Birla Institute of Technology & Science, Pilani
Work Integrated Learning Programmes Division
First Semester 2025-2026

Digital Learning Handout

Part A: Content Design

Course Title	Introduction to DevOps
Course No(s)	CSI ZG514/SE ZG514
Credit Units	4
Credit Model	3-1-0
Course Author	Prof. Sonika Rathi
Lead Instructor	A. ABDUL RAHMAN
Version No:	1.0
Date:	25/02/2025

Course Description:

Continual Service - continuous integration and continuous delivery; Scaling: automating infrastructure and infrastructure as code; DevOps and Cloud: platform-as-a service and DevOps, use of virtual machines and containers for deployment, Micro-services; application lifecycle management: deployment pipeline and application deployment, continuous deployment pipeline; stack management - life cycle of stack and events, resource and event monitoring, auto healing; Security: security of deployment pipeline, policy-as-code.

Course Objectives

No	**Course Objective**
CO1	To learn the key ideas and techniques to bring development and operations together to produce higher-quality software and deliver it more quickly
CO2	To learn the core principles, business and technical terms used in DevOps from perspective of business and IT teams
CO3	To gain knowledge of the Principles and practices of the DevOps Lifecycle including Continuous Integration, Continuous Inspection, Continuous delivery, Continuous deployment and Continuous monitoring
CO4	To understand the usage of tools and technologies used for implementing DevOps

Text Book(s):

T1	DevOps: A Software Architect's Perspective (SEI Series in Software Engineering) by Len Bass, Ingo Weber, Liming Zhu , Publisher: Addison Wesley (18 May 2015)
T2	Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation by Jez Humble, David Farley. Publisher: Addison Wesley, 2011

Reference Book(s) & other resources:

R1	Effective DevOps: Building A Culture of Collaboration, Affinity, and Tooling at Scale by Jennifer

	Davis , Ryn Daniels. Publisher: O'Reilly Media, June 2016
R2	The DevOPS Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations by Gene Kim, Patrick Debois, John Willis, Jez Humble, John Allspaw. Publisher: IT Revolution Press (October 6, 2016)
R3	Web Resources: 1) https://jenkins.io/ 2) https://xebialabs.com/solutions/devops/ 3) https://www.ibm.com/ibm/devops/us/en/casestudies/#all 4) https://git-scm.com/ 5) https://hub.docker.com/ 6) https://www.atlassian.com/git/tutorials/comparing-workflows

Learning Outcomes: Students will be able to

LO1	Explain the need for DevOps and list down the primary benefits of DevOps from perspective of business and IT teams
LO2	List the ways in which DevOps uses new tools/technologies to deliver quality software more rapidly
LO3	Illustrate the practices of version control and configuration management
LO4	Summarize the essentials of continuous integration (CI) and outline the principles and practices of continuous delivery (CD)
LO5	Implement an automated deployment pipeline and create a DevOps toolchain

Modular Content Structure

1. **Foundational Terminology and Concepts**
   - Software development lifecycle
   - The Waterfall approach
   - Agile Methodology
   - Operational Methodologies: ITIL
   - Development, Testing, Release, and Deployment Concepts
   - Provisioning, Version Control
   - Test Driven Development, Feature Driven Development
   - Behaviour-driven development
2. **Why and What is DevOps?**
   - Problems of Delivering Software
   - Principles of Software Delivery
   - Need for DevOps
   - Evolution of DevOps
   - DevOps Practices
   - The Continuous DevOps LifeCycle Process (Continuous Integration, Continuous Inspection, Continuous Deployment, Continuous Delivery, Continuous Monitoring)
   - DevOps Culture
   - Case Study- (IBM/Facebook/NetFlix)
3. **DevOps Dimensions**

- Three dimensions of DevOps – People, Process, Technology/Tools
- DevOps- Process
    - DevOps and Agile
    - Agile methodology for DevOps Effectiveness
    - Flow Vs Non-Flow based Agile processes
    - Choosing the appropriate team structure: Feature Vs Component teams
    - Enterprise Agile frameworks and their relevance to DevOps
    - Behaviour driven development, Feature driven Development
    - Cloud as a catalyst for DevOps
- DevOps – People
    - Team structure in a DevOps
    - Transformation to Enterprise DevOps culture
    - Building competencies, Full Stack Developers
    - Self-organized teams, Intrinsic Motivation
- Technology in DevOps (Infrastructure as code, Delivery Pipeline, Release Management)
- Tools/technology as enablers for DevOps

4. Source Code Management

- Version control system and its types
- Introduction to GIT
- GIT Basics commands (Creating Repositories, clone, push, commit, review)
- Git workflows- Feature workflow, Master workflow, Centralized workflow
- Feature branching
- Managing Conflicts
- Tagging and Merging
- Best Practices- clean code

5. Continuous build and code quality

- Manage Dependencies
- Automate the process of assembling software components with build tools
- Use of Build Tools- Maven, Gradle
- Unit testing
- Enable Fast Reliable Automated Testing
- Setting up Automated Test Suite – Selenium
- Continuous code inspection - Code quality
- Code quality analysis tools- sonarqube

6. Continuous Integration and Continuous Delivery

- Implementing Continuous Integration-Version control, automated build, Test
- Prerequisites for Continuous Integration
- Continuous Integration Practices
- Team responsibilities
- Using Continuous Integration Software (Jenkins as an example tool)
- Jenkins Architecture
- Integrating Source code management, build, testing tools etc., with Jenkins - plugins
- Artefacts management
- Setting up the Continuous Integration pipeline

- Continuous delivery to staging environment or the pre-production environment
- Self-healing systems

7. Continuous Deployment

- Deployment pipeline
- Human-free deployments
- Implementing and Automating the deployment process
- Deploying it to testing environments
- Releasing software into production
- Environment-based release patterns
- Rolling Back Deployments and Zero-Downtime Releases
- Blue/Green Deployment
- Rolling Upgrade
- The canary release pattern- Dark Launches

8. Continuous Monitoring

- Need for continuous monitoring
- Goals of monitoring
- Challenges of monitoring under continuous change
- Alert management
- Analytics
- Continuous customer feedback and optimization
- Use of ELK (Elasticsearch, Logstash, and Kibana) Stack

9. Configuration Management

- Infrastructure as code
- Managing Infrastructure and Environments (Production, pre-production, Test, Developer Environment)
- Environment provisioning
- Automating and Managing Server Provisioning
- Configuration management tools- Chef, Puppet
- Managing on-demand infrastructure, Auto scaling

10. Virtualization and Containerization

- Virtualization
- Virtualization vs Containerization
- Containerization using Dockers
- Docker Images
- Micro-services and Containerization
- Current Trends- Kubernetes, DevOps on Cloud, Function-As-A-Service (AWS Lambda), Security of Deployment Pipeline and Policy-as-Code

Part B: Learning Plan

Contact Session	List of Topic Title	Sub-Topics	Reference
1	**Foundational Terminology and Concepts**	• Software development lifecycle • The Waterfall approach • Agile Methodology • Operational Methodologies: ITIL • Development, Testing, Release, and Deployment Concepts • Provisioning, Version Control • Test Driven Development, Feature Driven Development • Behaviour-driven development	T2-Chapter 1 R1-Chapter 4
2	**Why and What is DevOps?**	• Problems of Delivering Software • Principles of Software Delivery • Need for DevOps • Evolution of DevOps • DevOps Practices • The Continuous DevOps LifeCycle Process (Continuous Integration, Continuous Inspection, Continuous Deployment, Continuous Delivery, Continuous Monitoring) • DevOps Culture and Team Structure • Case Study- (IBM/Facebook/NetFlix)	T1- Chapter 1 T2- Chapter 1 R1- Chapter 2,3
3	**DevOps Dimensions**	• Three dimensions of DevOps – People, Process, Technology/Tools • DevOps- Process • DevOps and Agile • Agile methodology for DevOps Effectiveness • Flow Vs Non-Flow based Agile processes • Choosing the appropriate team structure: Feature Vs Component teams • Enterprise Agile frameworks and their relevance to DevOps • Behaviour driven development, Feature driven Development • Cloud as a catalyst for DevOps	T1- Chapter1, 2,3 T2 – Chapter 6 R1- Chapter 11 R2 – Chapter 1, 3
4	**DevOps Dimensions**	• DevOps – People • Team structure in a DevOps • Transformation to Enterprise DevOps culture • Building competencies, Full Stack Developers • Self-organized teams, Intrinsic Motivation	T1- Chapter1, 2,3 T2 – Chapter 6 R1- Chapter 11 R2 – Chapter 1, 3

		• Technology in DevOps (Infrastructure as code, Delivery Pipeline, Release Management) • Tools/technology as enablers for DevOps	
5	**Source Code Management**	• Version control system and its types • Introduction to GIT • GIT Basics commands (Creating Repositories, clone, push, commit, review) • Git workflows- Feature workflow, Master workflow, Centralized workflow • Feature branching • Managing Conflicts • Tagging and Merging • Best Practices- clean code	T2-Chapter 2,14 R3- 4, 6
6	**Continuous build and code quality**	• Manage Dependencies • Automate the process of assembling software components with build tools • Use of Build Tools- Maven, Gradle • Unit testing • Enable Fast Reliable Automated Testing • Setting up Automated Test Suite – Selenium • Continuous code inspection - Code quality • Code quality analysis tools- sonarqube	T1- Chapter 5 T2- Chapter 4, 6, 13 R2-Chapter 3
7	**Continuous Integration and Continuous Delivery**	• Implementing Continuous Integration-Version control, automated build, Test • Prerequisites for Continuous Integration • Continuous Integration Practices • Team responsibilities • Using Continuous Integration Software (Jenkins as an example tool)	T2- Chapter 3, 15 R2- Chapter 3
8	**Continuous Integration and Continuous Delivery**	• Jenkins Architecture • Integrating Source code management, build, testing tools etc., with Jenkins - plugins • Artefacts management • Setting up the Continuous Integration pipeline • Continuous delivery to staging environment or the pre-production environment • Self-healing systems	T2- Chapter 3, 15 R2- Chapter 3 R3-1
9	**Continuous Deployment**	• Deployment pipeline • Human-free deployments • Implementing and Automating the deployment process • Deploying it to testing environments	T1- Chapter 6, 12 T2- Chapter 10 R2- Chapter 3, 4
10	**Continuous Deployment**	• Releasing software into production • Environment-based release patterns	T1- Chapter 6, 12

		• Rolling Back Deployments and Zero-Downtime Releases • Blue/Green Deployment • Rolling Upgrade • The canary release pattern- Dark Launches	T2- Chapter 10 R2- Chapter 3, 4
11	**Continuous Monitoring**	• Need for continuous monitoring • Goals of monitoring • Challenges of monitoring under continuous change • Alert management • Analytics • Continuous customer feedback and optimization • Use of ELK (Elasticsearch, Logstash, and Kibana) Stack	T1- Chapter 7 R1- Chapter 11
12	**Configuration Management**	• Infrastructure as code • Managing Infrastructure and Environments (Production, pre-production, Test, Developer Environment) • Environment provisioning	T2- Chapter 2, 11 R1- Chapter 14
13	**Configuration Management**	• Automating and Managing Server Provisioning • Configuration management tools- Chef, Puppet • Managing on-demand infrastructure, Auto scaling	T2- Chapter 2, 11 R1- Chapter 14
14	**Virtualization and Containerization**	• Virtualization • Virtualization vs Containerization • Containerization using Dockers • Docker Images	T1- chapter 13 R3- 5
15	**Virtualization and Containerization**	• Micro-services and Containerization • Current Trends- Kubernetes, DevOps on Cloud, Function-As-A-Service (AWS Lambda), Security of Deployment Pipeline and Policy-as-Code	T1- chapter 13 Lecture notes
16	**Revision**	• Recap of sessions 1 to 15	

Experiential Learning Components:

1. Lab work: 6
2. Project work: 0
3. Case Study: 0
4. Simulation: 0
5. Work Integrated Learning Assignment- 0
6. Design work/ Field work: 0

Objective of Experiential Learning Component:

Hands on sessions on implementation of
a) Basic git commands, branching, and merging
b) Continuous Integration using Jenkins
c) Continuous Inspection of code quality using SonarQube
d) Continuous Delivery or Deployment using Jenkins

Scope of Experiential Learning Component:

Tools and libraries: git, Jenkins, SonarQube, Docker Desktop, Docker Hub, Minikube

Lab Infrastructure:

Online/ Open source/ Virtual Lab

List of Experiments:

Lab No	Lab Objective	Session Reference
1	Exercises to demonstrate the use of GIT operations and commands (Push, pull, clone etc.,). Creating branches and merging branches using GIT	5
2	Installation of Jenkins and configuration of Jenkins to work with different version control, build and testing tools	7
3	Create jobs and projects in Jenkins	7
4	Demonstration of continuous integration with Jenkins through source code polling and build triggers	8
5	Demonstrate continuous inspection with Jenkins using sonarqube to ensure code quality	9
6	Demonstration of continuous deployment/delivery to staging/production environment with Jenkins	10

Evaluation Scheme:

Legend: EC = Evaluation Component; AN = After Noon Session; FN = Fore Noon Session

Evaluation Component	Name (Quiz, Lab, Project, Mid-term exam, End semester exam, etc.)	Type (Open book, Closed book, Online, etc.)	Weight	Duration	Day, Date, Session, Time
EC – 1*	Quiz (2 quizzes)	Online	10%	1 week	September 01-10, 2025
	Assignment/Lab Assignment (1 assignment)	Online	20 %	30 days	October 10-20, 2025
EC - 2	Mid-Semester Test	Closed Book	30%	2 hours	21/09/2025 (EN)
EC - 3	Comprehensive Exam	Open Book	40%	2 ½ Hours	30/11/2025 (EN)

EC1* (20% - 30%): Quiz (optional): 5-10 %, Lab Assignment/Assignment: 20% - 30%
Syllabus for Mid-Semester Test (Closed Book): Topics in Contact session: 1 to 8
Syllabus for Comprehensive Exam (Open Book): All topics

Important Links and Information:

eLearn Portal: https://elearn.bits-pilani.ac.in

Students must visit the eLearn portal regularly and stay updated with the latest announcements and deadlines.

Contact Sessions: Students should attend the online lectures as per the schedule provided on the eLearn portal.

Evaluation Guidelines:

1. EC-1 consists of either two Assignments or three Quizzes. Students will attempt them through the course pages on the eLearn portal. Announcements will be made on the portal in a timely manner.
2. For Closed Book tests: No books or reference material of any kind will be permitted.
3. For Open Book exams: “open book” means text/ reference books (publisher copy only) and does not include any other learning material. No other learning material will be permitted during the open book examinations. For Detailed Guidelines refer to the attached document.
   EC3 Guidelines
4. If a student is unable to appear for the Regular Test/Exam due to genuine exigencies, the student should follow the procedure to apply for the Make-Up Test/Exam, which will be made available on the eLearn portal. The Make-Up Test/Exam will be conducted only at selected exam centres on the dates to be announced later.

It shall be the responsibility of the individual student to be regular in maintaining the self-study schedule as given in the course handout, attend the online lectures, and take all the prescribed evaluation components such as Assignments/Quizzes, Mid-Semester Tests and Comprehensive Exams according to the evaluation scheme provided in the handout.

****