GitLab

Ditching DIY DevOps for GitLab's Single Platform

The evolution of DevOps: From toolchains to GitLab's DevOps platform

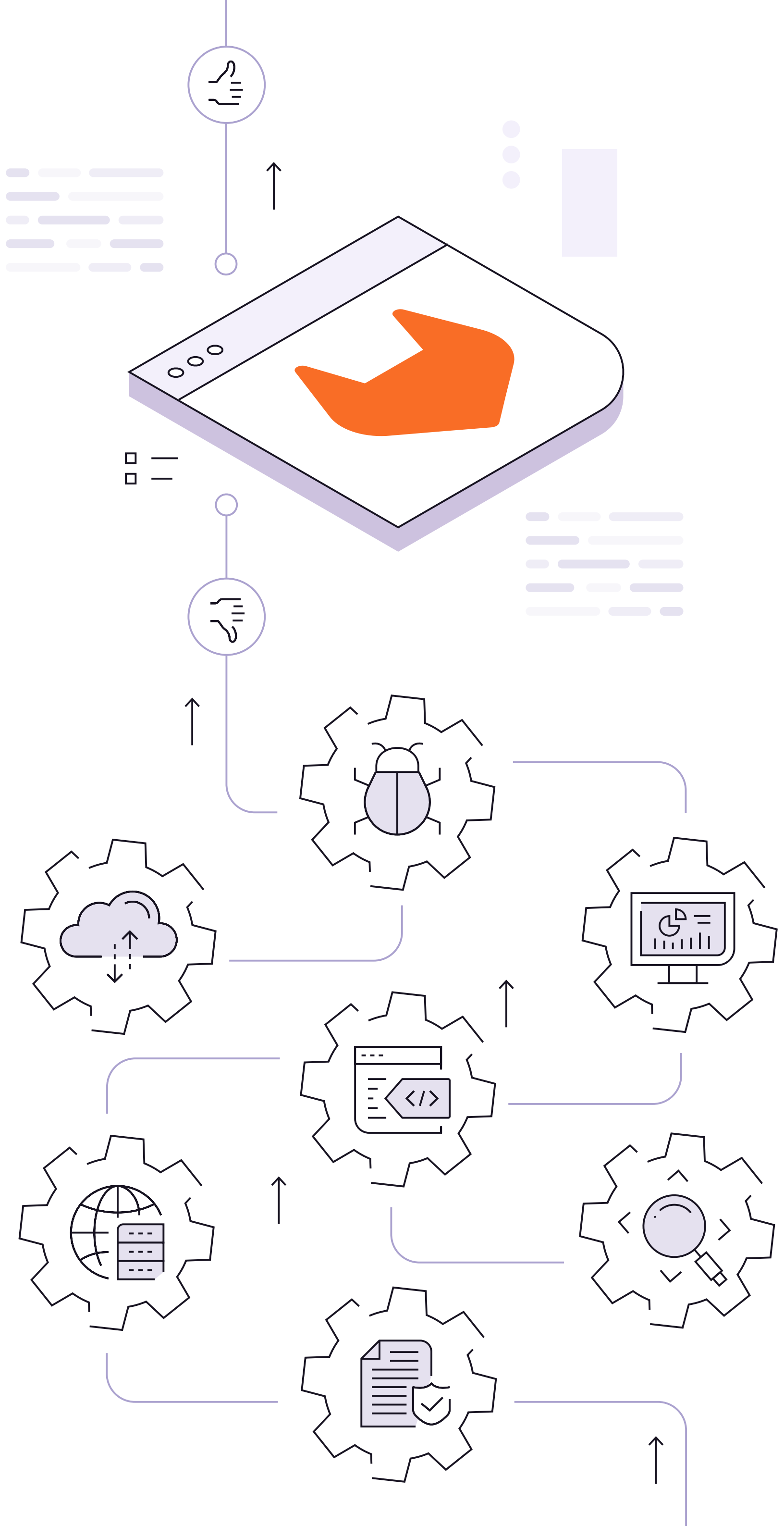


Table of contents

Introduction

The natural evolution of DevOps has organizations moving from a complicated and expensive do-it-yourself (DIY) toolchain to adopting an end-to-end DevOps platform. After all, Franken-system toolchains are stitched together in an attempt to aid software creation, but they actually end up costing companies time, money, and productivity.

Businesses are increasingly **turning to DevOps** to make the development and deployment of business-critical software more efficient and reliable. The problem is that many organizations have been piecing various DevOps tools together — one tool, for instance, to manage containers, one to automate deployment, another for code review. The list of different tools can be lengthy, with companies racking up five, 10, or even more in their toolchain. These stitched-together tools create a clumsy, snarled, and **time-consuming DIY toolchain** that can waste far more time and money than it was intended to save.

DevOps is the answer — but cobbling together a toolchain is not. Adopting a single end-to-end DevOps platform instead can help eliminate duplicate tasks, reduce costs, and drive critical advantages for both DevOps teams and the overall business.

In a **2022 Forrester Consulting Total Economic Impact™ (TEI) study** commissioned by GitLab, Forrester interviewed six GitLab customers and aggregated and combined their results into a composite organization with 5,000 initial users and an annual revenue of $60 billion. The study showed a three-year, 427% return on investment after adopting GitLab's single application. The study also noted that their number of annual releases for revenue-generating applications jumped by 12 times, with an 87% improvement in development and delivery efficiency times.

Having a comprehensive platform that boosts productivity, and at the same time can save money, is a superpower in economically challenging times because it can help deliver value to customers more quickly.

"We deliver value when we put software in front of our customers," says Emanuele Blanco, CTO of **Moneyfarm**, an online wealth management company with offices in the United Kingdom and Italy. "Having the infrastructure and a tool that (operates) seamlessly means developers can just focus on building features and making code that works. We have a tool that supports that in production (now) and it made a difference."

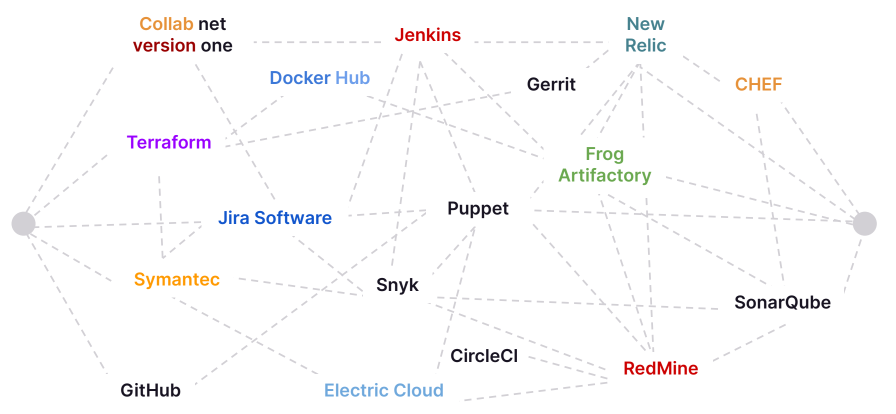
This guide will help you understand the complications and expenses that come with a DIY toolchain. It also will look at the benefits of migrating to GitLab's single platform, including improvements in **security**, **collaboration and communication**, efficiency, and project visibility. We'll also focus on how GitLab's platform can benefit not just software, but the overall business as well.

Let's dive in.

Problems with a DIY toolchain

It’s true that many organizations have begun to use DevOps tools and that’s a good start. But it’s only a step in the DevOps evolution because creating a clumsy, jerry-rigged toolchain out of disparate tools can create expensive problems that slow down development and deployment. And that works against business needs.

The problem: The “Toolchain Tax” adds cost, time, and risk



Problem:
A complex toolchain can cause bottlenecks, increase expenses and slow development and deployment.

Solution:
A DevOps platform can create efficiencies, reduce hands-on tasks, increase security, and reduce costs and aggravations.

The base of the problem is that these DIY toolchains are built with parts that were never designed to work together, which can lead to the creation of silos, a lack of visibility, more onboarding time, and added security challenges. It also can create a chaotic environment that leads to bottlenecks and requires constant management, tweaking, updating, and switching between interfaces — instead of enabling DevOps teams to simply focus on delivering code that drives the organization’s bottom line. That was clear in **GitLab's 2023 Global DevSecOps Survey**, which shows that toolchain sprawl weighs especially heavily on security professionals. For instance, 57% of security respondents said they use six or more tools, and they’re using more tools than in previous years

“With a toolchain, you simply don’t have everything in one place. With multiple tools, you’re managing versions and updates. You’re also paying for multiple tools and you don’t have one place to go for support. And you’re paying multiple companies for that support and training when you’re doing updates and onboarding. Just adding people to your team suddenly costs more and takes more time.”

Fatima Sarah Khalid
Developer Evangelist at GitLab

Getting rid of those disparate tools can free up those resources and all of that money. And that can lead to a return on investment, or ROI.

Benefits of moving from DIY to GitLab's platform

"GitLab has provided all the tools on one platform to maintain and deploy code, while aligning with our ISO quality management system," said Robert Brown, lead software engineer at Acertara Labs, a U.S. medical equipment manufacturing company.

Some DevOps vendors offer a kit of tools that teams still have to assemble themselves. GitLab covers all stages of the DevOps cycle because it is a complete platform, delivered as a single application that does everything from project planning and source code management to CI/CD, monitoring, and security. All the features seamlessly work together. One interface. One license. One source of truth. No stitching needed.

GitLab helps companies eliminate potentially costly integration work that comes with using point solutions and helps them create and release software faster while strengthening security and compliance. This combination shortens cycle times and increases productivity, enabling developers to build software with velocity, trust, and visibility.

And that creates value for customers.

“GitLab was the first to have that concept of the single application and that’s big,” says Brendan O’Leary, Staff Developer Evangelist at GitLab. “There are companies that have built platforms that are not fully integrated. They’re still stitched together pieces and when there’s stitching, there are breaking points. There’s no stitching with GitLab. It’s one piece of fabric. It’s the only platform with a single data store, one overview, and a single source of truth.”

There are many ways companies can realize the promise of DevOps by moving to a platform approach.

Building more secure and compliant software

In today's environment, security is a business imperative. And with GitLab, security is integrated throughout its platform, and not just bolted on as an afterthought.

With capabilities like dynamic and static testing, vulnerability management, and dependency and container scanning, DevOps teams can find vulnerabilities earlier in the process when they often can be more easily fixed. By shifting security left this way, teams can perform threat and vulnerability analysis as developers create the code, not after it's been deployed. That can create more secure software.

There also is no need to trade speed for security with GitLab. By solving a fragmented security experience, development is faster and more efficient. And GitLab's comprehensive security capabilities enable customers to consolidate their spend since they don't have to integrate with other security vendors.

Another security benefit of using a single platform is that when a team member leaves the company, their access only has to be removed from the system once, instead of from each and every tool where they had access. If someone forgets to delete a former employee's access from just one tool, there's the potential for a security breach. That risk is reduced with a single application.

"With a toolchain, companies have multiple points of entry so if one part is compromised, it can compromise everything else," says O'Leary. "With one platform, there's so much less complexity because there aren't multiple points of failure."

A single application also allows teams to verify the compliance of their code without leaving their workflow. In GitLab, compliance confirmation lives within the platform and is automated. This can remove the need for compliance managers to require developers to context switch among different point solutions, which can lead to the loss of productivity and efficiency. And GitLab also can show change traceability. For instance, a change in production code can be traced back to the requirement, as well as who made the change, and who approved it. This type of traceability can be difficult to achieve in a toolchain.

"Companies have a lot to navigate with regulatory issues or compliance requirements," says Khalid. "With a toolchain, they have to set up policy rules with every tool in the chain. If they have to remove a piece of data, they have to ensure that data comes off every single tool. With GitLab's platform, there's one policy management piece so it handles changes, like user roles and permissions, across the entire platform."

Increasing collaboration across the company

At its core, a DevOps platform is all about empowering teamwork. After all, developing and deploying software is a team sport.

By **breaking down silos** and increasing visibility across the entire software lifecycle, teammates can better share responsibility, **communicate in one place**, and work together to move software projects forward. And this goes beyond DevOps teams. People from around the company — whether they're in finance, marketing, sales, or the C-suite — also are able to collaborate on everything from planning to customer feedback. This can create more, and more diverse, project input, potentially leading to the creation of better, more secure, and more well-rounded products.

A DevOps platform **bridges collaboration gaps** by offering a seamless, end-to-end communication pipeline, fostering conversations instead of just pushing information in one direction.

When teams can work together to build, test, and deploy all in one place, they can solve problems and share knowledge, replacing traditional silos with a community work area.

According to GitLab's 2023 Global DevSecOps Survey, 39% of respondents said communication and collaboration skills are important for professionals in their industry — beating out subject-matter expertise (32%) and programming (32%)

"With a platform, someone working on a project's design and someone working on code can have conversations in one place so things aren't getting missed, balls aren't being dropped," says Khalid. "In GitLab, everyone is working in the same place and seeing the same things. Developers can collaborate with each other and branch off to create other features. No one is siloed."

Boosting efficiency

One of the benefits of migrating to a DevOps platform is **increasing efficiency**, which enables organizations to more quickly deliver high-quality software that answers business needs. And with increased efficiency, DevOps teams may have more time and attention to focus on bigger, and potentially more innovative, projects. It also means businesses can respond more nimbly to customer requests and have a better chance of staying ahead of competitors.

More efficient DevOps directly feeds the business

There are many parts of GitLab's end-to-end DevOps platform that can increase efficiency. How many of these would help your organization?

1. Running GitLab means that there is only one application to install, maintain, scale, back up, network, and secure. That allows teams to focus on delivering software instead of navigating a potentially chaotic development environment.
2. With one data model and a single interface, there can be a more efficient and engaging user experience.
3. Teams will see a **reduction in functional silos** that can increase teamwork, innovation, and productivity.
4. Automation minimizes the need for a lot of extra hands-on and time-consuming work, like backup, installation, and maintenance. It also can reduce the potential for human error and provide consistency. With GitLab, automation is a system feature and not something that has to be added in.
5. GitLab uses a single data store so users have easy access to information about the entire software lifecycle, instead of just parts of it.
6. Using a single application means teams don't have to integrate multiple products. This saves time, reduces risk, increases reliability, and lowers the bill of external integrators.
7. It's no longer necessary to ask for access to each separate tool; everyone is able to make use of all capabilities.
8. With single authentication, people only have to login to one application. They only have to set up one account.

9 A DevOps platform fosters collaboration and communication, which can increase the sharing of best practices and data across teams and projects, saving time and money.

10 By supporting continuous documentation, a DevOps platform can unify best practices between projects and DevOps teams. There's a direct correlation between creating clear, comprehensive, searchable, up-to-date, and well-organized documentation and a DevOps team's efficiency and success.

11 By using a platform's proactive monitoring capabilities for everything from planning to development, testing, and operations, teams are given an overarching view of the process so they can respond in a more timely manner to any problem that arises along the lifecycle.

12 New employees are no longer required to learn multiple tools during onboarding.

13 User training becomes less complex.

14 With a single vendor, companies only have to work with one point of contact, instead of multiple vendors pointing to each other.

Adopting a full DevOps platform is all about empowering teams to securely and efficiently turn a vision into software, meet customer needs, and reduce the time between designing new, higher-quality features and rolling them out into production.

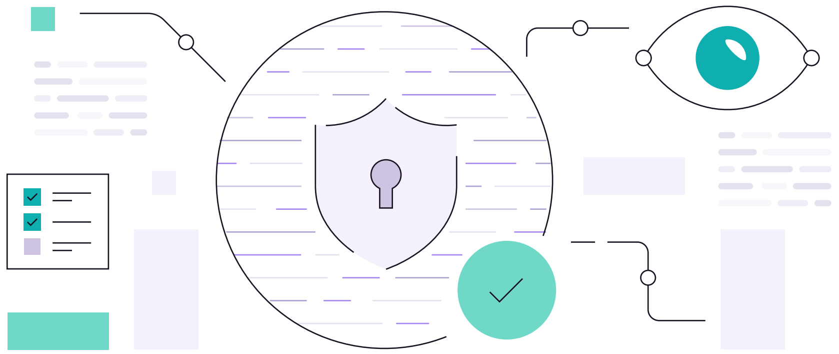
The GitLab platform enables teams to more efficiently make iterative deployments. And all of that means DevOps teams can deliver products that can drive the company's bottom line.

Supporting the overall business

Many companies depend on software. That means enabling an organization to collaboratively create, deliver, and manage code quickly and continuously is directly correlated to business value. What GitLab delivers is a faster cycle time, which has the potential to improve revenue.

Simply put, organizations may be able to generate more revenue by releasing customer-facing applications faster, more efficiently, and more securely.

Of course, having a single platform means companies aren't paying licensing fees for multiple tools. And costs quickly add up as teams add to a toolchain. To calculate what a company could save by replacing a toolchain, use this ROI calculator. *Please note that ROI may vary depending on many factors, and the ROI calculator does not reflect actual results as results may vary.*



"Using GitLab consolidates your software spend," says O'Leary. "That alone is a massive add-on for the business. But building software more efficiently, reliably, and quickly also will get a company new customers, keep the old customers, retain talent, and build its brand. Those are big business results."

In addition to getting rid of multiple licensing fees, moving to a single platform can have other more-hidden cost savings. For instance, not having to integrate and support multiple tools means it's likely that:

1. Organizations aren't paying to onboard new team members on many different tools
2. Engineering time and costs aren't wasted building and maintaining integrations for duplicative tools
3. Companies don't need to hire instructors or consultants from multiple vendors to train new hires or to train entire teams on software updates
4. IT has a single vendor point of contact and won't have to waste time and money figuring out the right vendor to reach out to

Getting the migration started

Migrating from a DIY toolchain to GitLab's end-to-end platform can seem like a big job to take on. There are a lot of moving parts to any adoption, but there are a series of steps that can ease the transition. Here are some non-technical basics to get organizations started, followed by some helpful technical how-tos.

Get your teams on board

One of the first tasks in beginning any migration is for IT leaders to evaluate their teams, as a whole, as well as the individuals on them. Whether they work on security, operations, quality assurance, coding, or product management, they'll take part in the migration and will have to change how they work in a fundamental way. Don't just spring this big move on them. Involve them in the planning.

Part of doing that is figuring out who the innovators are — the people who are going to be interested in and excited about this change. It's key to bring them on board first and enable them to help plan the migration and act as project champions with the rest of their teams. This will help people see the migration as a positive and powerful opportunity worth their time and effort.

Plan what to migrate first

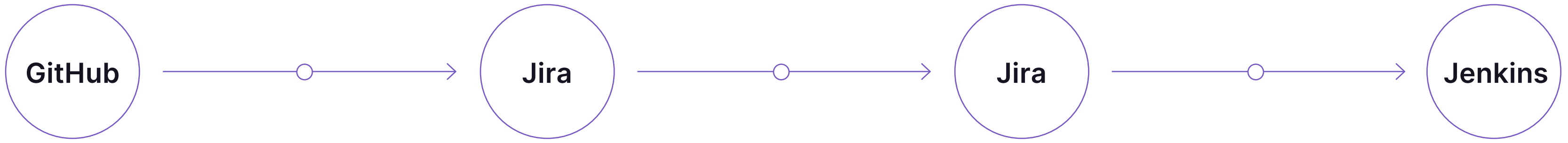
Early on, the company's migration team needs to outline what projects and teams to move first. Don't overwhelm people by starting out with something complex and business critical. It's easier to test the water, get some experience, and learn migration lessons by first moving smaller and more agile teams. Also look to first migrate discrete, greenfield projects, which lack constraints and rules imposed by previous builds. Run short test cycles. This will pave the road for larger and more involved projects, and the larger teams that handle them.

Carve out the needed time

A key time consideration is to make sure a migration isn't scheduled during or right before a busy time for the business or when new software is scheduled to be rolled out. If you don't, it could add unnecessary stress to your team and compromise the rollout.

IT leaders also need to ensure that everyone has the time they need to make the migration. Do a pre-migration audit to assess what is being worked on, what will be migrated, and what needs to get across the finish line first. Also consider who will be dedicated to the migration and who will be handling normal day-to-day development work. Carve out time for the people who will handle the migration to focus on just that.

Technical tips for migrations



Migrating from GitHub

You can import your GitHub repositories, including commit histories and branches, pull requests, issues, labels, comments, wiki pages, and milestones, into GitLab.

Step-by-step guide for importing from GitHub:

1. Create a new project in GitLab, select import project and then GitHub
2. Enter GitHub credentials
3. Select specific repositories to import and their destination in GitLab
4. Validate import in GitLab
5. Go to this site for more information

Integrating with Jira

To shrink your toolchain by creating a sync between GitLab and Jira, here are some steps to take.

Step-by-step guide for integration:

1. Download the GitLab app for Jira on the Atlassian Marketplace
2. Set up the Jira integration in GitLab
3. Use the name of the issue in Jira while committing changes so they are automatically reflected in Jira
4. Validate integration in Jira
5. Go to this site for more information

Migrating from Jira

You can import all your issues from Jira and work exclusively in GitLab.

Step-by-step guide:

1. Download the GitLab app for Jira on the Atlassian Marketplace
2. Set up the Jira integration in GitLab
3. Select the Jira project you want to import issues from
4. Validate import in GitLab
5. Go to this site for more information

Migrating from Jenkins

Many users migrate from Jenkins to GitLab to simplify their CI/CD workflow.

Step-by-step guide:

1. Use Auto DevOps as a starting point and customize to match your Jenkins pipeline
2. Add runners to your GitLab instance
3. Add review apps to test on the go
4. Migrate the deployment jobs and create environments
5. Go to this site for more information

What benefits look like in the real world

For some insights into how one company benefited from replacing a DevOps toolchain with GitLab's single platform, check out how **Airbus Intelligence**, an international provider of commercial satellite imagery and premium geospatial data services, widely adopted GitLab.

Airbus Intelligence, an arm of Airbus Defense and Space with 130,000 employees, needed to rid itself of some of the challenges that come with distributed teams and disconnected toolchains that were causing workflow inefficiency and slow production. The company also needed a way to help their teams, which are located around the world, collaborate and work more efficiently together. By adopting GitLab's DevOps Platform, teams could take advantage of version control and project management capabilities, as well as best-in-class continuous integration — all in a single application. Visibility, communication, and collaboration also improved. Deployments became less stressful. And now developers have more time to create better features in their software.

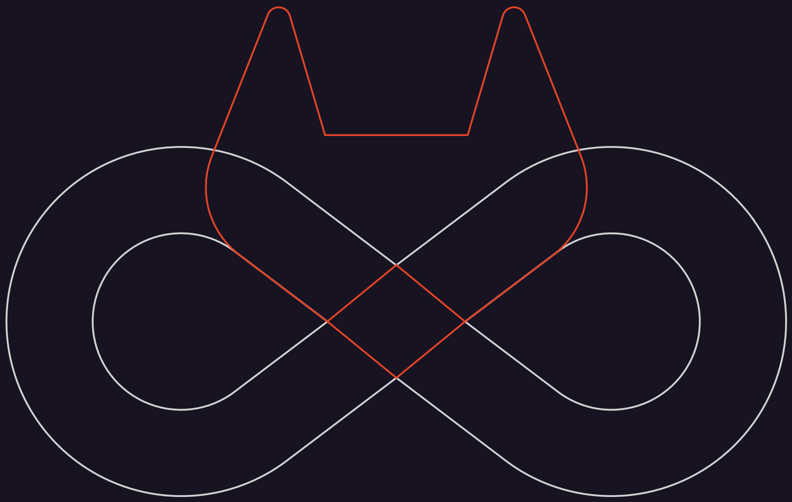
Because they're using GitLab, Airbus Intelligence also has seen a boost in recruiting talented developers.

Trading up from DIY to GitLab

In today's highly competitive landscape, organizations are under more pressure than ever to deliver software faster and more securely. They need a more mature, all-encompassing platform to improve their time to market.

GitLab answers that need with a single DevOps platform. Organizations can do away with their DIY DevOps toolchains and bring teams together to collaboratively plan, develop, ship, and monitor cutting-edge software all from one application. That increases speed, compliance, security, and business agility

Go **here** to learn more about GitLab's single, end-to-end platform.



About GitLab

GitLab is the most comprehensive, AI-powered DevSecOps Platform for software innovation. GitLab provides one interface, one data store, one permissions model, one value stream, one set of reports, one spot to secure your code, one location to deploy to any cloud, and one place for everyone to contribute. The platform is the only true cloud-agnostic end-to-end DevSecOps platform that brings together all DevSecOps capabilities in one place.

With GitLab, organizations can create, deliver, and manage code quickly and continuously to translate business vision into reality. GitLab empowers customers and users to innovate faster, scale more easily, and serve and retain customers more effectively. Built on open source, GitLab works alongside its growing community, which is composed of thousands of developers and millions of users, to continuously deliver new innovations.

GitLab

GitLab