Krisnadana Journal (Journal of Computers, Control Systems & Networks)
Vol 5, Issue 3, May 2026, pp. 543-549
https://doi.org/10.58982/4dj8p556

E-ISSN: 2808-3563
P-ISSN: 2808-3520

Implementation of Containerization on Village Information Systems (OpenSID) Using Docker for Server Resource Efficiency

I Wayan Yudik Pradnyana[1*], I Wayan Ady Juliantara[2]

[1*] Institut Teknologi dan Kesehatan Bintang Persada
[2] Universitas Tabanan

[1*]yudikpradnyana@bintangpersada.ac.id ; [2]adyjuliantara1@gmail.com

ARTICLE INFO

Article history:
Received 5 June 2026
Revised 22 June 2026
Accepted 29 June 2026

***Keywords**:*
Container-Based Deployment;
OpenSID;
DevOps;
Infrastructure as Code;
Village Governance

ABSTRACT

The digitalization of village governance in Indonesia faces significant challenges in the form of inconsistencies in the server environment (environment drift) that hinder the stability and scalability of the Village Information System (OpenSID). This research aims to implement the Container-Based Deployment method as an operational approach based on the DevOps paradigm to standardize village digital infrastructure. The method used is Infrastructure as Code (IaC) by integrating OpenSID applications and MariaDB databases into isolated Docker containers, with service access through specific port mapping for both primary application access and database management. The results showed that the use of containerization successfully overcame the complexity of manual configuration, with a 100% success rate of meeting system dependencies. This architecture allows for identical system replication, deployment time efficiency, and ease of service access. This research produces a reliable, portable, and efficient system architecture model for village governments in supporting the transformation of digital public services that are more stable and accessible.

1. Introduction

The digitalization of village governance in Indonesia is now an urgent need to accelerate the transformation of public services [1]. This effort is in line with the strategic policy direction contained in the national digital transformation blueprint for the regional and village levels [2]. In its development, the adoption of technology open-source such as the Village Information System (OpenSID) is one of the main pillars to improve the transparency and efficiency of population data in order to realize the ecosystem Smart Village [3]. However, at the technical implementation stage in the field, the digitization process faces a major obstacle in the form of inconsistencies in the server environment (Environment Drift) that often disrupts system stability when moved between infrastructures [4]. This obstacle is increasingly complex due to the limited quality of human resources in the field of information technology who manage administration at the village level [5]. Traditional

operational management patterns that rely on manual server configuration have proven to be highly vulnerable to the risk of human error (Human Error) [6].

Previous literature studies have suggested that although the concept of Cloud Computing and containerization began to be adapted to support the progress of digital villages [7], automation-based integration Infrastructure as Code (IaC) in applications with modern frameworks still needs in-depth studies to maintain long-term stability [8]. Therefore, accelerating the digitalization of public services requires a touch of an adaptive DevOps paradigm for the public sector [9], particularly through the use of Continuous Integration and Continuous Deployment (CI/CD) structured [10]. The operational role of DevOps is proving crucial in cutting the bureaucracy of setting up information system infrastructure [11]. Technical evaluation of community service application architectures shows that conventional deployments are often hit with the problem of complexity of fulfillment dependencies [12]. Web server incompatibilities and local resource allocation restrictions often degrade the performance of Laravel-based applications [13] [14]. In addition, the increasingly dynamic management of data volumes demands a reliable relational database scalability management strategy in an isolated ecosystem [15].

Although previous literature has documented the effectiveness of containerization and DevOps paradigms in information system optimization, there are significant gaps in the application of these methods in the context of village governance in Indonesia. The majority of current studies tend to focus on large-scale industrial environments supported by qualified IT experts, so there is no model that specifically bridges enterprise-grade technology with infrastructure limitations and unique characteristics at the village level that have connectivity constraints and limited technical human resources. In addition, there have not been many studies that evaluate how container-based deployment architectures can be implemented in a portable and resource-efficient manner, but still easily managed by village operators with limited digital literacy. This creates a research gap on how the standardization of digital infrastructure can be designed to be technically resilient but still adaptive to the operational realities of village government.

In order to anticipate these obstacles, this study proposes the application of the Container-Based Deployment to design isolated service automation, including MariaDB database performance optimization in containers [16] . This approach is reinforced by the architecture Microservices distributed to ensure the flexibility of the village-level information system [17]. Through a practical guide to containerization setup using Docker [18], environmental standardization can be achieved while improving the security aspects of public service infrastructure from potential external interference [19]. Ultimately, this OpenSID deployment technical standardization scheme is expected to be able to provide a portable, resilient, and sustainable digital infrastructure model in accordance with the developer community's documentation standards [20].

2. Method

This study uses the Container-Based Deployment approach as part of the DevOps paradigm to standardize the digital infrastructure of the Village Information System (OpenSID). This study applies Container-Based Deployment as an operational standard to ensure system consistency. This method excels through three main aspects: environmental standardization that eliminates environment drift by isolating all application dependencies. Resource optimization that allows for more efficient workload allocation than conventional

virtualization so that the system remains responsive on servers with limited specifications. As well as infrastructure automation (IaC) that replaces manual configuration with standard scripts to minimize human error and speed up system recovery time. The research steps are carried out systematically through the development phase and the production phase to ensure the consistency of the system environment.

The stages of research carried out are as follows:

- Source Code Preparation : The OpenSID source code is obtained from the official repository and managed using Git as version control.
- Dockerfile Setup : Dockerfile is structured to define all system dependencies, including PHP versions, GD libraries, Tidy, Zip, and Exif to address environment drift issues.
- Deployment Configuration : This stage involves using Docker Compose to orchestrate between the application container and the MariaDB database container. Infrastructure configurations are defined in code through docker-compose.yml files as a form of Infrastructure as Code (IaC) implementation.
- Testing and Implementation : The system is tested by verifying the fulfillment of server requirements, then implemented on a server with localhost access via port mapping 8080.

Data is collected through testing the functionality of the system in a container environment. The data analysis technique used is a comparative analysis between conventional installation methods and container-based deployment methods to measure deployment efficiency and system consistency. The entire workflow of this research is visualized in the framework presented in the Figure 1.

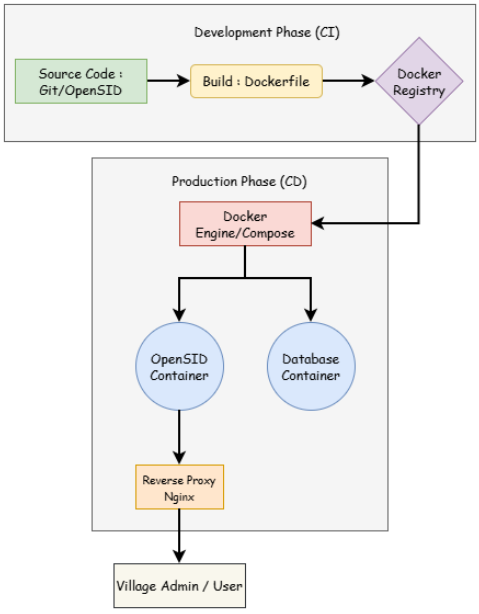

Figure 1. CI/CD Architecture Diagrams

System infrastructure configurations are defined in docker-compose.yml files to ensure consistent replication can be seen on Figure 2.

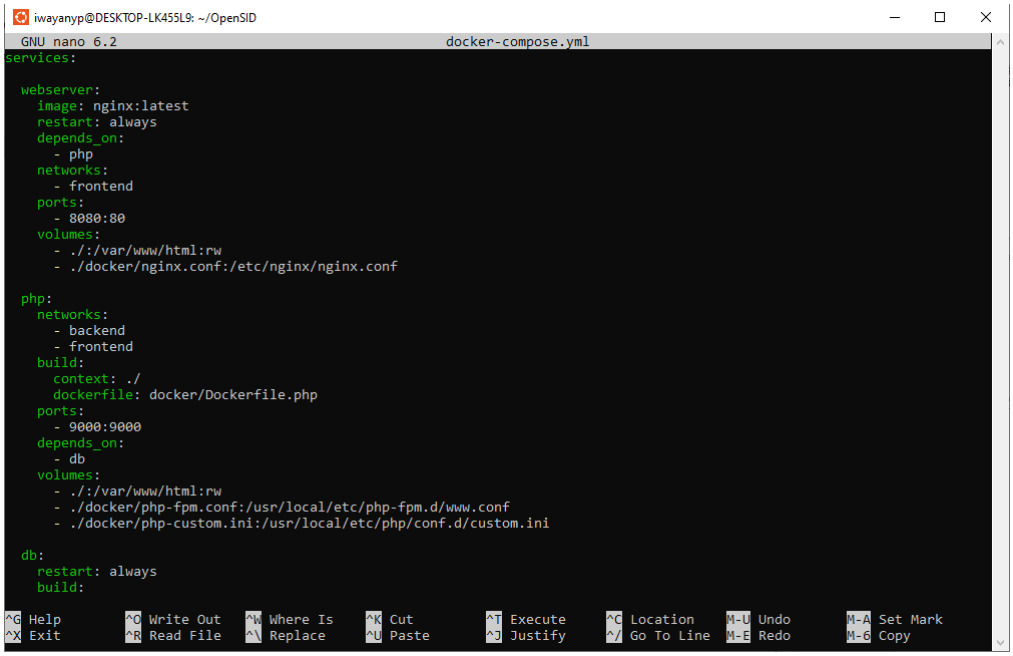

Figure 2. Docker-compose.yml

The success of the service orchestration in a production environment is verified through the state of the container running. The results of system monitoring showed that all four containers, operating stably with the appropriate port mapping configuration as presented in the Figure 3.

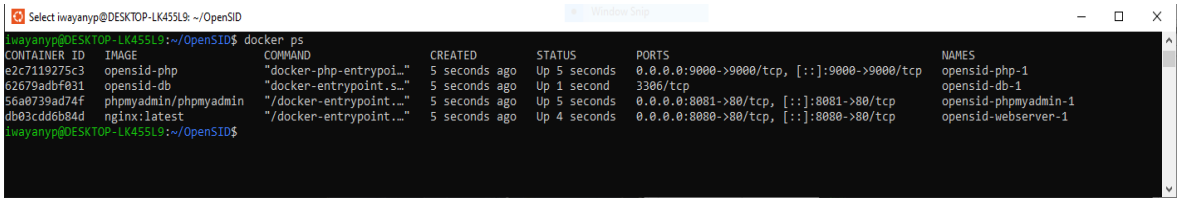

Figure 3. Container List

3. Result and Discussions

The results of the study show that the implementation of Container-Based Deployment on the OpenSID system has succeeded in mitigating the constraints of server environment inconsistency (environment drift) which has been the main obstacle in village digitalization. By using Dockerfile as Infrastructure as Code (IaC), all application dependencies are automatically configured, standardized, and isolated from the host environment.

3.1 Deployment Efficiency and System Consistency

Functionality testing showed that the use of containerization successfully met all system technical requirements (100%), including dependencies on PHP extensions and other system libraries. The integration of MariaDB and applications in a single Docker Compose ensures that system replication can be done quickly across a variety of different village servers with identical results. Deployment processes that previously took hours due to manual configuration can now be completed in minutes via automated build commands. The Comparative Analysis Table can be viewed at Table 1.

Table 1. Comparative Analysis

Features	Conventional Deployment	Container-Based Deployment
Installation Time	Old (Manual)	Fast (Automatic)
Environmental Consistency	Low (Environment Drift)	Height (Identical)
Resource Usage	High Overhead	Low/Efficient

3.2 Accessibility Analysis via Public IP and 8080 Port

The deployed system is accessed through localhost on port 8080. This method has proven to be effective in providing easy access for village devices without requiring complex reverse proxy configurations in the early stages of development. The stability of the system during the trial period was monitored very well, with efficient use of memory resources due to the absence of overhead from additional services outside the main container. The OpenSID application image can be seen in the image Figure 4.

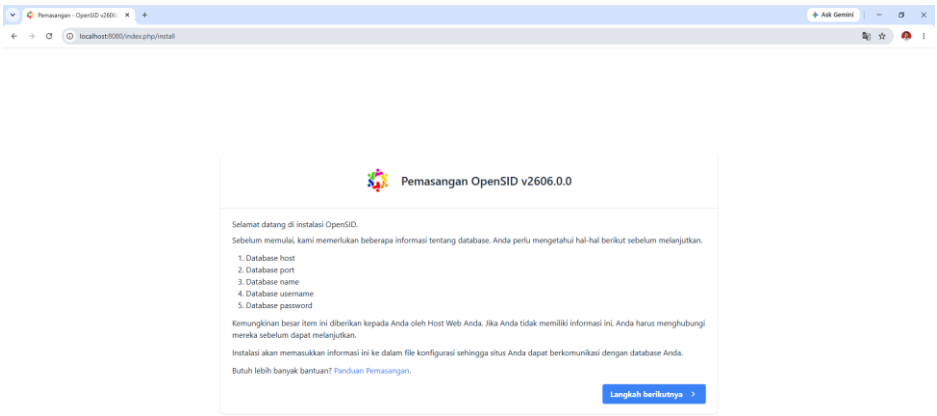

Figure 4. OpenSID Application

3.3 Discussion of Findings

The findings of this study confirm that the Infrastructure as Code (IaC) approach is able to significantly increase the reliability of public services at the village level. Compared to conventional installation methods, this approach offers a high level of portability, allowing the system to be moved to another server without the risk of malfunction due to different versions of the operating system or libraries. This result is in line with the Smart Village principle, where digital infrastructure must be able to support scalability and ease of maintenance by technical personnel at the regional level. The use of the 8080 port proved sufficient for the initial operational needs in this study, providing a balance between ease of implementation and system functionality.

4. Conclusion

Based on the results of the tests and implementations that have been carried out, this study concludes that the application of the Container-Based Deployment method using the Docker ecosystem is a very effective solution in standardizing the Village Information System (OpenSID) infrastructure. The use of Infrastructure as Code (IaC) through configuration file orchestration docker-compose.yml proven to eliminate manual

installation processes that are prone to human error, while significantly speeding up deployment times. Furthermore, breaking down the application architecture into isolated microservices services —which include PHP-based application processing, MariaDB databases, Nginx web servers, and phpMyAdmin management interfaces— comprehensively solves the problem of environmental drift. This approach ensures that all application dependencies are met absolutely, both in the development and production phases, without being affected by the basic operating system used on the local server. Overall, this standardized and portable infrastructure model provides a stable, scalable, and reliable operational foundation for village governments to accelerate the process of digitizing public services and supporting the realization of a smart village ecosystem. As a refinement step, further research is recommended to integrate a real-time monitoring system to monitor the utility of resources in each container. In addition, future development also needs to implement an automated backup mechanism and an advanced network security scheme to strengthen the resilience of village public service infrastructure in the long term.

Acknowledgment

Praise be to Ida Sang Hyang Widhi Wasa for completing the research on the implementation of Container-Based Deployment in the Village Information System (OpenSID). The author would like to express his deepest gratitude to the leadership and academic community of the Bintang Persada Institute of Technology and Health (ITEKES) for the support of the facilities and academic climate provided. Appreciation was also expressed to the OpenSID Developer Community for its dedication to providing comprehensive source code and technical documentation. Hopefully, this research will make a real contribution to accelerating the digitalization of public services in Indonesia.

References

[1] Ministry of Villages and Rural Development, "Technical Guidelines for the Digitalization of Village Governance in the Era of Digital Transformation," 2025.

[2] Ministry of Communication and Information of the Republic of Indonesia, "Blueprint for Digital Transformation of Local and Village Governments," 2024, *Ministry of Communication and Information Technology*.

[3] R. Wijayanti, "Adopting Open Source Technology to Realize Smart Villages: A Case Study of OpenSID Integration," *J. Sist. Inf. Manaj.*, vol. 10, no. 3, pp. 210–225, 2025.

[4] I. Setiawan, "Environment Parity Method to Overcome Deployment Problems in Village Applications," *J. Tek. Inform. and Sist. Inf.*, vol. 11, no. 3, pp. 310–325, 2024.

[5] T. Suryadi, "Digital Transformation of Village Government: Infrastructure Challenges and Human Resource Readiness," *J. Government Governance.*, vol. 5, no. 2, pp. 88–102, 2023.

[6] A. Nugroho, "Analysis of Server Infrastructure Efficiency Using Containerization Technology," *J. Techno. Inf.*, vol. 9, no. 4, pp. 150–165, 2022.

[7] S. Wibowo, "Implementation of Cloud Computing and Containerization in Supporting Smart Villages," *J. Techno. Common Inf.*, vol. 6, no. 2, pp. 95–110, 2022.

[8] S. Rahmawati, "The Application of Infrastructure as Code (IaC) for Village Information System Stability," *J. Inform. and Governance*, vol. 6, no. 3, pp. 200–215,

2024.

[9] B. Santoso and H. Wijaya, *DevOps for the Public Sector: Deployment Automation on Village Service Applications*. Informatics Publishers, 2025.

[10] R. Fauzi and A. Permana, "Implementation of Continuous Integration and Continuous Deployment (CI/CD) in the Development of Public Information Systems," *J. Compute. and Inform.*, vol. 15, no. 2, pp. 120–135, 2024.

[11] D. Lestari, "The Role of DevOps in Accelerating the Digitalization of Public Services in Indonesia," *J. Manaj. Technology.*, vol. 9, no. 1, pp. 12–28, 2023.

[12] L. Hakim, "Performance Evaluation of Container-Based Monolithic Architecture in Community Service Applications," *J. Software Engineering*, vol. 8, no. 1, pp. 45–56, 2023.

[13] A. Hidayat, "Comparative Analysis of Nginx and Apache Web Server Performance in a Docker Environment," *J. Techno. Computer.*, vol. 12, no. 1, pp. 45–58, 2024.

[14] A. Budiman and others, "Efficiency of Laravel-Based Application Deployment Using Docker on Local Servers," *J. Ilm. Tek. Inform.*, vol. 10, no. 1, pp. 30–42, 2023.

[15] D. Firmansyah and others, "Relational Database Scalability Management using Docker Compose in an Isolated Environment," *J. Computing Science. and Technology. Inf.*, vol. 12, no. 2, pp. 175–188, 2024.

[16] R. Pratama, "MariaDB Database Optimization in a Containerized Environment," *J. Data and Inf.*, vol. 7, no. 2, pp. 55–67, 2025.

[17] D. Kurniawan, "Microservices and Containerization Architecture in Supporting Village Digital Transformation," *J. Sist. Government Inf.*, vol. 8, no. 2, pp. 112–125, 2025.

[18] E. P. Putra, *Docker and Kubernetes Practical Guide for Indonesian Developers*. Elex Media Komputindo, 2022.

[19] M. Zulfikar, "Container-Based Infrastructure Security in the Digital Public Service Ecosystem," *Int. J. Cyber Secur.*, vol. 4, no. 1, pp. 10–22, 2026.

[20] OpenSID Community, "Technical Documentation and Standardization of OpenSID v24.x Deployment," 2026, *OpenSID Project*.