**[Usersnap](https://usersnap.com/?utm_source=blogmenu&utm_medium=blog&utm_campaign=blogmenu" \o "Home)**

* [Features](https://usersnap.com/features/feedback-widget-bug-tracking/?utm_source=blogmenu&utm_medium=blog&utm_campaign=blogmenu)
* [Pricing](https://usersnap.com/pricing/?utm_source=blogmenu&utm_medium=blog&utm_campaign=blogmenu)
* [Success Stories](https://usersnap.com/success-stories/?utm_source=blogmenu&utm_medium=blog&utm_campaign=blogmenu)
* [Integrations](https://usersnap.com/integrations/?utm_source=blogmenu&utm_medium=blog&utm_campaign=blogmenu)
* [Blog](https://usersnap.com/blog/?utm_source=blogmenu&utm_medium=blog&utm_campaign=blogmenu)
* [LOGIN](https://usersnap.com/a/?utm_source=blogmenu&utm_medium=blog&utm_campaign=blogmenu)
* [TRY IT FOR FREE](https://usersnap.com/?utm_source=blogmenu&utm_medium=blog&utm_campaign=blogmenu#signup)

https://secure.gravatar.com/avatar/763cd3b0ee57dab0833be6cd5eb6db27?s=45&d=mm&r=gby [Thomas Peham](https://usersnap.com/blog/author/thomas/)

**GitLab vs GitHub: Key differences & similarities**

*This article is brought to you by*[*Usersnap*](https://usersnap.com/?utm_source=blog&utm_medium=blog&utm_campaign=blogpost)*. Usersnap helps you to communicate visually. And the best part? It connects with*[*GitLab*](https://usersnap.com/gitlab?utm_source=blog&utm_medium=blog&utm_campaign=gitlab-github_blogpost)*and*[*GitHub*](https://usersnap.com/github?utm_source=blog&utm_medium=blog&utm_campaign=gitlab-github_blogpost)*.*[*Get a 15-day free trial here*](https://usersnap.com/?utm_source=blog&utm_medium=blog&utm_campaign=gitlab-github_blogpost#signup)*.*

Version Control repository management services are a key component in the software development workflow. In the last few years, GitHub and GitLab positioned themselves as handy assistants for developers, particularly when working in large teams.

With the latest release of GitLab 10.0, GitLab took a major leap forward from code management, to deployment and monitoring. GitLab calls it Complete DevOps. They aim for the entire software development, deployment and DevOps market.

That means when talking about the differences and similarities of GitLab vs GitHub, we need to look beyond code repositories and take a look at the entire process.

Ready?

*If you like this article, give it some sharing*

**20**

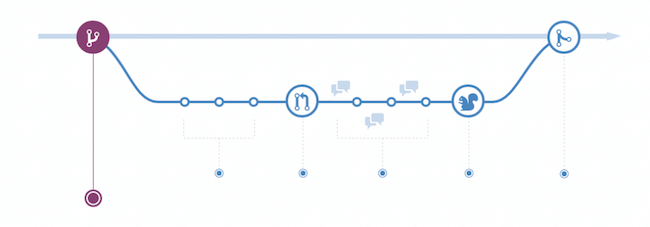
**Git explained**

Both, GitLab and GitHub are web-based Git repositories.

The aim of Git is to manage software development projects and its files, as they are changing over time. Git stores this information in a [data structure called a repository](https://www.sbf5.com/~cduan/technical/git/git-1.shtml).

Such a git repository contains a set of commit objects and a set of references to commit objects.

A git repository is a central place where developers store, share, test and collaborate on web projects.



**More than a Git repository: How to Complete DevOps**

Nowadays, GitLab and GitHub are more than “just” git repositories for developers.

GitLab says about their recently announce [Complete DevOps vision](https://about.gitlab.com/2017/10/09/gitlab-raises-20-million-to-complete-devops/):

*Now, we’re taking it a step further to unite development and operations in one user experience.*

GitLab realized the need for better and deeper integrations between development and DevOps toolchains. With the latest release of 10.0, GitLab rethinks the scope of tooling for both developers and operation teams.

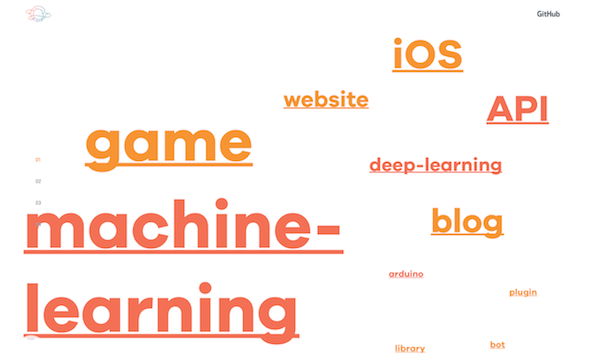
**The Basics of GitHub and GitLab**

Let’s start with the basics. GitHub is a Git based repository hosting platform with 26 million users (March 2017). Originally, GitHub launched in 2008 and was founded by Tom Preston-Werner, Chris Wanstrath, and PJ Hyett.

GitHub projects can be made public and every publicly shared code is freely open to everyone. You can have private projects as well, though, they require a paid GitHub plan.

Public repositories on GitHub are often used to share open source software. Besides the basic code repository, GitHub can be used for issue tracking, documentation and wikis.

Overall, 26 million people and 1.5 million organizations created 67 million repositories on GitHub in 2017.



*Most used repository tags on Github.com*

Similar to GitHub, GitLab is a repository manager which lets teams collaborate on code. Written in Ruby, GitLab offers some similar features for issue tracking and project management as GitHub. Founded by Dmitriy Zaporozhets and Valery Sizov in 2011, GitLab employs more than 150 people and has more than 1,400 open source contributors. According to Wikipedia, GitLab has 100,000 users (March 2017) and is used by enterprises such as IBM, Sony and NASA.

**Get great content directly delivered to your inbox.**

People from companies like Facebook, Google, and Microsoft are reading our articles.

Top of Form



Bottom of Form

**Key differences and similarities: GitLab vs GitHub**

According to various sources and our own experience, we identified the following **key differences** you should know when making the decision: ***GitLab vs GitHub***.

**Authentication Levels**

With GitLab you can set and modify people’s permissions according to their role. In GitHub, you can decide if someone gets a read or write access to a repository.

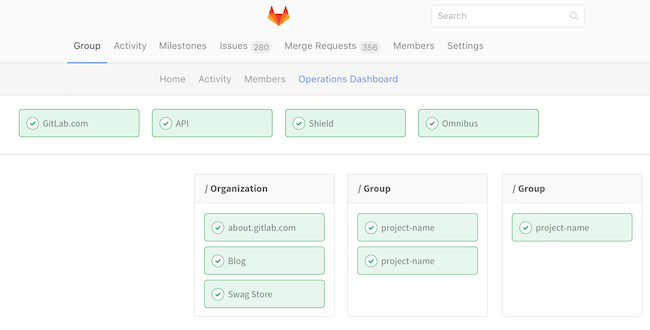
With GitLab you can provide access to the issue tracker (for example) without giving permission to the source code. This is obviously great for larger teams and enterprises with role-based contributors.

**Built-in CI / CD & going Beyond CD**

One of the big differences between GitLab and GitHub is the built-in Continuous Integration/Delivery of GitLab. CI is a huge time saver for many development teams and a great way of QA (nobody likes pull requests that break your application).

GitLab offers their very own CI for free. No need to use an external CI service. And if you are already used to an external CI, you can obviously integrate with Jenkins, Codeship and others.

As GitLab mentioned with its latest release of 10.0, GitLab is clearly addressing the DevOps market as well offering a operations dashboard that lets you understand the dependencies of your development and DevOps efforts.



*Operations Dashboard, source: gitlab.com*

It doesn’t stop there, however, as GitLab is addressing the topic of Auto CI and how to automatically run CI/CD without a human being actually setting it up.

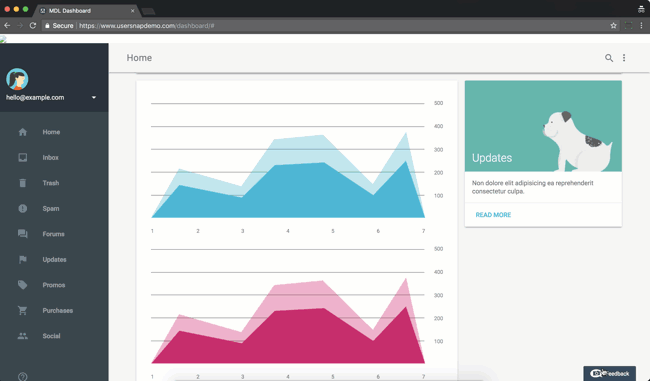
*But, really, every project should be running some kind of CI. So, why don’t we just detect when you’ve pushed up a project; we’ll just build it, and we’ll go and test it, because we know how to do testing.  
– Mark Pundsack, source: gitlab.com*

So, how does CI / CD work inside the GitHub universe? Obviously, GitHub offers various 3rd party integrations – such as Travis CI, CircleCI or Codeship – for running and testing your code. However, there’s no built-in CI solution at the moment.

**Issue Tracking**

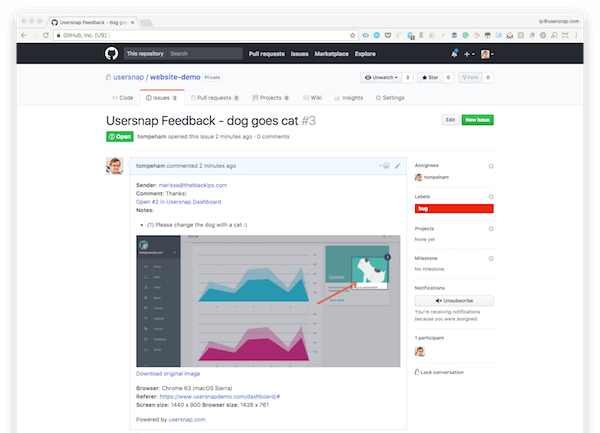
GitLab as well as GitHub provide a simple issue tracker that lets you change status and assignee for multiple issues at the same time.

Both are great issue trackers, especially when connected with a visual bug tracker like [Usersnap](https://usersnap.com/?utm_source=blog&utm_medium=blog&utm_campaign=gitlab-github_blogpost). While your developers still enjoy the great issue tracking interface of GitLab and GitHub, your testers, colleagues and clients can simply report bugs through the Usersnap widget.

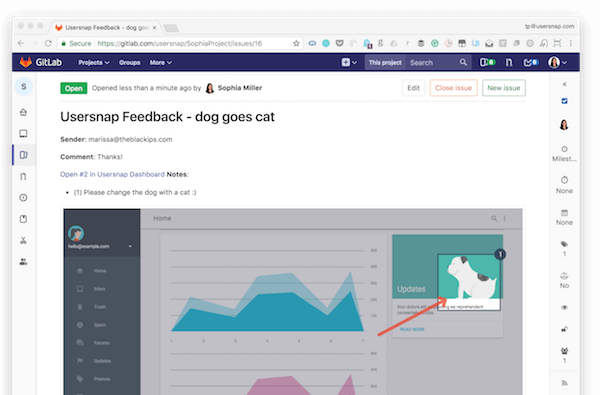


*Usersnap’s bug reporting widget in action*

Bug reports and user feedback can automatically be sent to [GitLab](https://usersnap.com/gitlab?utm_source=blog&utm_medium=blog&utm_campaign=gitlab-github_blogpost) or [GitHub](https://usersnap.com/github?utm_source=blog&utm_medium=blog&utm_campaign=gitlab-github_blogpost). Or you can pre-filter those tickets inside Usersnap and manually send it to your development project.



*Incoming bug reports displayed in your GitHub repository*

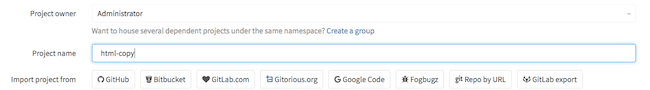


*Incoming bug reports displayed in your GitLab repository*

**Import & Export**

When thinking about moving to GitLab or GitHub, you should also consider the setup costs and resources needed for getting started. In that regard, the topic of available import and export features is pretty important.

GitLab offers a [detailed documentation](https://docs.gitlab.com/ee/user/project/import/index.html) on how to import your data from other vendors – such as GitHub, Bitbucket – to GitLab.



*Import projects from 3rd parties, source:gitlab.com*

GitHub on the other hand, does not offer such detailed documentation for the most common git repositories. However, GitHub offers to use GitHub Importer if you have your source code in Subversion, Mercurial, TFS and others.

Also when it comes to exporting data, GitLab seems to do a pretty solid job, offering you the ability to export your projects including the following data:

* Wiki and project repositories
* Project uploads
* Configuration including webhooks and services
* Issues with comments, merge requests with diffs and comments, labels, milestones, snippets, and other project entities

GitHub, on the other hand, seems to be more restrictive when it comes to export features of existing GitHub repositories.

**Integrations**

Both GitLab and GitHub offer a wide range of 3rd party integrations. Integrating your version control system with other application enriches your workflows and can boost productivity for your developers and your non-developers.

In order to check out if your favorite apps are compatible with GitLab and GitHub, I recommend checking out the documentation of GitLab and GitHub.

Besides the available integration partners, GitHub launched their GitHub marketplace in May 2017 offering you selected tools and applications.

GitLab took a similar path, and offers multiple integrations for development and DevOps teams.

**The GitHub community**

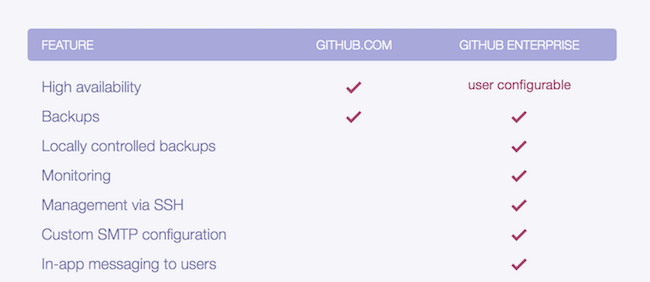
GitHub positioned itself among its community of developers. And its popularity is mainly driven by the highly active GitHub community of millions of developers. On the other hand, GitLab undertook some great activities, such as hosting community events and connecting open source contributors.

If you’re looking for the biggest community of developers, chances are high that GitHub is the better place to be.

**GitLab Enterprise vs GitHub Enterprise**

On an enterprise level, you should consider further factors when making an informed decision of whether to use GitLab vs GitHub.

GitHub is highly popular among developers, and over the last few years it gained popularity among larger development teams and organizations too.



*Custom enterprise features of GitHub*

On the other hand, GitLab is pretty strong on enterprise features, too. With different enterprise plans available, GitLab is particularly popular among larger development teams.

Here is, how GitLab and GitHub compare on pricing.

While GitHub’s enterprise plan starts at 2,500 USD per 10 users per year (= 250 USD per user), GitLab’s enterprise starter plan is 39 USD per user/per year.

| **Features** | **GitLab** | **GitHub** |
| --- | --- | --- |
| released | September 2011 | April 2008 |
| Free plans | Unlimited public and private repositories | Free for public repositories only |
| Paid plans | Starts at $39 per user per year | Starts at $84 per user per year |
| Code review features | yes | yes |
| Wiki | yes | yes |
| Bug & issue tracking | yes | yes |
| Private branch | yes | yes |
| Build system | yes | yes (with 3rd party service) |
| Import projects | yes | no |
| Export projects | yes | no |
| Time tracking | yes | no |
| Web-hosting | yes | yes] |
| Self-hosting | yes | yes (with enterprise plan) |
| Popularity | 546.000+ projects | 69.000.000+ projects |

**Wrapping it up.**

Undoubtedly, GitHub is still the most popular git repository with the largest number of users and projects. However, GitLab is doing a fantastic job offering your entire development (and DevOps) teams great tools for more efficient workflows.

**Bonus tip: Get user feedback & bug reports with Usersnap**

Last but not least, I wanted to give you a heads-up on Usersnap, which is our very own visual user feedback and bug tracking tool, used by companies like Facebook, Google, and Microsoft.

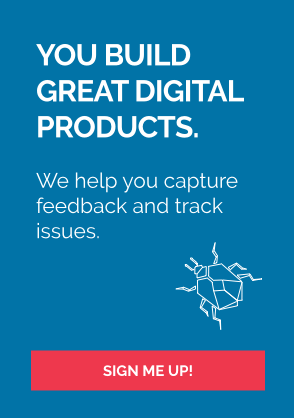
And the best part? You can connect [GitHub issues](https://usersnap.com/github?utm_source=blog&utm_medium=blog&utm_campaign=gitlab-github) or [GitLab issues](https://usersnap.com/github?utm_source=blog&utm_medium=blog&utm_campaign=gitlab-github_blogpost) with Usersnap to get visual bug reports directly sent to your preferred system.

Get great user feedback & bug reports in 2017 with a [free Usersnap trial](https://usersnap.com/?utm_source=blog&utm_medium=blog&utm_campaign=gitlab-github_blogpost#signup).

[](https://usersnap.com/?gat=blog-post&utm_source=blog&utm_medium=blog&utm_campaign=gitlab-github_blogpost#signup)

*If you like this article, give it some sharing*

**20**

* [](https://usersnap.com/?utm_source=blog&utm_campaign=sidebar&utm_medium=blog#signup)

© 2013 - 18 Usersnap GmbH | [Imprint](https://usersnap.com/legal?gat=blog) |