Name:

Enrolment No:

UPES

End Semester Examination, May 2025

Course: DevOps Fundamentals and SCM **Semester: IV**

Program: B.Tech CSE DevOps **Time : 03 hrs.**

Course Code: CSDV2009P **Max. Marks: 100**

Instructions: Please attempt according to the time provided and given weightage.

S. No.		Marks	CO
	SECTION A (5Qx4M=20Marks)		
Q1.	Explain the benefits to an organization that adapts DevOps.	**4**	**CO1**
Q2.	Differentiate between End User, Infrastructure and Application Monitoring.	**4**	**CO2**
Q3.	Compare and contrast between SAST and DAST.	**4**	**CO2**
Q4.	Bring out the benefits of CD.	**4**	**CO3**
Q5.	Summarize Alert Triage Process.	**4**	**CO1**
	SECTION B (4Qx10M= 40 Marks)		
Q6.	Write a short note on Ansible highlighting core concepts and using appropriate diagrams.	**10**	**CO3**
Q7.	With the help of appropriate diagrams, explain agent based and agentless monitoring.	**10**	**CO1**
Q8.	Write the commands for pushing local commits to a remote repository. Also, explain the difference between git push and git pull with examples.	**10**	**CO3**
Q9.	**“**Terraform is an open-source IaC software tool” Explain. **OR** Explain the following terms associated with GIT: Working Directory, Head, Staging Area, CVCS and DVCS.	**10**	**CO2**
	SECTION-C (2Qx20M=40 Marks)		
Q10.	You are working in a feature branch **feature/login**, and you need to merge it into the **main** branch. Demonstrate the complete process with reasoning using Git commands, how you will: **Check out the correct branches, Merge, Handle conflicts, Push the final result to the remote repository**.	**20**	**CO4**

Q11.	You are in the middle of implementing a new feature, but an urgent bug has been reported in production. You need to: **a. Save your current work** **b. Switch to the main branch** **c. Fix the bug** **d. Commit and push the fix** **e. Return to your previous work** Showcase the usage of git and related commands you may use to handle this situation. **OR** You are working on a project called **e-commerce-app**. You've just made some changes to the **checkout.js** and **payment.js** files, but you haven't committed them yet. Before proceeding, you want to check the current status of your files. After verifying the status, you review the project's commit history to confirm when the **Apply Coupon** feature was added. You find that a recent commit caused an issue in the coupon logic. This commit has already been pushed and is shared with the team. You need to undo this commit safely without affecting the rest of the team's history. Later, you realize two of your **local commits** are unnecessary and haven't been pushed yet. You decide to **remove them completely** from your local history. Using this scenario, answer the following giving appropriate reasons and command syntax: a. What Git command will you use to view which files are modified or untracked? b. How will you list recent commits and identify the one that introduced the Apply Coupon feature? c. How will you safely undo the shared commit that broke the coupon logic? d. How can you completely remove the last two commits from your local history (before pushing)?	**20**	**CO5**