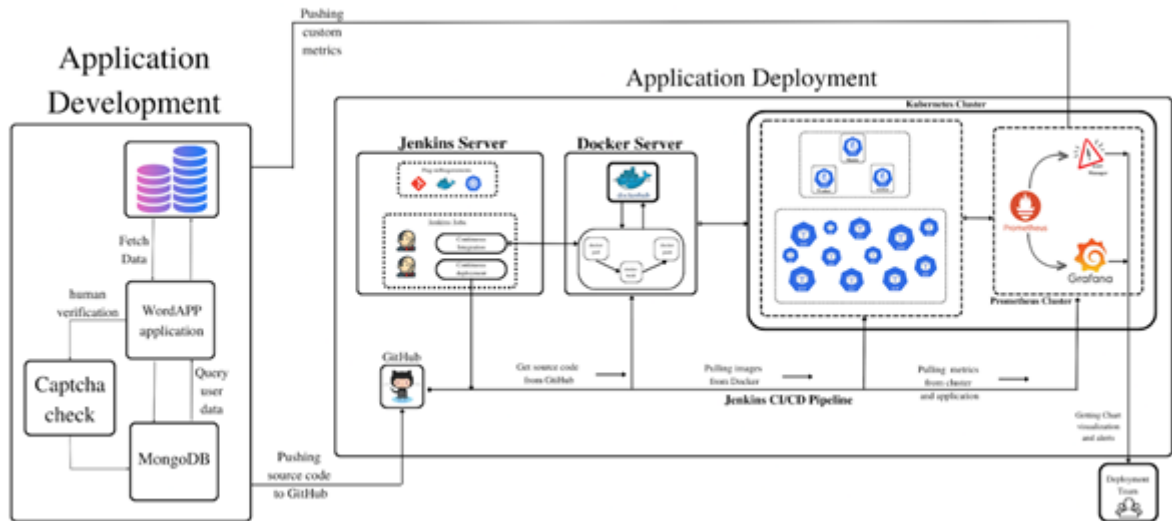
The most crucial element in determining customer satisfaction and company expansion is quality. Program quality characteristics are used to assess program quality. Software quality characteristics are a group of qualities used to define and rate the quality of software products [2].

An international standard for measuring program quality is ISO 9126. Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability are the six primary quality attributes listed by the ISO 9126 software quality paradigm. Twenty-seven sub-characteristics for both internal and exterior quality are divided into the main characteristics [2].

## **3. System Architecture**

Creating a web application using DevOps technology involves setting up a continuous integration and delivery pipeline that automates the software development and deployment process. This pipeline typically uses several tools such as Git, Docker, Kubernetes, Jenkins, and Prometheus. Developers use Git to collaborate on the project and manage code changes, while Docker enables them to package the application and dependencies into a portable

container. Kubernetes automates the deployment and management of these containers, while Jenkins automates the building, testing, and deployment of the application. Finally, Prometheus monitors the application's performance and availability.



**Figure.1.** System Architecture

## 4. Proposed Work

### 4.1 Application Development

Word Application is a great tool for discovering new and relatable words. With just a single word input, we can generate a list of 20 more words that are related to our initial search term. This feature can help you find synonyms, antonyms, and other related words that can help you express yourself more precisely.

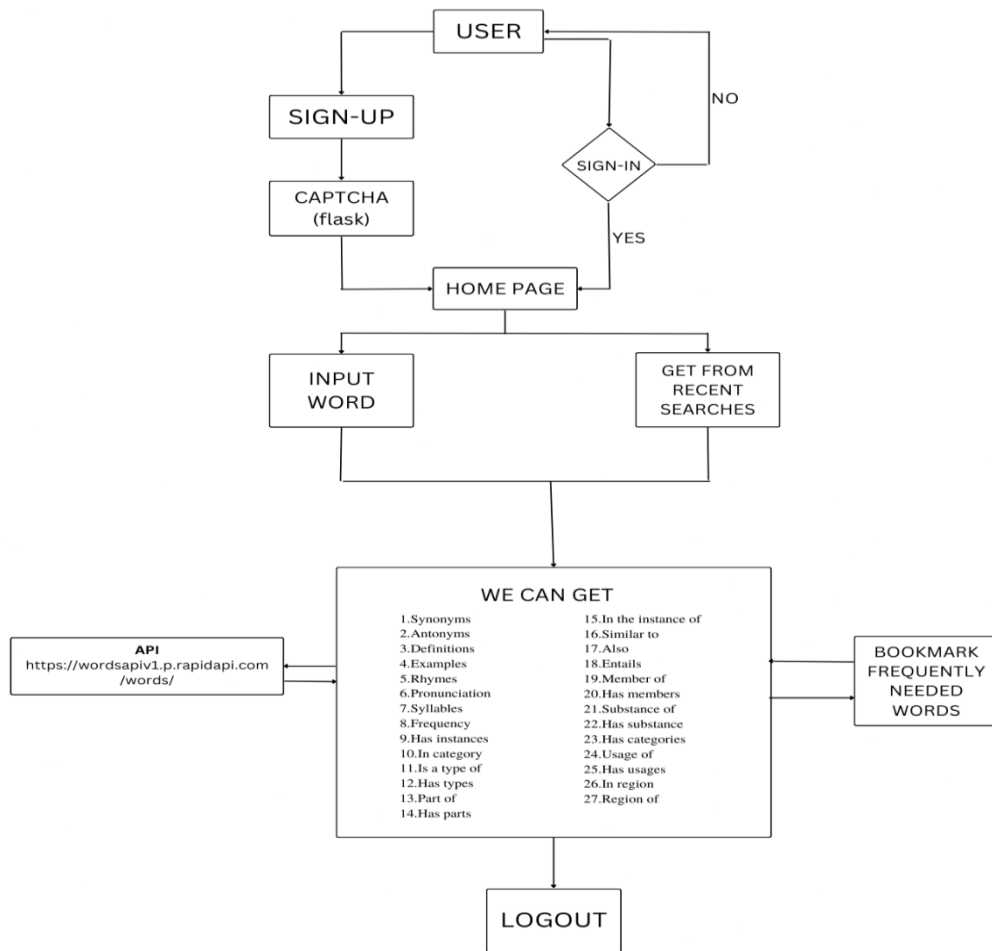
For example, let's say you want to find more words related to the term "happy." With Word Application, you can input "happy" and receive a list of 20 other words that are related to happiness, such as "joyful," "ecstatic," "delighted," "pleased," "content," and "gleeful".

In addition to helping you find new words to add to your vocabulary, Word Application can also be a useful tool for writers, students, and professionals who need to communicate

effectively in their work. By using the application to explore related words, you can find new and more precise ways to express your ideas and thoughts.

Bookmarking frequently needed words in the Word app can be a useful time-saving technique. By bookmarking the words or phrases that you use frequently, you can quickly find and insert them into your documents without having to type them out each time.

The history of words in the Word app can be a valuable tool for writers, students, and professionals who frequently work with text. By accessing the history of words, you can easily find and reuse words or phrases that you have previously typed in your document. This can save time and reduce errors, as you don't have to type out the same words repeatedly. The history of words can also help you improve your writing by allowing you to identify commonly used words or phrases that you may want to vary or replace.



**Figure.2.** Data flow diagram – Application Development

## 4.2 Setting up Jenkins Server

Jenkins is an open-source automation server that allows us to define and manage various stages of our software development process.. Before the Docker image is uploaded to Docker Hub, this pipeline may also include steps for testing and quality control [6]. The declarative configuration of Kubernetes can be used to launch the Docker image once it has been uploaded to Docker Hub. This configuration specifies the number of replicas, resource allotments, and environmental variables that should be used when deploying the Docker image. Additionally, Kubernetes includes features for handling containerized networking and storage, making it simple to set up deployed containers to cooperate with the rest of your infrastructure. It is possible to guarantee that the application is consistently updated and released in a dependable



and consistent manner by setting up a Jenkins server to automate the build and deployment process [3].

### **4.3 Dockerizing web application**

When developing software, it is important to have a reliable way to package and deploy our code. One way to do this is by using Docker, a platform that allows us to package the application and its dependencies into a portable container.

By wrapping up all of the application source code into a Docker container and pushing it to Docker Hub, we can ensure that our application is easily deployable and can be shared with others. Updating the version increment allows us to track changes and ensure that everyone is working with the latest version of the application [3].

### **4.4 Deploying containers into the Kubernetes cluster**

Deploying application and database containers into Kubernetes is a common practice for orchestrating containerized workloads. Kubernetes provides a platform for managing and scaling containerized applications and their dependencies, such as databases [1]. By deploying containers into Kubernetes, we can ensure that the applications and databases are highly available, scalable, and resilient. To deploy containers into Kubernetes, we first need to create container images and store them in a container registry. Once you have container images, we can use Kubernetes' declarative configuration to define how the containers should be deployed, including the number of replicas, resource limits, and environment variables. Kubernetes also provides built-in features for managing containerized storage and networking, making it easy to configure database containers to work with application containers. By deploying application

and database containers into Kubernetes, we can simplify our infrastructure management and ensure that the applications are running smoothly and reliably [5].

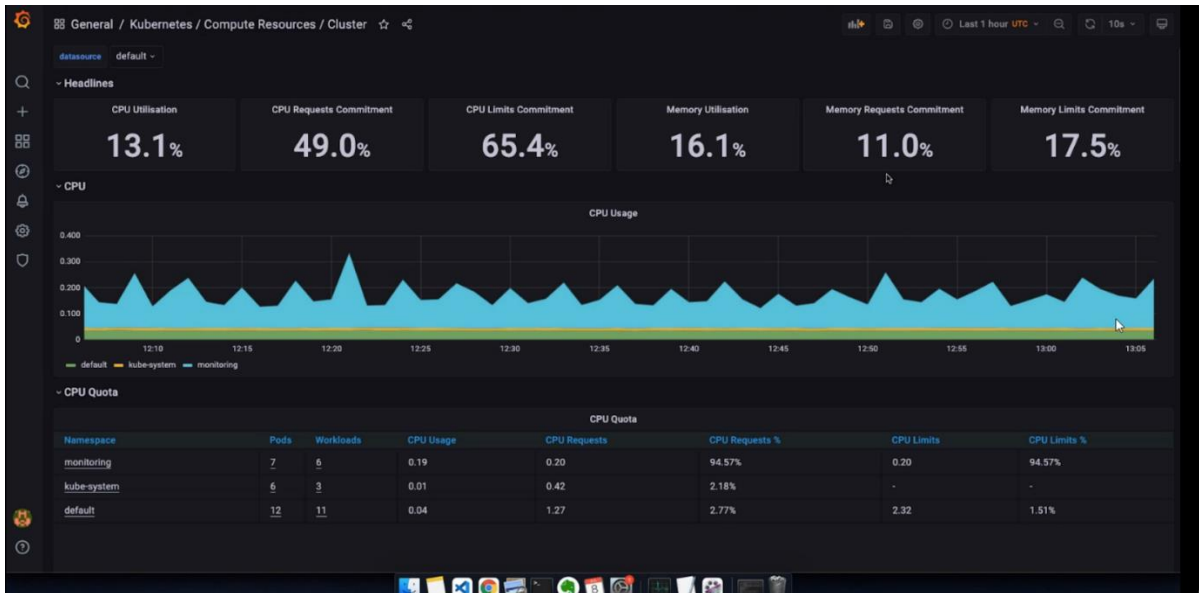
#### **4.5 Deploying Prometheus inside the Kubernetes cluster**

We can track the functionality and health of our containerized apps by deploying Prometheus inside a Kubernetes cluster. Prometheus is an open-source monitoring system that gathers and stores metrics from different sources, including applications, databases, and system resources. Using the official Prometheus Helm chart, the installation and configuration procedure becomes simpler when deploying Prometheus inside a Kubernetes cluster. Once Prometheus is installed, we can set up the applications to use either a Prometheus exporter or a client library to send metrics to Prometheus. Application-specific measures, such as request latency, error rate, and throughput, can be a part of these metrics.

#### **4.6 Visualizing metrics in Grafana**

Metrics from the application that are pushed to Prometheus and displayed in a Grafana interface can offer important information about the functionality and health of our application. Prometheus is a free and open-source monitoring tool that gathers and stores metrics from different sources, including software applications. We can use a Prometheus exporter or a client library to send metrics to Prometheus from our program. We can specify unique metrics and gather information from the application using these tools, including request latency, error rates, and throughput. After the metrics are forwarded to Prometheus, we can use Grafana to design unique displays and visualizations that present the metrics in a user-friendly manner. Grafana provides a range of visualization options, including graphs, tables, and heat maps, which can help us identify trends and anomalies in our data. We can also set up alerts in Grafana to notify us when specific metrics reach a certain threshold or when unusual behavior is detected. By

pushing metrics from our application to Prometheus and visualizing them in Grafana, we can gain valuable insights into the performance and health of our application, and take proactive steps to improve its reliability and availability.

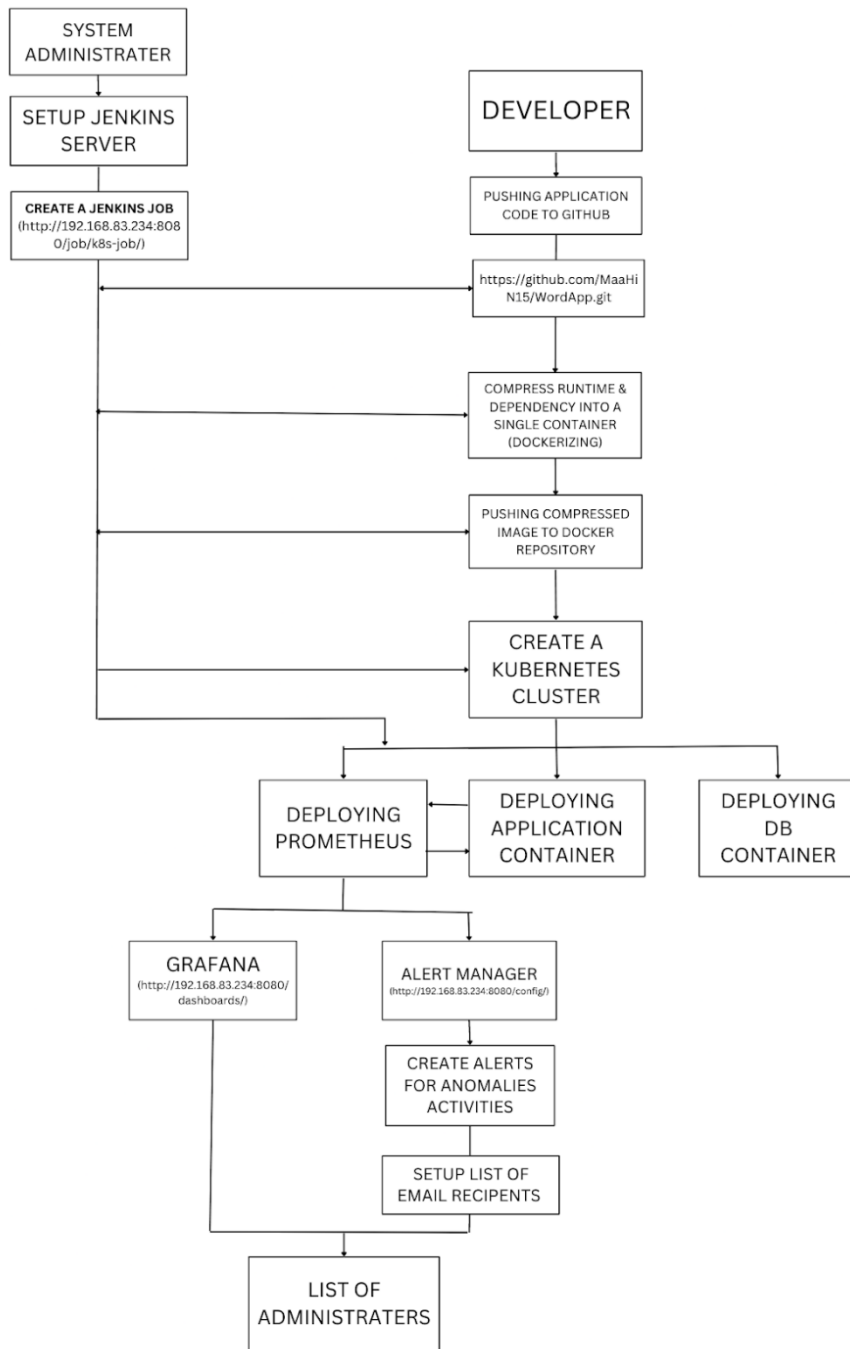


**Figure.3.** Visualizing metrics in Grafana

#### 4.7 Alerts set up using Prometheus Alert Manager

An open-source alerting system called Prometheus Alert manager enables you to create and manage alerts based on Prometheus metrics. In Alert manager, we can define alerting rules that describe the circumstances that cause an alert to be generated. For instance, we can configure a warning to let you know when our application's error rate goes above a predetermined limit. When an alarm is raised, the Alert manager can notify users via email, PagerDuty, and Slack, among other integrations. We can set up the Alert manager webhook integration to send notifications to a particular Slack channel if we want to receive them there. With this integration, we can take an instant action to resolve the problem by receiving real-time notifications in Slack whenever an alert is triggered. By setting alerts in Prometheus Alert

manager and receiving them in a Slack channel, we can proactively monitor our application and infrastructure, and ensure that any issues are quickly identified and resolved.



**Figure.4.** Data flow diagram – Application Deployment

---

## 5. Conclusion

The proposed system is to create a web application and automate its deployment process with DevOps tools such as Git, GitHub, Docker, Kubernetes, Jenkins, and Prometheus is an effective approach to software development. These tools allow developers to collaborate on code, manage version control, and streamline the deployment process. Git and GitHub provide a reliable and efficient version control system that allows multiple developers to work on a code base simultaneously. Docker enables developers to package their applications into containers, making them portable and easily deployable. Kubernetes provides a powerful orchestration platform for managing containerized applications and scaling them as needed. Jenkins is a popular automation server that can automate the entire software delivery process, from building and testing to deployment and monitoring. Finally, Prometheus provides a powerful monitoring and alerting system that allows developers to track the performance and health of their applications in real time. Overall, utilizing these DevOps tools enables developers to streamline their development process and increase the efficiency and reliability of their application deployment.

## 6. Future Enhancement

As technology evolves, there are always opportunities for improving the current processes and methodologies used in web applications using DevOps technologies. Here are some potential future enhancements:

**Implement Infrastructure as Code:** Infrastructure as Code (IaC) is a methodology that enables the automation of infrastructure provisioning and configuration using code. This approach can bring more consistency and reproducibility to the infrastructure and reduce the time and effort required for infrastructure management.

**Implement ChatOps:** ChatOps is a methodology that integrates chatbots and collaboration tools into the DevOps workflow, enabling teams to perform various operations and tasks through chat interfaces. Implementing ChatOps can improve communication and collaboration among the development, operations, and other teams involved in the Word App deployment.

**Use AI for Continuous Improvement:** Artificial Intelligence (AI) and Machine Learning (ML) can be used to analyze the Word App performance metrics and logs and identify patterns and anomalies. This can help teams to proactively detect and resolve issues and optimize the Word App performance.

**Integrate Security into DevOps:** Security is a critical aspect of any software development and deployment process. Integrating security into the DevOps workflow can help teams to identify and remediate security vulnerabilities and compliance risks earlier in the development lifecycle.

#### REFERENCES

- [1]. Cukier, Daniel. "DevOps patterns to scale web applications using cloud services." Proceedings of the 2013 companion publication for conference on Systems, programming, & applications: software for humanity. ACM, 2013.
- [2]. Roche, James. "Adopting DevOps practices in quality assurance." Communications of the ACM 56.11 (2013): 38-43.
- [3]. Humble, Jez, and David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010.
- [4]. Elliot, Stephen. "DevOps and the cost of downtime: Fortune 1000 best practice metrics quantified." International Data Corporation (IDC) (2014).
- [5]. Bang, Soon K., et al. "A grounded theory analysis of modern web applications: knowledge, skills, and abilities for DevOps." Proceedings of the 2nd annual conference on Research in information technology. ACM, 2013.
- [6]. Schaefer, Andreas, Marc Reichenbach, and Dietmar Fey. "Continuous integration and automation for DevOps." IAENG Transactions on Engineering Technologies. Springer Netherlands, 2013. 345-358.
- [7]. Sacks, Matthew. "Devops principles for successful web sites." Pro Website Development and Operations. Apress, 2012. 1-14.