International Journal of Engineering Trends and Applications (IJETA) – Volume 13 Issue 3, May-Jun 2026

RESEARCH ARTICLE OPEN ACCESS

DevSync: Continuous Integration and Delivery for Scalable Web Applications

Mohammed Afshan Khan*, Ratna Ram Tanwar**

*B.Tech Student, Department of Computer Science & Engineering, Global Institute of Technology, Jaipur
**Assistant Professor, Department of Computer Science & Engineering, Global Institute of Technology, Jaipur

ABSTRACT
This Today, most web applications need to be updated and deployed quickly while staying stable and easy to scale. Managing all this manually can be time-consuming and risky. DevSync is a project that helps make this process easier through automation. It uses Docker to create lightweight containers for different parts of the application, Nginx to manage incoming traffic and balance loads, and PM2 to monitor and restart processes automatically on AlmaLinux servers. Unlike other systems that depend on paid tools or complex setup steps, DevSync is completely free and simple to use. It automatically handles building, testing, and deploying applications, which reduces errors and saves developers a lot of time. It can also adjust to different workloads by scaling the application when needed, ensuring that everything runs smoothly even during high traffic. This research shows how combining open-source DevOps tools can create a fast, reliable, and efficient deployment system. DevSync makes continuous integration and delivery easier for developers and can be used in many environments from college projects to startups and larger companies.
Keywords — Continuous Integration; Continuous Delivery; Docker; Nginx; PM2; AlmaLinux; DevOps; Automation; Web Scalability.

I. INTRODUCTION

In today's world, almost every organization and business depends on web applications. From online shopping to education and entertainment, web apps play a major role in daily life. As these applications grow, it becomes difficult for developers to manage frequent updates, bug fixes, and new features manually. Every time a developer makes a small change, the application needs to be built, tested, and deployed again. Doing all this manually takes a lot of time and can lead to mistakes. That's where the idea of Continuous Integration and Continuous Delivery (CI/CD) comes in[5]. CI/CD helps developers automate this process so that the application can be updated quickly and safely. The main goal of my project, DevSync, is to build a system that makes this process easier using open-source tools.

DevSync is designed to automate the development and deployment cycle of web applications. It uses Docker for containerization, which means the whole application can run in a separate and consistent environment no matter which system it's deployed on. This avoids the common "it works on my machine" problem faced by developers. The project also uses Nginx as a reverse proxy and load balancer to manage traffic efficiently, and PM2 to handle process management making sure the application runs continuously, even if one process crashes. The system runs on AlmaLinux, which is a stable and secure Linux-based operating system often used for production environments. One of the main reasons for developing DevSync was to reduce the amount of manual work developers do during deployment. Many teams, especially in smaller organizations or academic projects, cannot afford paid CI/CD platforms or cloud services. DevSync provides a simple, free, and efficient alternative that automates most of the routine tasks. It ensures that the latest code can be deployed smoothly without affecting the live application, saving time and reducing human error. The project also focuses on scalability, meaning it can handle increased traffic or new application instances easily without reconfiguring the entire system.

This project not only helped in understanding how CI/CD works but also provided practical experience in configuring servers, managing containers, and automating deployment pipelines. While working on DevSync, I learned how different tools like Docker, Nginx, and PM2 can work together to create a complete DevOps environment. The system can be used by students, startups, or even professional developers who want a lightweight, automated solution for continuous deployment. In conclusion, DevSync aims to make web application deployment simple, reliable, and scalable. It combines several powerful open-source technologies to build an automated workflow that saves time, increases productivity, and ensures better application performance. The project shows that with the right approach and tools, even complex deployment processes can be handled efficiently without expensive

setups or advanced technical knowledge.

II. LITERATURE REVIEW

Earlier, most web applications were deployed manually,which required uploading files to servers and restarting them after every update . This approach often caused errors, downtime, and inconsistencies between environments. To solve these issues, tools like Git, GitHub, and GitLab were introduced for version control, followed by automation tools such as Jenkins, Travis CI, and GitHub Actions for Continuous Integration and Continuous Delivery (CI/CD)[8][11]. These tools helped automate testing and deployment but were often complex to set up or required paid subscriptions for advanced features. With the rise of Docker, developers started using containerization to create consistent environments for applications. Nginx became popular for managing web traffic, and PM2 helped in monitoring and maintaining Node.js applications. However, using all these tools together still needed advanced DevOps knowledge and time-consuming setup. The DevSync project aims to simplify this process by integrating these open-source tools Docker, Nginx, PM2, and AlmaLinux into a single, automated CI/CD system. It eliminates the need for costly platforms and makes deployment faster, easier, and more reliable, especially for students and small-scale developers who want to manage scalable web applications efficiently[11][14][13].

III. PROBLEM DEFINITION

In web application development, one of the biggest challenges developers face is managing frequent updates and deployments without breaking the system or wasting time on repetitive manual work[6]. Most small development teams and students still use manual methods like uploading files through FTP or restarting servers after every change. This process is slow, error-prone, and not suitable for applications that require regular updates or need to handle multiple users at once.[12]

Although there are many CI/CD tools available, most of them are either paid, complicated to configure, or require advanced technical knowledge. For small projects or educational environments[[2], setting up tools like Jenkins, GitLab CI, or AWS CodePipeline can be overwhelming and time-consuming. As a result, developers struggle to maintain consistent environments and smooth deployment processes[7].

The main problem this project addresses is the lack of a simple, cost-free, and automated deployment system that integrates easily with existing technologies. DevSync solves this by combining Docker, Nginx, PM2, and AlmaLinux to create an efficient Continuous Integration and Delivery pipeline that automates builds, testing, and deployment without expensive tools or complex setup.

IV. METHODOLOGY

The DevSync project is built by integrating multiple open-source tools to create an automated Continuous Integration and Delivery (CI/CD) environment. The system follows a step-by-step approach, starting from code development to final deployment on the server. The main goal is to make the process simple, fast, and reliable without depending on paid cloud platforms. First, the web application code is managed through Git for version control. Whenever a developer pushes new changes, Docker creates containerized environments to ensure the application runs consistently on any system. These containers package the code along with all necessary dependencies. Nginx is configured as a reverse proxy and load balancer to handle incoming requests and distribute traffic efficiently. To manage and monitor the running processes, PM2 is used. It keeps the application live, restarts it automatically if it crashes, and provides logs for better tracking. The system is hosted on AlmaLinux, which offers a stable and secure environment. By combining these tools, DevSync achieves smooth automation for building, testing, and deploying web applications. This approach saves time, reduces manual errors, and ensures that developers can focus more on coding rather than deployment issues.

V. RESULTS AND DISCUSSION

After developing and configuring DevSync, the system was tested by deploying a simple HTML web application on AlmaLinux using Docker and Nginx. The setup worked smoothly, showing how even basic web projects can benefit from automation. The HTML code was containerized using Docker, ensuring that it ran in a clean and consistent environment. Nginx acted as a web server and reverse proxy to serve the HTML page efficiently, while PM2 monitored the process to keep it active and automatically restart[11] it if needed. The deployment process became much faster compared to manual methods. Each time a change was pushed to the repository, the container automatically rebuilt, and the updated HTML[3] page was served instantly. There were no manual uploads or restarts required. This automation reduced both time and human error. The testing also proved that DevSync can handle

lightweight projects effectively and can easily be scaled for larger applications. The combination of Docker, Nginx, and PM2 on AlmaLinux provided a stable, reliable, and cost-free environment, making DevSync an excellent CI/CD solution for students and small teams[1][11][8][7][5].

VI. FUTURE ENHANCEMENTS

While DevSync currently provides a stable and automated CI/CD system for deploying web applications using Docker, Nginx, PM2, and AlmaLinux, there are several areas where the project can be improved in the future. One major enhancement is the integration of database services such as MySQL or MongoDB. This would allow developers to manage both frontend and backend components within the same automated pipeline. Automating database migrations and backups could further increase reliability and data safety. Another future improvement is the inclusion of automated testing and code quality checks before deployment. By integrating tools like Jest or Selenium, DevSync could automatically run tests whenever new code is pushed, ensuring that only error-free applications are deployed. This would make the system more professional and suitable for larger development teams[12][14][15]. Adding a real-time monitoring and analytics dashboard is another enhancement that can make DevSync more user-friendly. Such a dashboard could display system performance, container health, and deployment status in real time, helping developers identify issues quickly. In the long term, DevSync could also be expanded to support multiple programming environments such as Python, PHP, and Java applications, making it a more universal CI/CD solution. Finally, improving security features—like SSL integration, user authentication, and role-based access—would make the system safer for production use.Overall, these future enhancements would transform DevSync from a lightweight student project into a complete, scalable, and professional CI/CD automation platform suitable for real-world deployment scenarios. [9][6][5][4]

VII. CONCLUSION

The DevSync project successfully demonstrates how different open-source tools can work together to create a smooth, automated, and reliable Continuous Integration and Delivery (CI/CD) system for modern web applications. The main goal of this project was to make the process of deployment faster, easier, and more consistent without depending on expensive or complex cloud-based tools. By using Docker, Nginx, PM2, and AlmaLinux,[9][11][4][2] DevSync provides a strong foundation for developers and students who want to understand and apply DevOps concepts in real-world scenarios. Throughout the development and testing phases, DevSync proved to be simple yet effective. A basic HTML web application was deployed using Docker containers on AlmaLinux servers, with Nginx serving as the reverse proxy[2][1] and PM2 handling process management. The system ran smoothly and required very little manual intervention. Each time changes were made to the application, the container was automatically rebuilt and redeployed, showing how easily DevSync can maintain consistency and reliability. This automation reduced both time and human error, two common challenges in traditional deployment processes[2][1][5].

The project not only highlights the advantages of automation but also emphasizes how open-source tools can replace costly enterprise solutions. For students and small teams, DevSync provides an opportunity to learn how professional CI/CD pipelines operate while using free and accessible technologies. It also shows how automation can lead to better resource management, improved stability, and faster delivery cycles in web application development[7][8].

Another important takeaway from this project is the understanding of how different tools complement each other. Docker provides isolation and consistency; Nginx ensures performance and traffic control; PM2 maintains process stability; and AlmaLinux offers a secure and reliable server environment. Together, these components form a balanced and efficient workflow suitable for both small-scale and production-level applications.

In the future, DevSync can be further enhanced with more advanced features like database integration, automated testing, continuous monitoring, and better security mechanisms. These improvements would make it even more robust and ready for professional use. To conclude, DevSync is more than just a college project it’s a practical demonstration of how DevOps automation can simplify real-world deployment challenges. It helps developers save time, reduce effort, and achieve better results with fewer resources. The project stands as a cost-effective, educational, and scalable approach that bridges the gap between learning and real-world application in the field of web development and DevOps.

REFERENCES

[1] Docker Documentation. "Get Started with Docker." Available at: https://docs.docker.com/get-started/

[2] Nginx Documentation. "NGINX Reverse Proxy Configuration Guide." Available at: https://nginx.org/en/docs/

[3] PM2 Official Documentation. "Advanced Node.js Process Manager." Available at: https://pm2.keymetrics.io/

[4] AlmaLinux Foundation. "About AlmaLinux OS." Available at: https://almalinux.org/

[5] GitHub Guides. "Understanding GitHub Actions for Automation." Available at: https://docs.github.com/

[6] Jenkins User Guide. "Building a Continuous Integration Pipeline." Available at: https://www.jenkins.io/doc/

[7] Red Hat Developer Blog. Containerization and Its Impact on Modern Deployment.

[8] Medium Article by DevOps Engineer. "Using Nginx and Docker Together for Web Deployment.

[9] DigitalOcean Tutorials. "How to Deploy Web Applications with Docker Compose." Available: at:https://www.digitalocean.com/community/tutorials

[10] FreeCodeCamp. "Introduction to CI/CD Concepts for Beginners." Available at: https://www.freecodecamp.org/news/

[11] Stack Overflow Discussion. "Best Practices for Node.js Deployment with PM2.

[12] Linux Journal. "AlmaLinux as a Reliable Server OS for Developers.

[13] DevOps.com. "Benefits of Automation in Software Deployment.

[14] TechTarget. "Understanding Continuous Integration and Delivery Pipelines.

[15] YouTube Tutorial. "Complete Setup of Docker, Nginx, and PM2 on AlmaLinux." Watched and followed for project implementation,

[16] R. Ajmera and D. Dharamdasani, "Comparative study of existing food delivery app," Global Research Journal, pp. 454–463, 2022.

[17] A. Kalwar and R. Ajmera, "ARQI: Model for developing web application," International Journal on Technical and Physical Problems of Engineering (IJTPE), vol. 13, no. 47, pp. 7–13, Jun. 2021.

[18] A. Kalwar, R. Ajmera, and C. S. Lamba, "An empirical study in small firms for web application development and proposed new parameters," International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 4, Feb. 2019.

[19] R. Ajmera, A. Kalwar, and C. S. Lamba, "A modern study on progressions and issues of web applications development in small firms," International Journal of Scientific Research in Science and Technology, vol. 3, no. 8, Nov.–Dec. 2017.

[20] P. Jha, M. Mathur, A. Purohit, A. Joshi, and A. Johari, "LibUno: A React-Based Digital Platform for Smart Library Management," International Journal of Global Research in Science and Technology (IJGRST), vol. 9, pp. 38–51, 2024.

[21] T. Kataria, T. Saxena, A. Joshi, S. Soni, and M. Mathur, "Web-Based Automation of Campus Examination and Placement Processes Using MERN Stack," International Journal of Global Research in Science and Technology (IJGRST), vol. 10, pp. 236–241, 2025.

[22] A. Bohra, K. Paliwal, and S. Soni, "Online Code Editor: A Cloud-Based Platform for Real-Time Web Development," International Journal of Global Research in Science and Technology (IJGRST), vol. 9, pp. 52–76, 2024.

[23] H. Sharma, A. K. Agrawal, K. Bishnoi, and P. Jain, "From HTML Pages to AI-Driven Platforms: A Review of Web Development Evolution," International Journal of Global Research in Science and Technology (IJGRST), vol. 9, pp. 120–124, 2024.