A Lightweight JupyterHub Testbed on k3s for AI and Networking Education in Low-Resource Settings: Experience from Senegal

Faly Gaye
faly.gaye@ucad.edu.sn
Université Cheikh Anta Diop
Dakar, Senegal

Bamba Gueye
bamba.gueye@ucad.edu.sn
Université Cheikh Anta Diop
Dakar, Senegal

Ibrahima Diane
ibrahima.diane@ucad.edu.sn
Université Cheikh Anta Diop
Dakar, Senegal

Damien Saucez
damien.saucez@inria.fr
INRIA
Nice, France

Nikos Makris
nimakris@uth.gr
University of Thessaly & CERTH
Volos, Greece

Abstract

Limited infrastructure and equipment hinder hands-on AI and networking education in resource-constrained environments. In this paper, we present a lightweight, sustainable testbed based on *JupyterHub* deployed on a *k3s*-managed edge/cloud continuum, designed to support practical training under low-bandwidth and intermittent connectivity conditions. The underlying *Blueprint* leverages a heterogeneous, low-cost architecture optimized for smartphone access and minimal bandwidth consumption. Compared to traditional *Docker*-based orchestration, our architecture significantly reduces *CPU* usage by 93 % for *Keycloak*, 82 % for *PostgreSQL*, and 74 % for *Redis* while maintaining reliable performance under constrained connectivity. Furthermore, the stack is replicable by any African university with modest hardware and connects natively to the SLICES-RI identity fabric via *Keycloak* federation. These results highlight the potential of lightweight, cloud-native infrastructures to democratize access to advanced digital skills and support scalable, context-aware education in underserved regions.

Keywords

Digital research infrastructure, Low-cost heterogeneous networking, Inclusive digital education, Reproducibility

1 Introduction

Despite the potential of digital research, the African continent is essentially faced with significant constraints such as limited access to reliable computing and connectivity, fragmented datasets, weak interoperability, and insufficient platforms to facilitate cross-border collaboration. Previous studies highlight persistent gaps in digital access and infrastructure readiness in developing regions [1, 2]. The *DIGITAfrica* [1] project, of which we are a partner, addresses this critical strategic gap by co-designing the foundations of a pan-African Research Infrastructure (*RI*) in Digital Sciences [2]. Its core objective is to enable the emergence of a pan-African *RI* in Digital Sciences, comparable in ambition, scope, and transformative impact to major European Research Infrastructures such as SLICES-RI [3] and SoBigData-RI [4].

Recently, multi-country assessments conducted under the *DIGITAfrica* project in five African countries (*i.e.*, Cameroon, Kenya, Senegal, South Africa, Tunisia) revealed that urban digital ecosystems remain fragmented, characterized by limited access to reliable connectivity, inadequate computing resources, and skills gaps [5]. Therefore, the *DIGITAfrica* project aims to deliver modular and open-source *Blueprints* [2] that serve as reference architectures which African research institutions can adopt, customize, and deploy according to their own priorities, operational contexts, and resource constraints. Furthermore, it can be connected with SLICES-RI [3], *NRENs*, and other African partner sites [2]. The proposed lightweight *Blueprint* backend, in conjunction with *JupyterHub*, facilitates multi-user access to notebooks. This enables students to engage in interactive computing environments for hands-on experimentation in networking and artificial intelligence, eliminating the need for local installation.

For instance, by leveraging the heterogeneous networking *Blueprint*, a lightweight cloud/edge educational laboratory with CI/CD automation is currently being deployed. The platform, designed for educational service delivery, is intended to be efficiently accessible from students' smartphones while

[1]https://www.digitafrica.eu/
[2]https://gitlab.inria.fr/digitafrica/blueprints/services/user-portal

minimizing bandwidth consumption and ensuring usability in resource-constrained environments. The proposed approach emphasizes affordability, resilience, and sustainability, while enabling scalable deployment of digital services and hands-on educational platforms in resource-constrained environments

The paper's contributions are twofold: (*i*) the design of a resilient, low-cost, and edge-enabled overall system architecture tailored to resource-constrained environments; and (*ii*) an experimental evaluation demonstrating the effectiveness of the proposed platform in enabling inclusive digital services, scalable hands-on learning, and digital skills development in underserved urban settings. The system is evaluated through real-world deployment scenarios, including a comparative analysis of container-based and lightweight orchestration approaches (*Docker* vs *k3s*) [6]. Experimental results demonstrate improved resilience, scalability, and accessibility, particularly in degraded connectivity conditions.

The paper subsequently details a system architecture for a Pan-African research infrastructure in digital sciences. It then outlines insights into system design trade-offs, deployment challenges, and user adoption in the Global South.

2 System architecture for a Pan-African Research Infrastructure in Digital Sciences

The heterogeneous networking *Blueprint* seeks to unify various communication technologies, including 5*G*, *Wi-Fi*, fiber, satellite, community mesh networks, and SDN-controlled switching into a coherent, manageable infrastructure. It enables the deployment and management of networks spanning these multiple technologies within a single, open, software-defined framework.

2.1 Brief overview of the approach

The *Blueprint* architecture follows a five-tier model ranging from micro-edge nodes (Tier 0) to a pan-African federation (Tier 4) to accommodate heterogeneous resource-constrained environments, from rural schools to national research infrastructures. At its core, a *k*3*s*-based *Kubernetes* cluster provides the orchestration backbone, enabling low-footprint, highly portable deployments that operate efficiently under intermittent connectivity and limited compute. The architecture combines lightweight *k*3*s* orchestration with Keycloak-based federated identity management to ensure service continuity under intermittent network connectivity, enabling automatic recovery of platform services within 15 seconds following transient network disruptions.

Each Tier maps to a specific *k*3*s* deployment profile: single-node lightweight control planes for Tier 0, multinode clusters with network policies for Tiers 1-2, and federated *k*3*s* instances across *WAN* for Tiers 3-4. This design ensures progressive scalability, offline-first operation, and minimal bandwidth consumption, directly supporting hands-on networking and AI/ML education in Senegalese and broader African contexts.

2.2 Blueprint architecture

Leveraging the *DIGITAfrica Blueprint*, this section describes the architecture of a lightweight cloud-edge platform deployed on a *k*3*s* multinode cluster, where *Keycloak* enables federated identity, authentication, and Single Sign-On (*SSO*) across distributed services.

Users access the system through a unified entry point (via dedicated portal), which exposes key tools such as *JupyterHub* for interactive experimentation. User environments are provisioned on demand as containerised workloads (e.g., notebook containers with sidecar services), providing isolation while enabling scalable and reproducible execution.

For instance, *k*3*s* and *JupyterHub* offer a comprehensive suite of capabilities, enabling users to: (*i*) conduct networking lab exercises, including simulations of Software-Defined Networking (*SDN*)/ Network Function Virtualisation (*NFV*) and cloud environments; (*ii*) execute AI models and develop distributed architectures; (*iii*) test federated learning scenarios. These capabilities are accessible without the need for an expensive laboratory or a stable high-speed internet connection. The designed and deployed *Blueprint* highlights the potential of lightweight, cloud-native infrastructures to democratize access to advanced digital skills and support scalable, context-aware education in the Global South.

Students at Cheikh Anta Diop University, called *UCAD*, can authenticate once and access SLICES-RI resources under their institutional identity, and vice versa for visiting European researchers accessing the *UCAD* platform. This bidirectional federation is implemented using *Keycloak* [7], an open-source identity and access management solution that provides *SSO*, authentication, and authorization across distributed applications and services.

3 Deployment Experience

This section presents the deployment experience and experimental evaluation, focusing on orchestration overhead, resource utilization, system behavior under load, reproducibility, and the operational differences between Docker Compose and a k3s-based infrastructure. All experiments were conducted on a virtual server running Ubuntu 22.04 *LTS* with 4 *vCPUs*, 7.8 *GB RAM*, and 100 *GB SSD* storage under a *KVM*-based virtualization environment. This configuration is representative of modest hardware commonly available in resource-constrained educational settings.

3.1 Docker-Based deployment

The initial learning infrastructure was deployed using Docker Compose with standalone containers for *Apache*, *Keycloak*, *PostgreSQL*, and *Redis*. The integration of *Apache*, *Keycloak*, *PostgreSQL*, and *Redis* ensures a balanced combination of security, persistence, and performance, providing a robust foundation for assessing responsiveness and latency in distributed systems. This lightweight model enabled rapid prototyping within a resource-constrained virtual infrastructure. Nevertheless, resource monitoring using `docker stats` revealed distinct runtime behaviors. *Apache* remained lightweight with negligible CPU usage and approximately 19 *MiB* memory consumption. *Keycloak* represented the primary resource consumer, maintaining stable memory usage around 676 *MiB* with occasional *CPU* peaks reaching 4.5%. *PostgreSQL* exhibited transient *CPU* spikes up to 10 % while maintaining approximately 49 *MiB* memory usage, whereas Redis showed short CPU bursts up to 5 % with minimal memory consumption around 7 *MiB*.

Although Docker Compose simplified deployment, several operational limitations emerged in multi-user scenarios. Containers shared host resources directly without Kubernetes-style resource governance, increasing the risk of workload contention under concurrent activity. Furthermore, *Docker Compose* lacked native orchestration capabilities such as automatic workload rescheduling, namespace isolation, and self-healing mechanisms.

3.2 k3s migration and identity federation

Docker Compose offers a lower entry barrier and is widely familiar to system administrators, making it particularly attractive for rapid prototyping and small-scale deployments. In contrast, while *k3s* introduces additional orchestration concepts, it remains lightweight, easy to deploy, and fully compatible with the Kubernetes ecosystem. This compatibility is a significant advantage, as it facilitates the adoption of cloud-native best practices and eases the transition from educational laboratories to production-grade research infrastructures. Consequently, our evaluation complements the quantitative analysis by recognizing the trade-off between operational simplicity and the advanced orchestration capabilities required for scalable, resilient, and reproducible digital infrastructures.

To overcome the limitations of the Docker Compose deployment, we migrated the platform to k3s. This migration was driven by the need for native orchestration capabilities, improved scalability, enhanced workload isolation, and greater deployment reproducibility in JupyterHub-based educational environments. Existing container images were preserved, while the deployment logic was reimplemented using *Kubernetes* manifests and *Helm* charts. Beyond enabling more efficient resource management, the migration also aligns the platform with cloud-native operational practices and facilitates future integration with federated research infrastructures and multi-node deployments.

The migration process required approximately two weeks, including networking adjustments, persistent storage configuration, and identity federation integration. *Keycloak* federation with the SLICES-RI Identity Provider through *OIDC* introduced several configuration challenges. The deployment required careful clock synchronization across machines and services due to *JWT* token expiration times depending on system time. This was necessary to avoid token expiration issues, ensure proper certificate trust validation between services, and correctly map attributes from the *UCAD* user schema to the claims expected by SLICES-RI. Documenting these deployment steps proved essential for ensuring reproducibility and reducing debugging complexity, thereby enabling other institutions to replicate the blueprint for their own educational infrastructure.

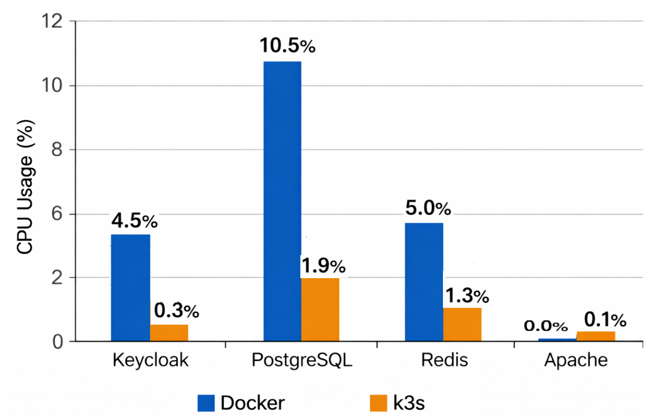

Figure 1: Comparative CPU usage between Docker and k3s.

3.3 Observed system behavior under load

Cluster-level monitoring using *kubectl* top confirms stable resource utilization under concurrent workloads. Specifically, *JupyterHub* pods consume approximately 195 *MiB* of memory, with *CPU* usage between 6–8 *m* (0.6–0.8%).

Additionally, Figure 1 depicts *CPU* usage comparison between *Docker* and *k3s*. Indeed, *Keycloak* pods maintain around 744 *MiB* of memory, with CPU usage ranging from 1–3 *m* (0.1–0.3%)). Besides, *PostgreSQL* pods peaked at 19 *m* (1.9%) CPU usage, whereas *Redis* pods reached approximately 13 *m* (1.3%).

Compared to traditional *Docker*-based orchestration as illustrated in Figure 1, our architecture significantly reduces *CPU* usage by 93 % for *Keycloak*, 82 % for *PostgreSQL*, and

74 % for *Redis* while maintaining reliable performance under constrained connectivity.

When contrasted with *Docker Compose*, the *k3s* deployment demonstrates both reduced and more stable *CPU* utilization. This improvement stems from *Kubernetes′* enhanced workload scheduling and resource management capabilities. Furthermore, namespace isolation and pod-level resource allocation provide cleaner workload separation, which is particularly beneficial for multi-user *JupyterHub* sessions where resource contention could otherwise degrade the user experience.

Although *Kubernetes* system services such as *CoreDNS*, *Traefik*, and the metrics server introduced additional memory overhead, the platform demonstrated improved operational stability and orchestration capabilities compared to the *Docker*-based deployment.

3.4 Resource utilization analysis

To evaluate the performance of our *k3s*-based architecture, we conduct cluster-level monitoring using *kubectl* top under concurrent multi-user workloads. Table 1 and Table 2 present the observed *CPU* usage for each service.

Table 1: Docker runtime resource consumption

Service	CPU Usage (%)	Memory Usage (MiB)	Observations
Apache	0.0	19	Lightweight frontend
Keycloak	0.06–4.5	676	Main memory consumer
PostgreSQL	Up to 10.5	49	Transient CPU peaks
Redis	0.8–5.0	7	Short CPU bursts

Table 2: k3s cluster resource consumption

Pod	CPU Usage (%)	Memory Usage (MiB)	Observations
JupyterHub Hub	0.6–0.8	195	Stable orchestration
Keycloak	0.1–0.3	744	Stable identity service
PostgreSQL	Up to 1.9	45	Reduced CPU usage
Redis	Up to 1.3	3	Lightweight runtime
CoreDNS + Traefik	≈ 0.2	40	Kubernetes overhead

Docker containers consistently showed higher *CPU* percentages, with *PostgreSQL* peaking at 10.5% and *Redis* at 5.0%. While this approach was sufficient for prototyping, the lack of resource governance led to contention under load. In contrast, *k3s* pods maintain significantly lower CPU utilization. Besides, *PostgreSQL* reaches only 1.9% and *Redis* 1.3%, reductions of 82% and 74%, respectively.

Compared to *Docker Compose*, the *k3s* deployment exhibited reduced and more stable CPU utilization, indicating improved workload scheduling and resource management enabled by Kubernetes orchestration. Furthermore, namespace isolation and pod-level resource allocation enhanced workload separation for multi-user *JupyterHub* sessions, preventing the resource contention observed in the Docker-based deployment.

These results demonstrate the efficiency of *Kubernetes* scheduling and resource isolation, validating that *k3s* provides a lightweight yet production-ready orchestration layer suitable for resource-constrained educational environments.

Table 3: Comparative analysis between Docker Compose and k3s

Feature	Docker Compose	k3s
Deployment Complexity	Low	Moderate
Resource Overhead	Low	Higher
Namespace Isolation	No	Yes
Automatic Scheduling	No	Yes
Self-Healing	No	Yes
Resource Quotas	Limited	Native Support
Multi-user Scalability	Limited	Improved
JupyterHub Integration	Basic	Kubernetes-native
Operational Stability	Moderate	Higher

Table 3 summarizes the main operational differences between *Docker Compose* and the *k3s* infrastructure. *Docker - Compose* provides a simple and lightweight deployment mechanism suitable for rapid prototyping. In contrast, *k3s* introduces orchestration capabilities that enhance workload isolation, scalability, and resilience. *Kubernetes*-native features such as namespace isolation, automatic scheduling, and *self-healingfacilitate* the management of concurrent *JupyterHub* sessions in a multi-user environment. Although *k3s* adds orchestration overhead through *Kubernetes* system services, it ensures more stable resource utilization and improved infrastructure management under concurrent workloads.

3.5 Replicability and broader impact

Each of the five African *DIGITAfrica* partners can deploy this stack as their local node, with *Keycloak* federating all nodes to SLICES-RI. Thus, the platform serves not merely as a local tool but as a blueprint for a federated African educational infrastructure. The complete replication package including *k3s* manifests, *Keycloakrealmexport* (exporting an identity management realm (users, roles, clients) from Keycloak), *JupyterHub Helm* (for *JupyterHub* deployment on *Kubernetes* via *Helm*) values, and a replication *README* is available at this URL [3].

4 Conclusion

We report on an ongoing real-world deployment, where the platform is being used to deliver large-scale hands-on training through a multi-access edge laboratory, supporting both AI and networking education. In Senegal, this deployment enables educational labs and CI/CD automation to provide *Jupyter* environments for students and reproducible training sessions.

[3]https://gitlab.inria.fr/digitafrica/blueprints/services/user-portal

By integrating *JupyterHub* with *k3s*, we establish a flexible and sustainable architecture that accommodates distributed workloads—including networking experiments and AI pipelines—while ensuring accessibility for users with limited bandwidth and heterogeneous devices such as smartphones.

Building on this open-source *Blueprint*, we are developing a multidisciplinary online learning and hands-on platform that supports practical training in cloud service automation, progressively simplified network deployment options, and modular AI/ML education.

Acknowledgments

The research leading to these results has received funding from the European Union, under the Grant Agreement No 101187966 (DIGITAfrica - Horizon Europe programme). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] P. D. Ba, B. Gueye, and I. Niang, "Adoption of distance learning at cheikh anta diop university during covid-19: responses and challenges," 2020, aCM Sigcomm Education Workshop and Community Discussion [Online] http://gaia.cs.umass.edu/sigcomm_education_workshop_2020/papers/sigcommedu20-final12.pdf.

[2] DIGITAfrica – Towards a Comprehensive Pan-African Research Infrastructure in Digital Sciences, "D1.1 demand analysis towards digitafrica," 2025, [Online] https://www.digitafrica.eu/wp-content/uploads/2025/07/DIGITAfrica_D1.1-Demand-analysis-towards-DIGITAfrica_web_disclaimer.pdf.

[3] S. Fdida, P. Andreou, N. Makris, D. Saucez, S. Gallenmüller, and B. Vermeulen, "Slices-ri pre-operation methodology and services," 2025, iEEE International Conference on Computer Communications and Networks (ICCCN).

[4] V. Grossi, B. Rapisarda, F. Giannotti, and D. Pedreschi, "Data science at sobigdata: the european research infrastructure for social mining and big data analytics," *International Journal of Data Science and Analytics*, vol. 6, no. 3, pp. 205–216, 2018.

[5] DIGITAfrica – Towards a Comprehensive Pan-African Research Infrastructure in Digital Sciences, "D4.1 guidelines female and young researchers," 2025, [Online] https://www.digitafrica.eu/wp-content/uploads/2026/01/DIGITAfrica_D4.1_Guidelines-female-and-young-researchers_web_disclaimer.pdf.

[6] S. Telenyk, O. Sopov, E. Zharikov, and G. Nowakowski, "A comparison of kubernetes and kubernetes-compatible platforms," *IEEE IDAACS Conference*, pp. 313–317, 2021.

[7] R. Hat, "Keycloak," 2023, [Online] https://www.keycloak.org.