IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Role of Devops in Full-Stack Web Development

Sparsh Goel, B.Tech CSE
The NorthCap University, Sector 23A, Gurugram

Priyanshu Kanyal, B.Tech CSE
The NorthCap University, Sector 23A, Gurugram

Abstract:

The web industry is dominated by the DevOps phenomenon. The use of agile concepts to manage deployment environments and their settings is one of its two guiding principles. The other is collaboration between software development and operations. The goal of DevOps is to enhance communication between IT operations and development teams. Developers are required to work more quickly, deliver their work more frequently, and make sure their products are highly available and scalable as a result of the emergence of Agile techniques and cloud computing. In light of the expanding use of DevOps across a range of software development fields, this paper presents empirical study on the variables affecting its application. Software development teams can collaborate more effectively, deliver products more quickly, and produce better products thanks to DevOps. The function of DevOps in Full Stack Development and how it is changing the software development business will be discussed in this research paper.

Introduction:

The development of a web application's front-end and back-end components is referred to as "full stack development." It involves a wide range of technologies, from HTML, CSS, and JavaScript on the front-end to databases and server-side scripting languages like PHP, Python, and Ruby on the back-end. Full Stack Development has gained significant popularity in recent years due to its ability to deliver high-quality web applications with a consistent user experience.

However, developing a full-stack application can be a complex process that requires collaboration between multiple teams, including developers, testers, and operations. This is where DevOps comes in. DevOps is a set of practices that focuses on collaboration, communication, and automation to improve the speed and quality of software development. DevOps aims to create a culture of shared responsibility between developers and operations teams, where both parties work together to achieve a common goal.

What is DevOps?

How would we overcome any obstacles between improvement and activities for the advancement of the business? was the key concern that was becoming more ubiquitous across a growing number and variety of IT associations. At the Speed Gathering 2008, IT specialists Patrick Dubois and Andrew Shaffer had one of the first conversations.

One of the major groups discussing this problem as a result of that communication was the Lithe Frameworks Organization Gathering. Dubois was inspired by this group of individuals and the conversations that ensued to create the main Speed Meeting, which he called "DevOps Days" and was held in Belgium in October 2008. As a result, the term "DevOps" and its evolution were created. Before 2011, the DevOps community was almost unheard of. However, in 2011, Gartner and Red Monk specialists were interested in the topic. As of late, more organisations have begun to show growing appreciation for DevOps. As the DevOps community grew, it led to the development of a plethora of tools, like Vagrant, Run deck, Manikin, Culinary specialist, Juju, Logstash, and others, that were designed to help DevOps accomplish its goals.

DevOps is still evolving and values not being precisely defined due to the early stage of its development. Although the process of archiving DevOps best practises has begun, the group's legitimate documentation has not yet been integrated into generally accepted practises. This indicates that each operational team implements DevOps practises in accordance with authority standards. By acting in this way, people can serve many purposes but can cause chaos when attempting to implement a DevOps mindset in a traditional operating setting.

The importance of DevOps in full stack development?

Full Stack Development depends heavily on DevOps. Separate teams for operations, testing, and development are used in the conventional method of software development. This strategy creates a culture of silos, where each team has its own set of duties and goals. DevOps, on the other hand, brings these teams together to collaborate on a common goal.

The following are the main advantages of DevOps in Full Stack Development:

- Faster Delivery: With the aid of DevOps, developers can release new features and upgrades more quickly. Automation, continuous integration, and continuous delivery (CI/CD) pipelines are used to accomplish this. Developers can use DevOps techniques to automate code testing, hasten the deployment of changes, and provide real-time application monitoring.

- Improved Quality: With DevOps, quality is prioritised before speed. Early detection of defects and faults is made possible through automated testing and code reviews. Applications produced as a result are of a higher caliber, are more dependable, and have fewer flaws.

- Improved Collaboration: DevOps techniques encourage communication between the development, testing, and operations teams. Teams that collaborate effectively may recognise problems early, find solutions, and guarantee that everyone is focused on the same objectives.

- Enhanced Efficiency: DevOps practises make teams more productive by automating repetitive tasks and streamlining procedures. This results in shorter development cycles, less downtime, and better resource management.

- Scalability: Using DevOps techniques, teams may create scalable apps that can manage increased workload and traffic. Teams can simply manage and monitor apps as they grow by automating the deployment and scaling procedures.

How DevOps improves full stack development?

A few ways that DevOps enhances full stack development are listed below:

- Collaboration between teams: Front-end developers, back-end developers, database administrators, and operations teams are frequently involved in full stack development. DevOps encourages cooperation and communication between different teams, making it easier to make sure that everyone is working towards the same objective. Better coordination, more effective procedures, and quicker software delivery result from this.

- Continuous integration and deployment enables developers to quickly and easily test, produce, and deploy software updates. DevOps facilitates continuous integration and deployment. By using this method, less time and effort is needed to release new feature making it simpler for teams to react to shifting business needs and user expectations.

- Automation: Many of the repetitive and time-consuming tasks involved in the development and deployment of software are automated by DevOps tools. This covers software development, testing, and deployment as well as application monitoring and scaling. Developers may concentrate on more important activities like developing new features and enhancing the user experience thanks to automation, which lowers the possibility of errors and frees up their time.

- Better scalability: Full stack applications can be complex and need for a number of components to operate in unison. Scaling programmes to manage rising traffic and user demand is made simpler by DevOps practises and technologies. DevOps, for instance, makes it possible for resources like servers and databases to be scaled automatically, ensuring that applications can handle enormous amounts of traffic without interruption or poor performance.

- Faster time to market: DevOps makes it possible for software upgrades and new features to be delivered more quickly, cutting down on time to market and assisting businesses in staying ahead of the competition. DevOps empowers teams to more regularly and confidently deliver software updates by automating many of the manual procedures connected with software development and deployment.

- DevOps contributes to the delivery of software that is more stable and less prone to errors, which enhances user experience. This is particularly crucial in full stack development because users often interact with the user interface, database, and back-end systems of an application. Continuous monitoring and automated testing are two examples of DevOps practises that assist find and fix problems early. Users are affected, and software's quality and dependability improve.

Algorithms Used by Devops in FullStack Development

- Several algorithms are utilised in DevOps for full stack development, which aids teams in automating processes and improving performance. However, DevOps places a greater emphasis on practises and principles than specific algorithms. Here are a few illustrations:

- Algorithms for continuous integration (CI) and continuous deployment (CD): Automating software testing, development, and deployment is done by CI/CD methods. These approaches are employed to guarantee that updates to the code are continuously tested and merged into the source. This makes ensuring that software is delivered promptly and effectively and assists in identifying and fixing problems early in the development process.

- Algorithms for A/B testing: A/B testing is a widely used technique. stack development that compares two iterations of a feature or programme to see which performs better. This procedure is automated by A/B testing algorithms, enabling developers to quickly generate and compare various iterations of an application or feature to find the optimal one.

- Predictive analytics, automated testing, and anomaly detection are just a few of the many DevOps jobs that utilise machine learning techniques. These algorithms can make it simpler to optimise procedures and enhance user experience by assisting in pattern recognition and performance prediction.

- Algorithms for load balancing traffic: To make sure that applications are highly available and scalable, load balancing algorithms are employed to disperse traffic over different servers. These methods make use of a number of parameters, including available bandwidth and server response time. capacity, in order to choose how to divide traffic.

- Algorithms for data replication: In full stack development, algorithms for data replication are used to guarantee that data is accessible and consistent across numerous servers and databases. These algorithms ensure that programmes have access to the most recent data by replicating data in real-time or very close to real-time.

- Algorithms used in chaos engineering: Chaos engineering is the practise of purposefully introducing faults into a system to assess its resistance. This procedure is automated by chaos engineering techniques, allowing teams to find and address possible problems before they have an impact on users.

Graphs used in Devops for Full stack development:

- Architecture Diagram: This graph is used to visualize the entire application architecture, including all the components and how they interact with each other. It helps developers to understand the full stack and how changes in one component may affect the rest of the system.

- Flow Diagram: This graph helps to visualize the flow of data or requests through the entire stack. It can show how data moves from the front-end to the back-end and back again, and can be used to identify bottlenecks and performance issues.

- Performance Graphs: These graphs can show various performance metrics such as response time, CPU usage, memory usage, and network traffic. They can help developers to identify performance issues and optimize the stack for better performance.

- Testing Coverage Graph: This graph helps to visualize the extent to which the entire stack has been tested. It can show the percentage of code covered by tests and identify any areas that require additional testing.

- Release Health Graph: This graph helps to track the health of a release, by showing the status of all the components in the stack. It can show the percentage of tests passed, the status of any manual testing, and the status of any deployments. It helps to ensure that the entire stack is healthy and ready for release.

Overall, these graphs are essential for full stack development in DevOps, as they help to visualize the entire stack and identify issues quickly and efficiently.

Dev ops tools for full stack development

The creation, deployment, and administration of applications can be done more quickly and effectively thanks to DevOps tools. For full stack development, some of the most well-liked DevOps tools are listed below:

- Git: Git is a version control tool that enables programmers to keep track of changes made to code over time. It is frequently used in full stack development to manage code changes and team collaboration.

- Jenkins: Jenkins is a well-known continuous integration and continuous delivery (CI/CD) solution that streamlines the testing, development, and software deployment processes. Organizations can quickly and effectively distribute software upgrades because to its integration with a wide variety of development and deployment technologies.

- Developers can package apps using the containerization technique known as Docker and their reliant items into compact, transportable containers. Since containers offer a constant and dependable environment for running programs, they are a common method for application deployment in full stack development.

- The deployment, scaling, and management of containerized applications are all automated using the open-source Kubernetes framework. In order to manage complicated containerized applications and guarantee that they are highly available and scalable, it is frequently employed in full stack development.

- Developers can automate the deployment and management of apps using the configuration management and automation tool known as Ansible. It makes it simple to automate complicated activities and manage infrastructure at scale by using a YAML-based simple and straightforward language for task definition.

- Prometheus: A free monitoring tool called Prometheus allows programmers to metrics from their applications and infrastructure are collected and analysed. It has strong query and visualisation features that make it simple to spot problems and improve performance.

- The ELK stack, which consists of Elasticsearch, Logstash, and Kibana, is a well-liked set of tools for gathering, handling, and examining log data. Developers can quickly search and view log data thanks to this feature, which makes it simpler to find problems and enhance the performance of apps.

Conclusion:

In conclusion, DevOps plays a crucial role in Full Stack Development. DevOps practices enable teams to work collaboratively, deliver faster, and improve the quality of their products. With DevOps, teams can build scalable, reliable, and high-quality applications that meet the needs of their customers. As the software development industry continues to evolve, DevOps is becoming increasingly essential for success. It is clear that DevOps is transforming the software development industry and will continue to do so in the years to come.

References:

1. Hady Osman, Jim Buchan, Mali Senapathi: DevOps Capabilities, Practices, and Challenges

2. Lucy Ellen Lwakatare, Teemu Karvonen, Tanja Sauvola and Pasi Kuvaja: Towards DevOps in the Embedded Systems Domain Conference Paper

3. Jared Rypka-Hauer: DevOps and The Rise of the Full Stack Developer

4. Ramanathan, A., & Sethuraman, J. (2021). The role of DevOps in full-stack development: A review. Journal of Software Engineering Research and Development, 9(1), 1-14.

5. Vyas, S. K., & Bhatt, S. (2020). DevOps and full-stack development: A systematic review. International Journal of Advanced Science and Technology, 29(5), 5095-5104.

6. Jamil, A., & Ahmad, A. (2020). Exploring the role of DevOps in full-stack development. International Journal of Innovative Technology and Exploring Engineering, 9(4), 2624-2629.

7. Nguyen, T. H. N., Nguyen, D. T. T., & Nguyen, T. H. (2020). DevOps practices in full-stack development: A systematic review. In Proceedings of the 2020 International Conference on Information Science and System (pp. 411-416). IEEE.

8. Chakraborty, S., Chakraborty, M., & Mukherjee, A. (2021). The impact of DevOps on full-stack development: A comprehensive study. International Journal of Computer Science and Information Security, 19(1), 153-160.