UPES
End Semester Examination, May 2025

Course: CICD Pipeline and Security	**Semester: VI**
Program: **B.Tech CSE, DevOps**	**Time : 03 hrs.**
Course Code: CHGS3030	**Max. Marks: 100**

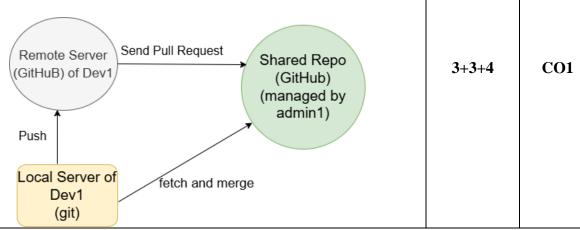**Instructions: All sections are compulsory. Write your responses according to their marks.**

SECTION A
(5Qx4M=20Marks)

S. No.		Marks	CO
Q 1	Write down the key differences between continuous delivery, continuous deployment and continuous development.	4	CO1
Q 2	What is staging in Continuous Deployment (CD)? Write down about four stages in CICD pipelines.	4	CO1
Q 3	Write the key difference between Zero-Downtime Releases and Rolling Back Deployments with an example.	4	CO1
Q 4	Explain four types of static code analysis with an example.	4	CO2
Q 5	Write down types of testing applied in the CICD pipeline.	4	CO4

SECTION B
(4Qx10M= 40 Marks)

Q 6	You are working as a developer (**Dev1**) on a collaborative project hosted on GitHub. The project is actively maintained and updated by multiple developers. Both your personal repository (<my_repo>) and the shared project repository (<shared_repo>) are based on the **main** branch. Using the provided image as a reference, complete the following tasks: 	3+3+4	CO1

	A . Explain the Role of Each Arrow
	Describe the meaning and purpose of each arrow shown in the diagram. What step in the GitHub workflow does it represent?
	B. Describe the Overall Workflow
	Explain the complete process of how Dev1 keeps their local repository synchronized with <shared_repo>, makes changes, and submits a **pull request**. Describe what is happening at each step in the collaboration process.
	C. Write the Git/GitHub Commands
	Provide the **ordered list of Git commands** that Dev1 should use to:
	
	  1. Clone the shared project
	  2. Create and push changes to <my_repo>
	  3. Create a pull request from <my_repo> to <shared_repo>
	  4. Pull updates from <shared_repo> if it gets updated
	  5. Merge the pull request if accepted
	
	Use placeholders like <my_repo> and <shared_repo> in the code where appropriate.		

Q 7	Explain the following OWASP with appropriate example: a. Injection b. Broken Authentication and Session Management c. Sensitive Data Exposure d. Broken Access Control e. Using Components With Known Vulnerabilities f. Insufficient Logging and Monitoring	**5 * 2**	**CO2**
Q 8	What is continuous monitoring in the context of CI/CD? Write a short note on either Prometheus or Nagios as a monitoring tool.	**5 + 5**	**CO3**
Q 9	What is Continuous Feedback in a CI/CD pipeline, and why is it critical to the success of DevOps? Explain how feedback should be managed for maximum effectiveness, including characteristics it should have and avoid. Also, mention the rules teams should follow when handling feedback during development.	**2+3+3+2**	**CO3**
SECTION-C (2Qx20M=40 Marks)			

SECTION-C
(2Qx20M=40 Marks)

Q 10	Explain a detailed CI/CD workflow using Jenkins as the CI/CD server to build, test, secure, and deploy a Maven-based Java project, integrated with GitHub for version control. This workflow also includes JUnit for unit testing, Snyk for static application security testing (SAST), JMeter for load testing, and Prometheus/Grafana for monitoring and feedback. Or	**4*5** **Or** **7+7+6**	**CO3**

	You are working on a software project that follows a Continuous Integration/Continuous Deployment (CI/CD) pipeline. The development team wants to ensure **automated testing and quality assurance** through the following tools. 1. JUnit 2. SonarQube 3. Snyk a. Identify at which stages of the CI/CD pipeline the above tools are typically used. Provide the reason for using each tool at that specific stage and describe what kind of testing or analysis each performs. b. Describe how the use of these tools contributes to software quality and security assurance in the pipeline. c. Provide commonly used metrics or outputs generated by each tool that help developers make informed decisions.		
Q 11	Explain the key challenges Compuware (or any other software company) faced before adopting DevOps and explain how their transition to a DevOps model, through improved agility, tool integration, and continuous delivery practices, helped them overcome those issues. Mention the tools used in their DevOps toolchain and highlight the main improvements in their performance metrics.	**5+5+10**	**CO4**