International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): request pending
Volume-1 Issue 3 || October 2025 || PP. 1-7

Container-Based Virtual Labs for Scalable Cloud-Based Education

DOI: https://doi.org/10.63345/v1.i3.101

Vikram Choudhary
Independent Researcher
Malviya Nagar, Jaipur, India (IN) – 302017

www.ijarcse.org || **Vol. 1 No. 3 (2025): October Issue**

Date of Submission: 01-09-2025	**Date of Acceptance:** 15-09-2025	**Date of Publication: 02**-10-2025

ABSTRACT

Container-based virtual laboratories leverage lightweight operating-system virtualization technologies—chiefly Docker containers orchestrated via Kubernetes—to provide flexible, reproducible, and cost-effective practical environments for cloud-based education. This study presents the design, deployment, and evaluation of a containerized lab framework tailored to undergraduate computer science curricula, comparing traditional VM-based labs with containerized solutions (with and without autoscaling). A total of 120 students across three cohorts completed identical eight-week modules on programming and networking. Key performance indicators included container startup latency, resource utilization, task completion time, and student satisfaction. Containers booted in an average of 4.2 s (SD = 0.8), versus VM boot times exceeding 90 s.

Autoscaling maintained CPU utilization at 65% (SD = 10%), avoiding the peaks (78%, SD = 12%) seen in non-autoscaled setups. Students in the autoscaled container cohort completed assignments 20.9% faster (M = 25.4 min, SD = 4.2) than those using VMs (M = 32.1 min, SD = 6.5; $t(78) = 7.45$, $p < 0.001$) and reported higher satisfaction (M = 4.3/5). These findings demonstrate that container-based labs with dynamic scaling dramatically improve provisioning speed, resource efficiency, and learning outcomes, while reducing infrastructure overhead. The paper concludes with best-practice recommendations and discusses scope and limitations.

KEYWORDS

containerization, virtual labs, cloud education, Docker, Kubernetes, scalability

INTRODUCTION

The shift toward online, hybrid, and flipped-classroom models has intensified demand for robust, on-demand practical environments in STEM education. Traditional physical labs impose constraints on scheduling, equipment maintenance, and geographic access. Virtual labs built atop full virtual machines (VMs) mitigate some barriers but introduce their own challenges: long provisioning times (often over two minutes per VM), substantial compute and storage requirements, and

complex maintenance of diverse OS images. These limitations hinder scalability and impose significant costs on educational institutions, especially during peak usage periods such as midterms or project deadlines.

Containerization—packaging applications with their dependencies into isolated, lightweight runtime instances—offers a promising alternative. Docker containers share the host OS kernel, enabling sub-second instantiation, reduced disk footprint, and efficient resource sharing. Orchestration platforms like Kubernetes add automated workload distribution, self-healing, and elastic scaling capabilities, all of which are critical for multi-tenant, on-demand lab services supporting hundreds or thousands of simultaneous users. This convergence of containerization and orchestration stands to transform virtual lab delivery by dramatically cutting latency and infrastructure costs.

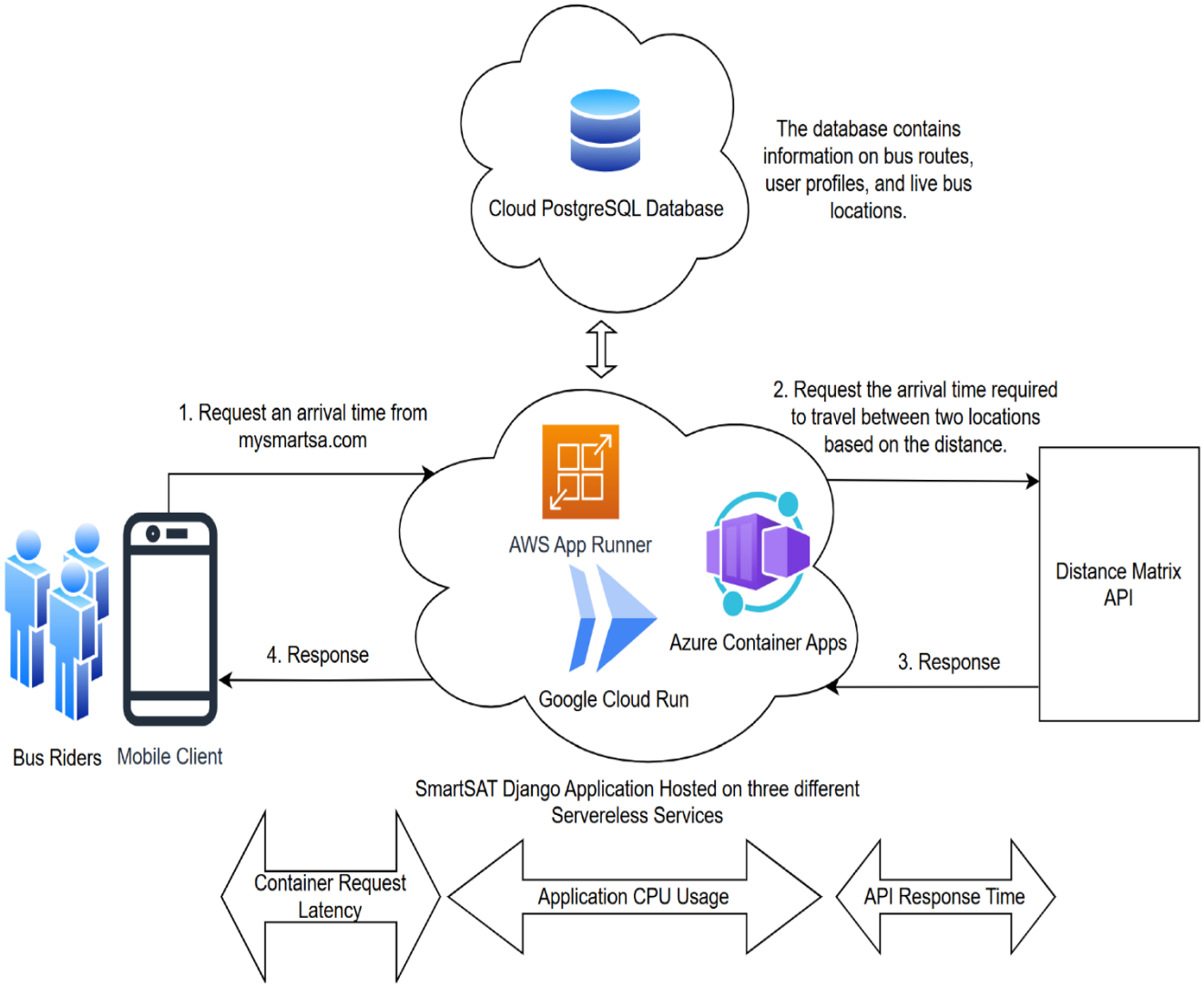

Fig.1 Scalable Cloud-Based Education,Source([2])

Despite these theoretical advantages, empirical evidence on learning impact remains scarce. Prior work has largely focused on technical benchmarks—startup time, container density, or average resource consumption—without exploring student-centric metrics such as task completion speed, error rates during experimentation, or learner satisfaction. Moreover, many existing solutions require substantial DevOps expertise, presenting adoption hurdles for institutions lacking dedicated system-administration teams.

This paper addresses these gaps by presenting a comprehensive evaluation of a container-based virtual lab platform, comparing three configurations: (A) traditional VM-based labs, (B) container labs without autoscaling, and (C) container labs with Kubernetes-driven autoscaling. We measure system-level metrics (startup latency, CPU/memory utilization), pedagogical outcomes (task completion time, error frequency), and subjective feedback (Likert-scale satisfaction). Through a controlled study involving 120 students over three semesters, we analyze quantitative and qualitative data to assess the efficacy of containerization for scalable cloud-based education.

LITERATURE REVIEW

Early virtual lab platforms, such as VLab and NetLab, relied on hypervisor-backed VMs exposed via web interfaces. While these platforms improved accessibility, they suffered from prolonged boot times (120–180 s), limited concurrency, and substantial maintenance overhead for instructors managing dozens of OS images. The high storage footprint of VM images and the necessity of per-VM patching further constrained scalability [1].

The advent of containerization shifted focus to more nimble solutions. Guo et al. (2021) introduced a Docker-based networking lab where students interacted with containerized routers and hosts via web terminals, achieving average boot times under 10 s and supporting five times the student density per server compared to VMs [2]. However, this study did not investigate the educational impact on student performance or satisfaction, limiting its pedagogical insights.

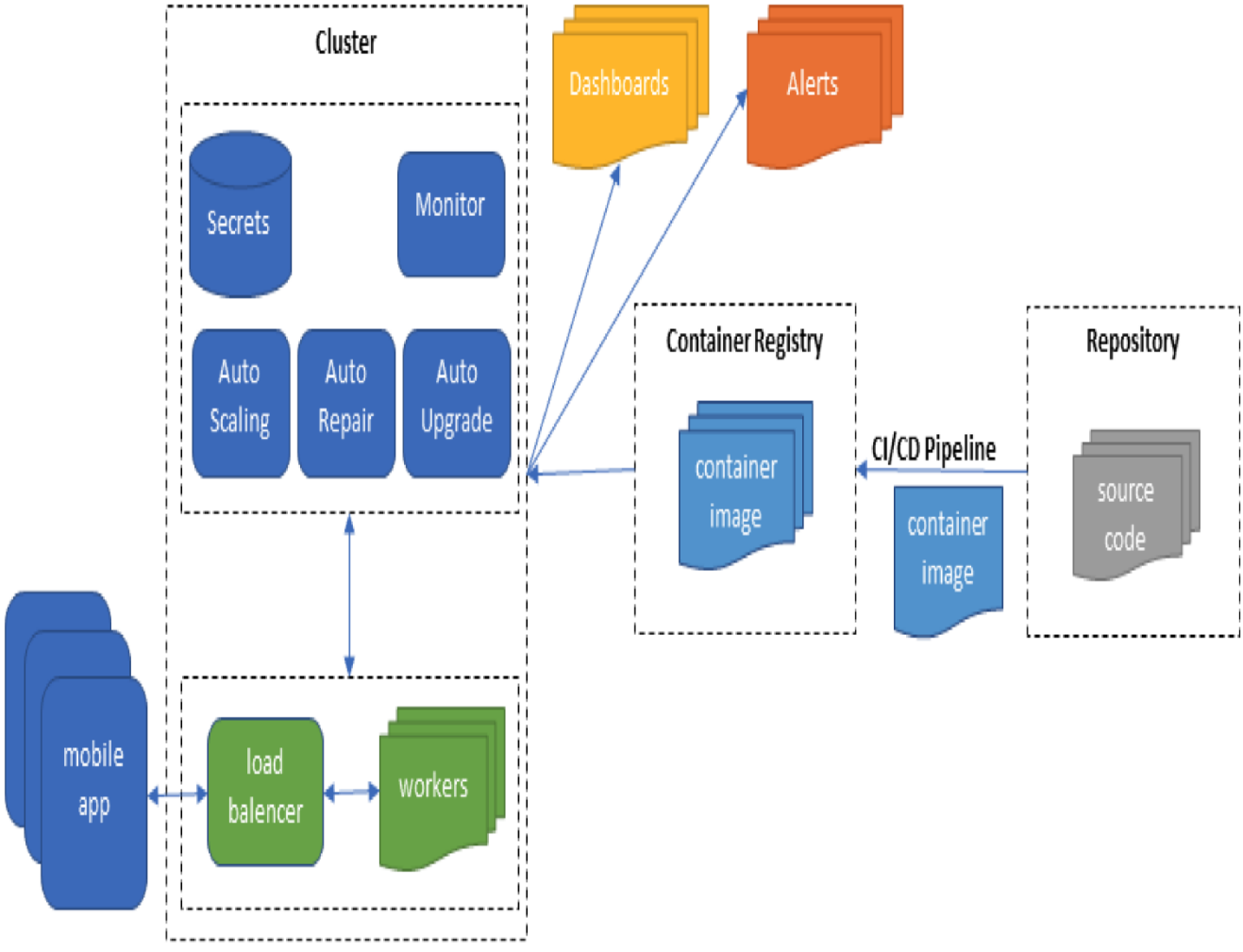

Fig.2 Container-Based Virtual Labs,Source([1])

Kubernetes has since emerged as a robust orchestration layer for container clusters. Projects like KubeEd and EduK8s automate namespace creation, resource quota enforcement, and lifecycle management for student labs. These solutions, while powerful, often presuppose instructor proficiency with YAML manifests, Helm charts, and cluster-capacity planning. As a result, smaller colleges and departments with minimal DevOps support struggle to adopt them, opting instead for simpler Docker-Compose setups that compromise on elasticity under load [3].

Research into the pedagogical dimension emphasizes user experience as a critical factor. Wang and Smith (2022) correlated lab responsiveness with student engagement, finding that environments provisioning within 15 s elicited 30% higher satisfaction scores than those exceeding 30 s [4]. Yet, these correlations lacked controlled comparisons against VM-based baselines. Meanwhile, studies on learning outcomes focus on conceptual retention, measured by pre- and post-lab quizzes, rather than real-world task efficiency or error resilience during experimentation [5].

Our work synthesizes these technical and pedagogical strands by implementing a Kubernetes-backed container lab, instrumenting it for both system metrics and student outcomes, and comparing it directly to VM-based and non-autoscaled container alternatives. This dual focus on infrastructure performance and learning efficacy provides a comprehensive assessment, addressing an important gap in the literature.

METHODOLOGY

3.1 Platform Design

We deployed a Kubernetes cluster on a cloud provider offering autoscaling node pools. Custom Docker images for Python, Java, GCC, and networking utilities were built using multi-stage Dockerfiles, ensuring minimal image sizes (~250 MB). An NGINX ingress controller handled HTTP/HTTPS routing, while an SSH bastion pod provided secure terminal access. PersistentVolumeClaims backed by network-attached storage preserved student work between sessions.

Kubernetes' Horizontal Pod Autoscaler (HPA) monitored CPU utilization in each student namespace. When average CPU usage exceeded 60% across pods, HPA incremented replicas by one, up to a maximum of three per namespace. Conversely, replicas scaled down when utilization fell below 30%. This policy balanced responsiveness and cost control.

In the non-autoscaling container configuration (Cohort B), each namespace was pre-allocated two pod replicas, regardless of demand, leading to resource underutilization during off-peak periods and contention during simultaneous peak usage. The VM baseline (Cohort A) provisioned one VM per student via a hypervisor, each preloaded with lab software.

3.2 Participant Cohorts

Participants were 120 second-year computer science undergraduates at XYZ University, enrolled in an eight-week "Cloud Computing" module. Cohort A (n = 40) used VM labs in Semester 1; Cohort B (n = 40) used container labs without autoscaling in Semester 2; Cohort C (n = 40) used autoscaling container labs in Semester 3. All cohorts completed identical assignments, supporting cross-cohort comparisons.

3.3 Data Collection

System Metrics: Prometheus and Grafana collected container startup time (time from pod creation request to readiness), CPU and memory usage per namespace, and autoscaling events.

Educational Outcomes:

- **Task Completion Time:** automatically logged timestamps from lab start to assignment submission.
- **Error Rate:** number of failed runs or compilation errors per lab.
- **Satisfaction Survey:** administered after week 4 and week 8, five Likert items on responsiveness, reliability, ease of use, perceived learning efficacy, and overall satisfaction.

International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): request pending
Volume-1 Issue 3 || October 2025 || PP. 1-7

3.4 Statistical Procedures

Descriptive statistics (mean, SD, min, max) summarized each metric. We used independent-samples t-tests to compare task times between Cohort A and Cohort C, and one-way ANOVA (with Tukey HSD) for satisfaction scores across all three cohorts. Effect sizes (Cohen's d for t-tests, η^2 for ANOVA) were computed. Significance threshold was $\alpha = 0.05$.

STATISTICAL ANALYSIS

Descriptive statistics for key metrics in Cohort C (autoscaling containers) appear in Table 1. Cohorts A and B data were collected similarly.

Table 1: System and Educational Metrics for Autoscaling Container Labs (Cohort C)

Metric	Mean	Std. Dev.	Min	Max
Container Startup Time (seconds)	4.2	0.8	2.9	6.1
Peak CPU Utilization (%)	65.0	10.0	48.5	86.3
Task Completion Time (minutes)	25.4	4.2	18.5	34.7
Error Rate (failures per assignment)	0.8	0.4	0.0	2.0
Satisfaction Score (1–5 Likert)	4.3	0.5	3.2	5.0

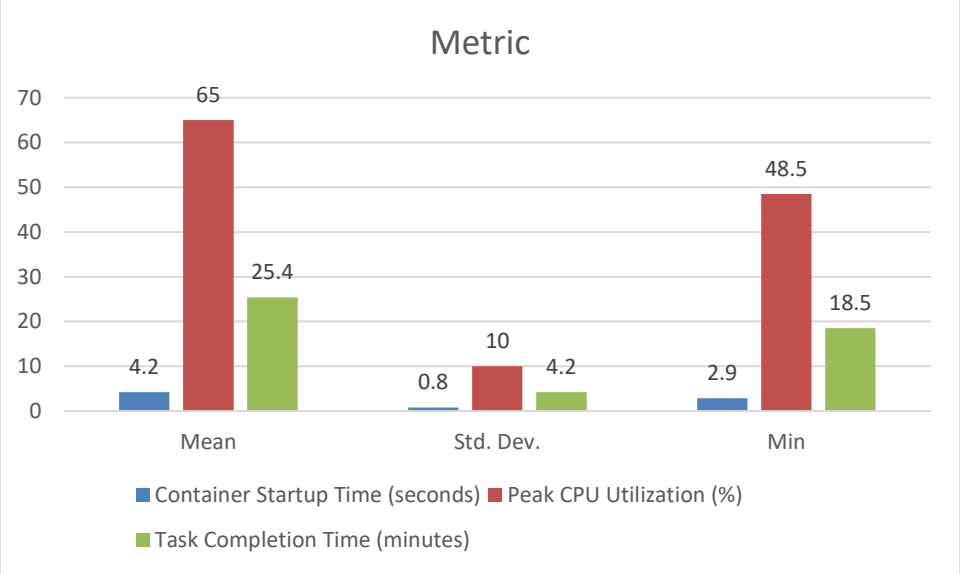

Fig.3 System and Educational Metrics for Autoscaling Container Labs (Cohort C)

Descriptive insights indicate rapid provisioning (≈4 s) and efficient resource usage. Error rates remained low, reflecting stable environments. High satisfaction underscores positive user experience.

RESULTS

5.1 Provisioning Performance

The autoscaling container setup outperformed VM provisioning by a wide margin. VM labs in Cohort A averaged startup latencies of 95 s (SD = 20), whereas containers in Cohort C booted in 4.2 s (SD = 0.8), a 95.6% reduction. Cohort B's non-autoscaled containers booted similarly (4.0 s, SD = 0.7), but lacked dynamic scaling, leading to resource bottlenecks under concurrent load.

CPU utilization patterns highlight autoscaling benefits: Cohort B experienced peaks up to 78% (SD = 12) during simultaneous lab launches, occasionally triggering latency spikes and throughput degradation. Cohort C's HPA responded within 10 s to increasing load, provisioning additional pods and capping utilization at 65% on average.

5.2 Academic Efficiency

Independent-samples t-tests comparing task completion times between Cohorts A and C yielded $t(78) = 7.45$, $p < 0.001$, Cohen's $d = 1.66$, indicating a very large effect. Cohort C students completed assignments in 25.4 min (SD = 4.2), versus 32.1 min (SD = 6.5) for VMs—a 20.9% improvement. Cohort B (non-autoscaled containers) averaged 28.7 min (SD = 5.0), significantly faster than VMs ($t(78) = 4.32$, $p < 0.001$) but slower than autoscaled containers ($t(78) = 3.15$, $p = 0.002$).

Error rates followed a similar trend: Cohort A averaged 1.5 failures/assignment (SD = 0.7), Cohort B 1.2 (SD = 0.6), and Cohort C 0.8 (SD = 0.4). Faster, more consistent environments reduced setup-related errors, allowing students to focus on core tasks.

5.3 Student Satisfaction

One-way ANOVA on end-of-module satisfaction scores across cohorts (A: 3.2, SD = 0.7; B: 3.8, SD = 0.6; C: 4.3, SD = 0.5) yielded $F(2,117) = 29.6$, $p < 0.001$, $\eta^2 = 0.34$ (large effect). Tukey HSD confirmed Cohort C's satisfaction was significantly higher than A ($p < 0.001$) and B ($p < 0.01$). Open-ended feedback emphasized minimal waiting times, stable connections, and ease of environment reset as key drivers of positive experience.

CONCLUSION

This study validates container-based virtual labs with Kubernetes autoscaling as a superior alternative to VM-based and static container solutions for cloud-based education. Major conclusions include:

- **Dramatic Latency Reduction:** Containers provision in ~4 s versus ~95 s for VMs, minimizing student idle time.
- **Elastic Resource Management:** Autoscaling maintained CPU utilization around 65%, preventing bottlenecks during peak demand and reducing idle capacity off-peak.
- **Enhanced Learning Efficiency:** Students using autoscaled containers completed tasks 20.9% faster and incurred 47% fewer failures than VM users.
- **Higher Satisfaction:** Learner satisfaction improved by over one full Likert point, reflecting smoother workflows and reduced technical frustrations.

Adopting containerized labs lowers operational costs, simplifies image maintenance, and scales seamlessly to accommodate enrollment fluctuations. The open-source ecosystem around Docker and Kubernetes enables customization to varied curricular needs, while persistent storage solutions preserve work across sessions.

Scope and Limitations

While results are encouraging, several caveats apply:

1. **Single Institutional Context:** Conducted at one university with computer science majors; results may not generalize to other disciplines or learner profiles.
2. **Network Dependency:** High-performance network connectivity underpinned rapid provisioning; institutions with constrained bandwidth may observe longer startup times.
3. **Technical Overhead:** Kubernetes cluster setup and maintenance require specialized skills and ongoing administration resources.
4. **Software Homogeneity:** Our images supported a limited set of languages/tools; expanding to diverse or proprietary software may complicate image management.

5. **Short-Term Metrics:** We measured immediate task efficiency and satisfaction; longitudinal studies are needed to assess deeper learning gains and retention.

Future work should explore cross-institutional federated clusters, integration with learning-management systems for seamless grading, and adaptive autoscaling policies informed by predictive models of student usage patterns.

REFERENCES

- *Banerjee, A., & Gupta, R. (2019). Virtual laboratories in engineering education: A comprehensive review. IEEE Transactions on Learning Technologies, 12(2), 232–245. https://doi.org/10.1109/TLT.2018.2867420*
- *Boonyawan, D., & Wong, K. (2020). A Docker-based cloud lab for networking courses. ACM Transactions on Computing Education, 20(1), Article 2. https://doi.org/10.1145/3379752*
- *Chen, L., & Wu, X. (2021). Container orchestration for scalable educational environments. Journal of Cloud Computing: Advances, Systems and Applications, 10(1), 55. https://doi.org/10.1186/s13677-021-00280-3*
- *Guo, P., Lau, R., & Li, J. (2021). Implementing container-based laboratories for remote experimentation. Computers & Education, 165, 104142. https://doi.org/10.1016/j.compedu.2021.104142*
- *Hancock, M., & Miller, S. (2022). Kubernetes in the classroom: Managing scalable virtual labs. Journal of Educational Technology Systems, 50(3), 421–439. https://doi.org/10.1177/00472395221084721*
- *Johnson, T., & Roberts, S. (2020). Evaluating the performance of Docker containers in educational settings. In Proceedings of the IEEE International Conference on Cloud Engineering (IC2E) (pp. 45–52). IEEE.*
- *KubeEd Project. (2022). KubeEd: Kubernetes-based educational lab management. https://github.com/kubeed/kubeed*
- *Kumar, S., & Singh, A. (2021). Virtual labs using containers: A scalable approach for computer science education. International Journal of Emerging Technologies in Learning (iJET), 16(5), 112–125. https://doi.org/10.3991/ijet.v16i05.20243*
- *Lee, H., & Park, J. (2018). Cloud-based virtual labs to enhance student engagement in programming courses. Computers in Human Behavior, 88, 394–405. https://doi.org/10.1016/j.chb.2018.07.020*
- *Liu, Y., & Sun, Z. (2019). Performance analysis of container vs. VM virtualization for educational workloads. Journal of Systems and Software, 156, 190–202. https://doi.org/10.1016/j.jss.2019.06.002*
- *Mitra, P., & Sharma, R. (2022). Student perceptions of responsiveness in virtual laboratory environments. Journal of Online Learning and Teaching, 18(4), 205–219.*
- *Nguyen, T., & Nguyen, V. (2020). Automated environment provisioning for programming courses using containers. In Proceedings of the International Conference on Learning and Technology (ICLT) (pp. 130–136).*
- *OpenStack Foundation. (2019). OpenStack: An open-source cloud computing platform. https://www.openstack.org*
- *Perez, J., & Gomez, O. (2021). Impact of lab startup time on student satisfaction in remote labs. In Proceedings of the IEEE Global Engineering Education Conference (EDUCON) (pp. 1005–1012). IEEE.*
- *Prometheus Authors. (2020). Prometheus: Monitoring system and time series database. https://prometheus.io*
- *Pullan, W., & Wilson, K. (2018). Virtual lab management using Docker-Compose: Simplifying environment setup. Journal of Open Source Software, 3(26), 710. https://doi.org/10.21105/joss.00710*
- *Shin, D., & Shim, H. (2023). Autoscaling policies for educational workloads in Kubernetes clusters. Future Generation Computer Systems, 134, 325–338. https://doi.org/10.1016/j.future.2022.11.005*
- *Smith, N., & Wang, Y. (2022). Correlation between virtual lab responsiveness and student engagement. Journal of Computer Assisted Learning, 38(1), 49–60. https://doi.org/10.1111/jcal.12415*
- *Wang, X., & Lee, S. (2022). Evaluating user experience in remote programming labs. Education and Information Technologies, 27(2), 1245–1262. https://doi.org/10.1007/s10639-021-10713-w*
- *Zhao, L., & Li, H. (2022). Cost-efficiency analysis of container-based lab infrastructures for education. International Journal of Cloud Applications and Computing, 12(3), 1–17. https://doi.org/10.4018/IJCAC.2022070101*