

Automated Container-Based Game Server Provisioning: Reducing Latency and Enhancing Security with Api- Driven Orchestration

Ketan Kirty¹, Mr. Milan Saxena²

^{1,2}*Final Year UG Student, AP Gr 3, School of Computer Science and Engineering, Galgotias University, UP, India*

Abstract- The growing popularity of sandbox and competitive gaming has increased the demand for configurable multiplayer game servers. Traditional provisioning methods rely on manual configuration, which adds complexity and leads to delays. This study examines how containerization and API-driven automation can significantly improve both provisioning efficiency and system security. The central research question asks whether automated container-based provisioning can reduce latency while maintaining scalability and security. Using a three-tier architecture built on AMD EPYC 9355 processors, the proposed solution achieves deployment times of approximately 12 seconds, compared to 15–30 minutes for conventional VPS setups. Security is enforced through token validation, firewall isolation, and multi-layered filtering. These findings confirm that the proposed approach is well-suited for real-time gaming infrastructure.

Abbreviations- API: Application Programming Interface; CSRF: Cross-Site Request Forgery; DDoS: Distributed Denial of Service, LEMP: Linux, Nginx, MySQL, PHP, VPS: Virtual Private Server, CPU: Central Processing Unit, EPYC: AMD Processor Family.

I. INTRODUCTION

As sandbox and competitive gaming settings have grown in popularity, so has the need for configurable multiplayer game servers. Conventional provisioning techniques require human configuration, which increases complexity and causes delays. This study explores how containerization and API-driven automation can greatly increase system security and provisioning effectiveness. How might automated container-based provisioning lower latency while preserving scalability and security? is the major research topic driving this study.

Container-based provisioning lowers overhead and enables nearly instantaneous deployment, in contrast to traditional virtualization. Previous methods have limited responsiveness to player demand and introduced overhead because they mostly relied on manual intervention or heavyweight virtual machines. The goal of this research is to provide a more safe and responsive provisioning architecture appropriate for real-time multiplayer game hosting by moving toward lightweight container orchestration and automated API operations.

II. LITERATURE SURVEY

1. Containerization Efficiency: Docker, a lightweight substitute for conventional virtualization with lower overhead and quicker deployment times, was first presented by Merkel (2014). Containerization dramatically increases resource utilization in cloud systems, according to later research by Pahl (2015).
2. Cloud Orchestration and Automation: Zhang (2018) and other surveys on cloud computing have emphasized how well automation systems work to manage scalable infrastructure. Lightweight container solutions for real-time game server provisioning, however, have not received much attention in research.
3. Hardware Optimization: According to AMD (2024), EPYC processors—especially the 9355 series—offer high core density and optimized parallel processing for applications that are containerized.
4. Infrastructure for Game Hosting: Contabo (2025) offers high-performance VPS and dedicated server solutions that are ideal for hosting game servers. These solutions feature strong network capabilities and DDoS protection, which are crucial for real-time gaming settings.

III. SYSTEM ARCHITECTURE

A three-tier concept is used in the demonstrated architecture:

- Web Layer: Manages payment processing and user interaction.
- Control Layer: Server administration and API orchestration.
- Containerized execution environment at the game node layer.

High core density and optimum performance for parallel container workloads are provided by the AMD EPYC 9355 processors that power the infrastructure and are housed on Contabo's high-performance VPS platform. Each tier may scale independently thanks to

this architecture, which guarantees the separation of concerns.

IV. METHODOLOGY

This study employs a mixed-methods approach, combining system implementation with quantitative performance measurement. The system is built on an LEMP-based stack with container orchestration tools, and it is deployed on Contabo VPS servers.

4.1 Provisioning Workflow

The user interacts with the Pterodactyl interface to start the server after the automatic provisioning procedure completes a secure, token-validated sequence. Figure illustrates the entire process.

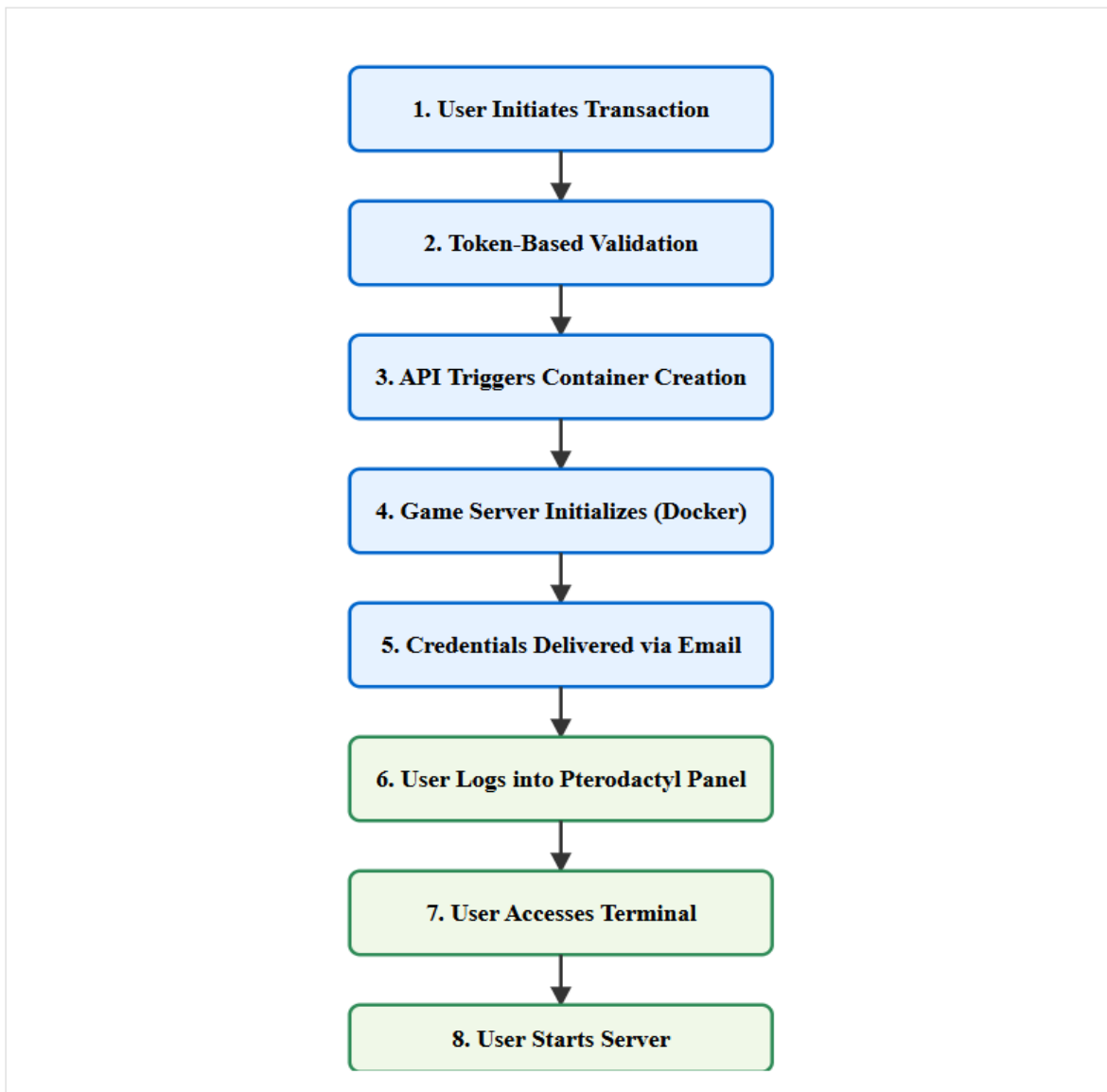


Figure 1: Complete Provisioning and User Activation Workflow

Using containerization and API orchestration hosted on Contabo infrastructure, the first five processes are completely automated. In order to access the container terminal and start the game server process, the user interacts with the Pterodactyl panel, a secure web-based administration interface, in steps 6–8. Strict firewall restrictions, token validation, and edge-based filtering during the automated phase are examples of security measures.

V. EXPERIMENTAL SETUP

AMD EPYC 9355 CPU-powered Contabo VPS instances, chosen for their high concurrent workload characteristics, were used for the experiments. Deployment time, CPU and memory usage, and system stability under concurrent provisioning requests were among the metrics assessed. Traditional VPS-based manual deployment was used as a baseline.

Metric	Traditional VPS	Proposed System
Deployment Time	15-30 minutes	~12 seconds
CPU Efficiency	Moderate	High (Container optimized)
Scalability	Limited	High

VI. RESULTS AND DISCUSSION

The suggested solution increases operational effectiveness and drastically lowers deployment latency. Even with concurrent provisioning workloads, container-based isolation guarantees consistent performance. Contabo's infrastructure's AMD EPYC 9355 architecture improves parallel processing capabilities, allowing for the effective management of numerous provisioning requests at once. These results extend the benefits of containerization into the real-time game server administration area and are consistent with previous studies on the subject.

Game server providers may now offer on-demand, transient environments without the burden of typical virtual machine provisioning thanks to the deployment time reduction from minutes to seconds.

VII. SECURITY ANALYSIS

The system's security relies on several layers working together: firewall isolation, edge filtering, and CSRF protection. Firewalls restrict access to only essential

services, while edge filtering blocks malicious traffic before it ever reaches the application. CSRF protection stops untrusted sources from making unauthorized changes. On top of that, token-based validation adds a trust check — resources only get allocated when requests are properly authorized. Together, these methods defend against common gaming infrastructure attacks like DDoS floods, session hijacking, and credential stuffing, all without hurting application-level performance.

VIII. LIMITATIONS

The system relies on third-party services like payment gateways and edge protection to operate effectively. While it works well under current conditions, it still needs to be tested under extreme workloads to ensure reliability at scale. At present, the implementation supports only one type of game server. Expanding it to handle multiple games will require additional development and configuration changes. This highlights a limitation in terms of flexibility. Future improvements will focus on increasing compatibility and scalability. Enhancing these areas will make the system more robust and widely usable.

IX. CONCLUSION

This study shows that the efficiency and scalability of gaming server provisioning are greatly increased when containerization and API-driven automation are combined. Performance under concurrent workloads is further enhanced by the usage of AMD EPYC infrastructure hosted on Contabo. To increase the system's applicability, future work will focus on developing predictive auto-scaling techniques and expanding multi-game functionality. This strategy provides a feasible route toward next-generation game hosting systems by resolving the latency and security issues present in manual provisioning.

REFERENCES

- [1] Merkel, D. (2014). Docker: Lightweight Linux Containers. *Linux Journal*.
- [2] Pahl, C. (2015). Containerization and the Cloud. *IEEE Cloud Computing*.
- [3] Zhang, Q. (2018). Cloud Computing: State-of-the-Art. *Journal of Internet Services*.
- [4] AMD. (2024). EPYC Processor Architecture Overview. *AMD Technical Documentation*.

- [5] Cloudflare. (2025). DDoS Mitigation Techniques. *Cloudflare Learning Centre*.
- [6] Contabo. (2025). High-Performance VPS and Dedicated Server Hosting. *Contabo Infrastructure Documentation*.