| Course Code | Course name | L | T | P | C |
|---|---|---|---|---|---|
| CSDV4009P | CICD Pipeline and Security | 4 | 0 | 0 | 4 |
| Total Units to be Covered: 06 | Total Contact Hours: 60 | | | | |
| Prerequisite(s): | Container Orchestration and Security-CSDV3019P | Syllabus version: 1.0 | | | |

**Course Objectives**

The course objective of "CICD Pipeline and Security" is to provide students with a comprehensive understanding of Continuous Integration and Continuous Deployment (CICD) principles, practices, and security considerations. The course aims to equip students with the knowledge and skills to design, configure, and implement efficient and reliable CICD pipelines. It focuses on integrating security practices into the CICD process, including automated security testing, vulnerability scanning, code analysis, and secure deployment practices. The course also covers DevSecOps principles, emphasizing collaboration and the integration of security throughout the software development and operations lifecycle. Students will learn about various tools and technologies used in CICD pipelines, as well as deployment strategies, automated testing, monitoring techniques, and industry best practices. The ultimate goal is to enable students to build secure and efficient CICD pipelines that ensure the continuous delivery of high-quality software.

**Course Outcomes**

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

**CO1.** Explore the CICD toolchain, including version control, build automation, and containerization by understanding the core concepts and benefits of CICD pipelines.

**CO2.** Integrate security practices into CICD pipelines, including automated security testing and vulnerability scanning.

**CO3.** Design and implement scalable and reliable CICD workflows.

**CO4.** Apply DevSecOps principles to foster collaboration and security throughout the software development process.

**CO-PO Mapping**

| Program Outcomes / Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | - | - | - | - | 3 | - | - | - | - | - | 3 | 3 | - | 2 | - |
| CO 2 | - | - | - | - | 3 | - | - | - | - | - | 2 | 3 | - | 2 | - |
| CO 3 | - | - | - | - | 3 | - | - | - | - | - | 2 | 3 | - | 2 | - |
| CO4 | - | - | - | - | 3 | - | - | - | - | - | 2 | 3 | - | 2 | - |
| Average | - | - | - | - | 3 | - | - | - | - | - | 2.25 | 3 | - | 2 | - |

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

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

**Syllabus**

**Unit I: Introduction to CICD Pipelines and Security** **10 Lecture Hours**

Overview of CICD principles and benefits, Introduction to security considerations in CICD pipelines, Understanding the software development lifecycle and the role of CICD

**Unit II: Version Control and Source Code Management** **10 Lecture Hours**

Introduction to version control systems (e.g., Git, SVN), Branching strategies and best practices, Integrating version control into CICD pipelines

**Unit III: Security Integration in CICD Pipelines** **10 Lecture Hours**

Identifying security vulnerabilities in software development, Secure coding practices and code analysis tools, Integrating security practices into CICD pipelines

**Unit IV: Automated Testing and Quality Assurance** **10 Lecture Hours**

Different types of automated testing (e.g., unit testing, integration testing), Implementing test automation in CICD pipelines, Continuous quality assurance and code quality monitoring

**Unit V: Deployment Strategies and Orchestration** **10 Lecture Hours**

Overview of deployment strategies (e.g., blue-green, canary, rolling deployments), Infrastructure as Code (IaC) and configuration management, Deployment orchestration and containerization (e.g., Docker, Kubernetes)

**Unit VI:** **10 Lecture Hours**

Security controls and practices in CICD pipelines, Vulnerability scanning and management, Compliance monitoring and reporting in CICD pipelines.

**Total lecture Hours 60**

**Textbooks**


1. Jez Humble and David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Addison-Wesley, 2010.
2. Julien Vehent, "Secure DevOps: A Practical Guide to Securing Your Software Delivery Pipeline", Manning, 2018.
3. Heather Adkins, Betsy Beyer, Paul Blankinship, and Piotr Lewandowski, “Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems", O′Reilly, 2020.


**Reference Books**


1. Gene Kim, Jez Humble, Patrick Debois, and John Willis, "DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations", It Revolution Press, 2016.


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

**Examination Scheme**

| Components | IA | MID SEM | End Sem | Total |
|---|---|---|---|---|
| Weightage (%) | 50 | 20 | 30 | 100 |

| Course Code | Course name | L | T | P | C |
|---|---|---|---|---|---|
| CSDV4109P | CICD Pipeline and Security Lab | 0 | 0 | 2 | 1 |
| Total Units to be Covered: 12 | | Total Contact Hours: 30 | | | |
| Prerequisite(s): | Container Orchestration and Security Lab - CSDV3119P | Syllabus version: 1.0 | | | |

**Course Objectives**

The course objective of "CICD Pipeline and Security" is to provide students with a comprehensive understanding of Continuous Integration and Continuous Deployment (CICD) principles, practices, and security considerations. The course aims to equip students with the knowledge and skills to design, configure, and implement efficient and reliable CICD pipelines. It focuses on integrating security practices into the CICD process, including automated security testing, vulnerability scanning, code analysis, and secure deployment practices. The course also covers DevSecOps principles, emphasizing collaboration and the integration of security throughout the software development and operations lifecycle. Students will learn about various tools and technologies used in CICD pipelines, as well as deployment strategies, automated testing, monitoring techniques, and industry best practices. The ultimate goal is to enable students to build secure and efficient CICD pipelines that ensure the continuous delivery of high-quality software.

**Course Outcomes**

At the end of this course student should be able to learn:

**CO.1**. Interpret advantages of using continuous integration and continuous development in Agile.

**CO.2**. Explain anatomy of continuous delivery pipeline to automate the testing within minimum constraints.

**CO.3.** Outline continuous integration by using various tools for continuous integration and automation.

**CO.4** Understand static code analysis like data flow analysis, taint analysis, lexical analysis.

**CO-PO Mapping**

| Program Outcomes / Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | - | - | - | - | 3 | - | - | - | - | - | 3 | 3 | - | 2 | - |
| CO 2 | - | - | - | - | 3 | - | - | - | - | - | 2 | 3 | - | 2 | - |
| CO 3 | - | - | - | - | 3 | - | - | - | - | - | 2 | 3 | - | 2 | - |
| CO4 | - | - | - | - | 3 | - | - | - | - | - | 2 | 3 | - | 2 | - |
| Average | - | - | - | - | 3 | - | - | - | - | - | 2.25 | 3 | - | 2 | - |

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

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

**List of Experiments**

**Experiment No 1** Installation of Jenkins and Execution of a simple Job in Jenkins

**Experiment No 2** Jenkins Integration with GitHub

**Experiment No 3** Jenkins Integration with GitHub and Maven

**Experiment No 4** Static Code Analysis using SonarQube

**Experiment No 5** Jenkins Integration with Sonarqube

**Experiment No 6** Create Pipeline using Jenkinsfile

**Experiment No 7** Create Pipeline using Blue Ocean Plugin

**Experiment No 8** Implementing Master/Slave Architecture in Jenkins

**Experiment No 9** Uploading Artifacts on Nexus Server using Command Line

**Experiment No 10** Nexus Integration with Jenkins

**Experiment No 11** Integration of Docker with Jenkins to generate an image of generated build

**Experiment No 12** Deployment of Docker Image on Cloud/ Local server (Nexus) using Jenkins

**Total Lab hours 30**

**Textbooks**

1. Jez Humble and David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Addison-Wesley, 2010.
2. Julien Vehent, "Secure DevOps: A Practical Guide to Securing Your Software Delivery Pipeline", Manning, 2018.

3. Heather Adkins, Betsy Beyer, Paul Blankinship, and Piotr Lewandowski, “Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems", O′Reilly, 2020.

**Reference Books**

1. Gene Kim, Jez Humble, Patrick Debois, and John Willis, "DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations", It Revolution Press, 2016.

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

**Examination Scheme:** Continuous Assessment

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