International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

DevHub – A Complete DevOps Management Platform

Shravani A. Bire[1], Rashmi V. Kaware[2], Minakshi S. Bobade[3], Gayatri A. Kanake[4], Sakshi G. Rathod[5], Divyani V. Yerme[6]

[123456] *Department of Computer Science and Engineering, P. R. Pote Patil College of Engineering and Management, Amravati, Maharashtra, India*
Under the Guidance of Prof. S. R. Kale

ABSTRACT

Modern software development organizations use multiple tools for source-code management, project coordination, CI/CD, system monitoring, security, and reporting. Although these tools provide specialized functionality, their separation can lead to tool fragmentation, context switching, and difficulty in obtaining a unified view of development and operational activities. Recent research on DevOps, platform engineering, and Internal Developer Portals (IDPs) highlights the importance of centralized platforms for simplifying developer workflows and improving visibility.

This paper presents DevHub – Complete DevOps Management Platform, a centralized web-based platform that integrates authentication, role-based access control, team and project management, deployment tracking, system monitoring, dashboard analytics, and security support. The deployment module uses background workers to simulate deployment lifecycle progression, while the monitoring module performs configurable URL health checks. DevHub is implemented using React, Vite, Node.js, Express.js, TypeScript, Prisma ORM, and SQLite. The reviewed literature indicates a need for lightweight platforms that combine development management, deployment visibility, and monitoring. DevHub provides a practical foundation for centralized DevOps management and future integration with real CI/CD platforms and advanced analytics.

Key Words: DevOps, IDP, Platform Engineering, CI/CD, RBAC, System Monitoring.

1. INTRODUCTION

1.1 Background

DevOps integrates development and operations to improve collaboration, automation, software delivery, monitoring, and continuous feedback. Modern software projects commonly use separate tools for source-code management, project coordination, deployment, monitoring, and security. Although these tools provide specialized capabilities, information distributed across multiple platforms can increase context switching and make it difficult to obtain a unified view of development and operational activities.

Platform engineering and Internal Developer Portals (IDPs) have emerged as approaches for bringing commonly required development and operational capabilities into a centralized environment. A centralized platform can provide developers and managers with a common interface for accessing project, deployment, monitoring, and activity information.

1.2 DevOps and Integrated Management

DevOps activities extend across development, deployment, and operational monitoring. An integrated management platform can provide centralized access to teams, projects, repositories, deployments, environments, monitoring information, and activity records. Such integration can reduce context switching and improve visibility across the software lifecycle.

Authentication and role-based access control further ensure that platform resources are accessed according to user responsibilities. DevHub applies this approach by combining development management, deployment tracking, system monitoring, and dashboard-based analytics within a single web-based environment.

1.3 Existing Challenge

Organizations often depend on separate tools for source-code management, project coordination, CI/CD, deployment tracking, monitoring, and security. This separation can result in fragmented information and make it difficult to track project progress, deployment status, system health, and team activities from a single location. Managers and developers may need to switch between multiple platforms to obtain information about different stages of the software lifecycle.

Therefore, there is a need for a centralized and secure platform that brings essential DevOps management activities together while maintaining controlled access to resources.

1.4 Proposed DevHub

DevHub – Complete DevOps Management Platform is proposed as a centralized web-based solution for managing essential development and operational activities. The platform integrates authentication and RBAC, team and project management, deployment lifecycle tracking, system monitoring, dashboard analytics, and security-related functions.

The deployment module uses application-managed background workers to represent deployment lifecycle states such as Queued, In Progress, Success, Failed, and Rolled Back. The monitoring module performs configurable URL-based health checks and maintains monitoring information. The platform therefore provides centralized visibility while leaving integration with real external CI/CD platforms as future work.

2. PROBLEM STATEMENT

Modern DevOps environments use multiple tools for source-code management, project collaboration, CI/CD, deployment tracking, monitoring, and security. Although these tools provide specialized functionality, their separate use can lead to tool fragmentation, context switching, scattered information, and difficulty in tracking project and deployment activities.

Developers and managers may need to access different platforms to check project progress, deployment status, system health, and team activities. The absence of a centralized dashboard can make it difficult to obtain a complete view of the software lifecycle and identify deployment or system failures efficiently.

Therefore, there is a need for a centralized and secure DevOps management platform that integrates essential development and operational activities. DevHub addresses this problem by providing authentication and RBAC, team and project management, deployment lifecycle tracking, URL-based system monitoring, and centralized dashboard analytics in a unified web-based environment.

3. OBJECTIVES

The main objectives of DevHub – Complete DevOps Management Platform are:

1. To implement secure authentication and role-based access control for controlled access based on user roles.

2. To provide centralized team and project management for better collaboration and organization.

3. To implement deployment lifecycle tracking using background workers to simulate CI/CD stages such as Queued, In Progress, Success, Failed, and Rolled Back.

4. To provide system monitoring through configurable URL health checks and monitoring status.

5. To maintain deployment, monitoring, and activity records for improved tracking and traceability.

6. To develop a centralized dashboard for project statistics, deployment metrics, monitoring information, and activity feeds.

7. To incorporate security mechanisms, testing practices, and Docker-based deployment support.

4. LITERATURE REVIEW

Recent DevOps research focuses on improving software delivery through collaboration, automation, continuous integration, continuous delivery, monitoring, security, and continuous feedback. However, the increasing number of development and operational tools can create challenges related to coordination, visibility, adoption, and developer experience.

Platform Engineering and Internal Developer Portals: Research on platform engineering and Internal Developer Portals emphasizes centralized platforms that provide developers with common access to development and operational resources. Anjum [1] presents a multivocal review of platform engineering and IDPs, highlighting areas such as platform capabilities, adoption, developer experience, and maturity. Van de Kamp et al. [7] propose a reference model for platform engineering, while Aslina and Nugraha [10] explore potential AI use cases in IDPs for improving developer experience. These studies support the use of centralized platforms for simplifying access to development capabilities.

DevOps Metrics and Lifecycle: Kumar et al. [2] examine DevOps metrics and identify measures that can be used to evaluate DevOps practices. Amaro et al. [3] study the relationship between DevOps capabilities and software lifecycle processes. These studies highlight the importance of monitoring development and operational activities and support the use of centralized dashboards for deployment and activity visibility.

DevOps Adoption and Human Factors: El Aouni et al. [4] investigate the integration of Agile, Cloud, and DevOps and discuss associated benefits and challenges. Pérez-Sánchez et al. [5] examine human factors influencing

DevOps adoption. Nayanajith and Wickramarachchi [11], Karunarathne et al. [12], Hamza et al. [14], and Azad and Hyrynsalmi [15] further examine challenges, adoption practices, and critical success factors. These studies indicate that successful DevOps environments require not only technical automation but also effective collaboration, organizational readiness, and usable workflows.

DevSecOps and Automation: Bautista Ramos and Yoo [6] examine cybersecurity in DevOps environments, while Zhao et al. [9] identify important dimensions of DevSecOps. Saxena et al. [8] discuss DevOps automation using Infrastructure as Code. These studies highlight the importance of integrating security and repeatable automation into modern development environments.

Research Gap: The reviewed studies provide strong coverage of individual areas such as platform engineering, DevOps metrics, adoption, security, and automation. However, these areas are frequently considered separately. Based on this review, there is a practical need for lightweight platforms that combine team and project management, deployment lifecycle visibility, system monitoring, security, and dashboard analytics within a unified environment. DevHub addresses this practical gap by integrating these capabilities while using background-worker simulation instead of claiming direct execution of external CI/CD pipelines.

5. PROPOSED METHODOLOGY

5.1 Proposed System Overview

DevHub – Complete DevOps Management Platform is designed as a centralized web-based platform for managing essential DevOps activities in a single environment. The system integrates authentication, role-based access control, team and project management, deployment lifecycle tracking, system monitoring, and dashboard analytics.

Users first authenticate into the system, after which access is controlled according to their assigned roles. Teams and projects can be managed centrally. The deployment module tracks the deployment lifecycle using background workers and maintains states such as Queued, In Progress, Success, Failed, and Rolled Back. The monitoring module performs configurable URL-based health checks and stores monitoring information. The dashboard combines project, deployment, monitoring, and activity data to provide a centralized view.

The system uses React, TypeScript, and Vite for the frontend, Node.js and Express.js for the backend, and Prisma ORM with SQLite for data management. Security mechanisms include JWT authentication, password hashing, RBAC, validation, rate limiting, and audit logging.

5.2 System Architecture

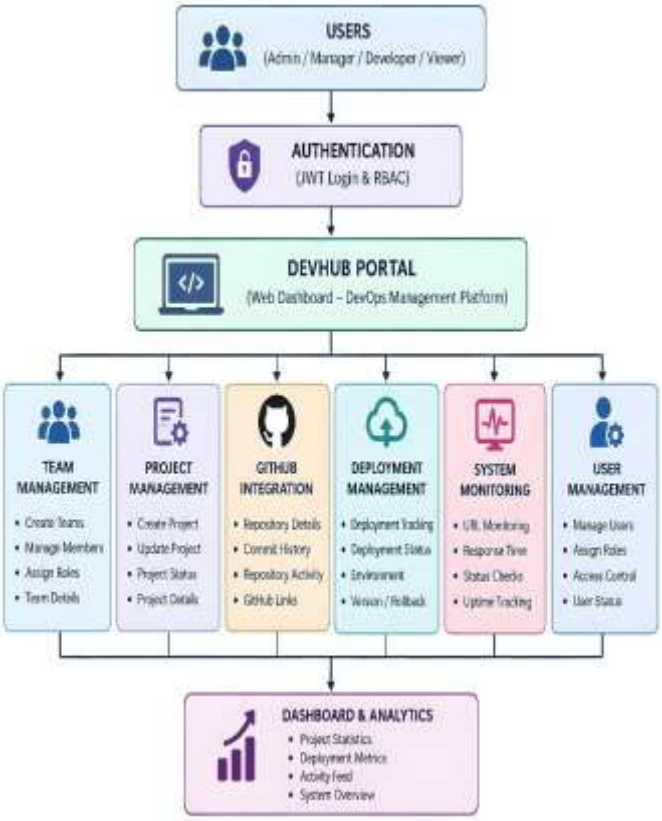


Fig -1: System Architecture of DevHub – Complete DevOps Management Platform.

The proposed DevHub architecture follows a three-layer architecture consisting of the presentation layer, application/backend layer, and database layer.

1. Presentation Layer:
The frontend is developed using React, TypeScript, Vite, Tailwind CSS, Zustand, and Recharts. It provides interfaces for login, dashboard, teams, projects, deployments, monitoring, and user management.

2. Application Layer:
The backend is developed using Node.js, Express.js, and TypeScript. It provides REST APIs for authentication, CRUD operations, team/project management, deployment tracking, monitoring, notifications, API keys, and audit activities. JWT-based authentication and RBAC control access to protected resources.

3. Data Layer:
The system uses Prisma ORM to communicate with the SQLite database. The database stores users, sessions, teams, projects, repositories, deployments, monitoring targets, ping history, audit logs, activities, notifications, and other system records.

5.3 System Modules

1. User Authentication and Authorization: Provides secure login, JWT authentication, password protection, and role-based access control.

2. Team and Project Management: Manages teams, members, projects, repositories, environments, and project activities.

3. CI/CD Deployment Tracking: Tracks deployment lifecycle states including Queued, In Progress, Success, Failed, and Rolled Back using background-worker simulation.

4. System Monitoring and Alerting: Performs configurable URL/HTTP health checks and maintains system status and failure information.

5. Dashboard and Analytics: Presents project, deployment, monitoring, and activity information through centralized dashboards and charts.

6. Security, Testing and Deployment: Provides validation, API security, rate limiting, audit logging, Jest-based testing, and Docker support.

6. RESULTS & DISCUSSION

6.1 Results

The implementation of DevHub successfully integrates major DevOps management activities into a centralized web-based platform. The completed system provides secure authentication and role-based access, team and project management, deployment lifecycle tracking, URL-based system monitoring, and dashboard analytics.

Functional testing was performed for major system areas including authentication, role-based access, team and project operations, deployment lifecycle states, monitoring operations, dashboard information, and security-related functions. The implemented modules operated according to their intended workflows during functional verification.

6.1 Discussion

The deployment module provides visibility into different lifecycle states through background-worker simulation, while the monitoring module provides system health information through configurable URL checks. The centralized dashboard brings project, deployment, monitoring, and activity information together, reducing the need to switch between separate interfaces.

The implementation demonstrates the practical feasibility of combining multiple DevOps management activities within a single lightweight platform. However, the current implementation should be considered a centralized management and visibility platform rather than a complete production-grade CI/CD execution system.

7. LIMITATIONS

7.1 Deployment & Monitoring Scope

The current system provides deployment lifecycle tracking through background-worker simulation rather than integration with real external CI/CD pipelines. Monitoring is primarily based on URL/HTTP health checks.

7.2 Limited Evaluation

The platform has been tested for functional correctness, but extensive performance, load, usability, and large-scale evaluation has not been performed.

7.3 Access Control Expansion

The current RBAC implementation provides role-based access, but more fine-grained permissions and resource-level access control can be added in future versions.

7.4 Scalability

The current implementation uses SQLite and is mainly suitable for development and small-scale usage. Large-scale deployment may require PostgreSQL, load balancing, caching, and performance optimization.

7.5 Overall Limitation

DevHub currently focuses on centralized DevOps management rather than complete production-grade CI/CD and observability. Future improvements can include real CI/CD integration, advanced monitoring, real-time notifications, and scalable infrastructure.

8. CONCLUSION

DevHub – Complete DevOps Management Platform provides a centralized web-based environment for managing essential DevOps activities within a single platform. The implemented system integrates secure authentication and role-based access control, team and project management, deployment lifecycle tracking, URL-based system monitoring, and dashboard analytics. The use of background workers enables the system to represent deployment lifecycle states and provides visibility into deployment progress without claiming direct execution of external CI/CD pipelines. Functional testing confirmed the intended operation of the major system modules and workflows. Overall, DevHub demonstrates the practical feasibility of combining development management, deployment visibility, monitoring, and security features in a lightweight platform. Future improvements can focus on integration with real CI/CD platforms, advanced monitoring, real-

time notifications, fine-grained access control, and scalable infrastructure.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Prof. S. R. Kale for her valuable guidance, support, and encouragement throughout the development of the DevHub project and preparation of this research paper. The authors also thank the Department of Computer Science and Engineering, P. R. Pote Patil College of Engineering and Management, Amravati, for providing the necessary academic support and resources. The authors are grateful to everyone who contributed directly or indirectly to the successful completion of this work.

9. REFERENCES

[1] M. A. Anjum, "Platform engineering and internal developer portals: A multivocal literature review," Frontiers in Computer Science, vol. 8, Art. no. 1814498, 2026, doi: 10.3389/fcomp.2026.1814498.

[2] A. Kumar, M. Nadeem, and M. Shameem, "A systematic literature review for investigating DevOps metrics to implement in software development organizations," Journal of Software: Evolution and Process, vol. 37, no. 1, Art. no. e2733, 2025, doi: 10.1002/smr.2733.

[3] R. Amaro, R. Pereira, and M. M. da Silva, "Mapping DevOps capabilities to the software life cycle: A systematic literature review," Information and Software Technology, vol. 177, Art. no. 107583, 2025, doi: 10.1016/j.infsof.2024.107583.

[4] F. El Aouni, K. Moumane, A. Idri, M. Najib, and S. U. Jan, "A systematic literature review on Agile, Cloud, and DevOps integration: Challenges, benefits," Information and Software Technology, vol. 177, Art. no. 107569, 2025, doi: 10.1016/j.infsof.2024.107569.

[5] J. Pérez-Sánchez, S. Rafi, J. M. Carrillo de Gea, J. N. Ros, and J. L. Fernández Alemán, "A theory on human factors in DevOps adoption," Computer Standards & Interfaces, vol. 92, Art. no. 103907, 2025, doi: 10.1016/j.csi.2024.103907.

[6] R. C. Bautista Ramos and S. G. Yoo, "Cybersecurity in DevOps environments: A systematic literature review," IEEE Access, vol. 13, pp. 191959–191979, 2025, doi: 10.1109/ACCESS.2025.3582892.

[7] R. van de Kamp, Z. Zhao, and K. Bakker, "Paving the path towards platform engineering using a comprehensive reference model," in Enterprise Design, Operations, and Computing, Lecture Notes in Business Information Processing. Cham, Switzerland: Springer, 2024, doi: 10.1007/978-3-031-54712-6_11.

[8] A. Saxena, S. Singh, S. Prakash, T. Yang, and R. S. Rathore, "DevOps automation pipeline deployment with IaC (Infrastructure as Code)," in Proc. 2024 IEEE Silchar Subsection Conference (SILCON), Agartala, India, 2024, pp. 1–6, doi: 10.1109/SILCON63976.2024.10910699.

[9] X. Zhao, T. Clear, and R. Lal, "Identifying the primary dimensions of DevSecOps: A multi-vocal literature review," Journal of Systems and Software, vol. 214, Art. no. 112063, 2024, doi: 10.1016/j.jss.2024.112063.

[10] Y. R. Aslina and I. G. B. B. Nugraha, "Exploring potential AI use cases in internal developer portals: A path to enhanced developer experience," in Proc. 2024 IEEE International Conference on Data and Software Engineering (ICoDSE), 2024, pp. 143–148, doi: 10.1109/ICoDSE63307.2024.10829893.

[11] A. Nayanajith and R. Wickramarachchi, "Challenges affecting the successful adoption of DevOps practices: A systematic literature review," in Proc. 2024 4th International Conference on Advanced Research in Computing (ICARC), 2024, pp. 311-315, doi: 10.1109/ICARC61713.2024.10499735.

[12] M. A. W. Karunarathne, W. M. J. I. Wijayanayake, and A. P. K. J. Prasadika, "DevOps adoption in software development organizations: A systematic literature review," in Proc. 2024 4th International Conference on Advanced Research in Computing (ICARC), Belihuloya, Sri Lanka, 2024, pp. 282–287, doi: 10.1109/ICARC61713.2024.10499789.

[13] J. A. V. M. K. Jayakody and J. I. Wijayanayake, "DevOps maturity: A systematic literature review," in Proc. 2024 International Research Conference on Smart Computing and Systems Engineering (SCSE), 2024, pp. 1-6, doi: 10.1109/SCSE61872.2024.10550493.

[14] U. Hamza, S. M. Syed-Mohamad, and N. L. Abdullah, "Exploring the benefits, challenges and guidelines of DevOps adoption: A systematic literature review and an empirical study," Journal of Mathematical Sciences and Informatics, vol. 4, no. 2, 2024, doi: 10.46754/jmsi.2024.10.007.

[15] N. Azad and S. Hyrynsalmi, "DevOps critical success factors—A systematic literature review," Information and Software Technology, vol. 157, Art. no. 107150, 2023, doi: 10.1016/j.infsof.2023.107150.