Project Report: Integration of Jira, Bitbucket, Git, and Docker

# 1. Introduction

In modern software development, integrating issue tracking, version control, and deployment pipelines is crucial for productivity and clarity. This project sets up a minimal full-stack workflow using Jira, Bitbucket, Git, and Docker, demonstrating how each tool fits into a streamlined development lifecycle. A basic Flask application was used as the core project, containerized with Docker, with all changes tracked through commits and linked to Jira issues.

# 2. Tools Used

## Jira (Atlassian)

Jira is a widely-used issue and project management platform. It allows teams to track work items such as user stories, tasks, and bugs. In this project:  
- Jira issues (e.g., DEV-1) represented tasks.  
- Commit messages were formatted to include these issue keys to link code activity with project management.

## Bitbucket (Atlassian)

Bitbucket is a Git repository management solution built for professional teams. It offers seamless integration with Jira. In this project:  
- A Bitbucket repository hosted the source code.  
- Each commit that mentioned a Jira issue key appeared under the Jira issue's Development section.

## Git & Git Bash

Git was used to version and control the application’s source code. Git Bash enabled command-line Git operations on Windows. Developers:  
- Cloned the Bitbucket repository.  
- Made commits including Jira issue keys (e.g., DEV-1).  
- Pushed code changes to Bitbucket.

## Docker

Docker is a platform used to package and run applications in isolated environments called containers. It was used here to:  
- Create a Dockerfile that containerized the Flask application.  
- Simulate a production-ready setup for the app.  
- Show that even infrastructure code changes (like Docker config) can be tracked in Jira through commits.

# 3. Integration Workflow

## Step-by-Step Integration

1. Setup Jira Project  
 - A new software project (Kanban-style) was created on Jira.  
 - Tasks such as “Create Dockerfile” or “Build Flask app” were tracked as issues.  
  
2. Create and Link Bitbucket Repo  
 - A Bitbucket repository was initialized and linked to Jira from repository settings.  
 - The repository showed active commits and development history in linked Jira issues.  
  
3. Clone Repo and Start Development  
 - Repository was cloned via Git Bash.  
 - Files like app.py, Dockerfile, and test files were created and committed with messages referencing Jira issue keys (e.g., DEV-1: Add Dockerfile).  
  
4. Observe Jira-Bitbucket Sync  
 - After pushing commits, Jira automatically displayed the corresponding commit under the issue.  
 - No manual updates were needed; the integration handled traceability.

# 4. Outcome and Benefits

- Traceability: Each code change is tied to a specific issue in Jira, enabling full traceability.  
- Seamless Collaboration: Team members can view commit history and code diffs from within Jira.  
- Realistic DevOps Simulation: By including Docker, the project reflects how real-world teams manage infrastructure alongside app code.  
- Productivity Boost: Reduced manual status updates—Jira automatically reflects commit and branch activity from Bitbucket.

# 5. Conclusion

This integration project successfully demonstrated the end-to-end linking of development tasks (via Jira) with code commits (via Git/Bitbucket) and containerization (via Docker). By using clear commit conventions and enabling Jira–Bitbucket synchronization, we created a transparent and maintainable development process. This workflow scales well for larger teams and projects requiring continuous delivery and agile tracking.