Mobile CI after Jenkins: 6 lessons from engineers who made the switch

bitrise

73.9% *(the number of CI practitioners that enjoy their jobs more or **much** more after switching from Jenkins to cloud CI)*

Engineers who are responsible for maintaining Jenkins are critical to their team's performance and often work years on building the specific knowledge required to support their organizations. However, more and more teams around the world are making the switch from self-hosted Jenkins to cloud CI to better equip themselves for the challenges of mobile development.

For this report, we gathered stories and experiences from engineers who used to be responsible for Jenkins, detailing how their roles changed when they moved to cloud CI. Do they enjoy their jobs more? Less? Do they feel more impactful? What does the rest of the team think? Read the insights based on a survey of hundreds of ex-Jenkins users and personal conversations with mobile teams from EPAM, Nextdoor, Pulselive, Delivery Hero, Signify, and many more.

"The tricky part is habit. Hitting the same thing over and over again will eventually become familiar. This can lead to unforeseen consequences – like using Jenkins when it is not really necessary. Do not be afraid to change things."

— Artur Dymorov,
Android Developer

"Leaving Jenkins behind doesn't mean that DevOps engineers' jobs will become obsolete, it just helps them get rid of tasks like manual tweaks and maintenance, allowing them to focus on more meaningful work."

— Moataz Nabil, Senior
Software Engineer, DeliveryHero

"Like lots of companies, we were still using the somewhat outdated Jenkins. Raise your hand if you think that Jenkins is a CI from a past era."

— Rémy Chantenay,
(then) Engineering Lead, Travelex

"Instead of spending hours troubleshooting CI/CD, our engineers can now focus on making our products better for our customers."

— Su Khai Koh,
Software Engineer, Nextdoor

1 "I'm no longer tied to physical build hardware"

Hosting on-premise CI means being responsible for a physical stack of machines. This is a huge burden — both for the business that needs to buy all that hardware and the engineer who's tasked with maintaining it. In interviews, Jenkins professionals recounted horror stories linked to physical hardware, such as office blackouts, or having to move the Macs with them whenever the team moved to a new location.

Infrastructure as code that's offered in cloud-based solutions allows engineers who previously spent most of their day in a closet full of Macs to focus on more meaningful tasks. Not depending on office hardware is a relief for the rest of the team, as well: they no longer have to send — and wait for — someone just to start the servers or fix connection issues. On top of that, since they are not tied to a specific location, it offers more flexibility and makes remote work easier.

Hatena

"One of our most dreaded tasks was having to restart the Mac, which caused a blackout in the entire office."

— Hiroki Kato,
Chief Engineer, Hatena

Pulselive

"Our local Jenkins setup was turning into a hindrance. We'd have it working relatively fine for a month or two, then something would change and it'd just stop working."

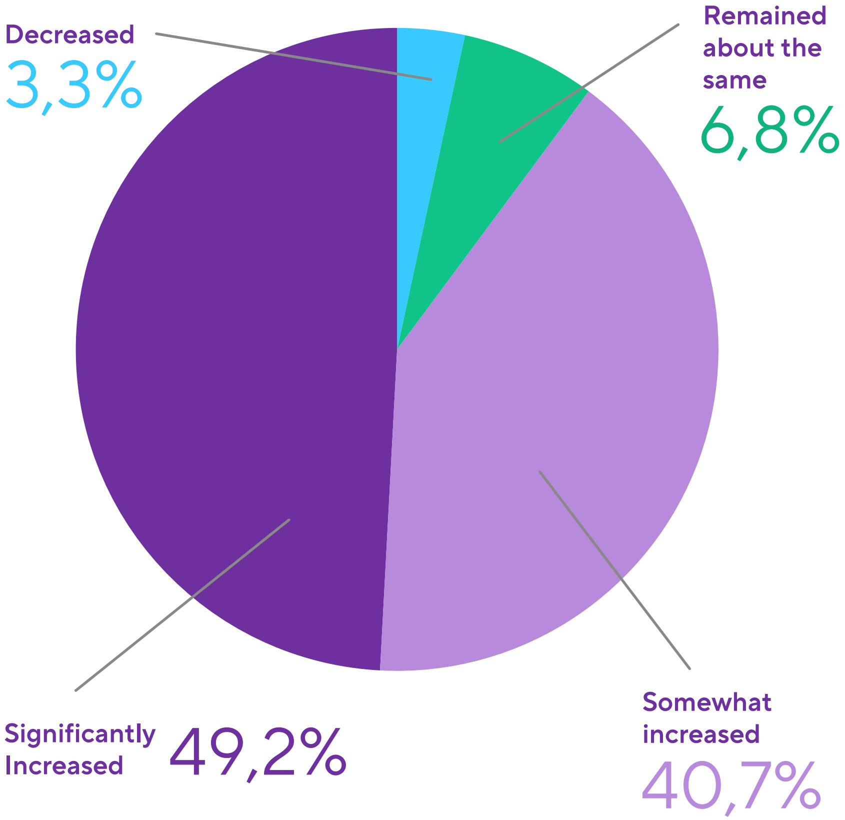
— Rasmus Larsson,
Product Director, Pulselive

nextdoor

"Because of the limited number of Macs we had, our queues grew larger and larger as we increased the number of features on our app and the engineers on our team. We had a hard time spinning up new machines."

— Su Khai Koh,
Software Engineer, Nextdoor

How has your team's performance changed since you switched to cloud-based CI?



"I no longer spend hours on firefighting and fixing errors" 2

Perhaps the biggest catch about using Jenkins is that someone has to maintain it. Time is a scarce commodity for mobile teams — lost connection, debugging, and manual maintenance can take valuable hours away from productivity. Even minor version updates or new plugins can break the system. In either of these cases, a dedicated engineer always needs to be involved to get everything up and running again.

After switching to a cloud-provider, engineers whose previous responsibility was mainly Jenkins setup and maintenance will be able to build infrastructure for scale and focus on other parts of the app development. They will also be able to break free from code signing procedures and other lengthy processes they used to have to deal with on a daily basis while the team used Jenkins.

COMPASS

"Our responsibilities have shifted from the team to the platform: we now have the entire process of building, testing, code signing, and shipping automated in Bitrise. As a result, we can focus on what's important."

— Russell Stephens,
Mobile Infrastructure Lead,
Compass

ibotta

"We started having concerns about a potential decrease in quality: in some cases, the Jenkins machine would begin failing and intervention was needed, occasionally taking several hours to debug. Not only did this take valuable time away from development, but manual maintenance wasn't something the team was comfortable dealing with in the long run."

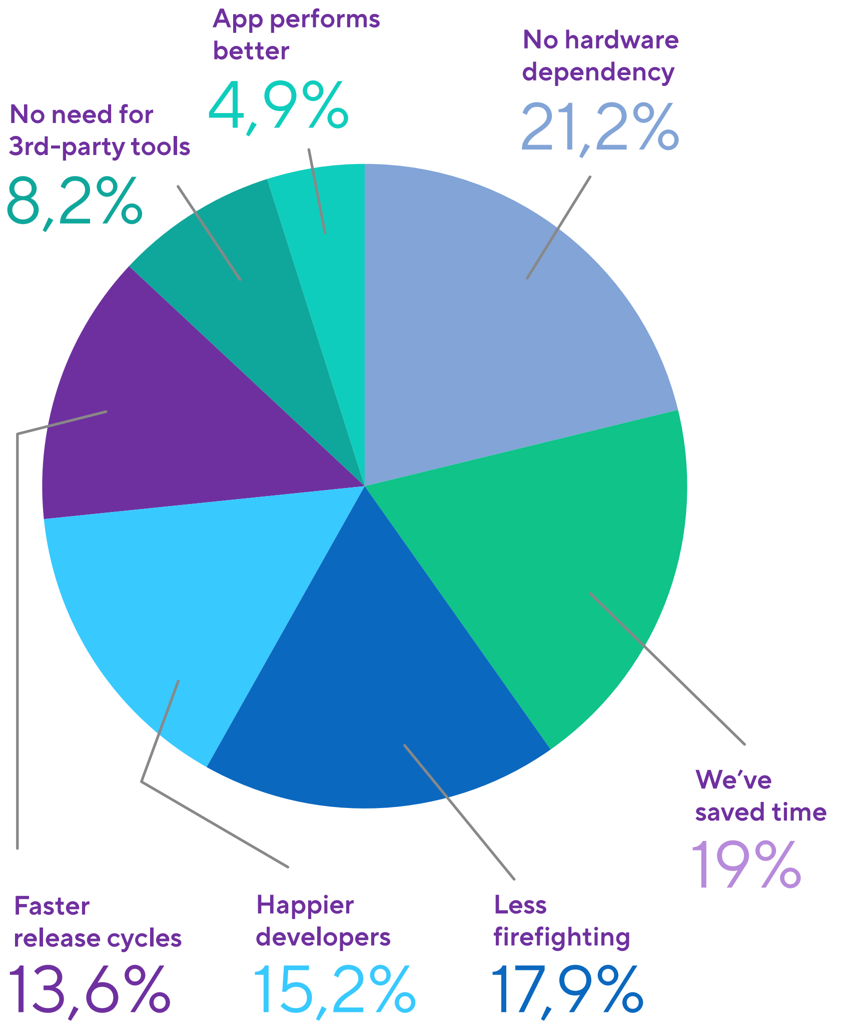
— Vince Delricco,
Android Engineer, Ibotta

nextdoor

"Keeping everything up to date on Jenkins was a huge overhead for us. Engineers spent hours fixing issues like network and hardware failures. Due to the lack of isolation between builds, the cache from an old build frequently made a new build fail. Small, unpredictable changes like network failures caused a lot of instability."

— Su Khai Koh,
Software Engineer, Nextdoor

What's the biggest benefit your team has experienced after migrating your app from Jenkins to a cloud-based solution?



3

“I don’t have to do manual updates anymore”

The shared library system in Jenkins means that users rely on admins to make changes. If developers cannot deploy because they have to wait for the operations team to download the latest Xcode or upgrade a plugin — which, by the way, might break everyone else’s pipelines — then the entire team is losing days’ worth of productivity.

In most cloud-based solutions staying up-to-date is quicker and easier, compared to Jenkins. The latest tech stacks can be automatically updated so there’s no need for manual downloads. When it comes to performance, teams always have the option to upgrade to machines with more capacity allowing them to release new features, and higher-quality apps, faster.

“We no longer have to worry about manual Xcode downloads, nor about office blackouts. Thanks to all of this, developers can focus on the code itself and create new features for users.”

**— Hiroki Kato,
Chief Engineer, Hatena**

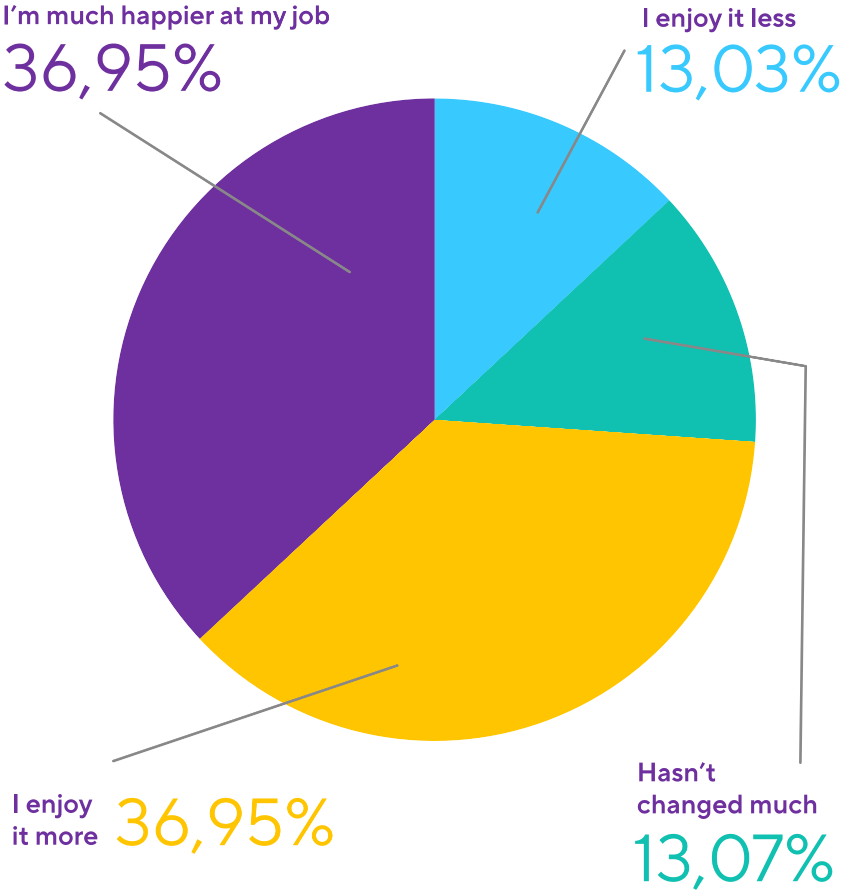
“The main downsides of Jenkins were that plugins often require updates with risk for stability, Xcode needed to be manually updated, and it wasn’t clear how to set it up without specific Jenkins knowledge. With Bitrise, we solved all of these issues.”

**— Antoine van der Lee,
iOS Engineer, WeTransfer**

“We were quickly convinced by Bitrise’s easy setup, CLI, open-source support, the ability to build definition as code, and the fact that it worked even without dedicated DevOps engineers”

**— Elvin Rakhmankulov,
Head Of Mobile Competency
Center, Epam Systems**

If you were responsible for maintaining Jenkins before switching to a cloud-based solution, how do you feel your job has changed since the switch?



“I no longer have to be the gatekeeper”

4

On Jenkins, access management is generally done by a single admin, which leads to tracking and accountability issues with the pushed code. This makes the overall release progress a rather difficult job for larger projects and causes a lot of trouble for the release manager. Even though there’s a charm to being the Jenkins savior who can unblock a team from doing their work, there’s a greater role to be played for ex-Jenkins practitioners when they transition to self-service, cloud-CI.

These tools make it much easier for all developers in the team to step in, contribute, and onboard new team members, as well. At the same time, they allow senior engineers who previously maintained Jenkins to make more meaningful contributions to the team’s performance, the end-product, and even help scaling the app.

Pulselive

“We now have the ability to set up a workflow that automatically builds different builds, distributes different versions, and updates these versions. Instead of having to spend hours waiting around for separate builds, with Bitrise, we can essentially just press a button and it builds all of them, and they are all ready to go.”

— Rasmus Larsson, Product Director, Pulselive

“The full control Bitrise offers over our Workflows, the simplicity of its interface combined with the auto-updating stacks, and quick build times make the platform even more user-friendly.”

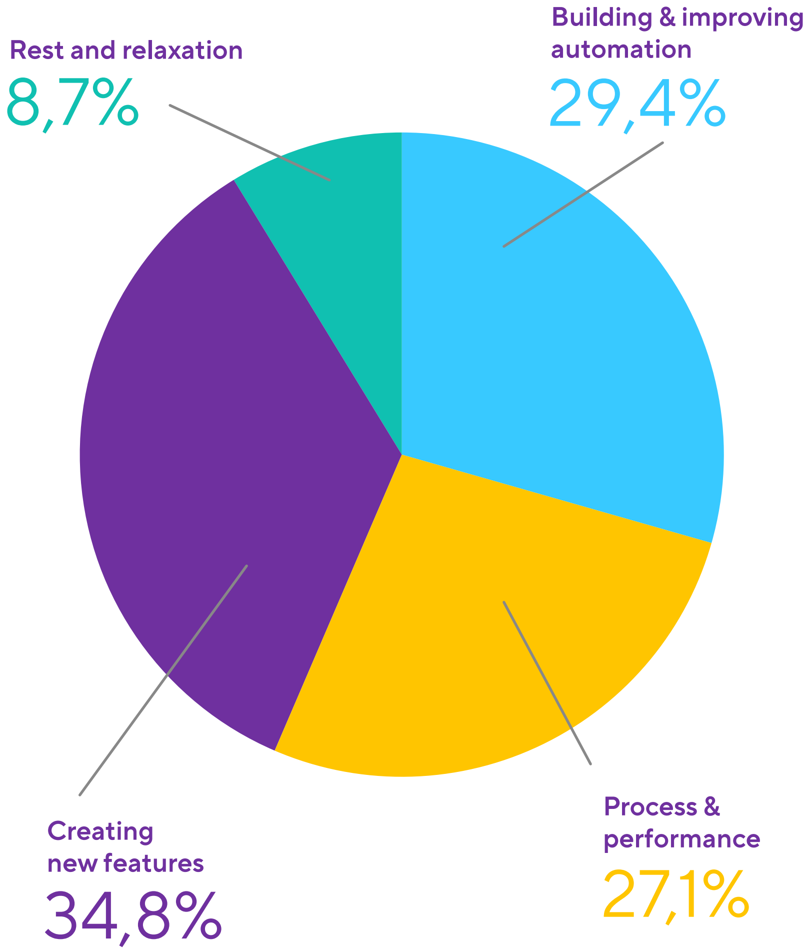
— Sage Young, Director Of Mobile Engineering, Fueled

wetransfer

“Not only can Bitrise Workflows be shared, which allows us to easily maintain all of our projects, they are also easy to understand for everyone in the team.”

— Antoine van der Lee, iOS Engineer, WeTransfer

On what other activities are you spending increased time now that you’re spending less time firefighting workflows?



5 "I'm contributing more to the organization's security"

Even though Jenkins wasn't designed with a 'security-first' mentality, most Jenkins maintainers care — and spend time on — security. Where its ecosystem of plugins holds an integration for almost every use case, its lack of formal vetting for most of these plugins is a nightmare for the security team. Integrations that are important for mobile development, such as device testing, iOS or Android simulators are also not built-in features of Jenkins — adding to the list of third-party tools that can make it vulnerable to attacks.

DevOps engineers, RTEs, and other CI maintainers are finding new ways to contribute to their team's security in cloud CI. Features like user access systems, enforced 2FA, SAML SSO, and VM destruction after a build all help facilitate the crucially important practice of DevSecOps take hold in mobile development teams.

signify

"Using Bitrise has even strengthened the security of our DevOps pipelines: we can easily connect to SAML SSO, so we don't have to worry about unauthorized access."

— Alexei Sintotski,
Lead Development Engineer,
Signify

nextdoor

"Setting up remote access to the build servers was a challenge for us. The workarounds we made to enable remote SSH were fragile and could have become security liabilities."

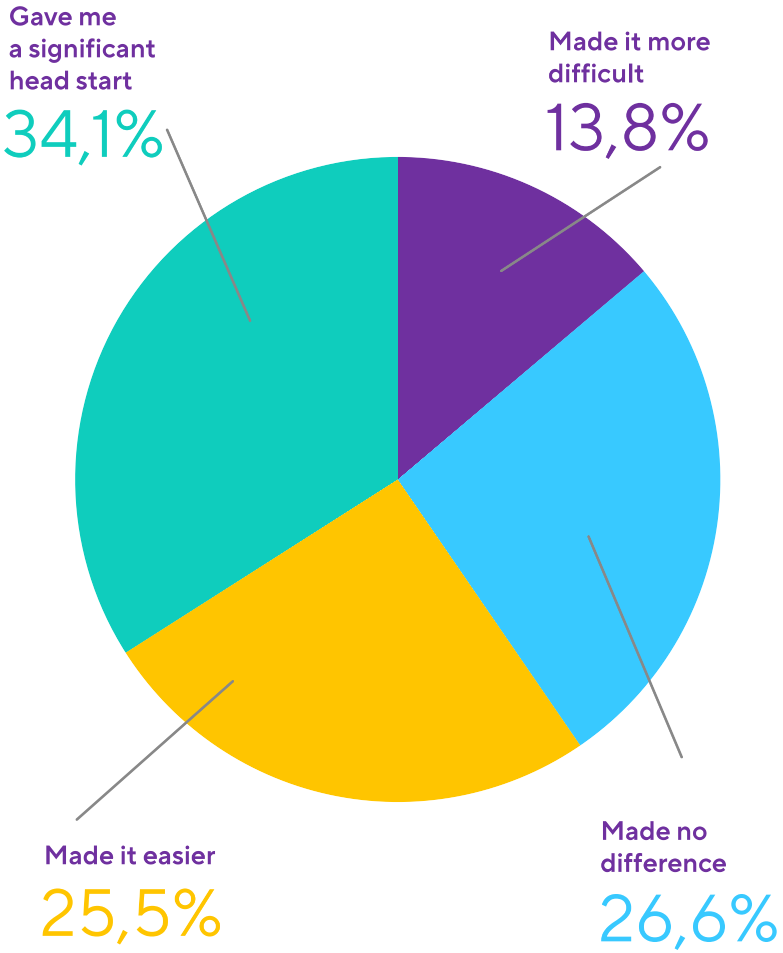
— Su Khai Koh,
Software Engineer, Nextdoor

Delivery Hero

"Managing Jenkins needs special attention: the security, roles, users, permissions are all the things you need to take care of. You also need an admin who can create the accesses."

— Moataz Nabil,
Senior Software Engineer,
Delivery Hero SE

Did you feel like your experience with Jenkins made it easier or more difficult to understand cloud CI?



“I don’t have to worry about unexpected costs anymore” 6

While the business often values Jenkins because they think it’s free, those working on their team’s Jenkins setups — and those responsible for the budget — know better. Common questions like “What will the server cost be for the second half of the year?“, or “How many man-hours should we calculate for maintenance?” are difficult — if not impossible — to answer.

With a cloud-based CI, you're not required to purchase hardware ahead of time, in anticipation of the team growing or ramping up their build activity. You can easily and regularly scale up, without being locked into whatever type of machine you bought years ago. Because cloud CI requires significantly less resources, the time and money needed for maintenance are easier to estimate. All in all, leaving Jenkins behind results in reduced cost, more predictability in resource requirement, and freed up capacity to do meaningful work.

Delivery Hero

“While Jenkins is free, the costs of hosting the server on which it will run cannot be easily calculated. It is not always possible to estimate the kind of load that the server needs to handle so the total costs remain unpredictable.”

— Moataz Nabil,
Senior Software Engineer,
Delivery Hero SE

“Do not forget that Jenkins needs to be run somewhere. Amazon AWS helps but still, someone has to maintain it. And there is no isolation out of the box. Need to cache something across jobs? Either install globally — and suffer consequences — or reinvent a Docker-driven wheel over and over again.”

— Artur Dryomov,
Android Developer

nextdoor

“Bitrise helped us reduce the cost of creating new instances by removing the maintenance burden of having our own infrastructure.”

— Su Khai Koh, Nextdoor

Conclusion

Life after Jenkins is actually pretty good, our research shows. The numbers show that those who used to be responsible for Jenkins are more productive and spend their time on more meaningful tasks after migrating to cloud CI. The office hardware pile is a thing of the past, firefighting is massively reduced and instead, they are able to focus on developing new features, building automation, and improving process and performance.

Turns out, in most cases, those years of Jenkins knowledge haven't gone to waste either, as the majority reports directly benefiting from their Jenkins expertise, during and after the migration. As for the rest of the team, the results suggest that the ability to step in, make meaningful contributions, and all the other benefits mentioned are making the developers happier, as well.

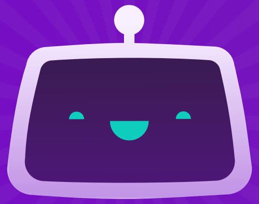
"We now deliver higher-quality apps, faster — without having to worry about the underlying infrastructure or needing DevOps engineers"

— Moataz Nabil, Senior Software Engineer, Delivery Hero SE

bitrise

How we can help

Mobile engineers behind some of the most popular apps rely on Bitrise to streamline their build-, test-, and deployment processes. By eliminating manual tasks like maintenance, fixing errors, and performance upgrades, we help customers iterate faster, release with confidence, and build apps that are used and loved by millions of users around the world.



Learn more about how you can start releasing with confidence at

www.bitrise.io

Trusted by the world's most sophisticated mobile teams

moz://a Expedia TONAL DeNA

ibotta TransferWise duolingo COMPASS

traveloka GET YOUR GUIDE PagerDuty Virgin mobile