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

RESEARCH ARTICLE OPEN ACCESS

LINUSFORGE: A Centralized and Secure DevOps Monitoring Panel for Linux Server Management

Dr. Gaurav Kumar Jain*, Divya Khansili, Garima Singh Naruka****

*Associate Professor, Department of Computer Science and Engineering, Global Institute of Technology, Jaipur

**B.Tech Student, Department of Computer Science and Engineering, Global Institute of Technology, Jaipur

ABSTRACT

Maintaining and monitoring multiple Linux servers are becoming more complicated as DevOps grows. In the deployment realm, we are deploying several times a week in environments that span thousands of servers. We require intelligent systems that are simple to operate and keep running 24 hours a day. Keeping it simple is a web based monitoring and management panel that accomplishes these goals. It currently serves as the central hub for guidelines, best practices, and tools. Its architecture is based on the API first; the front end is comprised of a single page application. LINUSFORGE is a unified console that centralizes server monitoring, alerting, and performance management critical to successful DevOps. Lastly, multi-server integration, intelligent alerting, TFA security, and HTTPS encryption are among the key features that make it a modern, secure, and powerful possibility for your monitoring needs..

Keywords — DevOps, Linux Environments, API-centric Design, Single Page Authentication (SPA), dynamic monitoring.

I. INTRODUCTION

With businesses increasingly shifting toward cloud-native and containerized infrastructures, DevOps teams have a much larger number of Linux servers to manage. While tools including Prometheus, Grafana, and Nagios offer excellent monitoring capabilities, they tend to operate in silos or require significant integration work bundles. This leads to disparate dashboards, spurious alerts, and slow response times. Therein lies the inspiration behind LinusForge – a centralized platform that monitors and intelligently manages several Linux servers. To this end, I envisioned creating a platform that integrates real-time visibility, alert intelligence using automation, and best-of-breed security protocols. LinusForge intends to be a one-stop platform for DevOps that offers a source of truth, eliminating tool fragmentation, and promoting automation to enhance infrastructure.

II. METHODOLOGY

LINUSFORGE's system design is designed in service-oriented because every link between client-side and server-side modules is done perfectly to avoid closing the link between databases and user requests. This architecture design aspect makes it flexible, scalable, easy to combine with other external sources.

i. The Client- side is designed by React framework, supports the SPA methodology, so the movement is smooth, and the user management is dynamic. The presentation of system statistics does not require any trimming on the move but dynamically. The user has no need for refresh. This is indicative of excellent usability. ii. Server-side: The system is constructed upon the Spring framework on this stage used to carry the service routes and course of incoming information from the Linux agents distributed and organize the backend protocol. Server-side additionally has PostgreSQL and MongoDB databases that it uses to handle the system metrics data persistently and on-the-run according to particular model wants. The PostgreSQL is useful because it may well handle the restricted, organized, or relational dataset, whereas the MongoDB databases deal with the non- or semi relational dataset and un-structured thus a more versatile database administration technique.

Moreover, the system design considers security as a critical element. All connections are secured with a secure socket layer communication; identity verification is further secured through Two-Factor Authentication mechanisms, which eliminates the potential unauthorized usage vulnerability. Moreover, for the development process, Agile Software Development Life Cycle model will be implemented. Through the phased development and feedback loops in real-time, this system will possess dynamic adaptability, which allows incorporation of new features and maintains elastic performance of evolving

DevOps management.

III. SYSTEM ARCHITECTURE

The architectureof LINUSFORGE is a distributed, service oriented framework intended to provide scalability, security, and real time performance. The system is divided into three major logical and physical layers: the client layer, the server layer, and the data management layer that communicate over the secure APIs. Its structure is designed as a service-based architecture to provide scalability, modularity, and reusability. It uses common communication standards to speed up and streamline the service integrations in applications. Each layer runs on its own infrastructure and can be developed simultaneously by a separate development team. Also they can be updated or scaled as needed without impacting the other tiers. The client layer represents the user-interface side of the platform. It is also called the presentation layer and users can observe the server actions in real time via a web dashboard, run visual reports of server operation, be alerted through message alert, and command multiple servers. This top-level tier can run on a web browser, as desktop application, or a graphical user interface . The server layer, also the middle layer work as the connection between Linux agents and the frontend. It processes incoming data, credential verifications, and sending of service requests. Safety in this layer is intertwined into the system through HTTPS lock and two-step login. Second logging ensures that private data remains private and that on regulated resources is locked. Additionally, by deterring illegal entry, two-step entry induces the usage of a VPN.

The data management layer also called database tier, data access tier or back-end, is where the information that is processed by the application is stored and managed. This can be a relational database management system such as PostgreSQL, MySQL, MariaDB, Oracle, Db2, Informix or Microsoft SQL Server, or in a NoSQL Database server such as Cassandra, CouchDB, or MongoDB. PostgreSQL is used to manage structured table data, like user information, login information, and configuration facts. MongoDB saves unstructured or partly-structured data such as the host remarks and pace logs. In a three-tier application, all communication goes through the application tier. The presentation tier and the data tier cannot communicate directly with one another.

By blending these levels, LINUSFORGE creates a flexible, modular setup. the chief benefit of three-tier architecture is its logical and physical separation of functionality. Each tier can run on a separate operating system and server platform for example, web server, application server, database server - that best fits its functional requirements. And each tier runs on at least one dedicated server hardware or virtual server, so the services of each tier can be customized and optimized without impacting the other tiers. One layer handles a unique task - yet combined, they form a unified system offering live tracking, secure operations and room for scaling. The layout keeps space for future upgrades, making it simple to plug in new DevOps tools or functions when business demands shift.

IV. APPLICATION

The uses of LINUSFORGE pop up everywhere - big companies use it, so do folks at home - all thanks to its API-first core that runs everything from live dashboards to built-in security without needing extras; compared to old-school tools like Nagios, Zabbix, Prometheus, or Webmin, which typically only monitor or manage basic tasks. Because it combines these features smoothly, LINUSFORGE offers a setup that bends easily to match varied needs and user types.

IT companies plus DevOps teams – Inside organisations,

LINUSFORGE keeps an eye on vital statistics like CPU usage, memory pressure, and network flow without stopping. Instead of working like Prometheus - built mainly for storing time-series data while relying on external apps such as Grafana to show visuals - LINUSFOREGE bundles storage right into live dashboards along with sharp alert systems, making problem detection plus resolution faster.

Data Centers – When handling large setups, LINUSFORGE keeps things running smoothly with instant alerts and real-time monitoring. Unlike Nagios, which needs lots of plugins to function, LINUSFORGE watches multiple servers right out of the box, so installation is faster while reducing daily maintenance.

Cloud Providers – Across multi-cloud setups like AWS, Azure,

or GCP, LINUSFORGE delivers unified monitoring so you see everything clearly no matter the platform. Zabbix might bring enterprise features too, yet setup tends to get complicated; LINUSFORGE skips that hassle by focusing on simplicity and security right out of the gate, which makes handling it way more straightforward.

Study Labs – Schools might try LINUSFORGE to help

students

pick up current DevOps routines through hands-on labs. Although Webmin usually demonstrates basic Linux configuration and management, LINUSFORGE goes further by emphasizing monitoring, notifications, or security workflows - letting learners gain real-world experience using tools that match actual job demands.

Small or mid-sized teams - those without deep pockets or tech know-how - can use LINUSFORGE as a budget-friendly, scalable option. Instead of wrestling with bulky enterprise-grade software, they get reliable, secure monitoring for essential servers. The system runs smooth without demanding high-end hardware or expert setup.

Personal Setups – When not used for work, LINUSFORGE still comes in handy for folks tinkering on their own, whether they're monitoring private servers, home labs, or tiny cloud systems. While Webmin focuses on getting servers running,

LINUSFORGE keeps an eye on things nonstop, sending live alerts so individuals can manage setups just as confidently as big teams do.

V. RESULTS AND DISCUSSION

A look at how current tools stack up against each other LINUSFORGE was built to upgrade today's Linux server management systems, particularly tools like Webmin and similar mainstream options that dominate the field right now.

A. Existing Systems:

Webmin – Even today, Webmin stays popular when managing Linux systems, especially setting things up or running services. Yet it struggles with real-time tracking, visual data displays, or smart warning systems. On top of that, critical protections such as encrypted connections and double-check logins need manual setup, which opens doors for errors. LINUSFORGE tackles these gaps by mixing setup tools with live oversight, automatic alerts, plus security turned on right away.

Nagios or Zabbix - each packs solid monitoring power with broad features. Still, setup and upkeep can get tricky, usually requiring folks who really know their way around tech. LINUSFORGE goes a different route, built around APIs from the start, focusing on smooth installation and daily use so smaller crews, research spots, or clubs without dedicated monitoring pros can actually use it without hassle.

Prometheus with Grafana – this combo's become a go-to for tracking and visualizing data. It works well, but getting things linked right takes time plus know-how. LINUSFORGE simplifies the process by offering ready-made dashboards and warnings, so you skip the tricky configurations or added software. Cloud-Native Dashboards (AWS CloudWatch, Azure Monitor) – While solid inside their own environments, they can't handle multiple platforms well. LINUSFORGE steps in here by offering a single control hub that monitors various setups - on-prem, across clouds, or hybrid systems. Sure, every option today has its strengths - but LINUSFORGE stands out by blending tracking, visuals, alongside protection into one solid system.

B. How This Stacks Up Against Earlier Studies:

Past research here usually focused on monitoring things such as CPU use, RAM, or storage space. Yet, two major drawbacks keep popping up:

Some research efforts pay little attention to protection - HTTPS or two-factor checks often get skipped, so setups end up exposed. Focused too narrowly most tools from past studies highlight just setup (like Webmin) or tracking (such as Prometheus), yet almost never combine both in a single solution.

LINUSFORGE stands out by tackling two problems at once. It brings in clever warnings that come with protection baked right in, so you don't just log data - your access stays locked down while alerts actually help. This blend of tracking, simplicity, and security gives a sharper, more usable setup compared to earlier research models.

C. What I Plan to Add

The key ingredients for LINUSFORGE include: however, they're not limited to these components Beyond Webmin Sure, Webmin works for basic setup, yet itlacks solid tracking and visual data displays. LINUSFORGE builds on that foundation, mixing in live dashboards, smart alert filtering, visual stats, on top of management tools. Secure-by-Design Unlike plenty of tools where safety gets tacked on later by admins, LINUSFORGE bakes it right into the core from day one. Features such as HTTPS encryption along with two-step login come ready-made, slashing room for user error. All-in-one system Instead of using different apps for tracking, warnings, visuals, or protection, LINUSFORGE brings them together. So users get simpler DevOps workflows, whether they're handling enterprise tasks, education setups, or solo work.

VI. CONCLUSION

LINUSFORGE is an all-in-one tool built for today's DevOps needs, designed around simplicity, reliability, and speed. Instead of juggling multiple apps, teams get a unified hub that's easy to navigate and quick to respond. The front end runs on React, whichteams can tweak on the fly, while Spring Boot powers the engine behind the scenes - this pairing boosts responsiveness and cuts delays. Because of this setup, users can monitor several servers at once, set up custom alerts that adapt to patterns, or track performance without constant manual checks. On the safety side, it doesn't rely on add-ons; encryption via SSL/HTTPS comes standard, and login requires two steps to block unauthorized access before it starts. Besides solid tech skills, LINUSFORGE shifts easily when conditions change or users demand more. Whether it's a business running critical systems, a cloud provider balancing loads, a school training DevOps beginners, or someone tinkering at home, it fits right in. Its room to grow and bend without breaking makes it more than just another monitor - it turns into a learning aid, career booster, a steady hand for speed and reliability. Merging tracking, warnings, and protection into one setup, LINUSFORGE stands out not just for fixing today's issues but helping shape tomorrow's IT world too.

REFERENCES

[1] React Documentation. [Online]. Available: https://reactjs.org/docs/getting-started.html

[2] Spring Boot Documentation. [Online]. Available: https://spring.io/projects/spring-boot

[3] PostgreSQL Documentation. [Online]. Available: https://www.postgresql.org/docs/

[4] MongoDB Documentation. [Online]. Available: https://www.mongodb.com/docs/

[5] Agile Alliance, "Agile methodology." [Online]. vailable: https://www.agilealliance.org/agile101/

[6] Chart.js Documentation. [Online]. Available: https://www.chartjs.org/docs/latest/

[7] Maven Documentation. [Online]. Available: https://maven.apache.org/guides/index.html

[8] IntelliJ IDEA Documentation. [Online]. Available : https://www.jetbrains.com/idea/documentation/

[9] Git Documentation. [Online]. Available: https://git-scm.com/doc

[10] Mozilla Developer Network, "HTTPS overview." [Online].Available: https://developer.mozilla.org/en-US/docs/Web/HTTPS

[11] Webmin Official Documentation. [Online]. Available:http://www.webmin.com/docs.html

[12] Nagios Documentation. [Online]. Available:https://support.nagios.com/kb/

[13] Zabbix Documentation. [Online]. Available:https://www.zabbix.com/documentation/current/manual

[14] Prometheus Documentation. [Online]. Available:https://prometheus.io/docs/introduction/overview/

[15] Grafana Documentation. [Online]. Available:https://grafana.com/docs/

[16] J. Humble and D. Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley, 2010.

[17] G. Kim, K. Behr, and G. Spafford, The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win. IT Revolution, 2013.

[18] L. Chen, "Continuous delivery: Huge benefits, but challenges too,"IEEE Software, vol. 32, no. 2, pp. 50–54, Mar.-Apr. 2015.

[19] OpenAI, "ChatGPT (Sept. 2025 version)," OpenAI, San Francisco, CA, USA. [Online]. Available: https://chat.openai.com/

[20] Google DeepMind, "Gemini (2025 release)," Google,

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

[22] A. Chauhan and R. Misra, "Outline of Web Development Life Cycle in Software Engineering," in Proc. International National Conference on Recent Trends in Engineering & Technology, 2023.

[23] 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.

[24] 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.

[25] 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.

[26] H. Fulwani, Harsh, H. Dhakad, A. Bohra, and M. Mathur, "Web Development: Technologies, Trends and Future Scope," International Journal of Global

Research in Science and Technology (IJGRST), vol. 10, pp. 161–165, 2025.

[27] 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.

[28] 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.

[29] 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.

[30] 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.