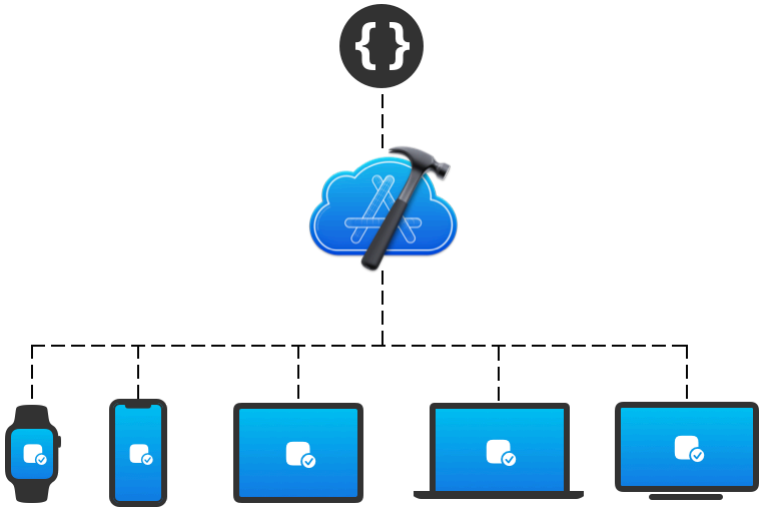
Xcode Cloud

Automatically build, test, and distribute your apps with Xcode Cloud to verify changes and create high-quality apps.

Overview

Xcode Cloud lets you adopt *continuous integration and delivery* (CI/CD), a standard software development practice that helps you develop and maintain your code and deliver apps to testers and users. Xcode Cloud is a CI/CD system that combines the tools you use to create apps and frameworks for Apple platforms: Xcode, TestFlight, and App Store Connect.



With Xcode Cloud, you can automatically and frequently:

- Build your project.
- Run tests and perform verifications.
- Distribute builds to testers and gather their feedback with TestFlight while protecting user privacy.

After successfully verifying a new version of your app with Xcode Cloud and TestFlight, you can quickly release it on the App Store.

For more information about continuous integration and delivery, see About continuous integration and delivery with Xcode Cloud. To learn more about configuring your project or workspace to use Xcode Cloud, see Configuring your first Xcode Cloud workflow.

> **Note**
>
> For additional information about Xcode Cloud that includes videos from WWDC21 and WWDC22, see The Xcode Cloud toolkit.

Topics

Essentials

- About continuous integration and delivery with Xcode Cloud

  Learn how continuous integration and delivery with Xcode Cloud helps you create high-quality apps and frameworks.

- Setting up your project to use Xcode Cloud

  Review account, project, and source control requirements before configuring your project or workspace to use Xcode Cloud.

- Configuring your first Xcode Cloud workflow

  Set up your project or workspace to use Xcode Cloud and adopt continuous integration and delivery.

Setup and maintenance

- Making dependencies available to Xcode Cloud

  Review dependencies and make them available to Xcode Cloud before you configure your project to use Xcode Cloud.

- Configuring Xcode Cloud for your team

  Start using continuous integration and delivery with Xcode Cloud as a team.

- Sharing macOS and Xcode versions across Xcode Cloud workflows

  Use custom aliases to share configurations with multiple workflows.

- Sharing environment variables across Xcode Cloud workflows

  Apply common configurations to multiple workflows by using shared environment variables.

- Building Swift packages and Swift Playgrounds app projects with Xcode Cloud

  Add your Swift package or Swift Playgrounds app project to an Xcode project to build it in Xcode Cloud.

- Setting the next build number for Xcode Cloud builds

  Start numbering builds from a custom build number for your existing Mac app to avoid version collisions.

- Including notes for testers with a beta release of your app

  Add text files to your Xcode project to provide notes to beta testers about what to test.

- Removing your project from Xcode Cloud

  Remove your project from Xcode Cloud to delete app and workflow data, disconnect your Git repository, and remove the Slack integration.

Usage data

- Reviewing Xcode Cloud usage data

  Access Xcode Cloud usage information to understand how you and your team use Xcode Cloud.

Workflows

- Developing a workflow strategy for Xcode Cloud

  Review how you can best create custom Xcode Cloud workflows to refine your continuous integration and delivery practice.

- Xcode Cloud workflow reference

  Configure metadata, start conditions, actions, post-actions, and more to create custom Xcode Cloud workflows.

- Creating a workflow that builds your app for distribution

  Configure a workflow to build and sign your app for distribution to testers with TestFlight, in the App Store, or as a notarized app.

Source code management

- Source code management setup

  Allow Xcode Cloud to access your Git repository.

- Configuring requirements for merging a pull request

  Protect stable branches by requiring a successful Xcode Cloud build or action before it's possible to merge a pull request.

Custom build scripts

- Writing custom build scripts

  Extend your Xcode Cloud workflows with custom build scripts that perform custom tasks or install additional tools.

- Environment variable reference

  Review predefined environment variables you use in custom build scripts.

Troubleshooting

- Resolving common configuration and build issues

  Review common configuration and build issues and learn how you can resolve them.

- Resolve GitHub Enterprise connection issues

  Verify that Xcode Cloud can access your GitHub Enterprise repository and fix configuration issues.

- Reporting feedback for Xcode Cloud

  Provide feedback on issues you encounter when building with Xcode Cloud.

Notifications

- Configuring webhooks in Xcode Cloud

  Configure webhooks that connect Xcode Cloud to other services and tools.

- Xcode Cloud webhook payload reference

  Review details of the webhook payload that Xcode Cloud sends, including the product, workflow, build, actions, results, and SCM metadata associated with it.

- Connecting Xcode Cloud to Slack

  Connect Xcode Cloud to Slack to keep your team informed about the latest Xcode Cloud builds.

REST API

Xcode Cloud Workflows and Builds

Automate reading Xcode Cloud data, managing workflows, and starting builds.

See Also

Distribution and continuous integration

Distribution

Prepare your app and share it with your team, beta testers, and customers.