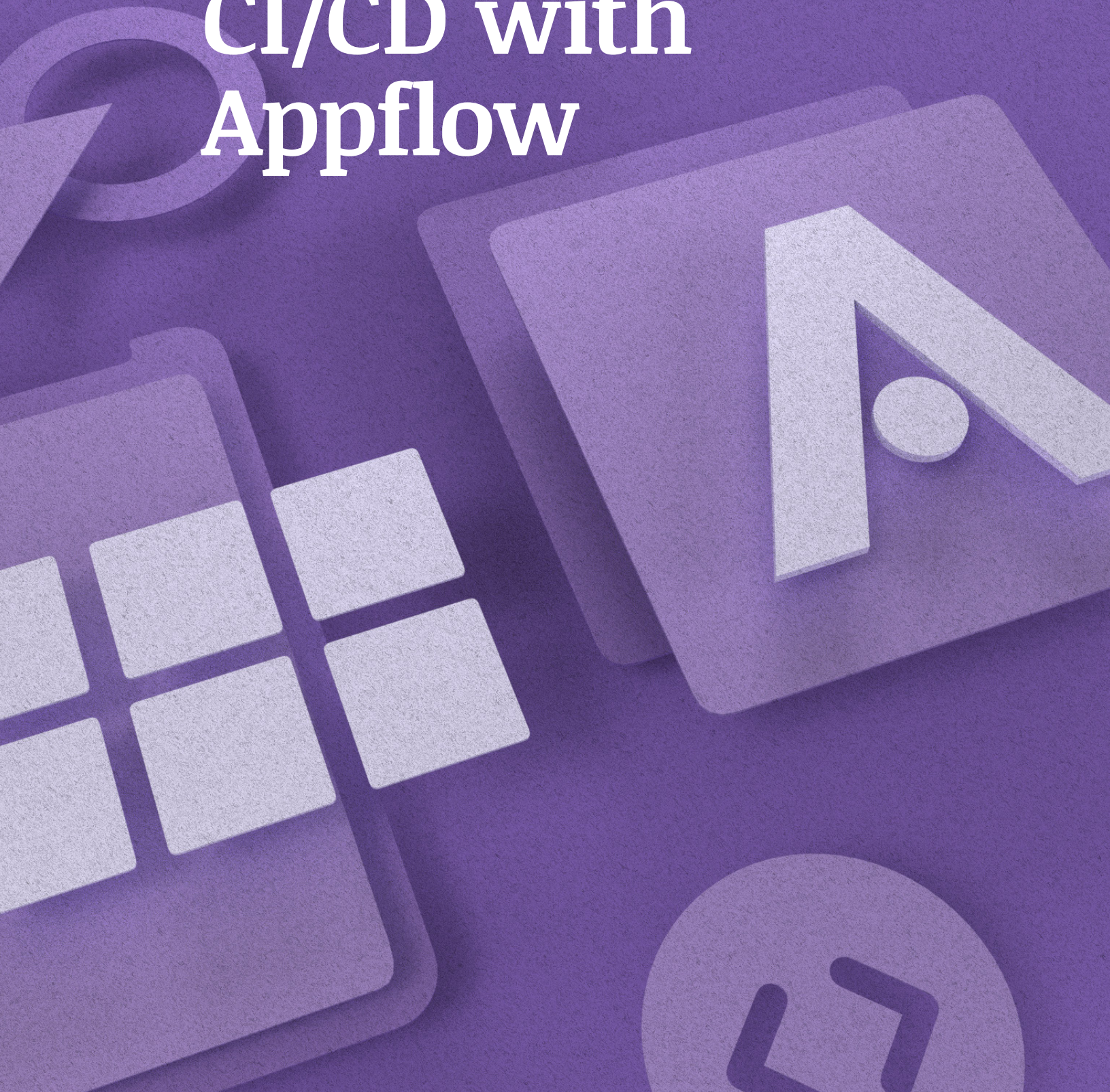


Solving Mobile CI/CD with Appflow



The Emergence of Continuous Integration & Delivery

Continuous integration and delivery (CI/CD) is one of the fastest growing sectors of the developer market. Focused on enabling teams to continuously test and integrate their apps (instead of doing it all at once before shipping), and then helping teams deliver those apps in an automated fashion, CI/CD has transformed the way developers build and ship software.

A staple in the backend and, more recently, frontend ecosystems, CI/CD services have proliferated to help developers perform a set of tedious tasks on every commit to their app, including:

- Running builds for every commit pushed to a git repo
- Running test suites and tracking failing and succeeding tests
- Performing webhooks and other actions after builds
- Launching or deploying apps to hosting destinations after successful builds

These tasks apply to nearly every type of software application, whether a backend, frontend web, or mobile app. As a result, the market has seen an influx of general purpose CI/CD services that integrate code into a shared repository, keeping code progression amongst a developer team seamless. That code is then continuously tested to keep problems from arising. From there, the code is deployed to a live production environment.

And that's just the beginning. Beyond a list of bulleted tasks, CI/CD is part of a much bigger Development and Operations (or [DevOps](#)) movement aimed at creating a culture of shipping often, involving stakeholders, and ensuring quality. Of course, as we'll explain in the sections below, achieving those outcomes in a mobile project is easier said than done.

The Challenges of Mobile CI/CD

For teams building and deploying app store apps, a mobile CI/CD solution is critical. As the CI/CD pipeline has many steps and continuously runs throughout the development cycle, its inherent complexity can cause countless problems, and the fact is that it remains a huge hassle for most development teams.

Even with the prevalence of general purpose CI/CD tools like GitLab and CircleCI, most aren't optimized for mobile, and many still require advanced DevOps expertise. For example, you have to know how to programmatically upload a mobile binary, entirely in the command line, without the help of a wizard or graphical interface.

There are tons of moving parts. And if you get just one thing wrong, it won't work.

Another major challenge is that the average mobile or web developer doesn't have the expertise needed to master DevOps and Mobile CI/CD. As a result, nearly a quarter of developers aren't using any type of DevOps solution and it shows in their release cadence.

According to Ionic's [“The State of Enterprise Mobile App Development”](#) research, 59% of teams without a CI/CD solution release software less than once per month, and 17% will release less than once per quarter.

The Mobile Delivery Gap

Supply and demand represents a big problem as enterprise teams are facing a growing app backlog.

The same research finds that about 68% of respondents, across company sizes, delivered one to three apps in the past year, with most others delivering between three and ten. While these numbers are not encouraging, the number of projects in the development backlog are worse: 84% reported a backlog of between one and nine applications. However, 16% — a significant percentage — have a backlog of 10 or more apps.

DevOps and CI/CD tools can have an enormous impact when it comes to accelerating the pace and frequency of software delivery. These strategies, and the tools that help teams enact them, streamline software development, testing, and deployment, increasing the delivery of the software and the frequency with which organizations release app updates — ultimately helping to close the delivery gap for mobile apps in the enterprise.

Infrequent Releases for Teams Without CI/CD

Development speed and release cadence pose a challenge for teams without a CI/CD tool.

In addition to the statistics above, development time for mobile apps was mostly measured in months. Over 60% of respondents said it takes them anywhere from three to six months to develop an app from start to finish while just 14% are more efficient (one to three months), and the rest have a process that takes longer than six months. There's little question that companies would benefit from an increase in the cadence of mobile app delivery, and a reduction in backlog.

Defining Mobile CI/CD

Mobile app development is one of the most complicated areas of software development.

Given that mobile apps utilize extensive native compilation, rigid operating system requirements, rigorous code signing, frequently-updated SDKs and tooling, complex operating system licensing, device testing, and app store submission, for a mobile CI/CD platform to perform at the highest level, it will need to have extra features to cover these unique demands. These include:

- Managed build environments for iOS, Android, and mobile web apps (for teams building Progressive Web Apps)
- Managed servers and server infrastructure as a service for legal, licensed builds on macOS and Linux
- Per-platform tooling: Xcode for iOS, Android SDK for Android, etc.
- Certificate and profile management for mobile code signing
- Deployment for web assets for hybrid mobile and Progressive Web Apps
- Multiple deployment channels to enable easy development, beta, and production tracks
- Automatic submission to Apple App Store and Google Play Store
- Frequent security patches and upgrades for safe builds on the latest Apple and Google approved tools and SDKs

These features are specialized mobile tasks that most traditional CI/CD services do not handle, so it's obvious that mobile teams will need mobile-specific CI/CD services and infrastructure to meet their development goals.

It stands to reason that as mobile apps continue to grow in importance, companies are searching for ways to efficiently deliver high-quality apps. The opportunity for business leaders to overcome the barriers to adoption is achievable. Actionable insights are readily available and plenty, but it's up to businesses to see past traditional challenges and deliver mobile apps that set them up for success, empowering them to withstand any number of changes the landscape presents. This can be done - as long as organizations understand the challenges and select solutions that make sense.

Appflow: Ionic's Mobile DevOps Platform

As a mobile DevOps platform, [Appflow](#) solves the entire app delivery process: from building a native binary to publishing on the app stores and updating apps over time. Development teams using Appflow can focus entirely on the unique features of their app, while outsourcing app delivery. Appflow powers mobile CI/CD at companies like Burger King, AAA, BCBS, and more.

If you read no further, the features below represent steps in the process of building and releasing an app that are not available from general purpose CI/CD tools:

- Setting up consistent, repeatable mobile build environments so apps can be built on every commit, using predictable dev tools, a blank slate, and operating systems
- Building app binaries and artifacts for iOS, Android, and web
- Running tests and other scripts on every commit to ensure apps stay working
- Connecting to existing GitHub, Gitlab, Bitbucket and other repos, to perform builds on every commit
- Releasing working updates to users in real-time (for web updates), or to the app stores (for binary updates)
- Managing different versions of apps between testers, stakeholders, and production users

The gaps left by these other general purpose tools are filled by Appflow, a robust tool which was designed to support these tasks and more.

Why Choose Appflow?

Appflow, first and foremost, is a managed mobile build environment. That means it provides consistent and frequently updated iOS, Android, and web build environments.

The Appflow team does the hard work of keeping mobile build tools and operating systems up to date and patched for security issues, whether that means updating to the latest versions of Xcode or the Android SDK, or new versions of macOS and Linux, plus a whole lot more.

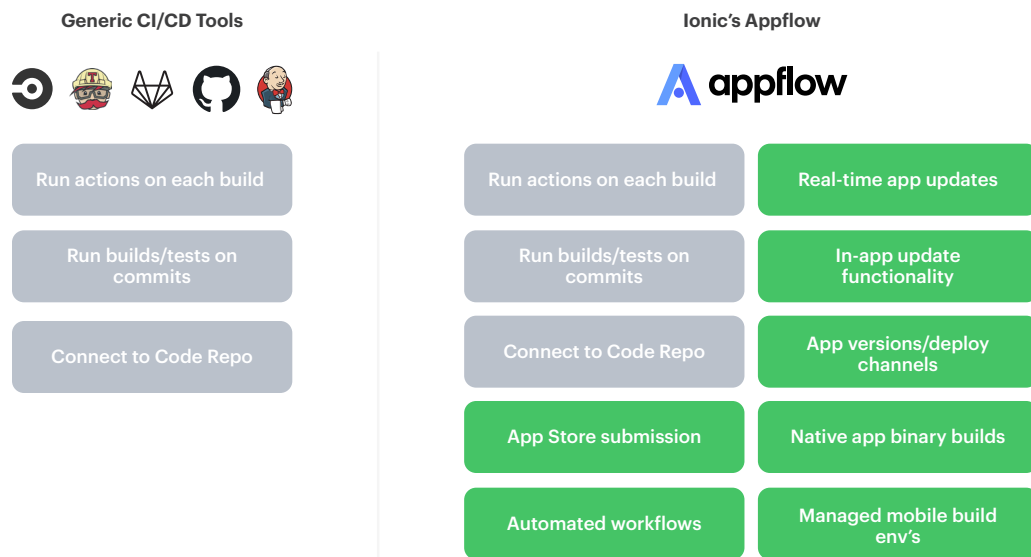
Builds are completed in a highly secure environment, so teams can feel confident offloading their mobile CI/CD process to Appflow, and on high performance hardware to make builds match the speed of development. Appflow also manages authentic, legal Mac build hardware for fully automating iOS builds in the cloud, benefitting development teams that primarily work on Windows machines, but wish to automate processes that would normally require a Mac.

On top of this environment, Appflow provides some pretty transformative features for hybrid app developers. Namely, the ability to remotely update an app without needing to re-submit to app stores (in a way that is App Store ToS friendly!), as well as publish app binaries directly to the Apple App Store and Google Play Store. Appflow can automate that whole process so it triggers on each commit, and enables powerful version splitting so testers, stakeholders, and customers can run on different versions of an app.

Appflow vs. General CI/CD Tools

Generic build services, like CircleCI, GitHub, and others, exist in the CI/CD space, but unlike Appflow, these services merely provide build servers - a way to trigger builds, and hooks for sending those builds elsewhere. But they do not provide the tools or pipeline for mobile-specific builds; they do not handle the deployment of your app, and they do not manage build environments.

Below is a look at the different approaches and how they compare:



In theory, teams could build their own Appflow-style pipeline on another generic CI/CD platform like CircleCI or GitHub Actions, but that would require teams to orchestrate and keep updated a very complex chain of tools, operating systems, build environments, app store submission features, remote app deployment, and more. This would require a full team of engineers, with a different skillset than the typical development team, to build and manage, especially as new mobile and desktop operating systems are released each year.

Here's the good news for companies that have invested in traditional tools: generic CI/CD tools can easily be used alongside Appflow.

In this scenario, a team might have their generic CI/CD tool trigger builds and push code to Appflow, while Appflow manages the hard work of performing mobile builds on managed mobile development infrastructure, while also handling the last mile app update and app store publishing.

The success rate for combining tools is so great, that approximately 68% of Appflow customers are using Appflow alongside a general purpose CI/CD tool, according to data from the [2020 Ionic Developer Survey](#). This delivers the best of both worlds: teams continue using their CI/CD tool of choice, while leaving the highly specialized, mobile-specific build and publishing requirements to Appflow.



What makes Appflow unique?

Appflow helps teams reach their business goals by releasing as often as they like. Our [data](#) shows that teams using Appflow are **90%** more likely to release multiple times per week or more, compared to teams that aren't using a CI/CD tool.



No CI/CD expertise required

- Appflow is easy to set up, with plenty of useful defaults to get you started. With CI/CD best practices already built in, you don't have to be an expert to start improving your team's agility and delivery immediately.
- Everything is accessible in a highly intuitive dashboard; no custom scripting necessary.



Tailored to the needs of mobile developers

- No need to maintain complicated iOS and Android build environments. Simply upload your mobile certificates to Appflow, start a build, and receive native app binaries in minutes.
- Automatic upgrades and optimizations of best-in-class build tools ensures that your production builds get faster over time.



Beyond CI/CD: Fully integrated DevOps

- Instantly send updates to users: deliver content changes, A/B tests, and bug fixes before or after your app is in the app store.
- Review and streamline approval workflows for ultimate flexibility: Increase your team's development agility while still maintaining a high level of quality and security.

Bottom Line

The very best engineering teams ship early and often, even on mobile. The overhead associated with building and publishing mobile apps is no excuse for highly productive mobile teams. In fact, most of our successful customers are shipping weekly.

Shipping frequently, even weekly, has a number of major benefits:

- New features drive business value and grow app adoption
- Frequent updates demonstrate a commitment to the app and a consistent maintenance schedule
- Engineering morale is tied strongly to shipping
- Keeps the app top of mind by appearing on the user's device in the App Store updates list

Appflow enables you to do this by solving the entire app delivery and release process: from building a native binary to publishing on the app stores and updating apps over time. Now, development teams using Appflow can focus entirely on the unique features of their app, and leave app delivery to us.

We think Appflow is pretty incredible. But don't take it from us - our customers' success is our success. Check out how one [neobank in Australia](#) solved their shipping problems with Appflow.



Get started - it's free to try. Or
contact us for more information.



Get started
for free today