2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #51053

Scalable Lab Redesign for Virtualization Technology Education

Dr. Peng Li, East Carolina University

Peng Li received a Ph.D. in Electrical Engineering from University of Connecticut. Dr. Li is currently an Associate Professor at East Carolina University. He teaches undergraduate and graduate courses in programming, computer networks, information security, web services and virtualization technologies. His research interests include virtualization, cloud computing, cyber security and integration of information technology in education.

Scalable Lab Redesign for Virtualization Technology Education

Abstract

Virtualization technology is evolving rapidly, driven by containerization and open-source hypervisors that dominate modern enterprise environments. This work-in-progress paper describes a comprehensive redesign of a senior-level virtualization technology lab course to ensure technological currency, pedagogical rigor, and scalability. Eight scaffolded lab modules integrate hosted hypervisors, container management, KVM-based virtualization, automation with Ansible, and enterprise monitoring using Zabbix. The design leverages Canvas New Quizzes for immediate formative feedback, supported by evidence-based learning strategies and aligned with the ACM/IEEE Information Technology Curricula 2017 (IT2017) competency framework [7]. Labs are deployable across personal, institutional, and cloud infrastructures, reducing vendor dependency and aligning with industry practices. The open-source technology stack eliminates recurring vendor licensing costs and supports long-term curriculum sustainability. Spring 2026 deployment will include evaluation through faculty peer review and student surveys assessing usability, relevance, and technical clarity. This approach demonstrates how open-source, modular curricula can sustain long-term modernization in information technology education while preparing students for DevOps and cloud-native roles.

1. Introduction and Motivation

The curriculum of virtualization is facing more challenges as markets shift toward containerization (Docker, Podman) and open-source hypervisors (KVM, Xen) alongside legacy proprietary solutions. Recent surveys confirm that enterprises increasingly adopt both containers and virtual machines as complementary technologies, with open-source platforms gaining significant market share [11]. In our Information and Cybersecurity Technology (ICT) Program, the course used to rely heavily on vendor-centric labs (VMware vSphere), creating high costs, limited flexibility, and student gaps regarding open-source alternatives now dominant in enterprise environments [1].

After the closure of VMware IT Academy, we secured a University Teaching Grant to revamp our virtualization technology lab course. This redesign addresses these challenges through 1) Technology Currency: comprehensive coverage of both container and hypervisor-based virtualization; 2) Pedagogical Evidence: immediate feedback supported by learning science research; 3) Scalability: modular labs deployable across platforms and reusable across courses; and 4) Automation: Infrastructure-as-Code approaches reflecting DevOps industry practice.

The need for this redesign is underscored by broader trends in information technology education. The ACM/IEEE Information Technology Curricula 2017 (IT2017) framework adopts a competency-based approach emphasizing that IT graduates must demonstrate knowledge, skills, and dispositions in professional contexts [7]. IT2017 identifies essential domains including Platform Technologies (ITE-PFT) and Networking (ITE-NET), as well as supplemental domains such as Virtual Systems and Services (ITS-VSS) and System Administration and Maintenance (ITS-SAD), all directly addressed by our redesigned lab sequence. Recent studies on container-based virtualization in educational settings further highlight the pedagogical value of exposing students to modern deployment tools and workflows [9]. Our redesign responds to these calls by integrating open-source tools that reflect current enterprise practice while maintaining a scaffolded pedagogical structure grounded in established learning theory.

2. Course Redesign: Lab Modules and Technology Stack

The course Virtualization Technologies is a senior-level course offered every spring. The lab redesign project started in summer 2025 and is expected to conclude in spring 2026. The new labs are composed of eight progressive modules (Table 1), scaffolded from foundational hosted hypervisors through advanced automation and monitoring. The technology stack, KVM, Podman, Buildah, Ansible, Zabbix, Google Cloud Platform, reflects current market trends and enables sustainability without vendor licensing constraints.

Labs 1-2 provide foundational understanding through hosted hypervisors and containers; Labs 3-5 deepen knowledge of container management and KVM hypervisor fundamentals; Labs 6-8 develop professional automation and monitoring competencies. This progression aligns with Vygotskian scaffolding, where students progress from modeling and coaching to articulation and reflection [4].

The labs explicitly address gaps in previous vendor-centric curricula: Podman/Buildah (Lab 3) expose students to open-source container alternatives; QEMU/KVM/libvirt (Labs 4-5) cover KVM-based virtualization architecture; Ansible (Lab 7) demonstrates Infrastructure-as-Code in realistic DevOps workflows; Zabbix (Lab 8) teaches enterprise-grade monitoring: all skills critical in current job markets [2]. This technology selection directly maps to IT2017 competency domains: Platform Technologies (ITE-PFT) covering operating system administration and scripting, Virtual Systems and Services (ITS-VSS) addressing virtualization architectures, and System Administration and Maintenance (ITS-SAD) encompassing infrastructure automation and monitoring [7].

Table 1: Eight Lab Modules: Scaffolded Progression from Fundamentals to Enterprise Practice

Lab	Title	Core Topics Addressed
1	Using Hosted VMware Hypervisor to Install Rocky Linux	Hosted hypervisor fundamentals on personal computers
2	Using Docker Containers on Google Cloud Platform	Containers on cloud infrastructure; KVM verification
3	Using Container Management Tools Buildah and Podman	Container image creation; rootless, daemon-free management
4	Installing QEMU/KVM and libvirt	Hardware virtualization; VM lifecycle; libvirt abstraction
5	KVM Virtual Machines: Installation and Configuration	VM installation, storage, NFS export, snapshots
6	Automating KVM VM Provisioning using Shell Scripting	Infrastructure-as-Code; automated provisioning; nested VMs; AI-assisted coding
7	Automating KVM VM Configuration Using Ansible	Configuration management; declarative playbooks; idempotency

8	Monitoring the KVM Virtualization Platform Using Zabbix	Enterprise monitoring; agent deployment; alerting; dashboards

When students perform the labs, they submit lab results using New Quizzes on Canvas. The lab quizzes are created together with lab manuals. Many tasks are automatically graded while some questions (e.g., file uploads) are manually graded. Educational research demonstrates that automated grading and immediate feedback significantly enhance learning outcomes, reduce anxiety, and improve retention in technical education [3], [8], [10]. Our design implements Canvas New Quizzes with advanced answer matching techniques: Regular Expression Match for validating command outputs, IP addresses, and configuration snippets; Levenshtein Distance for flexible matching of responses with minor typos or syntax variations [5]. This innovation addresses the longstanding challenge of scaling formative assessment in technical education without overwhelming faculty [8], [10].

By allowing students to iteratively correct mistakes and resubmit lab work, the design supports trial-and-error learning rooted in self-regulated learning theory, where error recovery and strategy adjustment develop metacognitive skills essential for independent problem-solving.

The lab sequence implements Vygotskian scaffolding [4] and Wood, Bruner, and Ross's tutoring framework [6]. Labs 1-2 model foundational concepts (hosted hypervisors, cloud containers), Labs 3-5 provide coaching through increasing complexity (container management, KVM fundamentals), Labs 6-8 require articulation and reflection through automation and monitoring design. This progression progressively releases responsibility to students, supporting deeper understanding and confidence development.

3. Technical Implementation and Industry Alignment

This section details the practical implementation of the redesigned lab modules and their alignment with current industry standards. Each lab is structured to progressively build technical competencies, starting from foundational virtualization concepts and advancing toward enterprise-level automation and monitoring practices.

Lab 1: Using Hosted VMware Hypervisor to Install Rocky Linux. Students install VMware Workstation or Fusion (available free for educational use) and deploy a Rocky Linux virtual machine on their personal computers. This introductory lab establishes foundational concepts of hosted (Type 2) hypervisors that run as software applications atop host operating systems, while gaining practical experience with VM lifecycle management, resource allocation, and guest OS installation, essential prerequisites for understanding more complex virtualization architectures. The hands-on experience with commercial hypervisor software provides valuable context for comparing against open-source alternatives introduced in later labs.

Lab 2: Using Docker Containers on Google Cloud Platform. Students provision Container-Optimized OS instances on Google Compute Engine while verifying that GCP's underlying infrastructure uses the KVM hypervisor. Through Docker container deployment (nginx, CirrOS), students learn operating system-level virtualization concepts, cloud networking, and firewall configuration, directly bridging personal computing experiences (Lab 1) to cloud infrastructure.

This lab demonstrates how containers and VMs coexist as complementary technologies in production cloud architectures.

Lab 3: Using Container Management Tools Buildah and Podman. Students use Buildah to create customized container images and Podman for rootless container execution. Unlike Docker's daemon-based architecture, Podman's daemonless design improves security for non-root users, a strategic advantage reflected in Red Hat's shift away from Docker toward Podman. This lab exposes students to production-standard tools increasingly dominant in enterprise container deployments.

Lab 4: Installing QEMU/KVM and libvirt. Students install QEMU/KVM and libvirt on AlmaLinux, transitioning from hosted to bare-metal (Type 1) virtualization. Through hardware detection, kernel module verification, and network configuration, students understand the architectural differences between hypervisor types while establishing hands-on competency with industry-standard open-source virtualization. They configure user permissions, manage virtual networks, create their first KVM-based VMs using virt-install, and develop foundational knowledge essential for infrastructure automation and monitoring in subsequent labs.

Lab 5: KVM Virtual Machines: Installation and Configuration. Building on Lab 4 foundations, students use virt-manager to deploy Rocky Linux virtual machines with advanced enterprise features: virtual storage management and NFS sharing, snapshot creation and rollback for system recovery. These hands-on experiences with production-grade VM operations demonstrate recovery and resilience patterns fundamental to enterprise infrastructure administration while preparing students for roles managing virtualized environments in data centers.

Lab 6: Automating KVM VM Provisioning using Shell Scripting. Students analyze comprehensive provisioning scripts demonstrating professional automation practices, idempotent operations safe to re-run without side effects, robust error handling, automated verification, and nested VM deployment, representing real DevOps workflows essential for modern infrastructure teams. Through examining scripts that validate prerequisites, configure KVM/libvirt automatically, deploy multiple Ubuntu VMs from cloud images, and implement automated testing, students practice iterative refinement with modern AI tools that accelerate script development. This lab bridges manual operations to Infrastructure-as-Code while introducing AI-assisted coding practices increasingly common in professional DevOps workflows.

Lab 7: Automating KVM VM Configuration Using Ansible. Students create Ansible projects with static inventories, configuration files, and playbooks to automate multi-server configuration on Ubuntu VMs. Through ad hoc commands and declarative playbooks, students learn infrastructure-as-code principles, privilege escalation, and idempotent task design that enables safe, repeatable deployments across diverse environments. Ansible represents production-grade open-source configuration management widely adopted in enterprise and cloud-native environments, directly preparing students for infrastructure automation roles.

Lab 8: Monitoring the KVM Virtualization Platform Using Zabbix. Students deploy Zabbix Server on KVM infrastructure and configure Agent 2 on managed hosts, implementing complete monitoring workflows: server deployment, agent installation, custom trigger creation for infrastructure alerts (CPU usage, disk space, memory thresholds), baseline establishment, and dashboard visualization. Zabbix represents production-grade open-source monitoring comparable to commercial tools and is increasingly adopted in enterprise environments. This lab prepares students for real-world infrastructure operations and on-call engineering roles by providing hands-on experience with enterprise monitoring platforms.

These eight labs demonstrate how the sequence progresses from foundational concepts (hosted hypervisors, containers) to enterprise-scale technical competencies (automation, monitoring) aligned with current job market demands.

4. Cost, Sustainability, and Maintenance Considerations

A key motivation for this redesign is the elimination of recurring vendor licensing costs. The previous VMware vSphere-based curriculum required annual academic licensing fees, and the closure of VMware IT Academy created both financial and logistical disruption. By transitioning to an entirely open-source technology stack (KVM, Podman, Buildah, Ansible, Zabbix), the redesigned curriculum incurs zero software licensing costs. The only external cost is Google Cloud Platform usage for Lab 2, which is covered by GCP for Education credits available to academic institutions at no charge.

The modular design supports long-term maintenance by isolating each lab as an independent unit that can be updated individually without disrupting the entire sequence. As technologies evolve, for example, if a new container management tool emerges, a single lab module can be revised or replaced while the remaining seven labs remain unchanged. This modularity also enables reuse across related courses (Cloud Computing, Systems Administration, Cybersecurity), maximizing the return on curriculum development investment.

Technical support requirements are minimized by packaging Labs 3-8 within a single AlmaLinux 10 virtual appliance. This pre-configured environment reduces setup complexity for students and standardizes the lab infrastructure across different deployment platforms (personal computers, institutional private cloud, public cloud). Faculty time is further reduced through the automated grading capabilities of Canvas New Quizzes, which handle the majority of routine assessment tasks and allow instructors to focus on higher-order pedagogical activities.

5. Deployment and Evaluation in Spring 2026

All eight lab manuals have been created by the beginning of the spring 2026 semester, and lab quizzes have been set up on Canvas. Lab 1 is to be performed on students' own personal computers. Lab 2 is to be conducted on Google Cloud Platform (GCP). Labs 3-8 are hosted inside a lab environment within a single AlmaLinux 10 virtual appliance, which can be deployed locally or remotely on x64 hosts with hardware virtualization support.

Deployment Timeline and Migration: The next phase in our deployment involves migrating the lab environment to the college's private cloud system (VMware Cloud Director), followed by comprehensive functionality and integration testing. This transition from local development to shared institutional infrastructure ensures scalability and supports multi-cohort deployment. Full course deployment is scheduled for Spring 2026 with one student cohort ($n \approx 25$-30), enabling detailed evaluation and refinement before broader adoption.

Faculty Peer Review: Two ICT faculty members with expertise in technical curriculum will conduct comprehensive reviews of all eight lab manuals and associated assessment materials. Using established educational rubrics aligned with ICT pedagogical standards, peers will evaluate: 1) Lab design quality and pedagogical soundness relative to learning objectives; 2) Learning objective clarity and measurability; 3) Technical accuracy and currency with industry practice; 4) Assessment instrument validity for measuring stated outcomes; and 5) Feasibility of learning objectives given typical student background, lab duration, and institutional resources. Faculty feedback will inform refinements for Year 2 offering.

Student Experience Survey: Anonymous, voluntary survey administered via Qualtrics in the final two weeks will assess student perceptions across four dimensions: 1) Usability: clarity of lab documentation, technical accessibility of virtual environment, adequacy of support materials; 2) Relevance: perceived industry alignment of labs, confidence in knowledge application, career preparation; 3) Technical Clarity: understanding of virtualization concepts, effectiveness of scaffolded progression, clarity of Canvas feedback; and 4) Virtual Environment: accessibility and responsiveness of infrastructure, integration with Canvas assessment tools. Survey employs 5-point Likert scales with open-ended follow-up questions.

The results of deployment and evaluation are pending and will be reported after they become available.

6. Conclusion

This lab redesign illustrates how technical education can remain current with industry trends while maintaining evidence-based pedagogy. Key innovations include: Immediate Feedback through Canvas New Quizzes with advanced answer matching (regex, Levenshtein distance) to scale formative assessment [8], [10]; Open-Source Adoption transitioning from vendor-centric labs to sustainable, enterprise-aligned tools such as Podman, Ansible, and Zabbix with zero recurring licensing costs; Scaffolded Progression structured using Vygotskian scaffolding and tutoring frameworks [6] to support progressive skill development; and Industry Alignment exposing students to production-standard workflows, automation, and monitoring practices critical for modern infrastructure roles.

Spring 2026 deployment with evaluation, faculty peer review and student surveys, will provide empirical evidence on usability, relevance, and learning outcomes. Findings will inform Year 2 refinements and contribute to scalable, evidence-based practices in virtualization and cloud computing education.

The modular lab design facilitates deployment across diverse infrastructure platforms (personal computers, institutional private cloud, public cloud) and enables reuse in related computing courses (Cloud Computing, Systems Administration, Cybersecurity). This maximizes return on curriculum development investment and supports institution-wide modernization efforts. This approach extends to other high-demand, rapidly evolving technical domains: DevOps, cloud computing, cybersecurity, data engineering. Open-source curricula provide institutional independence, reduced licensing burden, and long-term sustainability compared to vendor-centric programs. The assessment innovation (automatic grading with flexible answer matching) demonstrates scalable approaches to formative feedback in online technical education, addressing a persistent challenge in engineering education [8].

References

[1] H. Rajaei and E. A. Aldakheel, “Cloud Computing in Computer Science and Engineering Education,” presented at the ASEE Annu. Conf., San Antonio, TX, USA, 2012.

[2] P. Li, “Public and Private Clouds in Hands-on, Online Learning,” presented at the ASEE Mid-Atlantic Spring Conf., Washington, DC, USA, 2018.

[3] V. J. Shute, “Focus on formative feedback,” Rev. Educ. Res., vol. 78, no. 1, pp. 153–189, 2008.

[4] L. S. Vygotsky, Mind in Society: The Development of Higher Psychological Processes. Harvard Univ. Press, Cambridge, MA, 1978.

[5] S. Narciss, “Feedback strategies for interactive learning tasks,” in Handbook of Research on Educational Communications and Technology, 3rd ed. Springer, New York, 2008, pp. 125–144.

[6] D. J. Wood, J. S. Bruner, and G. Ross, “The role of tutoring in problem solving,” J. Child Psychol. Psychiatry, vol. 17, no. 2, pp. 89–100, 1976, doi: 10.1111/j.1469-7610.1976.tb00381.x.

[7] M. Sabin, J. Impagliazzo, H. Alrumaih, B. Viola, et al., “Information Technology Curricula 2017: Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology (IT2017),” ACM/IEEE-CS, 2017. [Online]. Available: https://dl.acm.org/doi/book/10.1145/3173161

[8] M. Messer, N. C. C. Brown, M. Kölling, and M. Shi, “Automated Grading and Feedback Tools for Programming Education: A Systematic Review,” ACM Trans. Comput. Educ., vol. 24, no. 1, pp. 1–43, 2024, doi: 10.1145/3636515.

[9] T. Tsuchiya, M. Matsunaga, H. Hirose, T. Yamada, H. Ichikawa, and Q. T. Minh, “Study on Web Based Virtualized Containers for Software Development Lectures in BYOD,” in Future Data and Security Engineering (FDSE 2023), Communications in Computer and Information Science, vol. 1925, Springer, 2023, pp. 289–302, doi: 10.1007/978-981-99-8296-7_21.

[10] J. C. Paiva, A. Figueira, and J. P. Leal, “Automated Assessment in Computer Science Education: A State-of-the-Art Review,” ACM Trans. Comput. Educ., vol. 22, no. 3, pp. 1–40, 2022, doi: 10.1145/3513140.

[11] A. Bhardwaj and C. R. Krishna, “Virtualization in Cloud Computing: Moving from Hypervisor to Containerization—A Survey,” Arab. J. Sci. Eng., vol. 46, pp. 8585–8601, 2021, doi: 10.1007/s13369-021-05553-3.