**UNIT I INTRODUCTION**

Introduction, Agile development model, DevOps, and ITIL. DevOps process and Continuous Delivery, Release management, Scrum, Kanban, delivery pipeline, bottlenecks, examples.

**UNIT II SOFTWARE DEVELOPMENT MODELS AND DEVOPS**

**Software development models and DevOps:** DevOps Lifecycle for Business Agility, DevOps, and

Continuous Testing. **DevOps influence on Architecture:** Introducing software architecture, The monolithic scenario, Architecture rules of thumb, The separation of concerns, Handling database migrations, Microservices, and the data tier, DevOps, architecture, and resilience.

**UNIT III INTRODUCTION TO PROJECT MANAGEMENT**

The need for source code control, The history of source code management, Roles and code, source code management system and migrations, Shared authentication, Hosted Git servers, Different Git server implementations, Docker intermission, Gerrit, The pull request model, GitLab.

**UNIT IV INTEGRATING THE SYSTEM**

Build systems, Jenkins build server, Managing build dependencies, Jenkins plugins, and file system layout, The host server, Build slaves, Software on the host, Triggers, Job chaining and build pipelines, Build servers and infrastructure as code, Building by dependency order, Build phases, Alternative build servers, Collating quality measures.

**UNIT V TESTING AND DEVELOPMENT**

Testing Tools and automation: Various types of testing, Automation of testing Pros and cons, Selenium - Introduction, Selenium features, JavaScript testing, Testing backend integration points, Test-driven development, REPL-driven development Deployment of the system: Deployment systems, Virtualization stacks, code execution at the client, Puppet master and agents, Ansible, Deployment tools: Chef, Salt Stack and Docker

Can you frame the 10 exercises that a third year engineering student must study on devops lab covering all the syllabus given above.

Git for source code management,

Capybara for automated testing, AWS, Docker, and Ansible

for automated virtual machine provisioning and conﬁguration,

and Jenkins for Continuous Integration.

Git for source code management,

Capybara for automated testing, AWS, Docker, and Ansible

for automated virtual machine provisioning and conﬁguration,

and Jenkins for Continuous Integration.

Git - source code management,

Capybara - automated testing,

AWS, Docker, and Ansible - automated virtual machine provisioning and conﬁguration,

Jenkins - Continuous Integration.