Bamboo Working Demo & Wrap-Up

# 1. Introduction

Bamboo is a Continuous Integration and Continuous Deployment (CI/CD) server developed by Atlassian. It automates the process of building, testing, and deploying applications. It integrates seamlessly with other Atlassian products like JIRA and Bitbucket, and supports many other tools used in software development pipelines.

# 2. Objectives of the Demo

- To demonstrate the working of a CI/CD pipeline using Bamboo.  
- To show integration with version control systems like Git.  
- To run automated tests and generate reports.  
- To deploy a sample application automatically.  
- To understand wrap-up procedures such as cleanup, reporting, and monitoring.

# 3. Bamboo Pipeline Overview

Components Involved:

- Plan: A Bamboo configuration to build and deploy code.  
- Stages: Logical divisions of work (e.g., Build, Test, Deploy).  
- Jobs: Contain tasks and run in agents.  
- Tasks: Individual operations like Git checkout, compile, test, etc.  
- Artifacts: Files produced and passed between jobs.  
- Deployment Project: Handles deployment after successful builds.

# 4. Real-Time Example: Web App Deployment

Scenario:

You have a Node.js web application in a Bitbucket repository. You want Bamboo to:  
1. Pull the latest code.  
2. Run unit tests.  
3. Package the application.  
4. Deploy it to a staging server.

# 5. Step-by-Step Working Demo

## Step 1: Create a Bamboo Plan

- Plan Name: WebApp\_Deployment  
- Repository: Bitbucket Git repository  
- Branch: main

## Step 2: Configure Stages and Jobs

### Stage 1: Build

- Job 1: Checkout & Install  
 - Task 1: Git Checkout  
 - Task 2: Run npm install  
 - Task 3: Archive node\_modules and source files

### Stage 2: Test

- Job 2: Unit Testing  
 - Task 1: Run npm test  
 - Task 2: Generate test reports (e.g., JUnit format)

### Stage 3: Package

- Job 3: Build Artifact  
 - Task 1: Compress app files into .zip  
 - Artifact: webapp.zip

### Stage 4: Deploy

- Job 4: Deploy to Staging  
 - Task 1: SCP upload to a remote server  
 - Task 2: SSH run deployment script (e.g., pm2 restart)

# 6. Wrap-Up Tasks

- Notifications: Email or Slack notifications on success/failure.  
- Cleanup: Clear old artifacts, Docker containers, temp files.  
- Logs: Store and monitor build/test logs.  
- Reporting: Send weekly Bamboo reports to stakeholders.

# 7. Key Features Highlighted in Demo

|  |  |
| --- | --- |
| Feature | Description |
| Git Integration | Automatically fetches latest code |
| Parallel Execution | Jobs run simultaneously on multiple agents |
| Build Artifacts | Automatically passed between stages |
| Test Reports | Results displayed directly in Bamboo |
| Deployment | SSH and SCP used for real-time deployment |
| Notification | Alert teams on build results |

# 8. Real-Time Benefits Observed

- Faster Release Cycles: Automated builds save time.  
- Early Bug Detection: Unit tests in CI pipeline reduce bugs in production.  
- Reliable Deployments: Consistent and repeatable deployment process.  
- Team Collaboration: Integration with JIRA keeps teams in sync.

# 9. Challenges and Solutions

|  |  |
| --- | --- |
| Challenge | Solution |
| Agent Unavailability | Use elastic agents or increase number of agents |
| Test Failures | Use test retry strategies, isolate failing modules |
| Deployment Issues | Automate rollback, add health check post-deployment |

# 10. Conclusion

This Bamboo demo showcased how powerful and flexible CI/CD pipelines can be. Through a practical deployment of a Node.js app, we observed how automation streamlines development and boosts productivity. Bamboo remains a vital tool for DevOps teams aiming for agility, quality, and speed.