

## **Lab sheet:- 5**

### **Title: Implement BitBucket Operations using Git.**

#### **Objective:**

The objective of this experiment is to guide you through the process of using Git commands to interact with Bitbucket, from creating a repository to collaborating with others through pull requests.

#### **Introduction:**

Bitbucket is a web-based platform designed to provide version control, source code management, and collaboration tools for software development projects. It is widely used by teams and individuals to track changes in code, collaborate on projects, and streamline the development process. Bitbucket offers Git and Mercurial as version control systems and provides features to support code collaboration, continuous integration/continuous deployment (CI/CD), and project management.

#### **Key Features of Bitbucket:**

- **Version Control:** Bitbucket supports both Git and Mercurial version control systems, allowing developers to track changes, manage code history, and work collaboratively on projects.
- **Repositories:** In Bitbucket, a repository is a container for code, documentation, and other project assets. It houses different branches, tags, and commits that represent different versions of the project.
- **Collaboration:** Bitbucket enables team collaboration through features like pull requests, code reviews, inline commenting, and team permissions. These tools help streamline the process of merging code changes.
- **Pull Requests:** Pull requests in Bitbucket allow developers to propose and review code changes before they are merged into the main codebase. This process helps ensure code quality and encourages collaboration.
- **Code Review:** Bitbucket provides tools for efficient code review, allowing team members to comment on specific lines of code and discuss changes within the context of the code itself.
- **Continuous Integration/Continuous Deployment (CI/CD):** Bitbucket integrates with CI/CD pipelines, automating processes such as building, testing, and deploying code changes to various environments.
- **Project Management:** Bitbucket offers project boards and issue tracking to help manage tasks, track progress, and plan project milestones effectively.
- **Bitbucket Pipelines:** This feature allows teams to define and automate CI/CD pipelines directly within Bitbucket, ensuring code quality and rapid delivery.
- **Access Control and Permissions:** Bitbucket allows administrators to define user roles, permissions, and access control settings to ensure the security of repositories and project assets.

#### **Benefits of Using Bitbucket:**

- **Version Control:** Bitbucket's integration with Git and Mercurial provides efficient version control and code history tracking.

- Collaboration: The platform's collaboration tools, including pull requests and code reviews, improve code quality and facilitate team interaction.
- CI/CD Integration: Bitbucket's integration with CI/CD pipelines automates testing and deployment, resulting in faster and more reliable software delivery.
- Project Management: Bitbucket's project management features help teams organize tasks, track progress, and manage milestones.
- Flexibility: Bitbucket offers both cloud-based and self-hosted options, providing flexibility to choose the deployment method that suits the organization's needs.
- Integration: Bitbucket integrates with various third-party tools, services, and extensions, enhancing its functionality and extending its capabilities.

#### **Materials:**

- Computer with Git installed (<https://git-scm.com/downloads>)
- Bitbucket account (<https://bitbucket.org/>)
- Internet connection

#### **Experiment Steps:**

##### **Step 1: Creating a Repository**

- Sign in to your Bitbucket account.
- Click the "Create" button to create a new repository.
- Choose a repository name, visibility (public or private), and other settings.
- Click "Create repository."

##### **Step 2: Cloning a Repository**

- Open your terminal or command prompt.
- Navigate to the directory where you want to clone the repository.
- Copy the repository URL from Bitbucket.
- Run the following command:

##### ***git clone <repository\_url>***

- Replace <repository\_url> with the URL you copied from Bitbucket.
- This will clone the repository to your local machine.

##### **Step 3: Making Changes and Creating a Branch**

- Navigate into the cloned repository:

##### ***cd <repository\_name>***

- Create a new text file named "example.txt" using a text editor.
- Add some content to the "example.txt" file.
- Save the file and return to the command line.
- Check the status of the repository:

##### ***git status***

- Stage the changes for commit:

##### ***git add example.txt***

- Commit the changes with a descriptive message:

***git commit -m "Add content to example.txt"***

- Create a new branch named "feature":

***git branch feature***

- Switch to the "feature" branch:

***git checkout feature***

#### **Step 4: Pushing Changes to Bitbucket**

- Add Repository URL in a variable

***git remote add origin <repository\_url>***

- Replace <repository\_url> with the URL you copied from Bitbucket.
- Push the "feature" branch to Bitbucket:

***git push origin feature***

- Check your Bitbucket repository to confirm that the new branch "feature" is available.

#### **Step 5: Collaborating through Pull Requests**

1. Create a pull request on Bitbucket:
  - Go to the repository on Bitbucket.
  - Click on "Create pull request."
  - Choose the source branch ("feature") and the target branch ("main" or "master").
  - Review the changes and click "Create pull request."
2. Review and merge the pull request:
  - Add a title and description for the pull request.
  - Assign reviewers if needed.
  - Once the pull request is approved, merge it into the target branch.

#### **Step 6: Syncing Changes**

- After the pull request is merged, update your local repository:

***git***

***checkout***

***main git***

***pull origin***

***main***

#### **Conclusion:**

This experiment provided you with practical experience in performing Bitbucket operations using Git commands. You learned how to create repositories, clone them to your local machine, make changes, create branches, push changes to Bitbucket, collaborate through pull requests, and synchronise changes with remote repositories. These skills are essential for effective collaboration and version control in software development projects using Bitbucket and Git.