# 1. Project Planning & Management

## Project Proposal

Overview: The project aims to implement an end-to-end DevOps pipeline using Git, Jenkins, Maven, Docker, Kubernetes, Ansible ,Terraform, and AWS.   
The objective is to automate application build, test, and deployment processes, enhancing software delivery speed and reliability.  
Scope: The project covers source code management, continuous integration, containerization, orchestration, and cloud deployment.

## Project Plan

Timeline: The project will be completed over a 12-week period with specific milestones including:  
- Week 1: Project setup and requirement analysis.  
- Week 2: CI/CD pipeline setup with Jenkins.  
- Week 3: Docker containerization and Kubernetes orchestration.  
- Week 4: Deployment on AWS and testing.  
- Week 5: Final presentation and documentation.  
  
Milestones:  
- Initial setup and configuration completed.  
- CI/CD pipeline operational.  
- Application containerized and orchestrated.  
- Deployment on AWS successful.  
- Project documentation and presentation ready.  
  
Resource Allocation:  
- Developer: 50% time on coding and testing.  
- DevOps Engineer: 70% time on CI/CD and deployment.  
- QA Tester: 30% time on test case creation and execution.

## Task Assignment & Roles

- Developer: Responsible for coding, application development, and source code management using Git.  
- DevOps Engineer: Manages CI/CD pipelines, containerization with Docker, and orchestration with Kubernetes.  
- QA Tester: Develops test cases, executes automated tests, and ensures application quality.

## Risk Assessment & Mitigation Plan

Identified Risks:  
- Integration Failures: Jenkins pipeline issues or deployment errors.  
- Security Vulnerabilities: Exposed secrets or misconfigured cloud services.  
- Performance Issues: Application not scaling properly under load.  
  
Mitigation Strategies:  
- Implement rollback strategies and staging environments.  
- Use secret management tools like Kubernetes Secrets.  
- Perform load testing and monitor resource utilization in Kubernetes.

## Key Performance Indicators (KPIs)

- Response Time: Application should load within 3 seconds under standard load.  
- System Uptime: Maintain 99.9% uptime for the deployed application.  
- Deployment Frequency: Aim for at least 2 deployments per week.  
- Lead Time for Changes: Reduce code change deployment time to under 1 hour.  
- User Adoption Rate: Increase active user engagement by 15% within the first month.