

GitHub vs GitLab Features Comparison



Software development collaboration is becoming more common. Irrespective of whether you are working alone or developing enterprise software, choosing the right version control system will have a huge impact on your productivity as a team. Some examples of version control systems include GitHub and GitLab. These two platforms have much in common as they both use Git but differ in features, development workflows, and team collaboration opportunities.

Despite the fact that many developers think that there is not much difference between GitHub and GitLab because they both have Git repositories, these platforms have some features unique to them. GitHub is popular due to the community of developers and wide open-source ecosystem whereas GitLab is famous due to having all the necessary DevOps tools including CI/CD, project management, etc.

[GitHub vs GitLab comparison](#) will help you see the main features that differ in these two platforms. Knowing about their features, collaboration tools, automation capabilities, security options, and deployment workflow you will be able to choose the platform that meets the requirements of your project.

What Is GitHub?

It is a Git repository hosting service that provides a platform for developers to make collaboration easier when developing software. It was founded way back in 2008, and today it

has become one of the biggest communities for [software development](#) in the world. GitHub is being used by millions of developers and organizations for source code management, review and issues tracking, as well as contribution to various open-source projects.

GitHub integrates version control systems with collaboration tools to provide an efficient process of software development. It is possible to create repositories, do coding in various branches, review codes through pull requests and merge approved codes in production.

Key GitHub Features

- Git repository hosting
- Pull requests
- GitHub Actions for CI/CD
- Code review tools
- Branch protection rules
- Issue tracking
- Project boards
- Wiki documentation
- GitHub Packages
- GitHub Copilot AI Assistance
- Security scanning
- Marketplace integrations

Many startups, freelancers, education-related bodies, and businesses that require efficient collaboration and need access to developer communities tend to use GitHub.

What Is GitLab?

GitLab is a [DevOps platform](#) based on Git repositories. It differs from GitHub in the sense that while GitHub mainly offers services related to code management, GitLab is a full package for managing every aspect of the software delivery cycle.

Programmers can write code, review it, test it, deploy and run it and manage its performance without using third-party tools.

DevOps features offered by GitLab make it preferred by those who would like to manage their development, testing, deployment and monitoring from one single platform.

GitLab can be deployed both as a cloud-hosted and self-managed platform, providing more options to its users concerning security and infrastructure.

Key GitLab Features

- Git Repository Hosting

- Merge Requests
- CI/CD Pipeline
- Full DevOps Lifecycle Management
- Security Scanning
- Container Registry
- Package Registry
- Kubernetes
- Infrastructure as Code
- Self-Managed Deployment
- GitLab Duo AI
- Monitoring

GitLab is utilized by companies, DevOps teams, governments, enterprises, etc.

Comparison of GitHub and GitLab Features

Whereas both websites use Git for source control management, there are certain differences between the two in terms of features which serve their purpose in different ways. Hereunder, the list of their features which make a difference is provided.

Repositories Management

Both services provide users with the opportunity to create both public and private repositories and provide such tools as branching, commits, tagging, forking, and repositories cloning.

GitHub provides a very easy-to-use interface making the process of repositories management very convenient and requires minimal configuration for its implementation.

User Interface

GitHub emphasizes simplicity.

It is quite simple and responsive to use. Thus, it makes things easier for the developers to discover the repositories, pull requests, issues, and workflows without any difficulty at all.

GitLab has more features in its interface compared to GitHub. Although it becomes easier for the experienced users to have better control over it, but new users might find the dashboard more complex initially.

The organizations having multiple DevOps processes find GitLab's interface helpful because it combines many development tools together.

Features of Collaboration

Collaboration has become very important for software development.

GitHub offers the following options for collaboration to developers:

- Pull Requests
- Code Reviews
- Discussion
- Issues
- Mentions
- Permissions
- Project Boards

GitLab supports all these options and adds the capability of planning to it.

- Milestones
- Epics
- Roadmaps
- Project Management

For the organizations having multiple teams, it becomes easier for them to use GitLab.

Pull Request vs Merge Request

The GitHub platform makes use of Pull Request. Whereas GitLab makes use of Merge Request.

The only difference is just in the names, since they serve the same purposes, that is reviewing the code before it merges into the master branch.

The GitHub pull request includes:

- Commenting on code
- Code suggestions
- Approval from reviewer
- Status checks
- Reviewers required

GitLab merge requests provide:

- Quality of code reports
- Pipelines
- Security issues
- Approvals to merge
- Dependency scanning

GitLab provides more information about the development within the merge request itself.

Branch Protection

Branch protection ensures that critical branches are not altered unintentionally.

GitHub lets administrators to:

- Mandate pull request review
- Disallow force push
- Enforce status checks
- Prohibit branch deletion

GitLab supports these options as well along with approval rules, protected environments, and deployment settings which are particularly helpful for enterprise users.

Issue Tracking

Issue tracking aids in tracking bugs, feature requests, enhancements, and development tasks.

GitHub Issues provide labels, milestones, assignees, templates, and workflow management. The interface is user-friendly and works great with pull requests.

GitLab takes issue tracking one step further by providing epic management, roadmapping, dependencies, time tracking, iterations, and planning functionality. Such tools will be more useful for managing software projects with multiple teams involved.

Small teams find GitHub Issues enough while enterprise users with more planning needs may find GitLab more flexible.

Project Management

The GitHub platform provides Project Boards for Kanban-style workflow management and task visualization.

GitLab goes one step ahead in project management by offering roadmap planning, epics, milestones, iterations, value stream analysis, and portfolio planning functionalities. This reduces the requirement for project management solutions from external sources.

Agile and DevOps-based teams can particularly benefit from the project management functionalities offered by GitLab.

Conclusion

According to GitHub vs GitLab Features Comparison, both these websites are great options for developing software in the contemporary world. In case if you are working on an open-source project, startup, or if you would like a simpler interface and a big community of developers, then GitHub should be your pick. If you have a company that requires CI/CD, DevOps tools, and enterprise security go with GitLab.