Name: **Enrolment No:**	

UPES

End Semester Examination, May 2025

Course: DevOps **Semester: VI**

Program: B. Tech CSE+FSAI **Time : 03 hrs.**

Course Code: CSDV 3006 **Max. Marks: 100**

Instructions: Attempt all Questions

SECTION A
(5Qx4M=20Marks)

S. No.		Marks	CO
Q 1	Write the purpose and process of Application Release Automation (ARA). What are the key components involved in ARA, and how does it contribute to streamlined application deployment?	**4**	**CO1**
Q 2	***XYZ Technologies, a growing software development company, has been facing issues with slow application deployments, frequent bugs in production, and poor collaboration between development and operations teams. The company is considering adopting DevOps to streamline its processes but is unsure how to build a business use case for the transition.*** As a DevOps consultant, explain why building a business use case for DevOps is essential for XYZ Technologies. What key factors should be included in the business use case to convince management to adopt DevOps practices?	**4**	**CO2**
Q 3	Give a brief about the role of CI in a DevOps pipeline. How does CI contribute to improving the overall quality of software, and what are some challenges faced when implementing CI in large-scale projects?	**4**	**CO3**
Q 4	**Outline** the Git branching model and how it helps manage features, bug fixes, and releases. What are the benefits of using branches in large-scale projects?	**4**	**CO3**
Q 5	Use the **docker exec command** to access a running container and perform diagnostic operations. Explain how this command helps in troubleshooting and debugging issues inside a container without stopping or restarting it.	**4**	**CO4**

SECTION B
(4Qx10M 40 Marks)

Q 6	**Differentiate between:**	**5+5 =10**	**CO2+ CO3**

	a) Continuous Integration (CI) and Continuous Deployment (CD) in a DevOps pipeline, mentioning their processes, objectives, and the tools used for each. **(b)** Git and Subversion (SVN) as Version Control Systems (VCS), highlighting their differences in terms of architecture, usage, and advantages in a collaborative development environment.		
Q 7	**An e-commerce company wants to adopt Continuous Deployment to improve the speed and reliability of releasing new features and fixes to its production environment.** **(a)** Investigate the key **business drivers** that justify the shift to Continuous Deployment. **(b)** Give details about the structure and purpose of a **Continuous Deployment Pipeline** and the support it provides for faster and more reliable software delivery. **(c)** Evaluate the benefits and challenges of using advanced deployment strategies like **Blue-Green Deployments** or **Canary Releasing** for achieving **zero-downtime releases**.	**3+3+4 =10**	**CO4**
Q 8	Define a hypervisor and compare **Type 1 (bare-metal)** and **Type 2 (hosted)** hypervisors with examples. Explain how a hypervisor manages virtual machines, allocates resources, and supports virtualization in a cloud or DevOps environment. Evaluate the benefits and limitations of using hypervisors in modern IT infrastructure, mentioning two advantages and two challenges.	**10**	**CO2**
Q 9	In a DevOps environment, selecting the right tools is crucial for the success of continuous integration, deployment, and monitoring processes. You are tasked with selecting appropriate tools for a project that involves **source code management, continuous integration, containerization, and continuous Deployment** Discuss how you would choose the tools for the following aspects of the DevOps pipeline: • **Source Code Management** • **Continuous Integration and Continuous Deployment,** • **Containerization** Explain the criteria you would use for tool selection in each category. **OR** You are asked to containerize a Java application using Ubuntu as the base image. The following Dockerfile was written:	**10**	**CO1**

```
FROM ubuntu:20.04
RUN apt-get update && apt-get install -y
openjdk-11-jdk
WORKDIR /usr/src/app
COPY . .
RUN javac Main.java
CMD ["java", "Main"]
```

(a) Evaluate the efficiency of this Dockerfile. Suggest any two improvements to optimize the image size and build process. **(4)**

(b) Give details on how Docker's layered architecture affects the image build time and storage. Use this example to support your answer. **(6)**

SECTION-C
(2Qx20M=40 Marks)

Q 10

a) You are working on a project that requires deployment across **Development (Dev)**, **Quality Assurance (QA)**, and **Production (Prod)** environments. Explain how you would manage configuration differences and deployment tasks in these environments. Discuss common challenges such as environment-specific settings, versioning issues, and configuration drift. Propose solutions to automate deployments and ensure consistency across all environments.

b) Explain the following Git commands, their purpose, and a scenario where each would be used:

- **git clone <repository-url>**
- **git commit -m "<message>"**
- **git push origin <branch>**
- **git reset --hard <commit>**
- **git rebase -i <commit>**

10+10 =20

CO4

OR

a) **Write about the following Docker commands and instructions:**

i) The significance of the `Docker images` command and its role in managing Docker images on a system.
ii) How the `docker exec` command works and the scenarios in which it is used with running containers.
iii) The purpose of the `CMD` instruction in a Docker file and how it defines the default behavior of a container.
iv) The role of the `ENTRYPOINT` instruction in a Dockerfile and how it differs from `CMD`.

	iv) The functionality of the `Docker ADD` command in a Dockerfile and how it compares to the `COPY` command. b) Discuss how various forms of virtualization, such as those used for hardware, software, networks, storage, and desktops, optimize resource usage, improve scalability, and contribute to system efficiency. Provide real-world use cases where these virtualization technologies are applied and critically assess their impact on modern computing environments.		

Q 11	**You are tasked with automating the deployment and monitoring of a pre-built web application. Your goal is to focus on containerization, continuous integration and deployment (CI/CD), logging, monitoring, and issue management using GitHub Actions workflows.**
	
	The following **Python Flask** web application code has been provided for your reference:
	
	`app.py`:
	
	`from flask import Flask`
	`import logging`
	
	`app = Flask(__name__)`
	
	`# Set up logging`
	`logging.basicConfig(level=logging.INFO)`
	`logger = logging.getLogger(__name__)`
	
	`@app.route('/')`
	`def home():`
	`  logger.info("Home route accessed")`
	`  return "Welcome to the Flask App!"`
	
	`@app.route('/fail')`
	`def fail():`
	`  logger.error("Simulated error on /fail route")`
	`  raise Exception("Simulated Error")`
	`  return "This will not be reached"`
	
	`if __name__ == '__main__':`
	`  app.run(debug=True)`
	
	`requirements.txt`:
	
	README.md ● ! data.yaml ● requirements.txt ●
	: > Users > mchugh > Downloads > requirements.txt
	1 Flask==2.0.1
	
	Answer the following questions:
	
	a) Containerization of the Web Application [4 Marks]
		4+6+6 +4=20	**CO4**

	1. Briefly explain the steps you would take to containerize this Python Flask application using Docker. 2. What would your Dockerfile look like? Provide a basic structure. **b) Set up CI/CD Pipeline with GitHub Actions [6 Marks]** 1. Describe the steps you would take to set up a GitHub Actions workflow for continuous integration and deployment (CI/CD) of the containerized Flask app. 2. Include the key components of the .yml file (e.g., triggering on main branch, building Docker image, running tests). 3. What would be the purpose of each step in the pipeline? **c) Logging, Monitoring, and Issue Management using GitHub Actions [6 Marks]** 1. How would you set up a GitHub Actions workflow to monitor the logs of the application and raise an issue in GitHub if an error (e.g., a 500 error) is detected in the logs? 2. What steps would you take to automatically create issues based on errors detected in the log files? 3. How can GitHub Actions help in tracking and managing issues? **d) Log Retention and Automated Issue Management [4 Marks]** 1. Briefly explain how you would manage log retention (e.g., automatically delete logs older than 7 days) using GitHub Actions. 2. How would you automate issue management in GitHub, such as assigning labels or closing issues based on log errors?		