Online Self-Service Learning Platform for Application-Inspired Cloud DevOps Curriculum

[1]Roshan Lal Neupane, [1]Prasad Calyam, [1]Songjie Wang, [1]Kiran Neupane, [1]Ashish Pandey, [1]Xiyao Cheng, [1]Durbek Gafurov, [1]Hemanth Sai Yeddulapalli, [2]Noah Glaser, [2]Kanu Priya Singh, [2]Yuanyuan Gu, [2]Shangman (Eunice) Li, [34]Sharan Srinivas
[1]Department of Electrical Engineering; [2]Information Experience Lab; [3]Department of Industrial and Systems Engineering, [4]Department of Marketing, University of Missouri; Email: {neupaner, calyamp, wangso, kngbq, hygw7, noah.glaser, srinivassh} @missouri.edu; {apfd6, xcheng, kanupriya.singh, sli, durbek.gafurov}@mail.missouri.edu; yggcc@umsystem.edu

***Abstract*—Cloud-hosted services are being increasingly used in hosting business and scientific applications due to cost-effectiveness, scalability, and ease of deployment. To facilitate rapid development, change and release process of cloud-hosted applications, the area of Development and Operations (DevOps) is fast evolving. It is necessary to train the future generation of application development professionals such that they are knowledgeable in the DevOps-enabled continuous integration/delivery automation. In this paper, we present the design and development of our "Mizzou Cloud DevOps platform", an online self-service platform to learn cutting-edge Cloud DevOps tools/technologies using open/public cloud infrastructures for wide adoption amongst instructors/students. Our learning platform features scalability, flexibility, and extendability in providing Cloud DevOps concepts knowledge and hands-on skills. We detail our "application-inspired learning" methodology that is based on integration of real-world application use cases in eight learning modules that include laboratory exercises and self-study activities. The learning modules allow students to gain skills in using latest technologies (e.g., containerization, cluster and edge computing, data pipeline automation) to implement relevant security, monitoring, and adaptation mechanisms. The evaluation of our platform features a knowledge growth study to assess student learning, followed by a usability study to assess the online learning platform as well as the curriculum content as perceived by instructors and students across multiple hands-on workshops.**

***Index Terms*—learning platform, microservices architecture, containerization, cluster management, infrastructure monitoring, continuous integration, continuous deployment**

I. Introduction

The term "DevOps," which has been in existence for nearly a decade, was originally coined to amalgamate established practices in software development (Dev) and IT operations (Ops) [1]). While various definitions of DevOps exist, its fundamental objective remains consistent: to automate and harmonize the processes of development and operations within agile software development and operational teams. This synchronization enables these teams to collaboratively create, test, and deploy software more swiftly and reliably. The overarching aim of DevOps is to condense the software development life cycle while ensuring that software of the highest quality, with minimal defects, is made available to users. This is achieved through the automation of integration, testing, maintenance, and delivery automation.

With the growth in digital transformation among many organizations, the DevOps practice is commonly performed through the usage of cloud computing resources. For example, about 80% of organizations are considering to use "Cloud DevOps" to transform their current manual and siloed processes used in evolving and scaling data/compute-intensive applications [2]. The overarching benefit of Cloud DevOps is that it enables essential components, including humans, culture, platform, and tools to work together in order to reduce friction in the processes between software/infrastructure development and release/operations. Given such a potential, the market size of Cloud DevOps is expected to grow from USD 7.4 billion in 2021 to USD 37.2 billion by 2030 at a Compound Annual Growth Rate (CAGR) of 20% [3].

Given the high demand for Cloud DevOps skills among current and future software developers and engineers, there are several avenues available for students to acquire these skills. These avenues encompass industry-specific certifications, Massive Open Online Courses (MOOCs), and dedicated course curricula within educational institutions (eLearning, cited in [4]). Notably, numerous public cloud providers offer comprehensive DevOps training and certification programs. These include offerings like DevOps on AWS [5], VMware Tanzu DevOps [6], QwikLabs DevOps essentials [7], and Azure DevOps [8], allowing students to gain expertise tailored to specific cloud ecosystems. Furthermore, MOOC platforms play a pivotal role in providing a diverse array of DevOps training courses. These courses are readily accessible through popular platforms such as LinkedIn Learning [9], Udemy [10], Udacity [11], Coursera [12], and Edx [13], ensuring flexibility and accessibility for learners seeking to enhance their Cloud DevOps proficiency.

In addition, there are several U.S. institutions that have recently started offering Cloud DevOps courses e.g., Harvard University [14], University of Minnesota [15], and John's Hopkins University [16]. However, current learning options come with significant expenses for accessing infrastructure, often

This project is supported by the VMware University Research Fund. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of VMware.

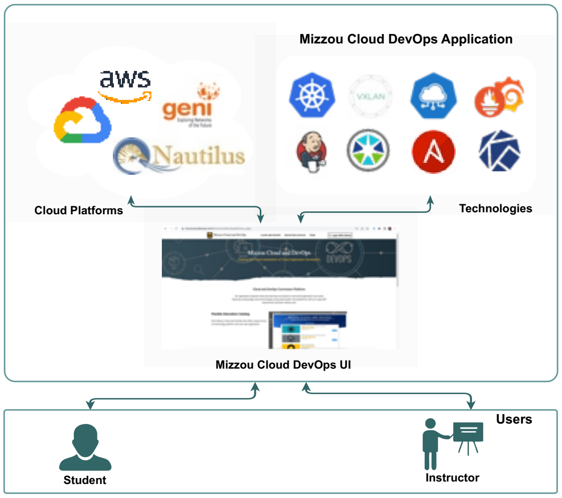


Fig. 1: Illustration of the Mizzou Cloud DevOps (MCD) platform featuring a user-friendly interface allowing access to students and instructors for diverse learning modules on cloud platforms and emerging technologies

lack comprehensive and integrated training materials, fall short on application-inspired learning experiences, self-assessment, and peer feedback mechanisms. This underscores the pressing need for an online self-service learning platform focused on Cloud DevOps concepts and hands-on skill development. This platform aims to fulfill the following objectives:

- Provide easy and unrestricted access to educational resources covering essential Cloud DevOps tools and technologies.
- Offer the capability and flexibility to utilize relevant cloud resources within the context of cutting-edge real-world "application-inspired" use cases.
- Empower students to receive peer ranking feedback
- Enable instructors to efficiently manage learning content and track students' learning outcomes.

This paper presents the design and development of the "Mizzou Cloud DevOps" (MCD) platform, an online self-service learning platform to teach students Cloud DevOps concepts, tools, and technologies. Our work builds upon and significantly expands the earlier work in [17], where an education catalog was developed to introduce basic Cloud DevOps technologies. Fig. 1 shows the MCD platform overview along with its components. The MCD platform offers free and easy access to learning modules with hands-on exercises that leverage different cloud infrastructures i.e., public clouds such as Amazon Web Services (AWS) [18], Google Cloud Platform (GCP) [19], and large research cloud platforms such as PRP Nautilus [20] (Kubernetes cluster resources) and GENI [21] (a future Internet experimentation facility). In addition, the MCD platform features an "application-inspired learning" methodology, which teaches students about popular Cloud DevOps concepts such as e.g., Kubernetes, VMWare Tanzu, KubeEdge, Jenkins, Ansible, Kubeflow, Prometheus, Grafana based on integration of real-world application use cases (e.g., cybersecurity, healthcare, video analytics). It also features a flexible and expandable catalog of eight learning modules, and is scalable for training any number of students.

The learning modules available in the MCD platform include: (i) Module-1—Kubernetes [22] with a Visual Cloud Computing application; (ii) Module-2—Software Defined Networking (SDN) with a Cyber Defense application; (iii) Module-3—KubeEdge [23] for edge computing with a Bioinformatics application; (iv) Module-4—CI/CD with Jenkins with a data-intensive Healthcare web application; (v) Module-5—Tanzu Community Edition [24] with a Drone Video Streaming and Analytics application; (vi) Module-6—Ansible [25] for Infrastructure as Code with Cybersecurity Operations in a Supply Chain/Logistics application; (vii) Module-7—Kubeflow [26] on AWS with a Health Informatics application; and (viii) Module-8—Prometheus [27] and Grafana [28] with a Video Transmission application.

Further, it allows students to obtain rewards and rankings through a peer standing dashboard, and provides telemetry for instructors to monitor and manage students' learning outcomes related to: containerization, trust-worthy high-performance/high-throughput computing workflows, data pipeline automation, microservices, containerized workloads, monitoring/management of infrastructure, clusters and apps across multiple clouds.

We conducted a series of community workshops to obtain learner experience feedback from the users of the Mizzou Cloud DevOps (MCD) platform. During these workshops, we administered a knowledge growth study survey, collecting data from over 200 participants both before and after they completed the lab modules. Our evaluation results show that there is significant growth in knowledge acquired by the participants and the benefits of the application-inspired learning of Cloud DevOps technologies is highly agreeable to the participants. In addition, we performed a usability study with participants who come from non-computer science backgrounds in order to identify how to make our learning modules more accessible to a wider audience. Lastly, we conducted a focus group study to obtain qualitative feedback to evaluate the benefits of our MCD platform.

The rest of the paper is organized as follows. In Section II, we discuss the related works. Section III outlines limitations in existing Cloud DevOps curriculum for learners, and challenges to develop novel/relevant curriculum. In Section IV, the design and development of MCD platform is discussed. In Section V, we detail our learning modules. Section VI discusses the evaluation results. Section VII concludes the paper.

II. Related Works

A. *Cloud DevOps for Application Workflows*

There have been prior efforts that have shown the benefits of Cloud DevOps in data/compute-intensive application workflows. Given that such workflows require expensive computing resources and custom workflow configurations, the work in [29] showed that users face challenges in lowering costs for development, and in the reproducibility of such workflows. Correspondingly, the work in [30] suggests the benefits in the use of Cloud DevOps concepts (e.g., configuration automation) for improving the reproducibility of data/compute-intensive workflows. It also highlights ideas such as parallel programming, and configuration automation through `YAML` files for better experience and results. Work in [31] explores the challenges and benefits of using DevOps concepts for improving software development in data-intensive application projects. Other works such as [32], [33] explored the advantages of using Cloud DevOps concepts for improving scientific software development projects in domains of information management, petroleum geology and geophysics.

Our novel work on application-inspired learning focuses on using Cloud DevOps for exemplar data/compute-intensive application workflows involving e.g., video transmission/processing, bioinformatics workflow management with edge computing, network security configuration and auditing, machine learning based model building for health informatics.

B. Online Learning Platforms for Cloud DevOps

Although there are benefits (e.g., decreased time of development, more stable operations, increased reproducibility and reliability) of using Cloud DevOps for application workflow management, available online learning platforms do not cater to learners from both computer science, and non-computer science backgrounds. A notable prior work on an online learning platform design for Cloud DevOps can be seen in [34], where Java developers learn skills for setting up a pipeline, testing, and deployment of a simple web application. Recommendations on software development curricula involving Cloud DevOps for students with computer science background are discussed in [35]–[37]. Work in [38] implements a cloud-based DevOps Lab Platform based on a Git-Lab code repository and a continuous integration framework.

The novelty of our proposed Mizzou Cloud DevOps platform is in the comprehensive development of a set of real-world application-inspired learning modules for online self-service learning to effectively/efficiently integrate Cloud DevOps within data/compute-intensive application workflows. Our MCD platform approach to use public cloud (e.g., AWS) and research cloud (e.g., Nautilus) resources, and supporting features such as self-assessment and peer feedback can elevate the learning experiences for students with, and without computer science backgrounds.

III. Motivation for Mizzou Cloud DevOps

In this section, we first outline the limitations in existing Cloud DevOps curriculum for students/instructors and application development professionals, when adhering to the conventional methods for developing Cloud-based applications. Following this, we present the challenges to develop novel/relevant curriculum.

A. Limitations of Traditional Workflows

Traditional workflows rely on manual and time-consuming processes, resulting in the following drawbacks:

- *Sequential and Siloed Processes*: The conventional approach follows a sequential, non-interconnected model akin to the waterfall methodology. This inhibits agile iteration across stages and impedes adaptability to evolving user needs. Modifications in one stage necessitate repeating multiple stages, consuming substantial time and resources [39].
- *Lack of Automation*: Manual integration of lifecycle stages introduces human errors, hindering clean experiment/development execution. This labor-intensive process is error-prone, time-consuming, and prone to software failures. Complex task/data dependencies elongate development timelines and undermine reproducibility [40].
- *Limited Data Security Operations*: Traditional workflows neglect early security considerations, obscuring vulnerabilities in subsequent stages. Manual inconsistencies and errors can compromise system security. Inadequate access controls and unpatched software escalate the risk of breaches and integrity attacks, endangering data confidentiality [41].
- *Inefficient Infrastructure Management*: The inability to adapt to evolving user requirements and lack of integrated application understanding lead to inefficient infrastructure management. Traditional deployments on physical servers with fixed operating systems result in resource allocation challenges, scalability issues, and application portability concerns [42].

By adopting our MCD approach, these limitations can be alleviated, revolutionizing domain specific workflows and enhancing their efficiency, security, and adaptability.

B. Challenges for Cloud DevOps Curriculum Development

The major challenges that learners face in their individual learning efforts of Cloud DevOps are as follows:

- *Lack of Comprehensive Training Materials*: Cloud DevOps is rapidly evolving, encompassing various tools and approaches like agile software development, Infrastructure as Code (IaC), version control, and continuous monitoring. Learners require easily accessible, comprehensive, and holistic training materials that facilitate rapid skill-building and motivate continued learning [43].
- *High Infrastructure Costs*: Learning Cloud DevOps demands significant computational, storage, and memory resources, incurring high costs. Acquiring both learning materials and necessary infrastructure can be prohibitively expensive, deterring individuals from embarking on their Cloud DevOps learning journey [44], [45].
- *Lack of Application-Inspired Learning*: Effective Cloud DevOps learning should extend beyond tool familiarity, preparing learners to apply best practices in real-world contexts. Students need insights into integrating cluster management, monitoring, and security for diverse application requirements. Professionals and domain scientists lack training in Cloud DevOps and struggle to incorporate it into their workflows [46].
- *Absence of Self-Assessment and Peer Feedback*: Feedback plays a pivotal role in shaping effective learning experiences. Cloud DevOps training should offer self-

assessment tools and peer feedback mechanisms to gauge knowledge growth and motivate learners to improve [47], [48].

To overcome these challenges, our MCD solution provides easily accessible, comprehensive, and cost-effective materials, address application-specific needs, and integrates self-assessment and peer feedback mechanisms in an online learning platform. Thus, we foster a supportive and encouraging environment for individuals seeking to gain expertise in Cloud DevOps for any domain application.

IV. MCD Platform Design and Development

In this section, we first list the technical requirements for achieving the learning objectives in our MCD platform. Following this, we describe the various MCD platform features. Lastly, we detail the MCD platform components and their workflow in the lab exercises within the learning modules.

A. Technical Requirements for Learning Objectives

To effectively educate students on Cloud DevOps, we designed our learning modules to cover a diverse range of Cloud DevOps tools and technologies, within the context of both public and research cloud infrastructures. Our intended audience encompasses cyberinfrastructure engineers, instructors, and online student learners. Each of our lab modules' content is independent, and students can choose and work on any module in their own pace. However, to successfully achieve the learning objectives, we outline the following technical prerequisites:

- *Cloud Infrastructures*: Our platform leverages widely recognized public cloud providers such as Amazon Web Services (AWS) [18] and Google Cloud Platform (GCP) [19] to offer adaptable and scalable cloud computing and networking capabilities. Additionally, we harness research clouds like the Pacific Research Platform (PRP) Nautilus [20], which provides accessible Kubernetes clusters with a plethora of available nodes, and GENI [21], a comprehensive experimental network testbed for the development and deployment of software-defined network applications.
- *Tools*: We employ Docker to furnish easily manageable virtual nodes for utilization in most of our lab exercises. In this context, we make use of DockerHub [49], a cloud-based container repository that is freely accessible, and GitHub [50], which serves as both a user authentication platform and a code repository.
- *Skill Set Background*: We anticipate that students engaging with the learning labs on the MCD platform possess prerequisite knowledge in fundamental areas such as computer networking, Docker container management, Linux scripting, and Python scripting. To further support their learning journey, students are provided with supplementary background reading materials that offer reference points on the technologies covered within each learning module.

B. The MCD Platform Features

The MCD platform accessible at [51] is designed to be versatile in supporting students' learning process and creation of new knowledge for various cloud services. It has the following features to maximize its impact on hands-on student training for delivery of conceptual knowledge and skill building:

1) Unique Education Catalog: The MCD platform offers a catalog comprising of eight distinct learning modules. These modules provide students with the opportunity to delve into cutting-edge technologies such as Kubernetes, KubeEdge, Prometheus/Grafana, Jenkins, OpenFlow, Ansible, and KubeFlow. Importantly, they do so by using open/public cloud infrastructures that are readily accessible, facilitating widespread adoption. Each module within the catalog focuses on a specific technology, cloud infrastructure, and use case application. For a detailed breakdown of these learning modules, please refer to Section V.

2) Extendable Course Content: The MCD platform employs a web services-based design and implementation approach, empowering instructors to seamlessly expand upon the course content by integrating new training elements. The user interface incorporates a dedicated 'Course Editor' section, enabling instructors to not only view, update, and remove existing modules but also to create entirely new modules effortlessly, without the need for coding expertise. This process encompasses tasks such as uploading lab manuals, incorporating HTML pages, and formulating assessment questions for student evaluation.

3) Real-world Application Use Cases: An essential aspect of hands-on learning in the MCD platform is to facilitate a learner with real-world problems to apply their learning. Towards this end, our MCD platform is supported by infrastructure configurations that are based on real-world application scenarios that include e.g., video transmission/processing, bioinformatics workflow management with edge computing, network security configuration and auditing, machine learning based model building for health informatics. Students can perform laboratory exercises using the infrastructure configurations facilitated by tutorials to explore, execute and evaluate various mechanisms on application infrastructures using cutting-edge technologies.

4) Easy Setup and Information Access: The MCD platform allows for easy-to-deploy cloud networks and DevOps practice scenarios by implementing a user-friendly interface with relevant instruction manuals. The manuals provide self-study steps to obtain in-depth understanding of various Cloud DevOps tools/technologies involved in the exercises with relevant and quickly accessible documentation. As shown in Fig. 2, instructions in the lab manuals have been developed with Markdown [52], a lightweight markup language for creating formatted text using a plain-text editor. This allows learners to conveniently navigate through various learning chapters pertaining to the modules, and use a code block copy feature for reducing typographical errors during execution of commands on terminal sessions.

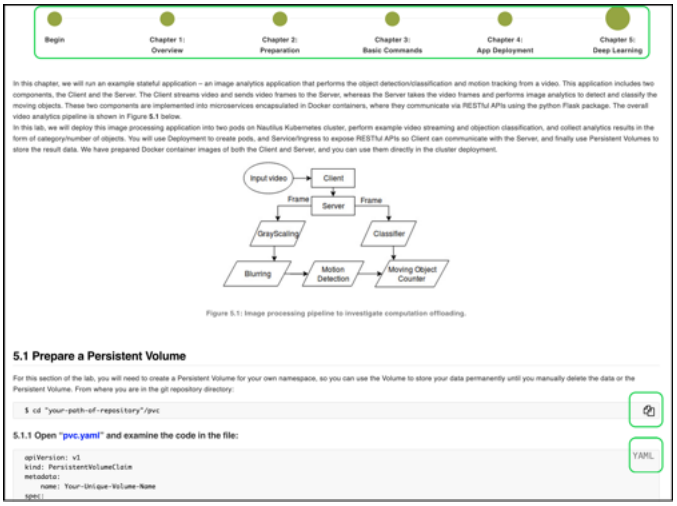


Fig. 2: MCD UI displaying a chapter in the Module-1 with the ability for students to switch different chapters of a module, copy codes from code blocks.

5) *Scalable and On-demand:* The learning modules offered on the Mizzou Cloud DevOps (MCD) platform use cloud infrastructures such as AWS, GCP, Nautilus, and GENI, which possess the capability to swiftly deploy on-demand infrastructure instances tailored to a multitude of applications. Resource allocations range from the launch of minimal infrastructure to accommodating provisioning of extensive infrastructure resources to support tens or even hundreds of students concurrently. The modular architecture of the MCD platform enhances its versatility and compatibility with various cloud infrastructures. Moreover, lab exercises can be seamlessly deployed on any public or research cloud infrastructure owing to the utilization of Docker containers.

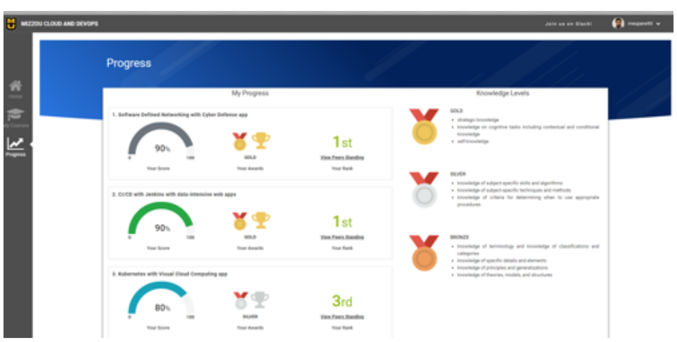


Fig. 3: MCD UI showing the 'Progress' page of a student, with awards, score, and rank including information on three knowledge levels: Gold, Silver, and Bronze.

6) *Ranking Dashboard and Peers Standing:* The MCD platform incorporates a built-in assessment that employs rankings to assess and monitor students' learning progress and achievements comprehensively. It introduces gamification elements, as illustrated in Fig. 3, enabling students not only to track their learning levels represented by medals (Gold, Silver, and Bronze) but also to gauge their performance relative to their peers across diverse learning modules.

7) *Telemetry for Instructors:* Instructors are empowered by the MCD platform to gain insights into students' progress, utilizing data accumulated throughout students' interactions with the learning modules. The platform offers interactive graphical views, including timelines of various activities within the MCD web portal. These visualizations encompass elements such as student ranking, enrollment statistics, and score analytics, facilitating data-driven assessments of student accomplishments and informing refinements to course content to enhance the overall student learning experiences.

C. MCD Platform Components

The MCD platform includes several supporting components that are related to the lab exercises in order to concurrently deliver the learning modules for multiple students. The components also allow for realistic application scenarios to be launched in the cloud infrastructure through easy-to-follow instruction manuals that guide students through all the steps involved in the process of completing the lab exercises.

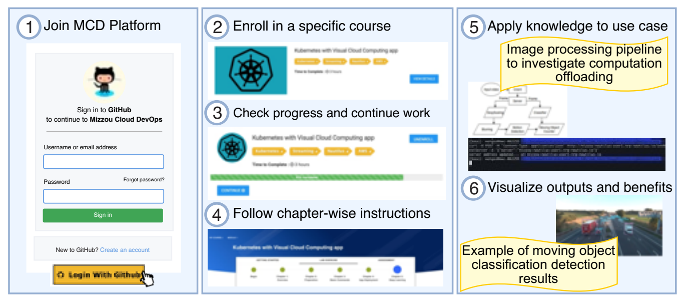


Fig. 4: Illustration of user steps in Mizzou Cloud DevOps platform.

1) *Web Portal:* Fig. 4 shows an illustration of our front-end implementation (developed using AngularJS) that guides a user through a series of steps to complete a set of learning module exercises. We made the platform access easy/secure by requiring users to use their GitHub credentials to access various learning modules. The MCD web portal is built as a central resource hub to provide a rich user experience to students and instructors. The user-friendly portal is designed with responsive views, simple navigation, contrasting color schemes, and effective visualizations. The user dashboard features two roles, i.e., instructor and learner. The instructor is able to manage the learning modules as *courses*, and can monitor student progress through a telemetry dashboard. Learners can enroll in a specific course, follow the hands-on lab exercises, and apply knowledge to a real-world application use case e.g., video analytics to visualize outputs and benefits. Moreover, the learner can take a technical assessment after completing the exercises to obtain a peer ranking score. The user authentication and enrollment is streamlined. Students use their GitHub accounts to log in, needing instructor approval for module enrollment. Prior to commencing exercises, students complete a pre-lab survey to assess initial knowledge/skill levels. The education catalog showcases module availability.

2) *Learning Evaluation API:* To enhance the assessment and grading process of student work, we have integrated

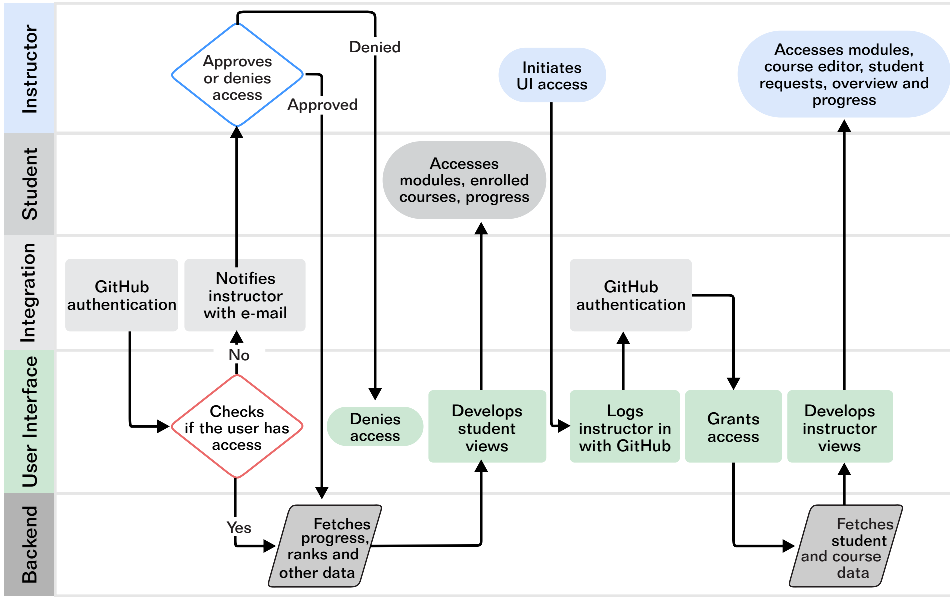


Fig. 5: MCD platform sequence flow for Student and Instructor user roles.

a set of Learning Evaluation APIs into the core of the Mizzou Cloud DevOps (MCD) platform. These APIs enable students to submit their technical assessments, which are then automatically graded. Additionally, the system assigns knowledge level medals based on individual performance. Within the *submitEvaluation* API, we've developed an auto-grading feature capable of evaluating responses to questions, each with varying levels of complexity. These questions primarily consist of multiple-choice items related to different lab exercises, ensuring a diverse range of assessments. After the evaluation, the *submitEvaluation* API not only determines knowledge levels but also bestows corresponding medals and ranks students in relation to their peers. Students can easily access their personal rankings and medals, as depicted in Fig. 3, and also view the rankings of their fellow students through the *getStudentRank* API. Medals are awarded based on the "Knowledge Dimension" concepts derived from Bloom's Taxonomy [53], with varying weights assigned to questions based on their complexity. These REST APIs have been meticulously implemented using *Spring Boot* [54], a well-established Java Web Framework. This component serves as the backbone of the MCD platform, offering students practical learning exercises mirroring real-world application workflows. Each exercise is thoughtfully accompanied by comprehensive instruction manuals, introducing essential technologies such as container orchestration for video processing, cloud networking for cyber defense, edge/cloud federation for trusted computing, data pipeline automation for knowledge discovery, bioinformatics workflow management with edge computing, network security configuration and auditing, and machine learning based model building for health informatics. The web portal, adopting a wiki-based format, provides invaluable guidance to help learners enhance their Cloud DevOps skills.

3) Lab Exercise Workflow: Fig. 5 depicts the sequence of actions taken by both students and instructors within the MCD platform. To begin their learning, students utilize their GitHub accounts for login, a process that requires approval from the instructor before they can enroll in any available learning module from the catalog. Prior to commencing a lab, approved students are prompted to complete a pre-lab survey, which assesses their initial knowledge and skill levels. Each learning module includes a dedicated section in the required reading materials, introducing students to relevant references, concept tutorials, and an overview of the technologies employed in the modules to help deliver background knowledge.

Upon completing the lab exercise steps, students are provided with a technical assessment associated with the corresponding learning module. This assessment gauges students' comprehension of the concepts, tools, and techniques at both fundamental and advanced levels. The submitted answers are graded using our automated grading system, which then assigns medals and peer rankings to students, as illustrated in Fig. 3. Medals represent the students' updated knowledge levels, categorized as follows:

- Bronze (Level 1): We combined the first two categories of the knowledge dimension, the factual and the conceptual knowledge, to form the first level, the Bronze level. At this level, students have knowledge of the terminologies, classifications and categories, as well as some of the principles, theories, relating to the Cloud DevOps concepts.
- Silver (Level 2): This level represents the procedural category of knowledge dimension. Student at this level will have knowledge of subject-specific skills, algorithms, and certain level of techniques, and possess knowledge about when to use appropriate procedures in application development.
- Gold (Level 3): This level refers to the metacognitive category, which is strategic knowledge. At this level, the student will have mastery of concepts and skills required for building applications using Cloud DevOps concepts.

4) Digitally Verifiable Credentials: We have integrated the MCD platform with the LinkedIn API to grant learners (virtual) certificates/badges that recognize their achievements and help them showcase their skills to their professional profiles. This MCD platform component is designed to motivate students by granting them certificates (digitally verifiable credentials) for individual courses or for completing the entire MCD learning modules catalog. The learners will be able to directly share the credentials to LinkedIn from the MCD platform in two ways: (i) open credential without verification ID, and (ii) credential with verification ID.

5) Infrastructure Provisioning Component: This component focuses on efficiently provisioning infrastructure for learning modules. Students will access on-demand instances with preconfigured tools and libraries via the online MCD platform. By automating infrastructure setup, students can dedicate their time to mastering module concepts and skills. The modular design allows utilization of diverse cloud infrastructure APIs, facilitating resource launch (VMs, storage, network slices, etc.) based on student workload needs.

The MCD platform sequence involves both students and instructors. Students log in using their GitHub accounts and must obtain instructor approval to enroll in available learning modules. Pre-lab surveys assess students' initial knowledge. Instructional materials and references prepare students for module exercises, which encompass a wide array of real-world application scenarios. The platform incorporates wiki-based content for comprehensive guidance. Students engage with cloud infrastructures like AWS and PRP Nautilus, enhancing their practical skills for application development. The platform's holistic approach aims to foster effective Cloud DevOps learning through seamless integration with cloud resources.

V. Cloud DevOps Learning Modules

The learning modules within the MCD platform are thoughtfully designed to provide students with a strong foundational understanding of popular Cloud DevOps tools and technologies, including Kubernetes, KubeEdge, Jenkins, and SDNs. These modules immerse students in real-world application scenarios, emphasizing automation and management principles, while simultaneously imparting practical problem-solving skills. Drawing from prior research findings, the content of these modules is grounded in areas such as video processing [55], cyber defense using pretense [56], cloud federation for trusted computing [57], data pipeline automation for knowledge discovery [58], and machine learning for intelligent healthcare appointment scheduling [59]. Fig. 6 provides an illustrative mapping of research-inspired application workflows to essential Cloud DevOps practices and tools, highlighting their application on both public and research cloud infrastructure resources. The lab exercise manuals and related documentation are readily accessible online at [60], offering students openly available learning resources.

A. Learning Modules Overview

All the lab exercises represent both new and existing capabilities on open/public cloud platforms, and go further to help students learn about their detailed integration that is useful for a broad set of application contexts. All the learning modules on the MCD platform address a unique focus relating to a specific technology, platform, and application use case (App). The available learning modules in the MCD platform include: Module-1) Kubernetes with Visual Cloud Computing App; Module-2) Software Defined Networking with Cyber Defense App; Module-3) KubeEdge for Edge Computing with Bioinformatics App; Module-4) CI/CD with Jenkins with Data-intensive Healthcare Web App; Module-5) Tanzu Community Edition with Drone Video Streaming and Analytics App; Module-6) Ansible for Infrastructure as Code with Cybersecurity Operations in a Supply Chain/Logistics App; Module-7) Kubeflow on AWS with Health Informatics App; and Module-8) Prometheus and Grafana with Video Tranmission App.

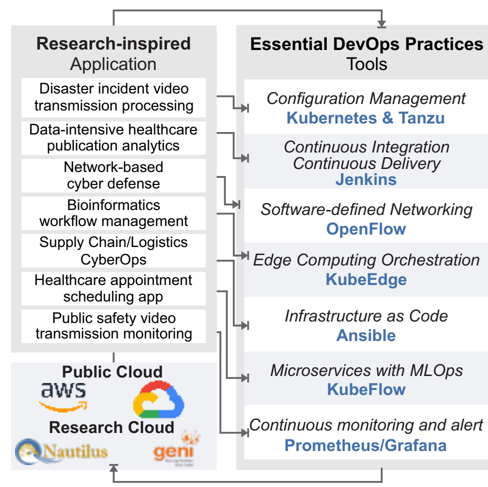


Fig. 6: Mapping of research-inspired applications to various DevOps practices and tools performed in public and research cloud platforms.

In the following, we will detail the eight learning modules in the MCD platform:

B. Module-1—Kubernetes with Visual Cloud Computing App

1) Technology: This module focuses on Kubernetes [22], a popular open-source system for automated deployments, scaling and management of containerized applications. Kubernetes offers many features for application development and operation, including configurability, scalability, flexibility, customizability, elasticity, and portability.

2) Platform: This module is available in PRP Nautilus [20] and AWS EKS [61]. PRP Nautilus is an openly accessible Kubernetes cluster supported by the National Science Foundation. PRP Nautilus runs up-to-date versions of the open-source Kubernetes software, so that students can use all the existing plugins and tools from the Kubernetes community. AWS EKS is a service for starting, running and scaling Kubernetes deployments.

3) Application Use Case: The application used in this module is a Visual Cloud Computing (VCC) application [55]. This application uses an image processing pipeline for object tracking in video streams. It follows a server/client architecture that is deployed on the Kubernetes cluster, and leverages deep learning algorithms using the Darknet with YOLO/COCO framework [62].

In this module, we guide the students to get familiar with the Nautilus Kubernetes cluster platform. They are provided with instructions to go through the fundamentals of Kubernetes, e.g., Pod, Deployment, Service, Load Balancing, Ingress, and Persistent Volumes. The students will deploy an example stateless application, the Guest Book application with MongoDB, to show how one can deploy an application without persistent data storage. The students will also deploy the VCC pipeline, and learn how to stream video data from Client pod, and analyze video frames with object tracking algorithms on the Server pod. They will then use a persistent volume to store results data and share them between the Client/Server.

C. Module-2—Software Defined Networking with Cyber Defense App

1) Technology: This module focuses on Software Defined Networking (SDN) using Virtual Extensible LAN (VXLAN). SDN is an approach to network management that enables dynamic, programmatically efficient network configuration. It is able to separate control plane from the data plane allowing a network to be programmed independent of the individual switches. We explore using VXLAN to enable SDN on Google Cloud Platform, allowing us to manage network configuration and routing, which can be achieved with SDN since cloud service providers typically do not allow users to manipulate their network switch hardware.

2) Platform: The cloud platforms we support for this module include both GCP and AWS. These cloud platforms allow us to integrate the application used on this module with ease of experimentation for various cyber defense by pretense scenarios and techniques.

3) Application Use Case: In this module, we use Dolus cyber defense by pretense system that was designed to perform attack detection and defense for various targeted attacks, such as distributed denial of service (DDoS), advanced persistent threats (APT), and Cryptojacking [56]. Dolus uses cyber defense using pretense principles to mitigate such targeted attacks. Using an SDN-based networking mechanism, this application allows re-direction of attacker's network connections into the system, which helps in thwarting targeted attacks through active defense.

In this lab module, students will learn how to setup SDN capability using the VXLAN protocol. They will create an SDN virtual network with a Pox Controller node as the control plane, and create two Open vSwitches, a Root Switch and a Slave Switch, which are connected to the Controller via a TCP/IP connection. The students then create a number of host VM instances that are connected to the two vSwitches via a VXLAN network overlay. These VM nodes work with various roles in the system. Some roles include benign users, attackers and controller. The SDN topology is used to allow experimentation with various Dolus cyber defense by pretense techniques on the cloud network infrastructure.

D. Module-3—KubeEdge for Edge Computing with Bioinformatics App

1) Technology: This module focuses on KubeEdge [23], an open-source system for extending native containerized application orchestration capabilities to hosts at the edge. It is built upon Kubernetes and provides fundamental infrastructure support for network, app deployment and metadata synchronization between cloud and edge computations.

2) Platform: This module is supported on GENI and AWS cloud platforms. GENI cloud is an openly accessible infrastructure supporting "at scale" research in networking, distributed systems, security, and novel applications. GENI is supported by the NSF and is available without charge (free) for research and classroom use. GENI offers various computing resource configurations from many locations around the United States, thus enabling distributed computing experimentation capabilities. AWS Educate accounts are also supported to perform this lab using KubeEdge.

3) Application Use Case: The application use case in this module is the volunteer edge computing for bioinformatics application. This application enables researchers to work with multi-cloud capabilities to better utilize compute resources available at local servers as well resources allocations on public cloud platforms. It gives researchers flexibility in creating multi-cloud architectures for their data-intensive and computation-intensive application requirements. Through a multi-cloud resource broker named OnTimeURB [57], this application allows the ideal compute resource allocations and stages the tasks on those resources during application execution.

In this lab, students will deploy a KubeEdge cluster on either the GENI cloud or on AWS. They will manage the cluster to include a Cloud Core node and multiple Edge Core nodes with various computing resources. Docker containers are then deployed with trusted bioinformatics workflows running HPC jobs based on various configurations. Volunteer edge computing algorithms are utilized to automatically select edge nodes to maximize performance while lowering related costs.

E. Module-4—CI/CD with Jenkins with Data-intensive Healthcare Web App

1) Technology: The technology students will explore in this lab is the Continuous Integration/Continuous Delivery (CI/CD) using Jenkins. Traditional integration comes with development practice that requires developers to integrate code into a shared repository several times a day, each of which is checked and then verified by build teams to detect problems early. This can be very tedious. Instead, a CI/CD system usually involves a tool that keeps a mandatory version control system. Whenever a change to version control system is detected, the system would automatically build and test the application, and deploy the newly built application on the server machine. In this module, we use Jenkins source automation server tool. Jenkins is the leading open-source automation server that provides hundreds of plugins to support building, deploying and automating any project application.

2) Platform: In this module, we use the AWS cloud infrastructure to create and configure a Jenkins server, and use the server to deploy example CI/CD applications and related data processing pipelines.

3) Application Use Case: This module uses the KnowCOVID-19 [58], an artificial intelligence (AI)-based publication analytics package that works through a guided web interface application. It is designed for COVID-19 clinical researchers to find helpful insights for decision-making on pandemic response by performing data mining of scientific publications. It enables filtering and ranking of data quality in the publications based on the evidence pyramid of scientific knowledge. The AI-based publication analytics is performed on published literature archives such as PubMed and LitCovid. This application also enables visualization of knowledge patterns to share with experts, drill down facts and study deep topic relationships.

In this module, the students will create and deploy a Jenkins server on the AWS cloud infrastructure. The instructions in the tutorials are explained such that the students get familiar with the basic functionalities of Jenkins. The KnowCOVID-19 application enables students in using simple application development increments (whenever model training is updated) to learn how to create and run CI/CD jobs on Jenkins.

F. Module-5—VMWare Tanzu for Kubernetes Cluster Management

1) Technology: This module focuses on VMware Tanzu Community Edition (TCE), using Kubernetes platform for working in small-scale or pre-production environments. TCE enables creation of application platforms by leveraging a cluster API to provide declarative deployment and management of Kubernetes clusters. TCE also allows learners to get bootstrapped by providing a set of opinionated building blocks. Additionally, it enables them to add or replace these with their own components.

2) Platform: In this module, we use the AWS cloud infrastructure where Ubuntu Linux is spawned in an EC2 instance and Tanzu managed clusters are deployed with two components: a management cluster and a workload cluster.

3) Application Use Case: The application used in this module is an image analytics application that performs object detection/classification from traffic video streams. This application includes two components, a client and a server. The client streams video and sends video frames to the server, whereas the Server takes the video frames and performs image analytics to detect and classify the moving objects. These two components are implemented using microservices encapsulated in Docker containers, where they communicate via RESTful APIs using the python Flask [63] package. The students will deploy this image processing application in two pods in the TCE cluster, perform example video streaming and objection classification, and collect analytics results in the form of category/number of objects. The students will use deployment to create pods, and service to expose RESTful APIs so that the client can communicate with the server. Docker container images of both the client and server have been prepared, and students can use them directly in the cluster deployment.

G. Module-6—Ansible for Infrastructure as Code with CyberOps in a Supply Chain/Logistics App

1) Technology: This module focuses on Ansible as an automation tool for managing servers in the cloud. The students will learn how to use the AWS cloud platform to bring up a simple Ansible topology that has 3 nodes - 1 Control node and 2 Managed nodes and learn how Ansible can be used to automate processes, e.g., user management and software/library management. The students also create a LAMP stack that runs a web server on the Managed nodes.

2) Platform: Students will create and launch the Ansible engine and nodes using AWS EC2 services, and then configure the nodes to speak to each other using Secure Shell (SSH) concepts.

3) Application Use Case: The application use case for this module is a cybersecurity breach scenario in a supply chain/logistics application [64], where a node is made vulnerable by executing a script that creates different user accounts and then creates a group named *attack* and makes the new users part of it. The script will then assign all the permissions in the system to this group enabling the users to execute super user commands. This allows collection of internal data into a file which is then moved to a different node. In this use case, the students will then perform an audit of the system configuration. The baseline information on the system configuration is gathered from the first node through the Ansible Audit Playbook execution.

This whole set up works with a 3-node infrastructure, a master node with Ansible Engine and 2 worker nodes. One of the worker node works as an attacker and the other worker node works as the victim, both nodes will be administrated by Ansible. Ansible is used to determine any suspicious action in the nodes through an audit.

H. Module-7—Kubeflow with Health Informatics App

1) Technology: The module focuses on Kubeflow which helps students learn about machine learning (ML) operation automation in the cloud using the Kubeflow framework. Kubeflow helps building, deploying, and managing multi-step ML workflows and ascribes to make it simple, portable and scalable. The complex systems are managed using Kubernetes and deployed to various clouds, local or on-premises platforms for experimentation for production usage. There are several components within Kubeflow such as Jupyter notebooks, ML training, hyperparameter tuning and serving workloads across multiple platforms.

2) Platform: Students will use the AWS cloud platform to perform this lab where they will use MiniKF, a third party service within the AWS platform to launch an instance to perform the ML tasks. In this module, students will work on Kubeflow that concentrates on the ML workflows accessed through MiniKF, which are used on top of Docker containers and uses Kubernetes to orchestrate them.

3) Application Use Case: The application use case for this module is an Health Informatics App that predicts the consultation length by accounting for its semi-continuous nature (i.e., zero in case of no-shows and positive otherwise, using ML algorithms), identifying important features for predicting no-shows, non-zero consultation lengths and assess the impact of integrating the ML-based prediction with the appointment scheduling system. The data used for this use case will be passed through a multi-step pipeline for various ML algorithms viz., Random Forest, Decision Trees. The model is designed such that the patients' data on their appointment set up and information on whether or not they have been coming for visits as per their schedule is utilized to make decisions on scheduling the future appointments. This also factors in double-booking of appointments and deferral of the patients based on the output.

I. Module-8—Prometheus and Grafana for Video Transmission App

1) Technology: This module focuses on Prometheus and Grafana monitoring tool stacks and their usability to monitor servers and applications. Prometheus is an open-source monitoring system with a dimensional data model, flexible query language, efficient time series database and alerting approaches. Grafana is a multi-platform open source analytics and interactive visualization web application.

2) Platform: The module deploys Prometheus and Grafana stack on the AWS cloud platform, familiarizes students with the basic concepts and functionalities using example Exporters to collect/visualize data from a sample application or server.

3) Application Use Case: The use case application for this module is based on a Video Transmission App. The students learn to develop an applicable Prometheus Exporter in Python, exporter will analyze the log file generated by video streaming server and exports the payload size, stream duration, throughput metrics for Prometheus and Grafana to scrape and visualize the data. The metrics to collect the data are configurable and temporal (i.e., in seconds, minutes, etc.) which are instantly reflected in Prometheus. Grafana is then connected to the Prometheus server to visualize the analytics.

VI. Evaluation

We organized a community workshop series for instructors and students from various universities in the United States to evaluate our MCD platform in terms of knowledge growth and learner outcomes. In this section, we summarize the results obtained from participants with diverse backgrounds in terms of age, technical knowledge and work experience across the workshops. In addition, we detail results from a usability study with students with non-computer science background, and a series of focus groups with MCD platform users.

A. Knowledge Growth Study

A knowledge growth study was performed to evaluate the students' progress in exposure to important Cloud DevOps technologies and application use cases. Kubernetes is one of the early learning modules hosted in the MCD platform, and Kubeflow is one of the latest learning modules of the platform. The Knowledge growth studies for these two learning modules are discussed in the following:

1) Module-1—Kubernetes: In order to evaluate knowledge growth of participants, we performed a pre-lab evaluation survey and a post-lab technical assessment for 122 learners who successfully completed the Module-1. According to participants' answers, they were assigned an initial knowledge level, ranging from Level 1 through Level 3. We also assigned a Level 0 for the participants that have no knowledge of the platform or the technologies used on a learning module to form a four knowledge dimensions from Bloom's taxonomy. We then performed a comparative study to compare their knowledge level before and after the learning module completion. Some example pre-lab questions for the learning module include:

- Are you familiar with Kubernetes?
- Have you used PRP Nautilus cloud platform before?
- Have you worked with Docker containers before?

The post-lab technical assessment featured a set of multiple choice questions prepared based on the concepts used in e.g., Module-1 along with technical questions that include:

- In the context of Kubernetes, what is a container?
- Which of the following commands is used to create a Kubernetes service?
- Given a set of pods, how can one enable network access between them using Kubernetes?
- How are pods created?

Based on the survey responses, participants were assigned knowledge levels as part of a statistical comparative study as illustrated in Fig. 7. The results illustrated in pie chart show that 83% of the participants in Level 0 and Level 1 were upgraded to at least Level 2, which demonstrates a notable knowledge growth in the participants. Moreover, many participants who were at Level 2 or below could enhance their learning and move to Level 3 as we see a rise from 4% to 41%. This strongly demonstrates that the learning module completion certainly helped the participants to improve their skill level in the technologies explored in the Module-1.

2) Module-7—Kubeflow: In order to evaluate knowledge growth of participants, we performed a pre-lab evaluation survey and a post-lab technical assessment for 97 learners who successfully completed the Module-7. We performed a study similar to Module-1. Some example pre-lab questions for the first lab include:

- Are you familiar with Kubeflow?
- Are you familiar with machine learning?

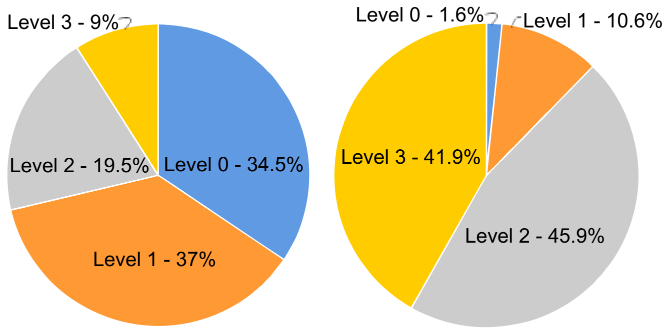


Fig. 7: Percentages of students in each knowledge level before (left) and after (right) successfully completing Module-1 based on pre-lab and post-lab assessments.

- Have you used Amazon Web Services before?

The post-lab technical assessment featured a set of multiple choice questions prepared based on the concepts used in Module-7 along with technical questions that include:

- What types of Machine Learning workloads can be handled by Kubeflow?
- What is the function of kfp.Client.create_experiment?
- The problem of predicting patient no-show risk will fall under which one of the following machine learning approaches?
- Given the class imbalance in the outcome variable (30% no-show, 70% show), which of the following metrics is not best suited for performance evaluation?

Based on the survey responses, participants were assigned knowledge levels as part of a statistical comparative study as illustrated in Fig. 8. The results presented in pie chart show that 57% of the participants in Level 0 and Level 1 were upgraded to at least Level 2, which demonstrates a notable knowledge growth in the participants. Moreover, many participants at Level 2 or below could enhance their learning and move to Level 3 as we see a rise from 0% to 34%. This strongly demonstrates that the learning module completion certainly helped the participants to improve their skill level in the technologies explored in the Module-7.

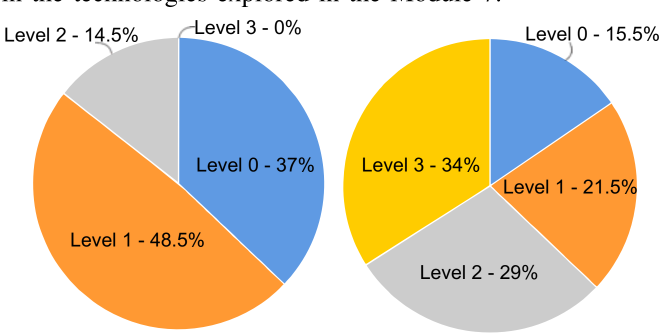


Fig. 8: Percentages of students in each knowledge level before (left) and after (right) successfully completing Module-7 based on pre-lab and post-lab assessments.

B. Feedback on Workshop Outcomes

In the community workshops that were open to the public, we performed a feedback survey involving 76 randomly chosen participants on their opinion and improvement suggestions for the MCD platform. In the following, we show a sample set of 5 questions that were part of the survey questionnaire:

- Our approach to using application-inspired real-world case studies to learn cloud platforms and DevOps technology is helpful
- Our labs featuring support for multiple cloud platforms are useful
- The technologies related to micro-services and containerized workflow within clusters are relevant and timely
- The ranking dashboard is helpful for visualizing and assessing your learning outcome
- Overall I would rate my experience as successful

Using the Likert Scale (range involves 'Strongly Disagree' to 'Strongly Agree'), we obtained the survey results presented in Fig. 9. We concluded that our approach of using application-inspired real-world case studies to learn the Cloud DevOps technologies was highly agree-able to the participants. The attendees also seemed excited with exposure to the various public and research cloud platforms we are supporting in the MCD portal. Most of the attendees also agreed that the technologies and concepts we included in the learning modules are timely and relevant to cloud application development.

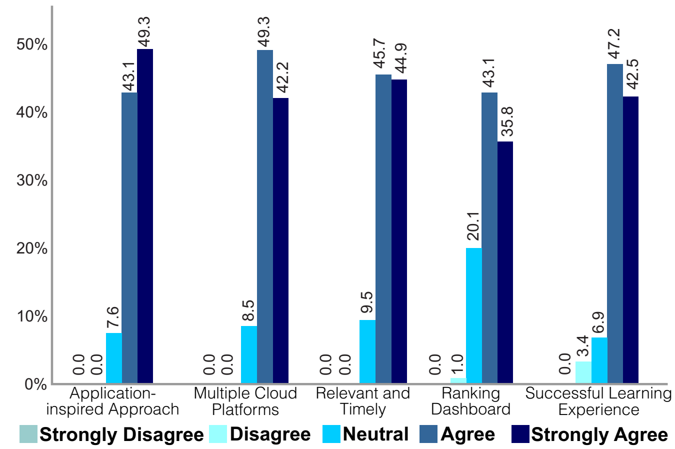


Fig. 9: Percentages of workshop attendees distributed with respect to their feedback on the provided questionnaire.

C. Usability Study with Students from Non-Computer Science Background

The usability study was performed by 4 independent evaluators from the Information Experience Lab at the University of Missouri, and involved students who were from industrial engineering and business backgrounds. The study involved participant observations and survey methods and included 8 (n = 8) randomly chosen participants. In the usability study, participants were asked to use Module-7 in the MCD platform (Kubeflow on AWS with Smart Scheduling App).

Table I lists the participants' demographic information, which shows that all of the participants were from non-computer science backgrounds. Participants for this evaluation were purposefully sampled from a Data Engineering and Predictive Modeling class at the University of Missouri, consisting

TABLE I Demographics of the Usability Study Participant (USP) pool with non-computer science background.

Participant	Age	Gender	Student Type	Department	Ethnicity
USP-1	21	Male	Full-time	Industrial engineering	White or Caucasian
USP-2	22	Male	Full-time	Industrial engineering	White or Caucasian
USP-3	22	Female	Full-time	Industrial engineering	White or Caucasian
USP-4	21	Female	Full-time	Industrial engineering	White or Caucasian
USP-5	22	Male	Full-time	Industrial engineering	Hispanic
USP-6	22	Female	Full-time	Industrial engineering	White or Caucasian
USP-7	23	Male	Full-time	Industrial engineering	White or Caucasian
USP-8	21	Female	Full-time	Industrial engineering	White or Caucasian

of undergraduate and graduate-level students in the Industrial Engineering program. After informed consent was obtained, the participants took part in a 75-minute session to complete a structured lab exercise within the module. Participants' interactions with the MCD platform were video recorded via Open Broadcaster Software (OBS), an open-source software for screen recording and live streaming. During the evaluation, trained researchers took field notes of participant interactions within the MCD platform. The video recordings and the field notes were stored for later analysis. After participants finished the lab component of the study, they then filled the System Usability Scale (SUS) [65] and rated the overall user-friendliness of the MCD platform on a single item adjectival ease-of-use scale.

Using a multi-methods approach, qualitative data were analyzed using inductive and deductive methods, and quantitative data were analyzed to provide descriptive statistics. For qualitative analysis, the deductive analysis was performed using the coding scheme outlined by Van den Haak et al. [66] which categorizes usability issues along the lines of comprehensiveness, feedback, data entry, terminology, and layout. Inductive analysis was also conducted to identify themes related to learner experience in video recordings and field notes [67]. Identified usability issues and the emergent themes were refined into four overarching themes of instructions in the form of text and images, system-generated feedback, and technical terms. For quantitative analysis, SUS scores were determined based on the methods outlined by the author in [68]. An additional single-item user-friendliness scale was used to interpret the SUS results [65].

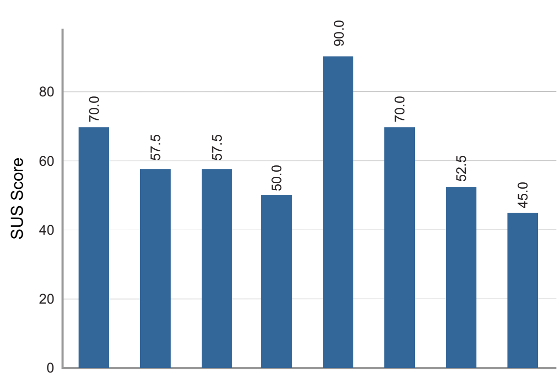


Fig. 10: Individual SUS scores of the participants of the usability test.

Quantitative results indicated an overall SUS score as 61.6 on a scale of 100 (where 100 is the highest). It means that the participants who were from non-computer science backgrounds rated the usability of the MCD platform as "D/Marginal" on a scale of A to F, where A is the best [69]. The "A-F" rating is a letter guide that explains the overall usability of the system. According to the SUS scores, participants rated the usability of the MCD platform as 63.7 and its learnability as 53.1. This suggests that the participants who did not have adequate pre-requisite computer science education found it difficult to learn the concepts, and note their experience of the MCD as an average learning system. Fig. 10 shows individual scores of each participant of the study. These ratings are slightly at odds with participants' ratings of the platform's user-friendliness, which was rated as "good" on a single-item adjectival ease-of-use scale [65]. These findings suggest that while participants found the MCD platform user-friendly, their lack of pre-requisite computer science background made the use of the MCD platform difficult to use or learn as they expected.

Meanwhile, qualitative findings from the usage test suggest that the participants liked the overall layout of tabs and chapters and the step-by-step format of instructions of the MCD platform. Across the sessions, no participant expressed dissatisfaction with the usage of the MCD platform or expressed a need to leave the study. All participants found the platform useful for their study curriculum and expressed a desire to use it in the future which suggests good acceptability. Overall, the participants' experience was satisfactory, although a few usability issues were encountered. The majority of the usability problems were related to the layout and design (font, color, images, etc.) of the instructions within the chapters, thoroughness of instructions in terms of step-by-step guidance of working with third-party applications (e.g., AWS), and notification for error prevention and regulation. For instance, the texts between headings and images needed to be more visual for learners. In this regard, USP-1 stated: "Some of these smaller texts are very easy to miss. I always read the big bold text."

Table II summarizes participants' likes and dislikes. Given that the participants were from non-computer science background, the participants also experienced difficulty understanding the technical terminology. USP-3 commented: "As someone who comes in with no background knowledge, maybe defining 'Docker', 'container' and the 'Kubernetes pods'."

TABLE II Participants likes and dislikes identified during usability test.

Likes	Participants
Figures and images	USP-1, USP-3, USP-4, USP-5, USP-6
Step-by-step instructions	USP-1, USP-4, USP-5, USP-8
Video tutorial	USP-5, USP-7, USP-8
Instructions on the code	USP-4, USP-5
Overall layout of tabs and chapters	USP-3, USP-4, USP-8
The green and blue circles show the visibility of the student's progress through the chapters	USP-3, USP-4, USP-8
Dislikes	**Participants**
The instructions on the top of the instructional image are not clearly visible	USP-1, USP-7, USP-8
Screenshots appear to merge with the background because of the same color and lack of border/frame	USP-3, USP-4, USP-7
The instructions (annotations) within the images were of the same font size and color as the content on the images	USP-3, USP-4, USP-7
It is difficult to read the instructions as the layout of headings, figures, and text is not clear	USP-2, USP-3, USP-4, USP-5, USP-7
The screenshots are not updated with the current issues and errors	USP-3, USP-7
The lack of flexibility in moving back and forth between the chapters	USP-2, USP-3, USP-4, USP-5, USP-8
Worried about losing the current execution of the task if go back and forth between chapters while waiting for the environment to set up	USP-3, USP-7

Further, it was observed that participants' interaction with third party applications (e.g., AWS Marketplace) for the purpose of completing lab exercise impacted their user experience with the MCD platform to a large extent. For instance, USP-1 stated, "I keep switching to different platforms. Maybe there is a way to all do that in the same website or like have that be set up for you or like have that be in a chapter."

D. Focus Groups Study

For the focus groups study, 6 users (n = 6) of the MCD platform were randomly selected to participate in the study. We divided them into two learner groups: Focus Group 1 included four participants who completed at least one course and Focus Group 2 included two participants who did not complete any course. In Focus Group 1, all participants (n=4) were computer science graduate students. For the Focus Group 2, we conducted a semi-structured interview with FGP-5 and FGP-6. See Table III for participants' demographic information. Both qualitative and quantitative data were collected using multiple methods, and the results of the study are described below.

Participants' responses were analyzed using inductive methods. Results from inductive analysis suggested themes related to user experience of the MCD platform. Results suggest that participants found the MCD platform user-friendly and indicated that they were excited to learn and gain hands-on experience with Kubernetes and third party applications, e.g., AWS. Participants in both focus groups were asked to provide general comments on their reason to choose the MCD platform over other platforms available in the market. According to the participants from both sessions, the MCD platform was more suitable for students' needs. The self-paced format and short length of the modules was a strength. For instance, FGP-3 commented that "the courses are shorter, and up to the point, The explanation [of Kubernetes] is up to the point." The shortness and relevance of the courses was cited as being a motivational factor for participants to engage and complete the modules. While the participants in the second focus group had not completed any of the courses, they both provided insight into why. For example, FGP-5 was simply exploring the platform as a tool that he could make available to students in his own classes. FGP-6, who also had not completed any of the courses, indicated that shorter modules would motivate him to complete more of the MCD platform. When asked why he had not completed any of the courses he said that he was a full-time worker and student. Due to this he was looking for modules that were, "shorter and more digestible." In FGP-5's review of available platforms, he indicated that MCD was highly relevant and was among the best options that he had found on the market. FGP-2 indicated the relevance of MCD for students as it was free of cost in comparison to other platforms available in the market. For example, FGP-2 commented, "AWS has some courses but sometimes you have to pay for those." In addition, participants found the flow of the chapters and instructions in the course adequate. For example, FGP-5 commented specifically on the availability of lists of the courses with short descriptions on the homepage "as they build you up with the content."

Participants agreed that the assessments helped them to learn the concept of Kubernetes. For example, FGP-1 specifically commented that the practical examples and challenges in the assessment helped him to learn "how to use the Kubernetes tool through a practical application," although he felt that more questions were needed to get a good average score. FGP-3 also noted that the MCD platform "mainly concentrated on the learning part" and the assessment had only a "few number of questions." Participants found the design of feedback received on the assessments was helpful to understand their learning progress. For example, FGP-4 indicated that the use of gold, silver, and bronze colors as knowledge level badges directed the attention of participants to where they stand in comparison to their peers. This was also highlighted by both FGP-5 and FGP-6 in Focus Group 2 with FGP-6 saying. "I liked that they kind of broke down the knowledge level... having gold, silver

TABLE III Demographics for the Focus Groups Participant (FGP) pool with various professions.

Focus Group #	Participant	Age	Gender	Profession	Ethnicity
1 (Group)	FGP-1	24	Male	Software development engineer	White or Caucasian
1 (Group)	FGP-2	25	Male	Full-time student in Computer Science	White or Caucasian
1 (Group)	FGP-3	23	Male	Full-time student in Computer Science	Asian or Asian American
1 (Group)	FGP-4	23	Female	Full-time student in Computer Science	Asian or Asian American
2 (Group)	FGP-5	58	Male	Faculty	White or Caucasian
2 (Group)	FGP-6	27	Male	Software Developer and Aerospace Engineering graduate student	White or Caucasian

and bronze levels can be easily understood, and they provided information of what makes it a gold knowledge level versus a bronze. So, yeah, I think it was nice".

Participants found the MCD platform "modern, responsive, and very intuitive to use" (FGP-1). FGP-2 indicated that the platform had a user-friendly interface, however it was "a struggle going back [to the previous chapters within a module]. The page has only a forward button and doesn't have any back button." Overall, some participants felt that there were not enough questions in the assessment, but they appreciated the practical examples and challenges included in the assessment. Participants also found the feedback received on the assessments to be helpful in understanding their learning progress, and appreciated the use of colors to indicate knowledge level badges. These findings suggest that the assessments and feedback provided on the platform are effective in helping users learn about Kubernetes, but that improvements to the navigation and user interface of the platform could be beneficial.

Participants provided a number of recommendations for improvement and implementation of the platform. In line with the design goal, FGP-5 and FGP-6 noted the relevance of using GitHub as a sign-in platform for cloud computing and recommended continued use of GitHub for quick access to the platform. However, in Focus Group 1, FGP-1, FGP-2 and FGP-3 expressed concerns with the sign-in option using GitHub only. FGP-2 suggested using Google as an additional sign-on because non-computer science users "might have difficulties [signing-in into the MCD platform] because they don't usually have Github." FGP-1 and FGP-3 noted that the requirement of instructor's authentication to get access to the courses was tedious. In this regard, FGP-1 suggested using the university's portal, e.g., Canvas as an additional sign-in option for students.

In the second focus group, FGP-6 commented that the interface sometimes had issues when adapting to different browser resolutions. He said, "images didn't adjust their size well…the text would be overlapping with the images next to the course titles." FGP-6 also indicated that they wished that links to outside resources would open in a new tab.

VII. Conclusion

In this paper, we presented the Mizzou CloudDevOps (MCD) platform, an interactive, self-service oriented and user-friendly platform for online learning of Cloud DevOps concepts. The MCD platform features learning to use of state-of-the-art tools/technologies such as e.g., Kubernetes, KubeEdge, SDN using VXLAN, CI/CD with Jenkins, VMWare Tanzu (CE), Ansible, Kubeflow and Prometheus/Grafana in the context of real-world applications. Eight learning modules are included in the MCD platform, and each module features lab exercises that have exemplar application use cases in container cluster orchestration, cyber defense, trusted edge computing, data pipeline automation, drone video streaming and analytics, cybersecurity operations for security auditing, health informatics machine learning pipelines, and system log analysis for video transmission. The target audience for MCD include students with computer science as well as non-computer science backgrounds (e.g., industrial engineering, healthcare). We described our MCD platform design principles, experiences, and findings (via knowledge growth, usability and focus group studies with over 200 participants) to achieve desired student learning outcomes pertaining to various real-world applications.

As part of future work, efforts can be focused on integrating various Cloud DevOps concepts for showing synergies between multiple DevOps technologies together to achieve diverse application workflow expectations (i.e., ensuring that the highest quality of software and infrastructure (with minimal defects) are routinely created and maintained for data/compute-intensive applications).

References

[1] "How DevOps can accelerate software development process," Available at https://www.embeddedcomputing.com/technology/software-and-os/ides-application-programming/how-devops-can-accelerate-software-development-process.

[2] "DevOps stats and facts – all the numbers you might ever need on DevOps in 2022 [online]," Available at https://kruschecompany.com/devops-statistics-and-facts/.

[3] "DevOps Market Size, Share, Analysis Report By Component (Solutions, Services), By Deployment (On-premise, Cloud), By Enterprise Size (Large Enterprises, Small Medium Enterprises), By Application (IT, Telecom, BFSI, Retail), And Region Forecast, 2022 - 2030 [online]," Accessed: February 10, 2023. Available at https://www.acumenresearchandconsulting.com/devops-market.

[4] "How DevOps is transforming higher education, June 2021." Available at https://news.elearninginside.com/how-devops-is-transforming-higher-education/.

[5] "DevOps and AWS." Available at https://aws.amazon.com/devops/.

[6] "VMware Tanzu DevOps," Available at https://tanzu.vmware.com/devops.

[7] "DevOps Essentials, by QwikLabs," Available at https://www.qwiklabs.com/quests/96.

[8] "Azure DevOps," Available at https://azure.microsoft.com/en-gb/services/devops.

[9] "LinkedIn Learning [online]," Available at https://www.linkedin.com/learning.

[10] "Udemy [online]," Available at https://www.udemy.com.

[11] "Udacity [online]," Available at https://www.udacity.com.

[12] "Coursera [online]," Available at https://www.coursera.com.
[13] "Edx [online]," Available at https://www.edx.com.
[14] "An Introduction to Cloud and DevOps Concepts, Harvard University. [online]," Available at https://scholar.harvard.edu/molloy/classes/introduction-cloud-and-devops-concepts.
[15] "DevOps II: Development Strategies, University of Minnesota. [online]," Available at https://ccaps.umn.edu/credit-courses/devops-ii-development-strategies.
[16] "DevOps and Secure Software Development, John's Hopkins University. [online]," Available at https://ep.jhu.edu/courses/605609-devops-and-secure-software-development/.
[17] S. Wang, R. Neupane, A. Pandey, X. Cheng, and P. Calyam, "Online learning platform for application-inspired cloud and devops curriculum," in *2021 IEEE 28th International Conference on High Performance Computing, Data and Analytics Workshop (HiPCW)*. IEEE, 2021, pp. 35–42.
[18] "Amazon AWS Cloud. [online]," Available at https://aws.amazon.com/.
[19] "Google Cloud Platform. [online]," Available at https://cloud.google.com/.
[20] "PRP Nautilus Cloud. [online]," Available at https://nautilus.optiputer.net/.
[21] "GENI - Open cloud infrastructure. [online]," Available at https://portal.geni.net/.
[22] "Kubernetes. [online]," Available at https://kubernetes.io/.
[23] Y. Xiong, Y. Sun, L. Xing, and Y. Huang, "Extend cloud to edge with kubeedge," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2018, pp. 373–377.
[24] "Tanzu Community Edition [online]," Available at https://tanzucommunityedition.io/.
[25] "Red Hat Ansible [online]," Available at https://www.ansible.com/.
[26] "Kubeflow [online]," Available at https://www.kubeflow.org/.
[27] "Prometheus [online]," Available at https://prometheus.io/.
[28] "Grafana [online]," Available at https://grafana.com/.
[29] J. B. Baillieul, L. O. Hall, J. M. Moura, S. S. Hemami, G. Setti, G. Grenier, M. B. Forster, F. Zappulla, J. Keaton, D. McCormick *et al.*, "The first ieee workshop on the future of research curation and research reproducibility," 2017.
[30] F. Zhao, X. Niu, S.-L. Huang, and L. Zhang, "Reproducing scientific experiment with cloud devops," in *2020 IEEE World Congress on Services (SERVICES)*. IEEE, 2020, pp. 259–264.
[31] M. De Bayser, L. G. Azevedo, and R. Cerqueira, "Researchops: The case for devops in scientific applications," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 1398–1404.
[32] A. Qumer Gill, A. Loumish, I. Riyat, and S. Han, "Devops for information management systems," *VINE Journal of Information and Knowledge Management Systems*, vol. 48, no. 1, pp. 122–139, 2018.
[33] L. de Aguiar Monteiro, D. S. M. P. Monteiro, W. H. C. Almeida, A. C. de Lima, and I. S. Sette, "Methods of implementation, maturity models and definition of roles in devops frameworks: A systematic mapping," in *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2020, pp. 1766–1773.
[34] R. Chatley and I. Procaccini, "Threading devops practices through a university software engineering programme," in *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 2020, pp. 1–5.
[35] Y. Demchenko, Z. Zhao, J. Surbiryala, S. Koulouzis, Z. Shi, X. Liao, and J. Gordiyenko, "Teaching devops and cloud based software engineering in university curricula," in *2019 15th International Conference on eScience (eScience)*. IEEE, 2019, pp. 548–552.
[36] M. Radenković, S. Popović, and S. Mitrović, "Project based learning for devops project based learning for devops: School of computing experiences," in *E-business technologies conference proceedings*, vol. 2, no. 1, 2022, pp. 127–131.
[37] F. Tuo, Y. Bai, S. Long, Y. Luo, T. Wang, R. Wang, and G. Yin, "A new model of docker-based e-learning in hadoop," in *Proceedings of the 2018 International Conference on Distance Education and Learning*, 2018, pp. 22–31.
[38] X. Bai, D. Pei, M. Li, and S. Li, "The devops lab platform for managing diversified projects in educating agile software engineering," in *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2018, pp. 1–5.
[39] W. W. Royce, "Managing the development of large software systems: concepts and techniques," in *Proceedings of the 9th international conference on Software Engineering*, 1987, pp. 328–338.
[40] N. H. Rostami, E. Kheirkhah, and M. Jalali, "Web services composition methods and techniques: A review," *International Journal of Computer Science, Engineering & Information Technology*, vol. 3, no. 6, pp. 10–5121, 2013.
[41] "5 ways DevOps can improve IT security [Online]," Available at https://devops.com/5-ways-devops-can-help-improve-security/.
[42] P. Narang and P. Mittal, "Performance assessment of traditional software development methodologies and devops automation culture," *Engineering, Technology & Applied Science Research*, vol. 12, no. 6, pp. 9726–9731, 2022.
[43] "Explore the 30 Best DevOps Tools to Look Out For in 2023 [online]," Available at https://kinsta.com/blog/devops-tools/.
[44] "All you need to know about DevOps cost [online]," Accessed: August 5, 2023. Available at https://thedailyguardian.com/all-you-need-to-know-about-devops-cost/.
[45] "Get around the biggest disadvantage of DevOps: Cost [online]," Accessed: August 5, 2023. Available at https://www.techtarget.com/searchitoperations/opinion/Get-around-the-biggest-disadvantage-of-DevOps-Cost.
[46] "How to Become a DevOps Engineer: Career Path, Skills, Jobs [online]," Accessed: August 5, 2023. Available at https://www.bestcolleges.com/engineering/how-to-become-devops-engineer/.
[47] S. Thoo, P. Maguire, and R. Moorhead, "Giving feedback to learners in the practice," *Australian family physician*, vol. 33, no. 9, 2004.
[48] P. Dawson, D. Carless, and P. P. W. Lee, "Authentic feedback: supporting learners to engage in disciplinary feedback practices," *Assessment & Evaluation in Higher Education*, vol. 46, no. 2, pp. 286–296, 2021.
[49] "DockerHub [online]," Available at https://hub.docker.com/.
[50] "GitHub [online]," Available at https://www.github.com/.
[51] "Mizzou Cloud and DevOps Platform," Available at https://www.mizzouclouddevops.net/.
[52] "What is Markdown? [Online]," Available at *https://www.markdownguide.org/getting-started/*.
[53] B. S. Bloom *et al.*, "Taxonomy of educational objectives. vol. 1: Cognitive domain," *New York: McKay*, vol. 20, no. 24, p. 1, 1956.
[54] "Spring Boot [online]," Available at https://spring.io/projects/spring-boot.
[55] C. Qu, P. Calyam, J. Yu, A. Vandanapu, O. Opeoluwa, K. Gao, S. Wang, R. Chastain, and K. Palaniappan, "DroneCOCoNet: Learning-based edge computation offloading and control networking for drone video analytics," *Future Generation Computer Systems*, vol. 125, pp. 247–262, 2021.
[56] R. L. Neupane, T. Neely, P. Calyam, N. Chettri, M. Vassell, and R. Durairajan, "Intelligent defense using pretense against targeted attacks in cloud platforms," *Future Generation Computer Systems*, vol. 93, pp. 609–626, 2019.
[57] A. Pandey, Z. Lyu, T. Joshi, and P. Calyam, "OnTimeURB: Multi-cloud resource brokering for bioinformatics workflows," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2019, pp. 466–473.
[58] R. Oruche, V. Gundlapalli, A. P. Biswal, P. Calyam, M. L. Alarcon, Y. Zhang, N. R. Bhamidipati, A. Malladi, and H. Regunath, "Evidence-based recommender system for a covid-19 publication analytics service," *IEEE Access*, 2021.
[59] S. Srinivas and H. Salah, "Consultation length and no-show prediction for improving appointment scheduling efficiency at a cardiology clinic: a data analytics approach," *International Journal of Medical Informatics*, vol. 145, p. 104290, 2021.
[60] "Mizzou Cloud and DevOps [Online]," Available at *https://tinyurl.com/48bzwuj4*.
[61] "Amazon Elastic Kubernetes System (EKS). [online]," Available at https://aws.amazon.com/.
[62] R.-C. Chen *et al.*, "Automatic license plate recognition via sliding-window darknet-YOLO deep learning," *Image and Vision Computing*, vol. 87, pp. 47–56, 2019.
[63] "Flask [online]," Available at https://flask.palletsprojects.com/en/2.2.x/.
[64] D. Smith and S. Srinivas, "A simulation-based evaluation of warehouse check-in strategies for improving inbound logistics operations," *Simulation Modelling Practice and Theory*, vol. 94, pp. 303–320, 2019.

[65] A. Bangor, P. Kortum, and J. Miller, "Determining what individual sus scores mean: Adding an adjective rating scale," *Journal of usability studies*, vol. 4, no. 3, pp. 114–123, 2009.

[66] M. Van Den Haak, M. De Jong, and P. Jan Schellens, "Retrospective vs. concurrent think-aloud protocols: testing the usability of an online library catalogue," *Behaviour & information technology*, vol. 22, no. 5, pp. 339–351, 2003.

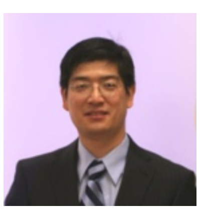
[67] J. W. Creswell and C. Poth, "Qualitative inquiry &research design: Choosing among five approoaches, ter," *Ahmad Lintang Lazuardi, Cet. I*, 2015.

[68] J. Broak, "Sus: a 'quick and dirty'usability scale, usability evaluation in industry," 1996.

[69] J. Sauro, *A practical guide to the system usability scale: Background, benchmarks & best practices*. Measuring Usability LLC, 2011.

Roshan Lal Neupane received his MS degree in Computer Science from University of Missouri-Columbia. He is currently pursuing PhD in Computer Science at the University of Missouri-Columbia. His research interests include cybersecurity, Blockchain, cloud/edge computing, Internet of Things, artificial intelligence, and cloud and network security. He is a student member of IEEE.

Prasad Calyam received his MS and PhD degrees from the Department of Electrical and Computer Engineering at The Ohio State University in 2002 and 2007, respectively. He is currently the Greg L. Gilliom Professor of Cybersecurity in the Department of Electrical Engineering and Computer Science at University of Missouri-Columbia. Previously, he was a Research Director at the Ohio Supercomputer Center. His current research interests include distributed and cloud computing, computer networking, and cybersecurity. He is a Senior Member of IEEE.



Songjie Wang received the MS degree from the Department of Electrical Engineering and Computer Science at the University of Missouri-Columbia in 2018, and the Ph.D. degree in plant science at the same university in 2009. He is currently a Cyber Infrastructure Engineer with the College of Engineering, University of Missouri-Columbia. His research interests include cloud and edge computing, computer networking, Internet of Things, cybersecurity, and big data.

Kiran Neupane is currently pursuing his MS degree in Data Science and Analytics from the University of Missouri-Columbia in the department of Data Science and Informatics. He is a Graduate Research Assistant at the Cyber Education, Research and Infrastructure Center (CERI) at the university. His research interests include Data Engineering, Artificial Intelligence and Machine Learning, Statistical Modeling and Visualization, Cloud Computing, and Cybersecurity.

Ashish Pandey received his B. tech degree in Mechanical Engineering from Indian Institute of Technology, Jodhpur, India. He is currently pursuing his PhD degree in the Department of Electrical Engineering and Computer Science at University of Missouri-Columbia. His current research interests include cloud computing, recommender systems, artificial intelligence, machine learning, and bioinformatics.

Xiyao Cheng received her MS degree in Computer Science from China Agriculture University. She is currently pursuing Ph.D. degree in the Department of Electrical Engineering and Computer Science at University of Missouri-Columbia. Her current research interests include knowledge graph based data modeling and conversational agents.

Durbek Gafurov received his Bachelor's degree from Politecnico di Torino. He is currently pursuing his MS degree in the Department of Electrical Engineering and Computer Science at the University of Missouri-Columbia. His current research interests include network and compute-intensive applications.

Hemanth Sai Yeddulapalli received his BTech degree from G. Pulla Reddy Engineering College, Kurnool. He is currently pursuing MS degree in Department of Electrical Engineering and Computer Science at University of Missouri-Columbia. His research interests include Cloud Computing and Machine Learning.

Noah Glaser received his MA and PhD degrees from the Department of Educational Studies at the University of Cincinnati in 2016 and 2020, respectively. He is currently an Assistant Professor at the University of Missouri-Columbia's School of Information Science & Learning Technologies, where he is also the director of the Information Experience Lab. His research interests include design research to create learning interventions that use innovative technologies such as virtual reality, video games, mobile devices, and artificial intelligence.

Kanu Priya Singh received her MS degree in Information Systems from Drexel University. Currently, she is a PhD candidate at the School of Information Science and Learning Technologies at the University of Missouri-Columbia. Before joining her PhD studies in 2019, she had worked as an Information Technology process consultant at Accenture Solutions in India. Her research interests include Human-Computer Interaction and Computer Supported Cooperative Work. Kanu is a student member of IEEE.

Yuanyuan Gu received her MA degree in TESOL (Teaching English to Speakers of Other Languages) from Stony Brook University in New York and MS degree in Learning Technologies and Design from University of Missouri-Columbia. She is currently pursuing Ph.D. degree in the School of Informational Science & Learning Technologies at University of Missouri-Columbia. Her research interests include game-based learning, virtual reality enabled learning environment and learning experience design.

Shangman (Eunice) Li received her Master of Education from Colorado State University. She is currently a Ph.D. candidate in the School of Information Science & Learning Technologies at the University of Missouri-Columbia. Her research interests include, user experience study, immersive learning, and accessibility. She is an active member of AECT (Association of Educational Communications and Technology) and AERA (American Educational Research Association).

Sharan Srinivas received his MEng and PhD degrees in Industrial Engineering and Operations Research from the Pennsylvania State University, USA. He is currently an Assistant Professor with a joint appointment in the Department of Industrial and Manufacturing Systems Engineering and the Department of Marketing at the University of Missouri. He is also a core faculty member at the MU Institute for Data Science and Informatics. His area of specialization is data analytics and operations research with research interests in healthcare operations management, transportation/logistics, smart service systems and supply chain.