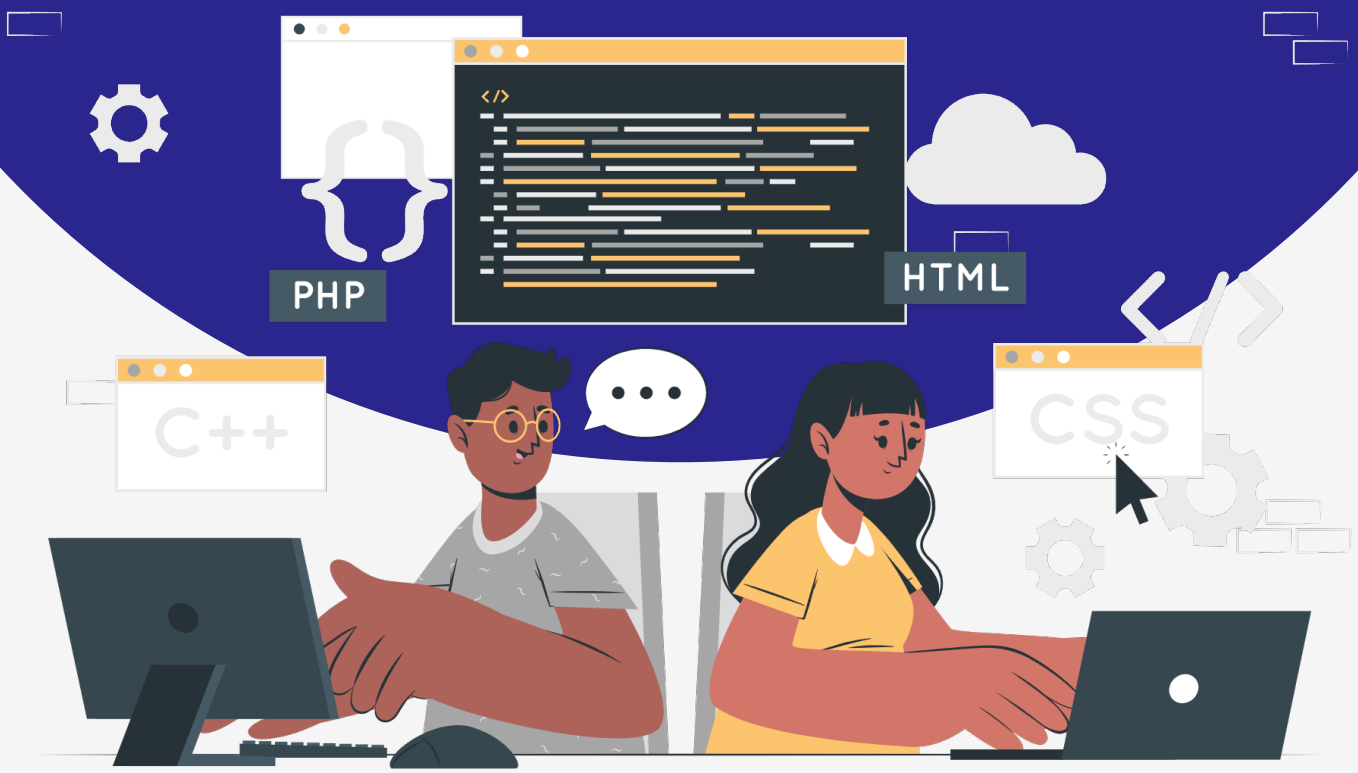
Lesson Plan

Git Repository Hosting Services



PW SKILLS

Git Repository Hosting Services

Introduction

In today's software development landscape, Git repository hosting services play a crucial role. They provide not only a place to store code but also tools for collaboration, continuous integration, issue tracking, and more. This document will explore three of the most popular Git repository hosting services: GitHub, GitLab, and Bitbucket. Each of these platforms has its own unique features, advantages, and use cases. We will delve into their specifics, comparing their functionalities and providing examples to illustrate their usage.

1. GitHub

Overview

GitHub is one of the most well-known Git repository hosting services. Founded in 2008, GitHub has grown to become the largest host of source code in the world. It offers a web-based interface, which makes it easy for developers to collaborate on projects. GitHub was acquired by Microsoft in 2018, which has further integrated it into the broader Microsoft ecosystem.

Key Features

- **Repository Hosting:** GitHub provides unlimited public repositories and a limited number of private repositories (depending on the plan).
- **Collaboration Tools:** Issues, pull requests, and project boards allow teams to work together efficiently.
- **GitHub Actions:** A powerful CI/CD tool integrated directly into the platform.
- **GitHub Pages:** Allows users to host static websites directly from a repository.
- **Security:** Advanced security features such as dependency scanning and secret management.

Example: Creating and Collaborating on a Repository

1. **Creating a Repository:**
   - Navigate to GitHub and log in.
   - Click on the "+" icon in the upper-right corner and select "New repository".
   - Fill in the repository name, description, and choose whether it should be public or private.
   - Click "Create repository".

2. **Cloning the Repository:**
   - Open your terminal.

```
git clone https://github.com/username/repository-name.git
```

3. **Collaborating:**

To contribute to the repository, create a new branch:

```
git checkout -b new-feature
```

Make changes and commit them:

```
git add .
git commit -m "Add new feature"
```

Push the changes:

```
git push origin new-feature
```

- **Create a pull request on GitHub to merge the changes.**

2. GitLab

Overview
GitLab, founded in 2011, is another popular Git repository hosting service. It is unique in that it provides both a hosted service (GitLab.com) and an open-source, self-hosted solution. This flexibility makes GitLab a popular choice for enterprises that require full control over their development infrastructure.

Key Features

- **Repository Hosting:** GitLab offers unlimited private and public repositories.
- **CI/CD:** GitLab CI/CD is built-in and provides powerful automation for testing and deployment.
- **Project Management:** Milestones, issues, and epics help in managing projects effectively.
- **Security:** Includes features like static and dynamic application security testing (SAST/DAST).
- **Integration:** Strong integration capabilities with other tools and services.

Example: Setting Up a CI/CD Pipeline

1. **Creating a Repository:**
   - Log in to GitLab and click on the "New project" button.
   - Fill in the project details and click "Create project".

2. **Setting Up CI/CD:**
   - In your repository, create a file named .gitlab-ci.yml.

Add the following content to the file:

```
stages:
  - build
  - test
  - deploy

build_job:
  stage: build
  script:
    - echo "Compiling the code..."
    - echo "Code compiled successfully."

test_job:
  stage: test
  script:
    - echo "Running tests..."
    - echo "All tests passed."

deploy_job:
  stage: deploy
  script:
    - echo "Deploying the application..."
    - echo "Application deployed successfully."
```

Commit and push the changes:

```
git add .gitlab-ci.yml
git commit -m "Add CI/CD pipeline"
git push origin main
```

- **GitLab will automatically detect the pipeline configuration and run the defined jobs.**

3. Bitbucket

Overview
Bitbucket is a repository management tool which is mainly designed for expert teams and professionals. Bitbucket is a Git repository management software that serves as a central hub for all the Git repositories. We use Bitbucket for access control, workflow control, pull request and integration for full rest API.
Bitbucket was launched in 2008 as an independent startup in Australia specializing in Mercurial project hosting. It was acquired by Atlassian, a fellow Australian corporation, in 2010, and nearly a year later added support for Git repository.

Key Features
- **Repository Hosting:** Bitbucket offers both Git and Mercurial repository hosting (Git-only after June 2020).
- **Collaboration Tools:** Pull requests, inline comments, and code reviews.
- **Pipelines:** Integrated CI/CD service.
- **Integration:** Seamless integration with Atlassian products and other third-party tools.
- **Security:** Features like IP whitelisting, two-step verification, and detailed access controls.

Example: Integrating Bitbucket with Jira

1. **Creating a Repository:**
   - Log in to Bitbucket and click on "Create repository".
   - Fill in the repository details and click "Create repository".

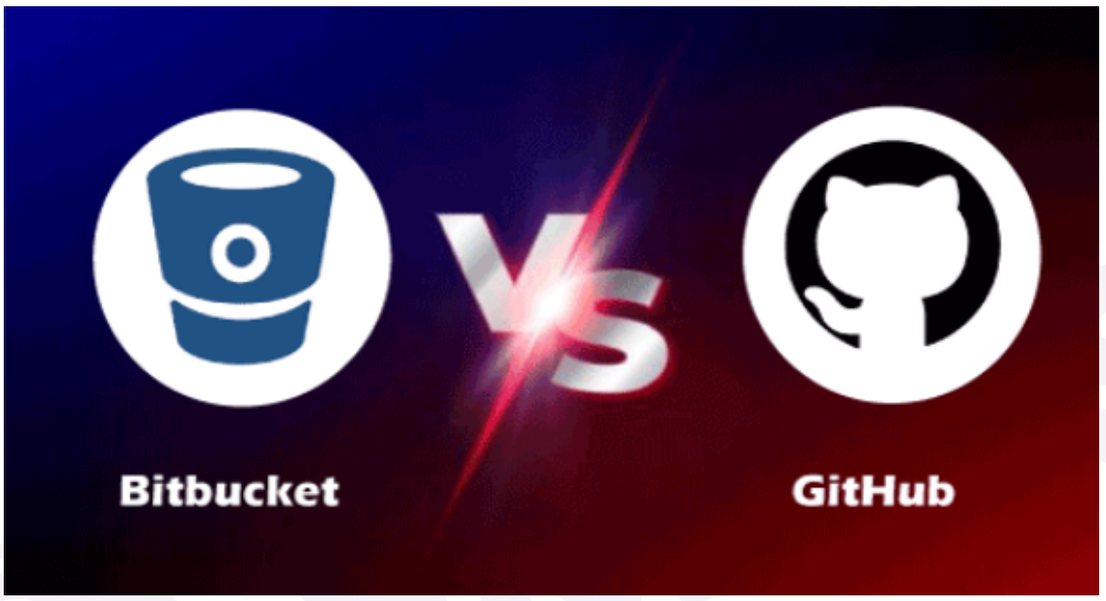
2. **Connecting to Jira:**
   - In Bitbucket, navigate to the repository settings.
   - Under "Integrations", select "Jira".
   - Follow the prompts to link your Bitbucket repository with a Jira project.

3. **Using Smart Commits:**
When committing changes, include a Jira issue key in the commit message:

```
git commit -m "PROJ-123: Fix the login bug"
```

- **This will automatically link the commit to the Jira issue, updating its status and adding the commit details to the issue.**

Difference Between Bitbucket and GitHub



S.NO	Parameter	Bitbucket	GitHub
	Developed by	It was developed by Jesper Noehr.	GitHub was developed by Chris Wanstrath, Tom Preston-Werner, P.J. Hyett, and Scott Chacon.
	Navigation	There is no feature for navigation in Bitbucket.	GitHub permits users to navigate usability.
	Version Control Systems	Mercurial and Git are supported by Bitbucket.	GitHub Supports only Git.
	Public Repository	It gives users the choice of having several free repositories.	It gives users access to an infinite amount of free storage space.

	Benefits	• It is adaptable with a variety of operating systems. • Bitbucket creates Authentication of social media support.	• It aids in the creation of a well-organized project document. • GitHub is used to share the work with the general public.
	Project Analysis	• With the help of Bitbucket, developers can visualize the analysis with charts.	• This feature is not yet available in GitHub; however, users can check the commit history.

Conclusion

GitHub, GitLab, and Bitbucket each offer a robust set of features for hosting Git repositories and facilitating collaborative software development. While GitHub is widely recognized for its community and extensive integration capabilities, GitLab stands out with its comprehensive DevOps tools and self-hosting option. Bitbucket is particularly appealing for teams already using Atlassian's suite of products. Choosing the right platform depends on the specific needs and workflows of your team, but all three provide powerful tools to enhance your development process.