

Reddit cuts mobile CI build times by up to 50% and improves developer experience with Buildkite

Migrating iOS and Android builds to Buildkite gave Reddit faster pipelines, greater stability, and improved developer experience without increasing infrastructure costs or losing scalability.



Industry
Software

Tech
GCP, Terraform, TypeScript

Founded
2012

Using since
2021

Performance gains

Build times for both Android and iOS apps dropped by about 30%, while queue times fell from several minutes to around five seconds, accelerating delivery.

Improved developer experience

Replacing unwieldy 6,000-line YAML configs with small, reusable pipeline steps reduced friction for test sharding, log customization, and parallel execution of iOS and Android tests.

Operational efficiency

Git and container caching cut checkout times from minutes to 30–40 seconds, while automatic build cancellation prevented wasted compute runs and helped control workloads.

Overview

By early 2024, Reddit's mobile engineering team, more than 200 developers strong, had reached the operational limits of its existing CI/CD system. Build queues stretched into minutes during peak activity. YAML configuration files ballooned, making changes time-consuming and error-prone. Concurrency limits throttled build throughput, while environment drift and dependency instability further eroded velocity and developer satisfaction.

Despite extensive custom tooling, inefficiencies persisted. Queue delays led to wasted developer time. Orchestration complexity increased the risk of failures. The inability to scale cost-effectively meant that supporting growing feature demands could quickly become prohibitively expensive. "There were just so many different points of failure, and we would get hit all the time, because we were hammering the services with our scale," said Geoff Hackett, Staff Android Engineer, Android Platform Team, at Reddit.

The Reddit leadership team recognized that incremental optimizations were no longer enough; they needed a CI/CD platform that could match the scale, pace, and complexity of Reddit's mobile codebase. The solution would have to integrate seamlessly with Kubernetes infrastructure, offer robust caching, improve configurability, and eliminate the roadblocks slowing iOS and Android delivery.

Opportunity

At Reddit's mobile engineering scale, developer velocity was directly gated by the capabilities of its CI/CD platform. "The primary issue was that we kept running up against our concurrency limits," said Hackett.

“

There were just so many different points of failure, and we would get hit all the time, because we were hammering the services with our scale.”

— **Geoff Hackett**,
Staff Android Engineer,
Reddit

"We were not able to control our build environment, so we didn't have the ability to even use a custom Docker image." This forced the Mobile Client Platforms team to maintain a fragile ecosystem of workarounds, including running a second CI system purely to handle build cancellations, building custom retry bots to patch over flaky runs, and heavily sharding YAML configurations to work around platform constraints. The cumulative effect was long feedback cycles, brittle pipelines, and an ever-growing maintenance tax on the platform and release engineering teams.

To sustain product momentum for millions of mobile users, Reddit's engineering leaders set stringent functional requirements for any replacement. The next system required first-class integration with Bazel and BuildBuddy, enabling iOS to utilize high-performance remote execution. Teams required full control over build environments, from Docker images to scripts, to ensure deterministic, reproducible results. It also had to deliver a clean developer experience, enabling pipelines to evolve dynamically without rewriting thousands of lines of YAML. It also needed hybrid hosting options that passed internal security and governance reviews for secrets handling and runner controls.

The business case was straightforward but compelling. A faster, more flexible CI/CD system would unlock engineering autonomy and drastically shorten iteration loops. If queue times could drop from minutes to seconds, if builds could speed up significantly, and if the complexity of sprawling, hard-to-maintain configurations could be eliminated, the result would be a happier and more productive developer base. Achieving this without increasing spend would mean not just a technical win, but a cost-neutral transformation of Reddit's ability to deliver high-quality mobile features at the pace its users expect.

Solution

Reddit's mobile CI/CD overhaul began with a rigorous evaluation of ten different platforms, including Buildkite, GitHub Actions, TeamCity, and Drone. After months of research, prototyping, and side-by-side trials, Buildkite emerged as the solution best aligned with Reddit's scale, Kubernetes-first infrastructure, and need for developer-friendly pipelines.

Buildkite's runtime-generated, composable pipelines were a notable breakthrough. "Our configuration had gotten out of hand," said Hackett. "We had like 6,000 lines of YAML that we eventually split across multiple files." Instead of hard-coding dozens of shards, engineers now define a small "seed" step in Buildkite that emits YAML dynamically at runtime. Using clean, maintainable logic written in Python or a similarly familiar language, they can compose reusable units on the fly, eliminating duplication and making structural changes without mass edits across multiple files. This replaced brittle, hard-coded orchestration with elegant, evolvable logic that scales smoothly with team needs.

For compute, Reddit deployed a hybrid approach that blends Buildkite's high-performance hosted agents with powerful Linux runners for remote execution. By tightly integrating with Bazel and BuildBuddy, both iOS and Android pipelines now run efficiently in a Linux environment, without the complexity or cost of bespoke Mac infrastructure for most stages. This architecture not only improves performance but also reduces operational dependencies, allowing teams to iterate on pipelines faster.

“

Buildkite's dynamic pipelines, Git caching, and container caching were game changers. Git checkouts dropped from minutes to under 40 seconds."

— **Ken Struys**,
Director of Developer Experience,
Reddit

"The decision was actually made for us when we decided we weren't ready to host our own builders," said Hackett. "Buildkite was the only option that gave us the flexibility to use the same system for hosted and on-prem builders, while improving the developer experience."

Buildkite also restored full control and extensibility to engineers. Teams can now build custom Docker images, manage dependencies deterministically, and leverage an approachable plugin model based on simple Bash scripts, steering clear of opaque vendor "actions" or rigid platform constraints. This results in predictable, reproducible environments and faster onboarding for new engineers modifying pipelines.

Performance-critical primitives have been another driver of Reddit's success with Buildkite. "Buildkite's dynamic pipelines, Git caching, and container caching were game changers," said Ken Struys, Director of Developer Experience at Reddit. "Git checkouts dropped from minutes to under 40 seconds."

Intelligent build cancellation automatically terminates obsolete builds when new commits land, cutting wasted concurrency and keeping CI throughput high. Log decoration and timing controls give developers clearer insight into failures, with rich formatting, clickable links, and visual cues improving debuggability and reducing time-to-fix. "Adding a new timed section to a build is as simple as echo '--- A section of a build'," said Hackett. "You can even add colors, images, clickable links and emojis to really customize your log output with some simple decorations."

The rollout was handled through a staged migration with "shadow builds". Buildkite pipelines were run in parallel to the existing production system, allowing for side-by-side validation of performance and stability without risking mobile release schedules. To support this, Buildkite's own engineering team developed a GitHub App for Reddit's GitHub Enterprise Server instance and provided hands-on guidance via Slack throughout the project. "We've had a support channel set up with them since day one, and their team is always responsive," said Hackett. "When we have questions, they've all been really quickly resolved, and they've been super transparent about causes and solutions."

Onboarding the 200+ developers was also extremely straightforward. "Teaching people Buildkite was easier than we expected," said Hackett. "We were able to write up a couple of wikis, and, for the most part, that did it. Everybody could really self-run once they understood the basics, because at the end of the day, we're all just writing Bash scripts."

Despite the scale, with two large mobile monorepos and more than 200 engineers, the core implementation was led by just two engineers, one for iOS and one for Android. Within a few months, the migration shipped ahead of schedule, without a major service disruption. The new pipelines require far less YAML maintenance, while delivering faster builds, shorter queues, and more stable environments from day one. "I think we were able to get everything migrated as quickly as we were because of dynamic pipelines," said Hackett. "Dynamic pipelines let us move really quickly and build a safer

Features leveraged

Dynamic pipelines

Runtime-generated, composable workflows that eliminate YAML duplication and enable rapid CI changes.

[Learn more →](#)

Hosted and Hybrid Agents

Flexible mix of Buildkite-hosted and Reddit-managed agents for scalable, secure builds.

[Learn more →](#)

Custom Docker Support

Full control over build images and dependencies for reproducible, deterministic environments.

[Learn more →](#)

Git and Container Caching

Dramatically cuts checkout and setup times—Git fetches now take just 30–40 seconds.

[Learn more →](#)

Automatic Build Cancellation

Instantly cancels obsolete jobs on new commits, saving cost and reducing wasted compute.

[Learn more →](#)

Bash-based Plugins

Instantly cancels obsolete jobs on new commits, saving cost and reducing wasted compute.

[Learn more →](#)

This combination of composable architecture, hybrid compute, greater control, and targeted performance optimizations have transformed Reddit's mobile CI/CD from a bottleneck into a competitive advantage, laying the groundwork for the substantial gains in speed, reliability, and developer experience measured after go-live. "After about a year of evaluations, a few months of prototyping and debate and another 5-8 months of intense cross-team collaboration, we managed to migrate Reddit's entire mobile CI system," said Hackett. "We've been up and running for almost 3 months and developer sentiment of CI is sky high."

Impact

Reddit's migration to Buildkite delivered measurable, across-the-board improvements. Builds on both iOS and Android now run roughly 30% faster overall, with benchmarking showing a 33% drop at the p50 and an even steeper 47% reduction at the p90. Queue latency, once measured in minutes, has fallen to about five seconds, and merge-queue cycle time was slashed from ~30 minutes to ~15 minutes, significantly shortening feedback loops.

Stability saw a dramatic lift thanks to deterministic, reproducible environments, richer logging, and dynamic orchestration, which reduced flakiness and improved developer sentiment. Git checkout time shrank to 30-40 seconds, while smarter automatic build cancellations cut wasted concurrency. "Buildkite automatically cancels builds when you push to a PR through the previous commit, or to the same branch for the previous commit," said Hackett. "All of these really simple features that we previously had to build tooling around to cancel manually to reduce costs."

In addition, the move away from sprawling YAML files to modular pipeline steps made configuration both simpler and more maintainable. "Buildkite dynamic pipelines have enabled us to build powerful CI systems with less cruft and less code repetition," said Hackett. "That's big. We could do these things on other platforms, but it would take a lot more code to do it."

Perhaps most impressively, all of these gains came without an increase in overall CI/CD spend, resulting in a far better price-performance ratio. For Reddit's mobile teams, Buildkite replaced a frustrating bottleneck with a responsive, efficient, and scalable delivery platform, one that now enables faster innovation, steadier releases, and a smoother experience for both developers and users. "We've been able to make pretty much every process easier just by implementing Buildkite," said Hackett. "Buildkite's flexibility enabled us to complete this complete mobile CI migration in record time, and their superior UI/UX has made our engineers happier and more productive. The speed helps too!"

“

We've been able to make pretty much every process easier just by implementing Buildkite. Buildkite's flexibility enabled us to complete this complete mobile CI migration in record time, and their superior UI/UX has made our engineers happier and more productive."

— **Geoff Hackett**,
Staff Android Engineer,
Reddit

Start turning complexity into an advantage

Create an account to get started with a 30-day free trial. No credit card required.

[Get started](#)
[View pricing](#)
