Course Code	Course name	L	T	P	C
	DevOps Lab	0	0	2	1
Total Units to be Covered:		**Total Contact Hours:**			
Prerequisite(s):	Elementary knowledge of DevOps	**Syllabus version: 1.0**			

Course Objectives

The student should be able to understand and apply concepts of horizontal scaling techniques and differentiate it from vertical scaling techniques.

Course Outcomes

On completion of this course, the students will be able to

CO1: Develop microservices-based applications.
CO2: Gain the skills to design and implement RESTful APIs
CO3: Understand and implement various communication patterns between microservices.
CO4: Containerize microservices using Docker and effectively deploy them.
CO5: Demonstrate proficiency in using Spring Boot Actuator and other monitoring tools.

CO-PO Mapping

Course Outcomes	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO 1**	**PSO 2**	**PSO 3**
CO1	1			1	1										3
CO2	1			1	1										3
CO3	1	2		2	1										3
CO4	1			3	1										3
CO5	1			1	1										3
Average	1	2		1.6	1										3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: To understand the core principles of DevOps and its significance in modern software development using DevOps tools and sample code repository.

Experiment 2: To explore the components and benefits of Application Release Automation (ARA) in the context of DevOps using ARA tools.

Experiment 3: To understand the concepts of Continuous Integration (CI) and how to set up a basic CI pipeline for a software project using CI tools.

Experiment 4: To explore the concepts and practices of Continuous Deployment (CD) and how they contribute to automated software delivery using CD tools.

Experiment 5: To understand the history, basic operations, and different types of Version Control Systems (VCS), and to compare their features and limitations using VCS software and sample code repository.

Experiment 6: To delve deeper into Git, a popular Distributed Version Control System (DVCS), and understand its advantages over centralized VCS using VCS software and sample code repository.

Experiment 7: To understand the concept of containerization, its advantages over traditional shipping methods, and the basics of container runtime and images using Docker and sample container images.

Experiment 8: To explore the internals of container runtimes, including underlying technologies like the chroot system, FreeBSD Jails, and Linux Containers (LXC), and to understand their role in containerization using Docker, VirtualBox and sample container images.

Experiment 9: To understand Docker architecture and how it can be used to deploy applications across different environments, and to explore Docker Swarm and Kubernetes for container orchestration using Docker Swarm and sample application code.

Experiment 10: To explore container monitoring and management capabilities on cloud platforms and to understand the architecture of services using sample dockerized applications and monitoring tools.

Total Lab hours 15

Textbooks

1. Building Microservices: Designing Fine-Grained Systems" by Sam Newman

2. "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Jez Humble and David Farley

Reference Books

1. "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win" by Gene Kim, Kevin Behr, and George Spafford

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %