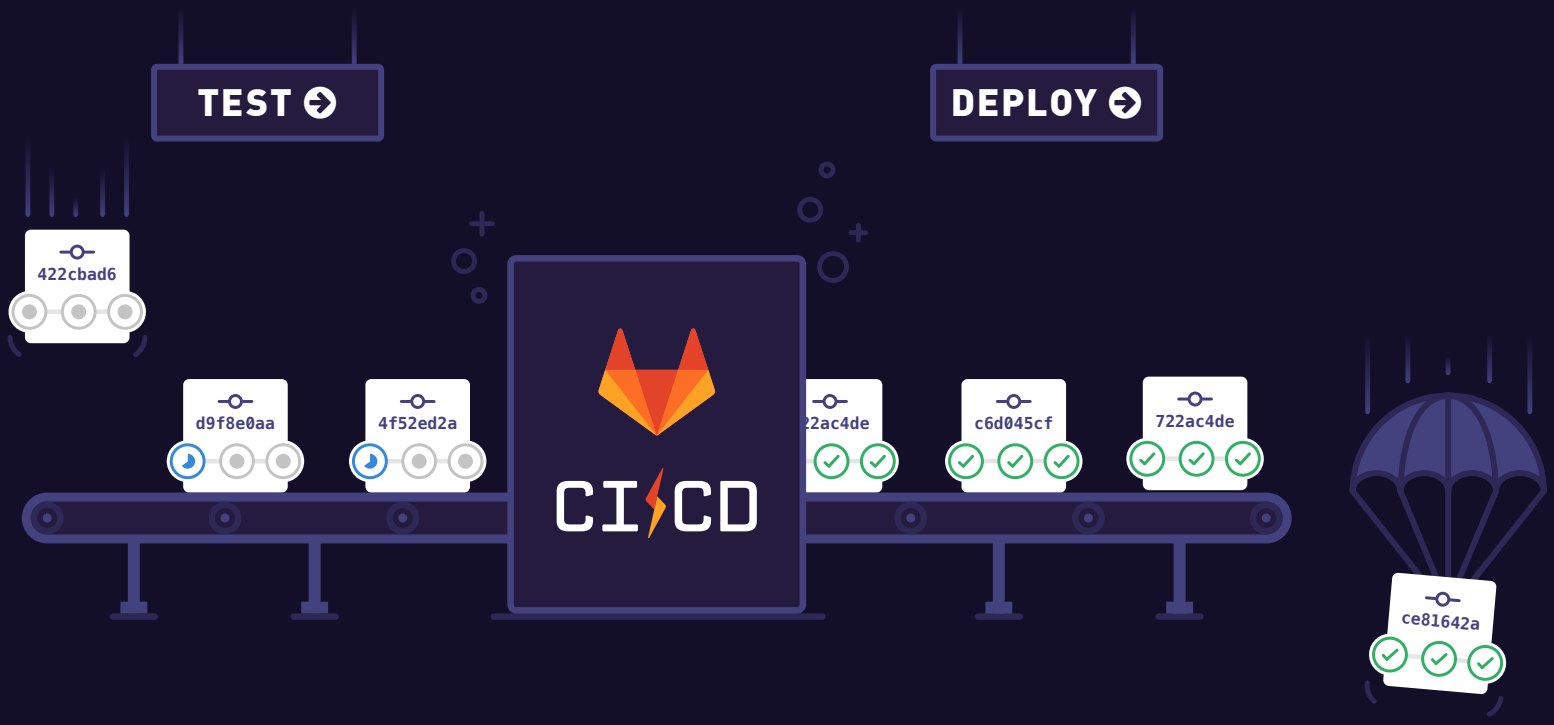
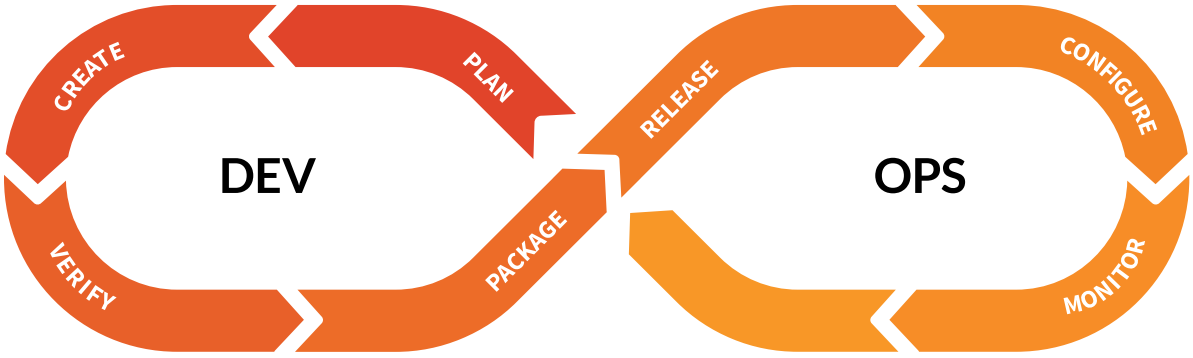
The benefits of single application CI/CD



GitLab

Continuous Integration and Delivery (CI/CD) changed the way we build, test, and deploy software. CI/CD tools automate these processes, reducing error rates and optimizing workflows. Code moves through each development stage while automated testing throughout the process ensures errors are caught and rolled back before they ever reach production.

The use of CI/CD tools will continue to rise and improve the way software is developed. Deployments no longer need to be an annual, quarterly, or even a monthly event. With CI/CD, DevOps teams can deploy several times a day.



10 Advantages of CI/CD

Now that we've outlined what CI/CD is, let's outline its advantages.

1. Fewer handoffs.

The more handoffs in your development pipeline, the more points of failure and the more complexity you add to the process.

2. Increased development speed.

All stages of development are faster with CI/CD. Faster iterations throughout the process make every team more efficient and developers can move onto other projects confidently.

3. More deployments.

Releases that happened every couple of weeks or more, can now be pushed up to six or more times a day.

4. Faster testing.

One of the more time-consuming parts of the development workflow is removed and developers can work on other high-value projects. Automated testing lets teams get feedback sooner and fail early rather than catching bugs in production – or worse – in the final release.

5. Fewer bugs.

With automated testing throughout the development process, bugs are caught as they arise and rolled back without ever making it into master. This ensures better code quality overall and that every release works just as intended.

6. Improved compliance.

Compliance tasks can be incorporated into the development lifecycle, reducing the risk of releasing non-compliant applications. Automated compliance makes it easier to complete audits and can prevent costly mistakes (especially in highly-regulated industries).

7. More time to innovate.

Less time spent on integration maintenance and undifferentiated IT spend means resources can be directed elsewhere.

8. Happier developers.

Developers can work confidently and fix things quickly instead of waiting for weeks to learn there was a bug.

9. Reduced overhead spending.

Organizations can only have so many developers at one time. Development hours are many times billable hours, and manually deploying or testing code can bust IT budgets. An automated workflow reduces manual tasks and makes budgets more efficient.

10. Consistent processes.

Having more automation in the development workflow means nobody ever forgets a step in the process. Builds are more consistent, it's easier to train new developers, and organizations have more control over how things are built and when they get released.

Are you doing optimized CI/CD?

To truly measure the speed of productivity, the entire SDLC needs to be evaluated, not just bits and pieces. Tech executive Gary Gruver in his book, Starting and Scaling DevOps in the Enterprise, recommends not just starting at the developer but to the business idea that influenced a new feature in the first place. Analyze the SDLC from the perspective of one developer: From idea, to development, to the test environment, and into production. What are all of the individual steps they must take in order to execute an idea? This process defines how value flows through organizations and identifies bottlenecks in the system.

Seamless continuous integration and delivery should have everything needed to get an app ready for deployment. Unfortunately, many organizations that use CI/CD are stuck in a less than optimal workflow caused by the tools they use – especially if something goes wrong.

In the event a pipeline fails, a developer may lack the visibility to fix the problem because they don't have access to a certain tool. Sometimes this requires two teams to coordinate with each other – one has the access to fix the problem while the other has access to see the problem. Even if both teams have access, they're still collaborating in different tools. For example, code review mechanisms won't exist in Jenkins CI, so teams will need to go back and forth between their SCM and CI tool to make the right documentation.

A complex toolchain relies on integrations and regular maintenance in order to work. Service instability and brittle configurations can result in interruptions that affect the entire lifecycle, resulting in long waits for builds to start and complete. It's not uncommon for well-intentioned admins to bring systems to a halt due to "simple" plug-in upgrades or installs. In a blog post, Jenkins creator Kohsuke Kawaguchi acknowledged some shortcomings of the platform, including its unreliability and reputation as a Frankenstein platform due to the sheer number of plug-ins it uses. This kind of transparency is refreshing and shows that Jenkins is actively working on the problem, but it still highlights a problem that many DevOps teams know all too well:

All of these integrations work great… until they don't.

Single application CI/CD

We believe a single application that offers visibility across the entire SDLC is the best way to ensure that every development stage is included and optimized. When everything is under one roof, it's easy to pinpoint workflow bottlenecks and evaluate the impact each element has on deployment speed.

Each step under a single DevOps platform offers a piece of information that makes for an informed decision about whether the code is ready to merge. While all of those stages are possible to perform without an all-in-one CI/CD solution, a comprehensive tool creates a single source of truth where each step can be monitored faster and with less potential for missed errors. Forrester research also found that IT professionals believe an out-of-the-box toolchain solution improves overall security.

Benefits of single application CI/CD:

» Visibility across the entire development lifecycle
» Single source of truth across all development stages
» No logging into multiple applications
» Simpler navigation on one interface
» Only one application to install, maintain, scale, backup, network, and secure
» Easier authorization management
» Lower operating expense

Continuous delivery goes hand-in-hand with CI, and if all code has been tested and is ready to go into production, that process should also be seamless. An efficient CI/CD strategy employs automation from start to finish, not just spots in-between.

Is all CI/CD created equal?

Teams have no shortage of CI/CD tools to choose from. Cost and features play a big role in evaluating the value one solution provides over another. There are also other factors in play, such as how a CI/CD tool might fit into your current needs vs. expected needs. A side-by-side feature comparison is one way that organizations look at the value of certain platforms.

Comparing GitLab CI/CD to Jenkins CI

Jenkins is one of the most popular build automation and CI/CD developer tools. It derives its flexibility by incorporating capabilities from hundreds of available plugins, which enable it to support building, deploying and automating projects.

Built-in CI/CD

CORE STARTER PREMIUM ULTIMATE
FREE BRONZE SILVER GOLD

GitLab collects and displays performance metrics for deployed apps, leveraging Prometheus. Developers can determine the impact of a merge and keep an eye on their production systems, without leaving GitLab.

Both Jenkins and GitLab offer CI/CD without a need to install it separately.

CI/CD Horizontal Autoscaling

CORE STARTER PREMIUM ULTIMATE
FREE BRONZE SILVER GOLD

GitLab CI/CD cloud native architecture can easily scale horizontally by adding new nodes if the workload increases. GitLab Runners can automatically spin up and down new containers to ensure pipelines are processed immediately and minimize costs.

Only GitLab offers Auto-scaling runners, which minimizes cloud costs and keeps pipelines running efficiently.

Step folding for CI/CD logs

CORE STARTER PREMIUM ULTIMATE
FREE BRONZE SILVER GOLD

Collapse the job log output for each command.

Only Jenkins offers collapsable job logs currently. GitLab plans on adding this functionality in a future release.

Transparency is one of our core values, which is why we list GitLab comparisons to other DevOps tools on our website. While Jenkins does offer CI/CD, it falls short in other built-in functionality, and the reliance on plug-ins means that Jenkins can also have a higher cost of ownership over time.
See the full comparison

Start projects with CI/CD out of the box

GitLab CI/CD is already built into the same application that contains source code management, planning, monitoring, etc. As a single application for the entire DevOps lifecycle, everything is in one conversation and visible across teams. With GitLab's out-of-the-box CI/CD, our goal is for DevOps teams to spend less time maintaining and more time creating.

Teams that made the switch to GitLab CI/CD

Ticketmaster

GitLab CI/CD supports ramp up to weekly mobile releases

> **"Since February we have had weekly releases of our mobile apps, and GitLab CI has been a huge part of our success over the past few releases. With the benefit of faster cycle time, and faster releases, we have seen other benefits. Since each release has a smaller change set, our crash-free rates and store ratings have improved. We have less time waiting for build and spend more time improving the quality of our products. Our fans are getting features into their hands more quickly and benefit from a higher-quality and a consistently improving product. The CI analytics available on GitLab are an additional scoreboard for our team to optimize and improve into the future."**
>
> *— JEFF KELSEY, LEAD ENGINEER - ANDROID DEVELOPMENT TEAM*

Read their story

Cloud Native Computing Foundation

Unified CI/CD system eliminates the complexity of managing multiple projects across many cloud providers

> **"Empowering your community to focus on their own project but also provide them the ability to combine it with other projects is a difficult problem. Having a group like the CNCF CI Working Group to provide that guidance, and have that conversation so each team is aware of their place and context within the larger group is something that we're still working out as we talk to the different groups within the CNCF," said McClimans. "Having GitLab**

> **available to us so quickly after we had those conversations allowed us to put something together that's meaningful and shows results. It's something that I've not seen happen as quickly in my life as a DevOps and CI guy."**
>
> *— CHRIS MCCLIMANS, CROSS-CLOUD CI PROJECT CO-FOUNDER*

Read their story

By automating processes and producing better quality code, CI/CD tools save organizations both time and money. The ability to self-test code, automate builds, catch bugs early, and monitor progress leads to optimal time efficiency and faster release cycles. The competitive advantage of CI/CD also extends beyond applications by creating a more cohesive DevOps culture that facilitates handoffs between teams automatically. Better processes, less manual tasks, and increased visibility also improves retention rates and helps organizations attract more talent. Single application CI/CD allows teams to expand on these benefits even further for a better and more seamless experience.

Start your free GitLab trial

About GitLab

GitLab is a complete DevOps platform, delivered as a single application. Only GitLab enables Concurrent DevOps, unlocking organizations from the constraints of today's toolchain. GitLab provides unmatched visibility, radical new levels of efficiency and comprehensive governance to significantly compress the time between planning a change and monitoring its effect. This makes the software lifecycle 200% faster, radically improving the speed of business.

GitLab and Concurrent DevOps collapse cycle times by driving higher efficiency across all stages of the software development lifecycle. For the first time, Product, Development, QA, Security, and Operations teams can work concurrently in a single application. There's no need to integrate and synchronize tools, or waste time waiting for handoffs. Everyone contributes to a single conversation, instead of managing multiple threads across disparate tools. And only GitLab gives teams complete visibility across the lifecycle with a single, trusted source of data to simplify troubleshooting and drive accountability. All activity is governed by consistent controls, making security and compliance first-class citizens instead of an afterthought.

Built on Open Source, GitLab leverages the community contributions of thousands of developers and millions of users to continuously deliver new DevOps innovations. More than 100,000 organizations, including Ticketmaster, ING, NASDAQ, Alibaba, Sony, and Intel trust GitLab to deliver great software at new speeds.


GitLab