



Couchbase

Automate Mobile Application Testing in the Cloud with Visual Studio App Center



Presented by Aaron LaBeau

Who am I?



Aaron LaBeau



@biozal



github.com/biozal

CODE

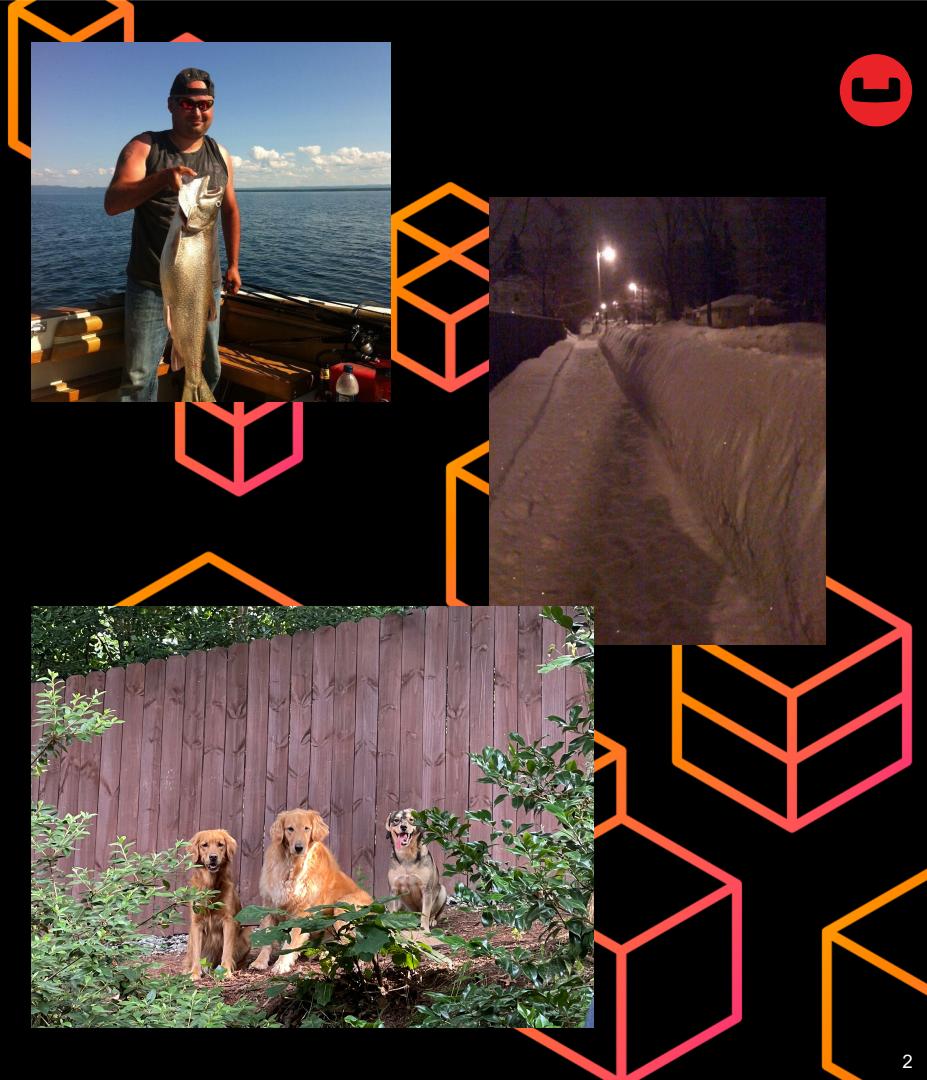
<https://github.com/biozal/appcenter-automation>

<https://github.com/couchbase-examples/>

Web couchbase.com

Twitter twitter.com/couchbase

Facebook facebook.com/couchbase





Mobile Application Experience

Mobile Application users come to expect more from applications, regardless if they are enterprise or from a public App Store

- User Experience
 - Rich, modern, and fast UI
 - Easy to use
 - if you need a manual to train someone how to use a mobile app, you're doing something wrong
- App Crashes
 - Market saturation of apps results in users not use your app or they will find a replacement
- Bugs
 - User will write negative reviews on your app, thus lowering your chances to appeal to new users



Bug Fix Loop for Mobile Apps

- App Store
 - Approval process from Google and Apple can be complex
 - Each time you release is a chance that it might be denied for some “new” reason that wasn’t discovered by the previous reviewer
- Bug Fix Loop
 - Users find the bug and complain – possibly leave poor reviews
 - Isolate the bug
 - Fix the bug
 - Test the fix for regression
 - Build new App – setup update on the App Store
 - Requires App Store approval, sometimes taking between 7 to 14 days





Solution for shipping apps with less bugs?



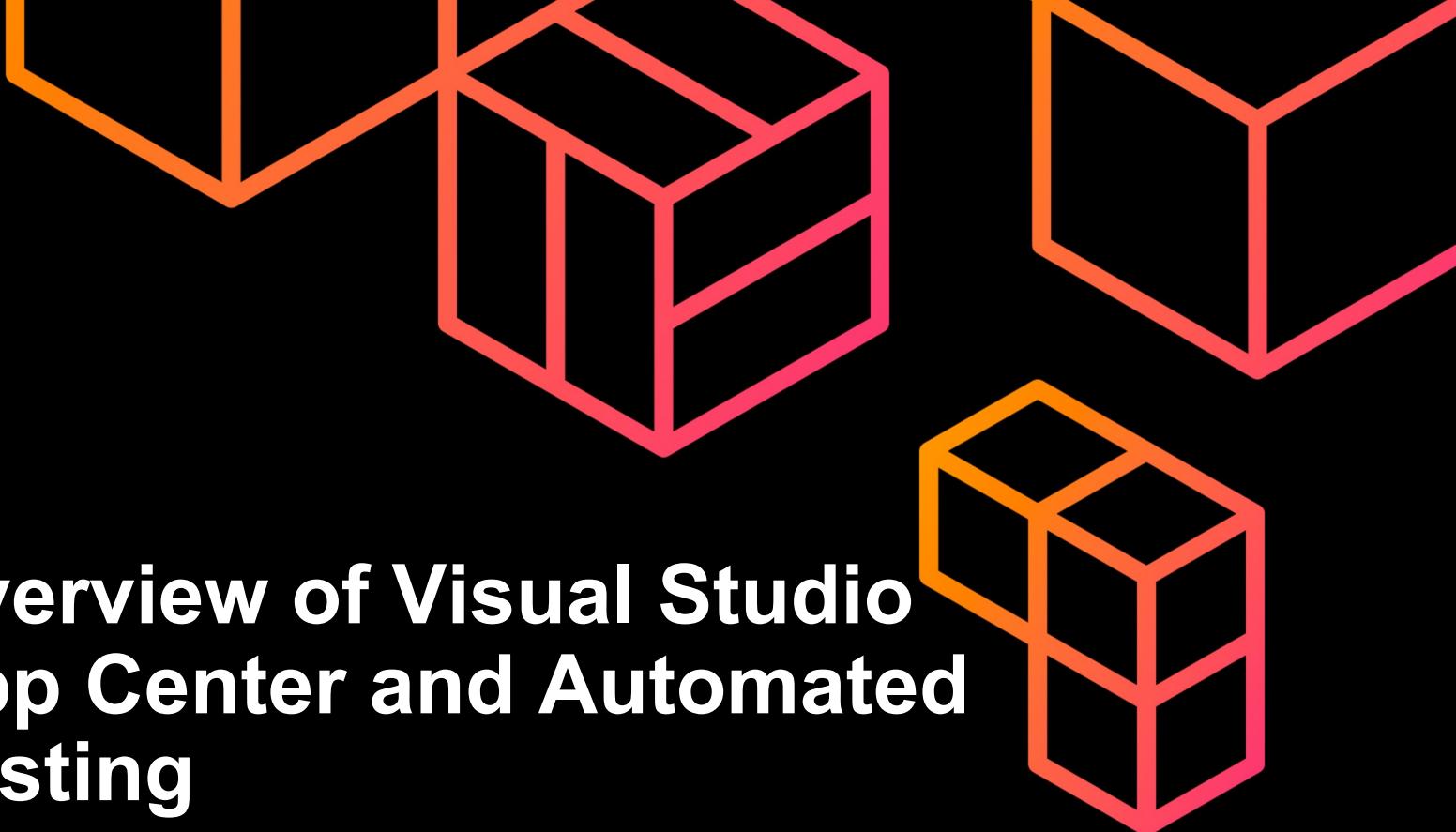
© Alexandersikov



Agenda

- 01/** Overview of Visual Studio App Center and Automated Testing
- 02/** How to Setup an App "Slug"
- 03/** Automation with App Center Builds
- 04/** Automation with GitHub Actions
- 05/** Automation Results in App Center
- 06/** Code Updates

1 Overview of Visual Studio App Center and Automated Testing





What is App Center?

- `Visual Studio` App Center brings together services for building, testing, and distributing applications along providing monitoring services like analytics and diagnostic services
- App Center building and distribution is based on Hockey App, a popular service used by iOS and Android developers to test mobile applications
- App Center testing solution is based on Xamarin Test Cloud, a solution for providing physical devices on the cloud to run automated tests



App Center Features

- **Build** – App Center build service allows you to connect your existing repos from GitHub or Azure Dev Ops and gives you some “lifecycle” scripts (post-clone, pre-build, and post-build). Supports code repos hosted in GitHub, Azure DevOps, Bitbucket, and GitLab
- **Test** – allows you to perform automated UI tests of iOS and Android apps on “hundreds” of configurations and “thousands” of real devices, including latest flagship phone and older hard to find phones
- **Distribute** – allows you to distribute your app to testers, Google Play, Apple’s App Store, and Microsoft InTune (MDM)
- **Analytics** – allows you to use the SDK to follow custom events along with information about your audience including active users, popular device models, and activity.
 - Warning: Information isn’t stored very long on the free account, need to export to Azure App Insights or Azure Blob storage
- **Crash** – get crash reports throughout the lifecycle of your app regardless of what you use to distribute



Automated Testing (UI Tests)

- UI Tests can run on real physical devices and grab screenshots for you to review
- Frameworks
 - iOS (10.0 and higher)
 - Appium (Java/JUnit)
 - Calabash (Ruby/Cucumber)
 - XCUITest (Swift/Obj-C)
 - Android (5.0 and higher)
 - Appium (Java/JUnit)
 - Espresso (Java or Kotlin)
 - Xamarin
 - Xamarin.UITest (C#/F#)

What other Automated Testing solutions exist?

- Perfecto (<https://www.perfecto.io>)
 - Works with Appium, Espresso, Quantum, XCUI Test
- App Center (Microsoft – <https://appcenter.ms>)
 - Works with Appium, Calabash, Espresso, Xamarin.UITest, XCTest
- AWS Device Farm (Amazon - <https://aws.amazon.com/device-farm/>)
 - Appium Java JUnit, Appium Java TestNG, Appium Python, Calabash, Instrumentation (Including JUnit, Espresso, Robotium), UI Automation, UI Automator, and XCTest (Including XCUI and KIF)
- Firebase Test Lab – Google (<https://firebase.google.com/docs/test-lab>)
 - Instrumentation tests, Robo test for Android, Game Loop tests for iOS and Android, and XCTest

2

How to Setup an App and "Slug" in App Center





Create Form

The screenshot shows the Couchbase App Center interface. On the left, there's a sidebar with 'All apps' and a list of existing apps: Aaron LaBeau, Couchbase, and Couchbase, Inc. The main area is titled 'Apps' and lists two entries: 'Cordova Test App - iOS' and 'UserProfile-Standalone-Android'. A search bar is at the top of this list. On the right, a modal window titled 'Add new app' is open. It contains fields for 'App name:' (set to 'Inventory-Demo-Compose'), 'Icon:' (a placeholder icon), 'Release Type:' (set to 'Enterprise'), and 'Owner:' (set to 'Couchbase, Inc.'). Below these are sections for 'OS:' (with 'Android' selected), 'Platform:' (with 'Java / Kotlin' selected), and an 'Add new app' button at the bottom right.

App Center Couchbase, Inc. / Apps

All apps

Aaron LaBeau

Couchbase

Couchbase, Inc.

Apps

Search

Name OS

Cordova Test App - iOS iOS

UserProfile-Standalone-Android Android

Add new app

App name: * Inventory-Demo-Compose Icon:

Release Type: Enterprise Owner: Couchbase, Inc.

OS: iOS Android Windows macOS tvOS Custom

Platform: Java / Kotlin React Native Xamarin Unity

Add new app



Setup your App

App Center Couchbase, Inc. / Inventory-Demo-Compose / Overview

Inventory-Dem... Android

Overview

Build

Test

Distribute

Diagnostics

Analytics

Settings

Getting started

Add App Center's SDK to your app.

For detailed instructions on SDK integration go to: [Getting Started with the Android SDK](#)

1 Add the SDK to the project

In your `app/build.gradle` add the following lines:

```
dependencies {
    def appCenterSdkVersion = '4.3.1'
    implementation "com.microsoft.appcenter:appcenter-analytics:${appCenterSdkVersion}"
    implementation "com.microsoft.appcenter:appcenter-crashes:${appCenterSdkVersion}"
}
```

2 Start the SDK

Open your app's main activity class and add the following import statements.

```
import com.microsoft.appcenter.AppCenter;
import com.microsoft.appcenter.analytics.Analytics;
import com.microsoft.appcenter.crashes.Crashes;
```



App App Token

- API tokens authenticate you to the app profile and all you to upload builds and run tests

The screenshot shows the Couchbase App Center interface for the 'Inventory-Demo...' app. The left sidebar has tabs for Overview, Build, Test, Distribute, Diagnostics, Analytics, and Settings. The Settings tab is currently selected. The main area displays 'App details' for the app, which is an Android application developed in Java / Kotlin by Couchbase, Inc. It shows 1 collaborator and 1 team, data management set to 90 days, and no services or webhooks. The 'Email notifications' section indicates a failure with the message: 'New version distributed and adding new devices to a release failed'. At the bottom, the 'App API tokens' section is highlighted with a red border, showing 'No tokens'.



Testing via App Center vs Your Solution (blue or red pill)

- If you aren't using the "Build" features in App Center, you will be required to install node and the appcenter-cli tool installed (and this must be done on your build host/build agent)
 - `npm install -g appcenter-cli`
- Once installed you can use the 'appcenter' command line tool
 - <https://github.com/microsoft/appcenter-cli>
- Tool updates OFTEN so make sure you have the latest when you update Xcode



What is an "Slug" / Needed info

- When running distribute or test commands from the cli you need the application slug or information about the app location in app center
- The format is usually
 - OrgName/AppName

Example:



← → C ⌂ appcenter.ms/orgs/Couchbase-Inc./apps/Inventory-Demo-Compose

Couchbase-Inc./Inventory-Demo-Compose



Services – Bug Tracker

The screenshot shows the 'Services' section of the Couchbase App Center. On the left, there's a sidebar with icons for Overview, Build, Test, Distribute, Diagnostics, Analytics, and Settings. The Settings icon is highlighted. In the center, there's a large blue 'C' logo for 'UserProfile-Standalone-Android'. To the right, under the heading 'Services', there's a 'Bug tracker' section showing a GitHub icon and the email address alabau@gmail.com. Below that is a button labeled '+ Add account or service'. Other service options listed include Webhooks, Export, Email notifications, and App API tokens.

Useful if you use the crash feature it can open a "bug" or "issue" for you automatically.



Authorization between App Center and GitHub

The image displays two screenshots illustrating the authorization process between App Center and GitHub.

Left Screenshot: GitHub OAuth Authorization Dialog

A user is授权 (Authorizing) an application named "App Center BugTracker" from "VSAppCenter". The application is requesting access to the user's GitHub account. The requested permissions include:

- Repositories**: Public and private
- Personal user data**: Email addresses (read-only)

The "Organization access" section shows the application has access to the "couchbase-examples" organization. A "Grant" button is visible at the bottom right.

Right Screenshot: GitHub Organization Settings - Third-party application access policy

The "Couchbase Examples" organization settings page is shown. Under the "Third-party application access policy" section, it indicates the "Policy: Access restricted" and lists the "App Center BugTracker" application as approved. A note states: "Only approved applications can access data in this organization. Applications owned by couchbase-examples always have access." A "Remove restrictions" button is present.

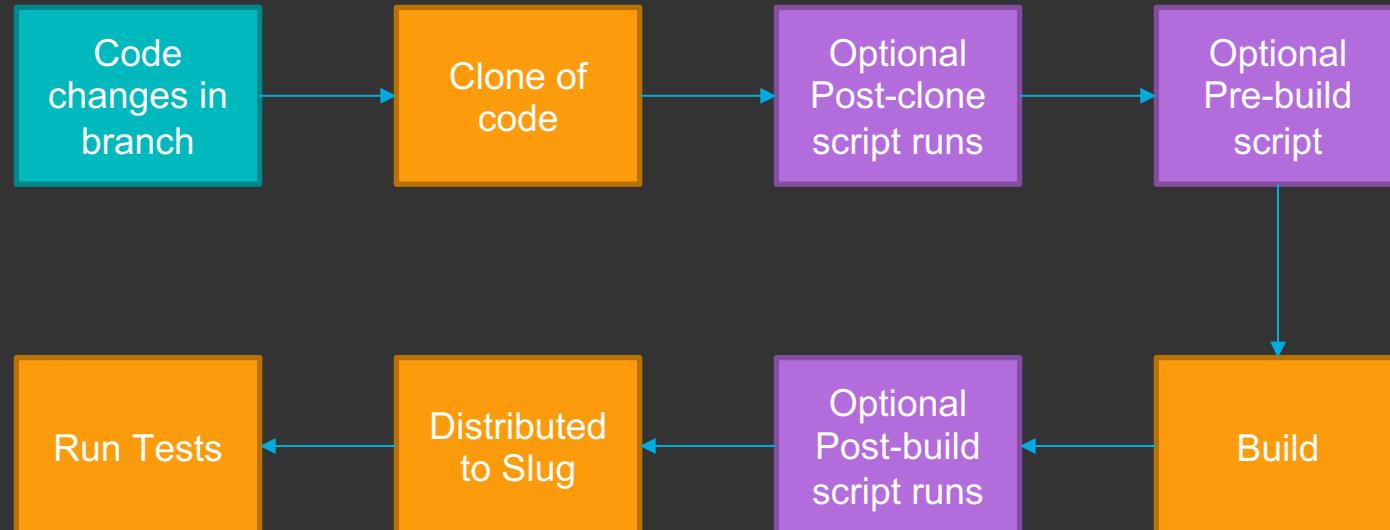
3

Automation with App Center Builds



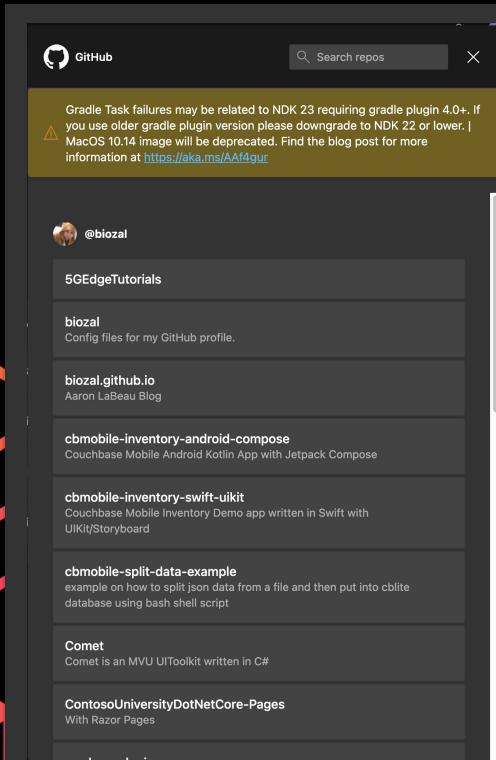


Workflow Using App Center Build





Setup Build Process



App Center can only see the GitHub org that you log into – i.e. if you login with an Azure Active Directory account you will only see repos that are connected to your user.

To set this up properly you need to be an Admin of both App Center and the GitHub org you are trying to connect to each other.

Once connected, App Center will install an App into the GitHub org to connect and allow OAuth authorization.



Setup Build Process – branches

App Center Couchbase, Inc. / Couchbid-Xamarin-Android / Build / Branches / main Go to dev

Couchbid-Xamarin-Android Android

Overview Build Test Distribute Diagnostics Analytics Settings

↳ Branches

- api-bid
- api-cb-lite
- cb-lite-3-memory-bug
- detail-screen-timer
- main
- mobile-bid

main



This branch has not been configured to build yet
It sure looks intriguing, though!

LAST COMMIT 4 days ago Configure build

merge fixes Aaron LaBeau



Setup Build Process – build configuration

The screenshot shows the 'Build configuration' screen for a project named 'main'. On the left, there's a sidebar titled 'Build app' with fields for 'Project' (Wavelength.Android.csproj), 'Configuration' (Release), 'SDK version' (Xamarin.Android 12.0), 'Build scripts' (None), 'Build frequency' (Build this branch on every push selected), and options for 'Build Android App Bundle' (Off) and 'Automatically increment version code' (On). Below these are settings for 'Build number format' (Timestamp) and 'Environment variables' (off). On the right, a vertical menu lists 'Build', 'Environment', 'Sign', 'Test', 'Distribute', and 'Advanced'. At the bottom are 'Save' and 'Save & Build' buttons.

The configuration wizard most of the time will know how to select proper files based on how you setup your app platform.

It supports auto changing of your version number, signing of applications and environment variables you can use in your custom scripts.

Setup Build Process – build configuration continued

The screenshot shows the 'Build configuration' screen for a project named 'main'. The 'Sign builds' section is turned 'On' with the note 'Builds must be signed to run on devices.' It includes a 'Keystore' section where a 'Keystore file' can be uploaded (using JKS format). 'Environment variables' are set to APPCENTER_KEYSTORE_PASSWORD, APPCENTER_KEY_ALIAS, and APPCENTER_KEY_PASSWORD. The 'Test on a real device' section is also turned 'On' with a 'Free' tier selected. A note says 'Verify that your build works on a real device by running a launch test.' Below this is a link to 'Do not use personal data in your tests'. The 'Distribute builds' section is turned 'On' with 'Groups' selected. A dropdown menu for 'Select destination' is open, and a checkbox for 'Do not notify testers' is checked. At the bottom are 'Save' and 'Save & Build' buttons.

To test builds and push them on a device on Android and iOS you must sign the builds. For Android you will need a keystore file and setup. For iOS you will need your cert/application profile files.

You can set up your Test from here along with distribution to internal testers.

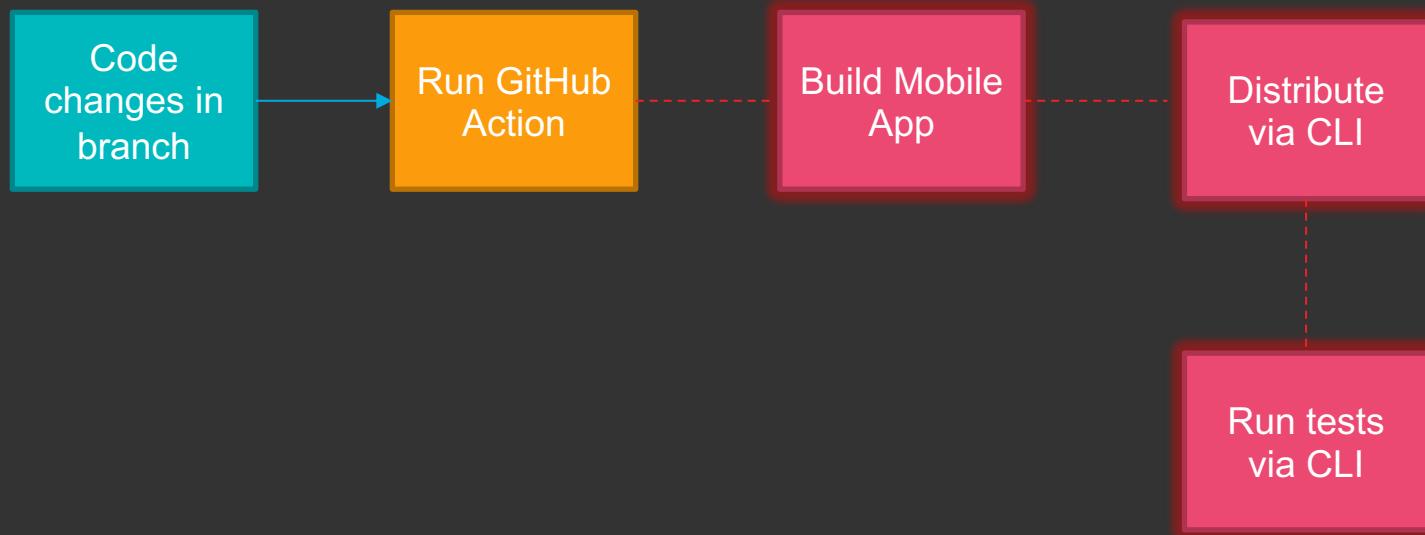
4

Automation with GitHub Actions





Workflow Using GitHub Actions





Live Demo

DEMO



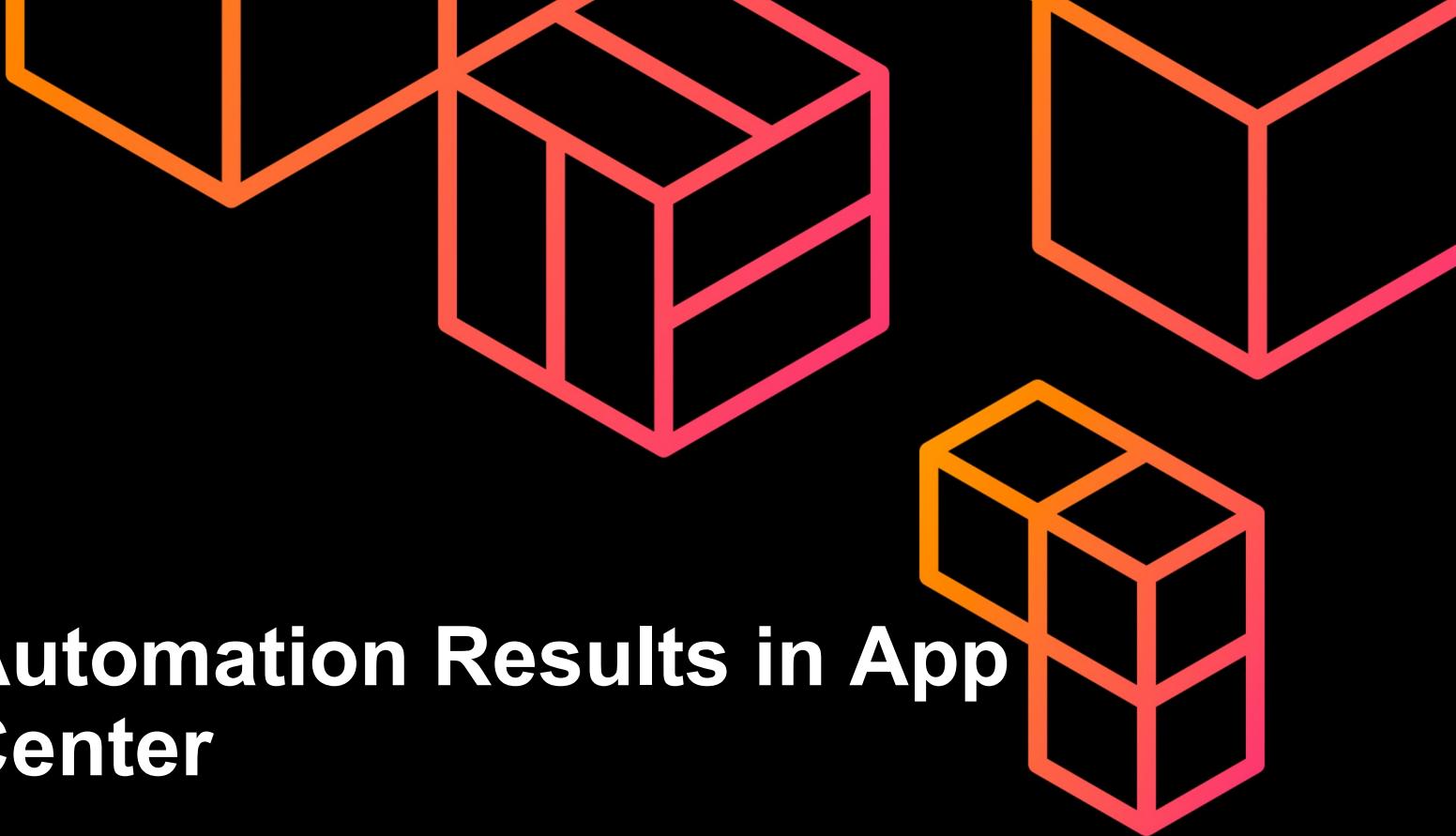


What about iOS and Certificates/Profiles

- With iOS it's highly recommended to review Microsoft's documentation
 - <https://docs.microsoft.com/en-us/appcenter/test-cloud/ios-sign-for-testing>
 - <https://docs.microsoft.com/en-us/appcenter/test-cloud/frameworks/xcuitest/>
- With iOS App Center can resign your application, which means it can run on there devices. They can only do this for apps that are signed with a Developer profile/Cert
- App Center can't resign apps that are signed with a App Store Profile/Cert

5

Automation Results in App Center





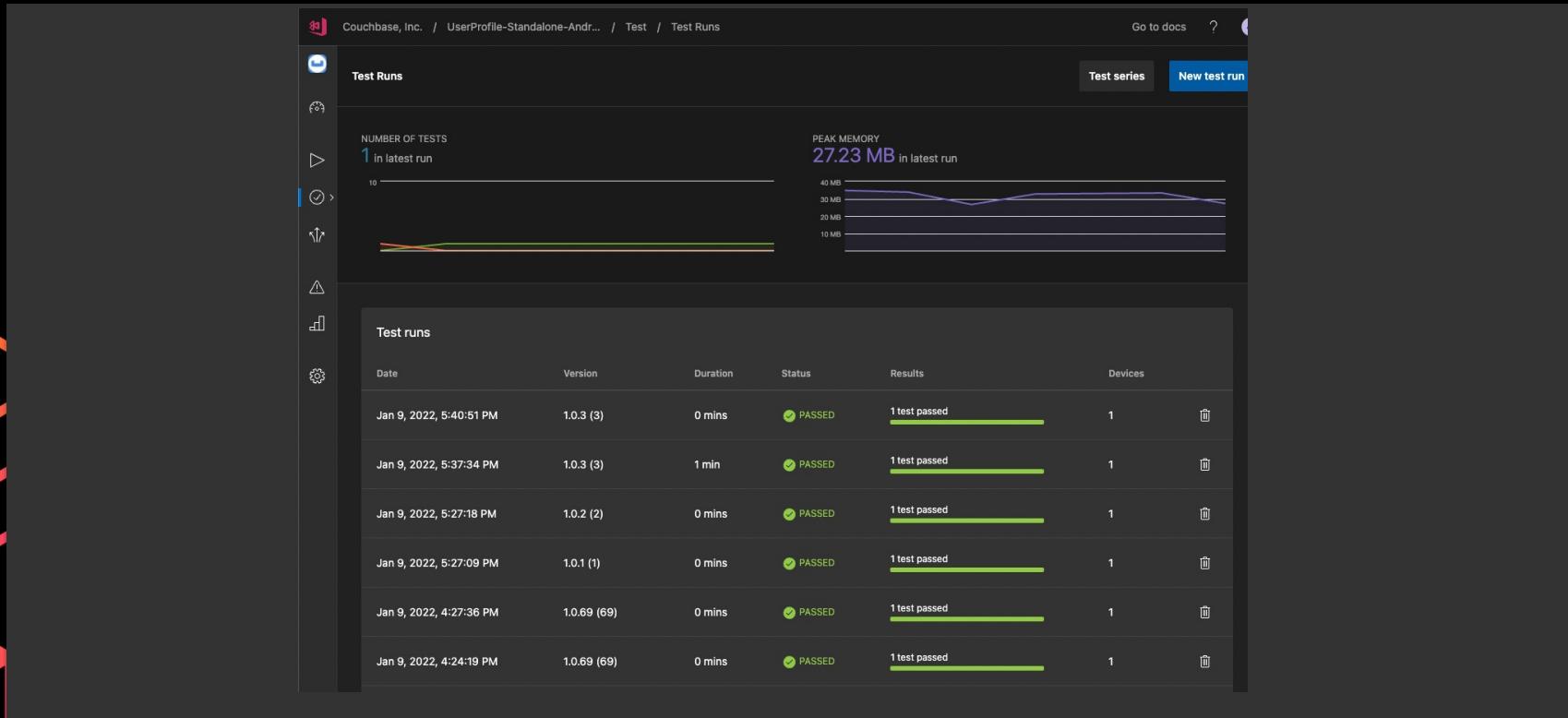
Live Demo

DEMO





Results Listing





Results Details

Couchbase, Inc. / UserProfile-Standalone-Andr... / Test / Test Runs / Jan 9, 2022, 5:40:51 PM Go to docs ?

Test Runs

Run	Date	Status
1	Jan 9, 2022, 5:40:51 PM	1 test
2	Jan 9, 2022, 5:37:34 PM	1 test
3	Jan 9, 2022, 5:27:18 PM	1 test
4	Jan 9, 2022, 5:27:09 PM	1 test
5	Jan 9, 2022, 4:27:36 PM	1 test
6	Jan 9, 2022, 4:24:19 PM	1 test
7	Jan 9, 2022, 4:07:33 PM	1 test

Jan 9, 2022, 5:40:51 PM
1 test

VERSION 1.0.3 (3) DEVICES 1 RUN TIME 5 min

100% 1/1 Test passed

100% 1/1 Device passed

Tests

Feature	Peak duration	Peak memory	Status
FunctionalTests	19.23 sec	27.23 MB	Passed
loginTest	19.23 sec	27.23 MB	Passed



Results Screenshots

Couchbase, Inc. / UserProfile-Standalone-Andr... / Test / Test Runs / Jan 9, 2022, 5:40:51 PM / loginTest Go to docs ? A

Jan 9, 2022, 5:40:51 PM 1 test loginTest data validation completed

DEVICE Samsung Galaxy S6 VERSION Android 5.1.1 SCREEN 5.1" RELEASE April 2015 X

FunctionalTests

loginTest

- Logging into the App
- Check email is correct from login
- Start updating user profile
- Save User Profile information
- Logout
- Log back into the App
- Start data validation
- data validation completed

SAMSUNG 100% 23:44

User Profile

Details Logs

Hardware usage

Memory	27.2 MB
CPU	20.7 %

DURATION Step ? Test

This device	0.25 sec	19.23 sec
Average	0.25 sec	24.06 sec

6 Code Updates





Report Helper

- Some frameworks will require helper classes like Espresso in order to take snapshots while the tests are running.
- Other frameworks like iOS automatically take snap shots as part of the framework at the end of each Method.
- RTM to see for your platform the steps you need to do in order to customize your snapshots.

<https://docs.microsoft.com/en-us/appcenter/test-cloud/frameworks/>





Live Demo Android

DEMO

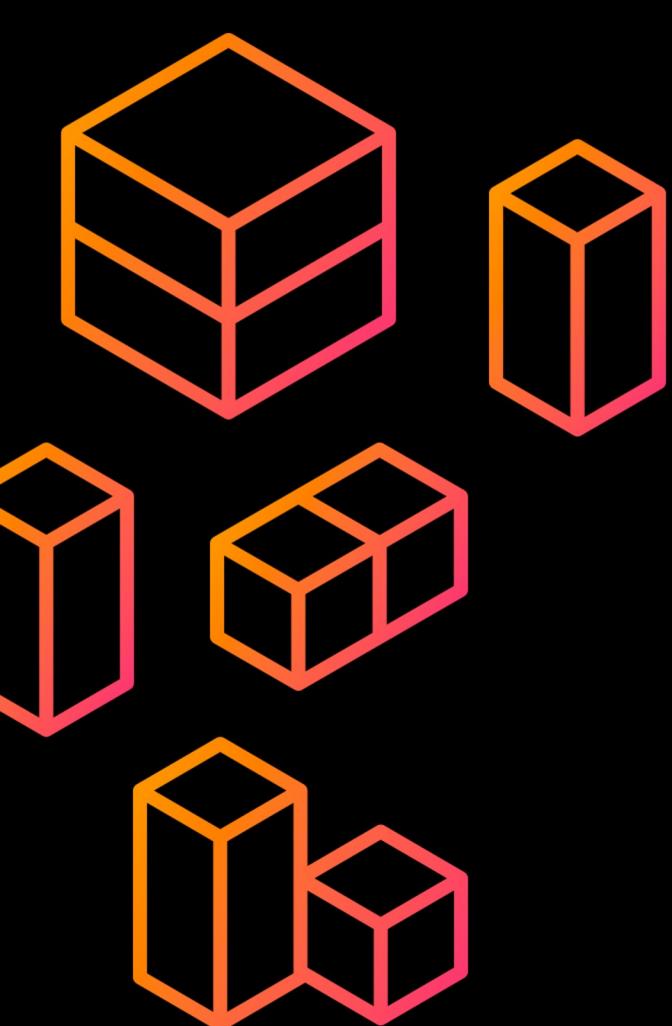




Summary

- There are several solutions for testing mobile applications with physical devices in the cloud
- App Center provides support for iOS and Android and supports older devices
- Some code changes may be required in order to get the optimal test experience
- Companies using MDM with device enrollment will find very limited support, if any and a lot of challenges





Questions?



Couchbase