

CI/CD

Codemagic



Joseph Co.

FLOW

Prepare your APK



Create Key Store



Upload Keystore to Codemagic

A photograph of three large, dark grey cooling towers of a nuclear power plant. They are set against a bright blue sky with scattered white clouds. The towers are cylindrical with a ribbed or fluted surface texture. Each tower has a red and white striped band near its top. The central tower is the largest and most prominent, while the two flanking it are slightly smaller.

PREPARATION
APK

Directory

File: android/gradle.properties

Add this line of code:

```
android.newDsl=false
```

Directory

File: android/app/build.gradle.kts

Update this line of code:

Note:

The applicationID is same
as your updated value

```
defaultConfig {  
    applicationId = "com.devops.test_deploy"  
    minSdk = flutter.minSdkVersion  
    targetSdk = flutter.targetSdkVersion  
  
    val buildId = System.getenv("CM_BUILD_ID")?.toIntOrNull() ?: 1  
    versionCode = buildId  
    versionName = "1.0.$buildId"  
}  
  
signingConfigs {  
    create("release") {  
        val keystorePath = System.getenv("CM_KEYSTORE_PATH")  
        if (!keystorePath.isNullOrEmpty()) {  
            storeFile = file(keystorePath)  
            storePassword = System.getenv("CM_KEYSTORE_PASSWORD")  
            keyAlias = System.getenv("CM_KEY_ALIAS")  
            keyPassword = System.getenv("CM_KEY_PASSWORD")  
        }  
    }  
}  
  
buildTypes {  
    release {  
        val keystorePath = System.getenv("CM_KEYSTORE_PATH")  
        if (!keystorePath.isNullOrEmpty()) {  
            signingConfig = signingConfigs.getByName("release")  
        }  
        isMinifyEnabled = false  
        isShrinkResources = false  
    }  
}
```

The screenshot shows a code editor interface with a dark theme. On the left, there is a sidebar titled "SEARCH" containing a search bar with the text "com.example.myapp" and a dropdown menu showing "com.devops.myapp". Below the search bar, it says "14 results in 6 files - Open in editor - Search with AI". The sidebar lists several files and folders with their counts: "build.gradle.kts android\app" (2), "MainActivity.kt android\app\src\main..." (1), "project.pbxproj ios\Runner.xcodeproj" (6), "CMakeLists.txt linux" (1), "ApplInfo.xcconfig macos\Runner\Config..." (1), and "project.pbxproj macos\Runner.xcode..." (3). A gear icon at the bottom of the sidebar has a blue circle with the number "1" on it.

The main area displays the content of the file "build.gradle.kts". The code is as follows:

```
plugins {
    id("com.android.application")
    id("kotlin-android")
    // The Flutter Gradle Plugin must be applied after the Android and Kotlin Gradle plugins.
    id("dev.flutter.flutter-gradle-plugin")
}

android {
    namespace = "com.example.myapp"
    compileSdk = flutter.compileSdkVersion
    ndkVersion = flutter.ndkVersion

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_17
        targetCompatibility = JavaVersion.VERSION_17
    }

    kotlinOptions {
        jvmTarget = JavaVersion.VERSION_17.toString()
    }

    defaultConfig {
        // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html)
        applicationId = "com.example.myapp"
        // You can update the following values to match your application needs.
        // For more information, see: https://flutter.dev/to/review-gradle-config.
        minSdk = flutter.minSdkVersion
        targetSdk = flutter.targetSdkVersion
        versionCode = flutter.versionCode
        versionName = flutter.versionName
    }

    buildTypes {
        release {
            // TODO: Add your own signing config for the release build.
            // Signing with the debug keys for now, so `flutter run --release` works.
            signingConfig = signingConfigs.getByName("debug")
        }
    }
}
```

The status bar at the bottom indicates "Ln 9, Col 18 (17 selected)" and other settings like "Spaces: 4", "UTF-8", "CRLF", and "Plain Text".

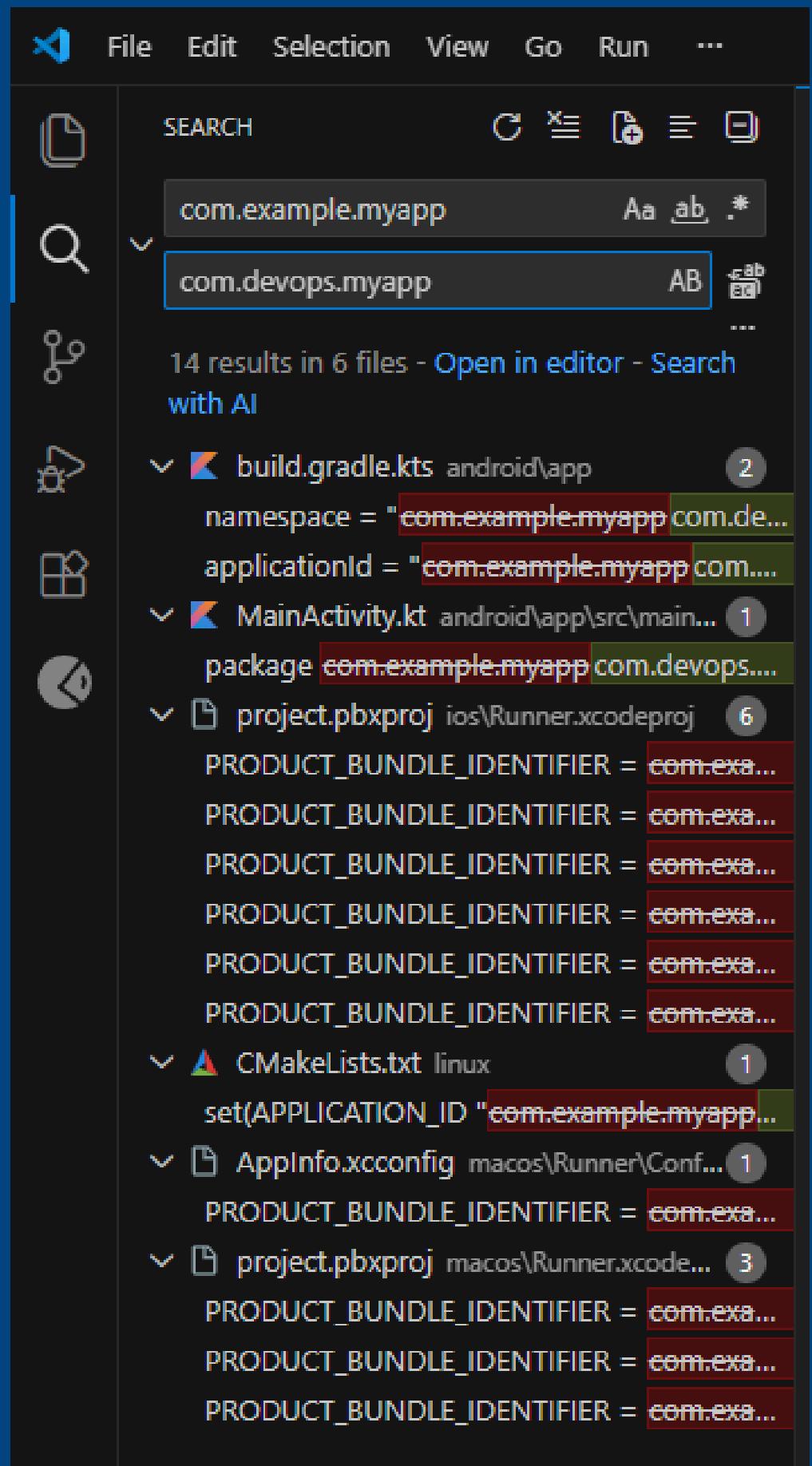
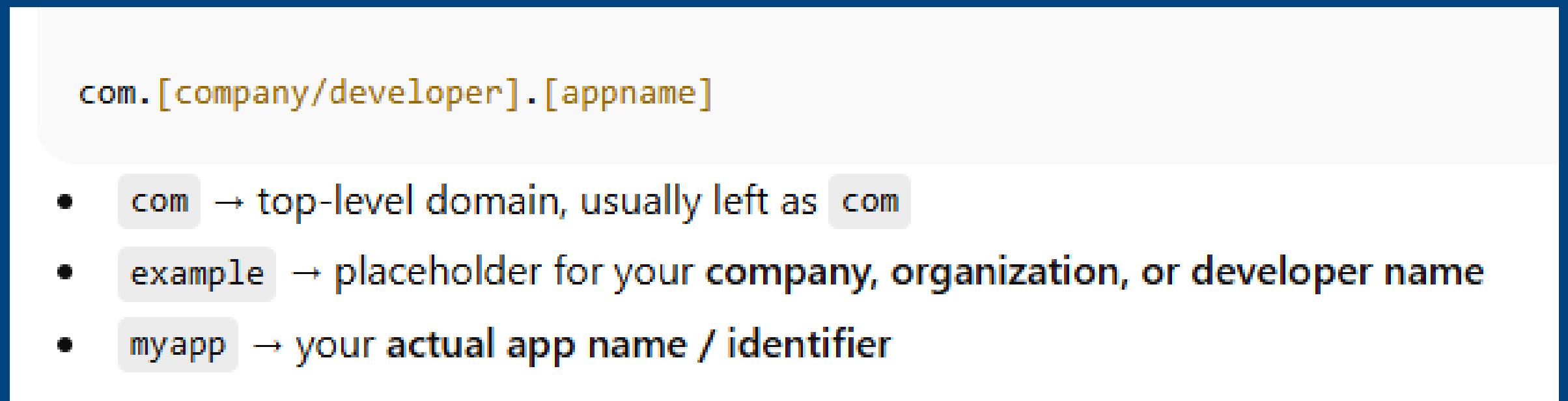
Next, copy the name from the namespace, search for it in your project files, and replace it with your new name.

Example:

Current: "com.example.myapp"
New: "com.devops.myapp"

Note: Do not change the com.

```
android {  
    namespace = "com.example.myapp"  
    compileSdk = flutter.compileSdkVersion  
    ndkVersion = flutter.ndkVersion
```





CREATE
KEYSTORE

Command Input

bash

 Copy code

```
keytool -genkey -v -keystore keystore.jks -keyalg RSA -keysize 2048 -validity 10000 -alias name
```

Explanation:

1. `keytool` → Java tool for managing keys and certificates.
2. `-genkey` → Tells keytool to generate a new key pair (public + private key).
3. `-v` → Verbose mode, shows detailed output while creating the key.
4. `-keystore keystore.jks` → The file where your key will be stored (`keystore.jks`).
5. `-keyalg RSA` → The algorithm used for the key; RSA is standard for Android signing.
6. `-keysize 2048` → Key length in bits (2048 is secure).
7. `-validity 10000` → Key is valid for 10,000 days (~27 years).
8. `-alias name` → The name you give to this key inside the keystore. You can reference this later in your Gradle build.

Command OutPut

```
PS D:\keystore\test> keytool -genkey -v -keystore keystore.jks -keyalg RSA -keysize 2048 -validity 10000 -alias keyness  
Enter keystore password:
```

```
Re-enter new password:
```

```
Enter the distinguished name. Provide a single dot (.) to leave a sub-component empty or press ENTER to use the default value in braces.
```

```
What is your first and last name?
```

```
 [Unknown]: keyness
```

```
What is the name of your organizational unit?
```

```
 [Unknown]: .
```

```
What is the name of your organization?
```

```
 [Unknown]: .
```

```
What is the name of your City or Locality?
```

```
 [Unknown]: .
```

```
What is the name of your State or Province?
```

```
 [Unknown]: .
```

```
What is the two-letter country code for this unit?
```

```
 [Unknown]: ph
```

```
Is CN=keyness, C=ph correct?
```

```
 [no]: yes
```

```
Generating 2048-bit RSA key pair and self-signed certificate (SHA384withRSA) with a validity of 10,000 days  
for: CN=keyness, C=ph
```

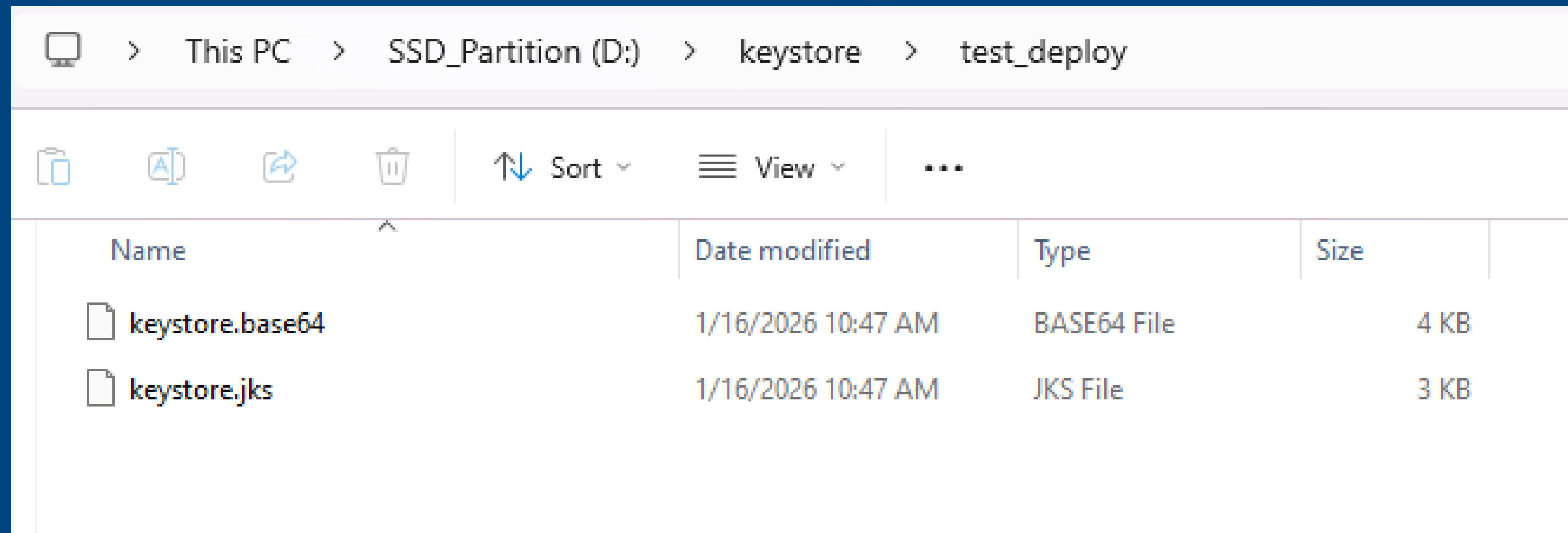
```
[Storing keystore.jks]
```

Optional Convert to base64:

```
[Convert]::ToString((Get-Content  
"keystore.jks" -Encoding Byte)) | Set-Content keystore.base64
```

```
PS D:\keystore\test> [Convert]::ToString((Get-Content "keystore.jks" -Encoding Byte)) | Set-Content keystore.base64
```

Folder Output:



A photograph of four large, dark grey cylindrical cooling towers of a nuclear power plant. They are set against a bright blue sky with scattered white and grey clouds. The towers have a ribbed texture and red and white safety markings near their tops.

Upload keystore
in CodeMagic

Build overview - Codemagic Application settings - Codemagic

codemagic.io/app/6969a63f03d8c799c6c317c9/workflow/6969a63f03d8c799c6c317c8/settings

codemagic

test_deploy

Start new build →

Workflow Editor Webhooks Scheduled builds **Repository settings**

Workflow **Default Workflow**

Switch to YAML configuration

Last updated: Jan 16th, 2026 at 10:51 AWST by (manansalajosephlee@gmail.com)

Build for platforms

Android iOS Web macOS Linux 

Windows  Run tests only

Publish updates to user devices using Shorebird

Disabled Release Patch

Run build on

All systems operational

User preferences

Log out

Workflow settings

Workflow name: **Default Workflow** 

Max build duration:

 Selected: 60 min 30 120

Duplicate workflow

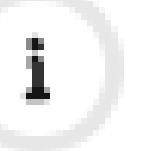
Make builds public 

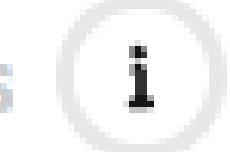
Build status badge 



Step 1: (Build for Android)

Build for platforms

Android **iOS** **Web** **macOS** **Linux** 

Windows  **Run tests only**

Step 2: (Build triggers for Android)

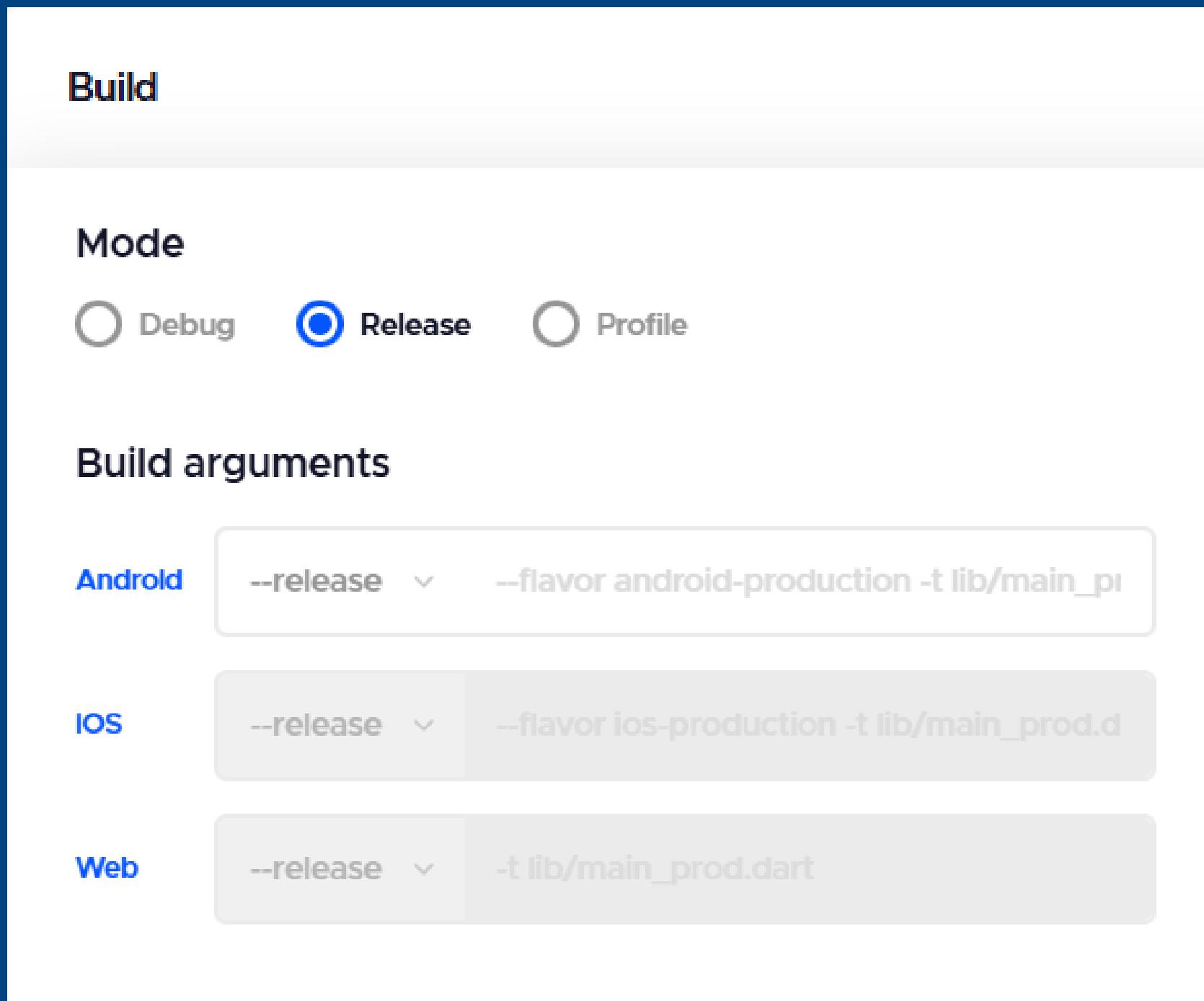
Build triggers

Select which branches to track and when to trigger builds.

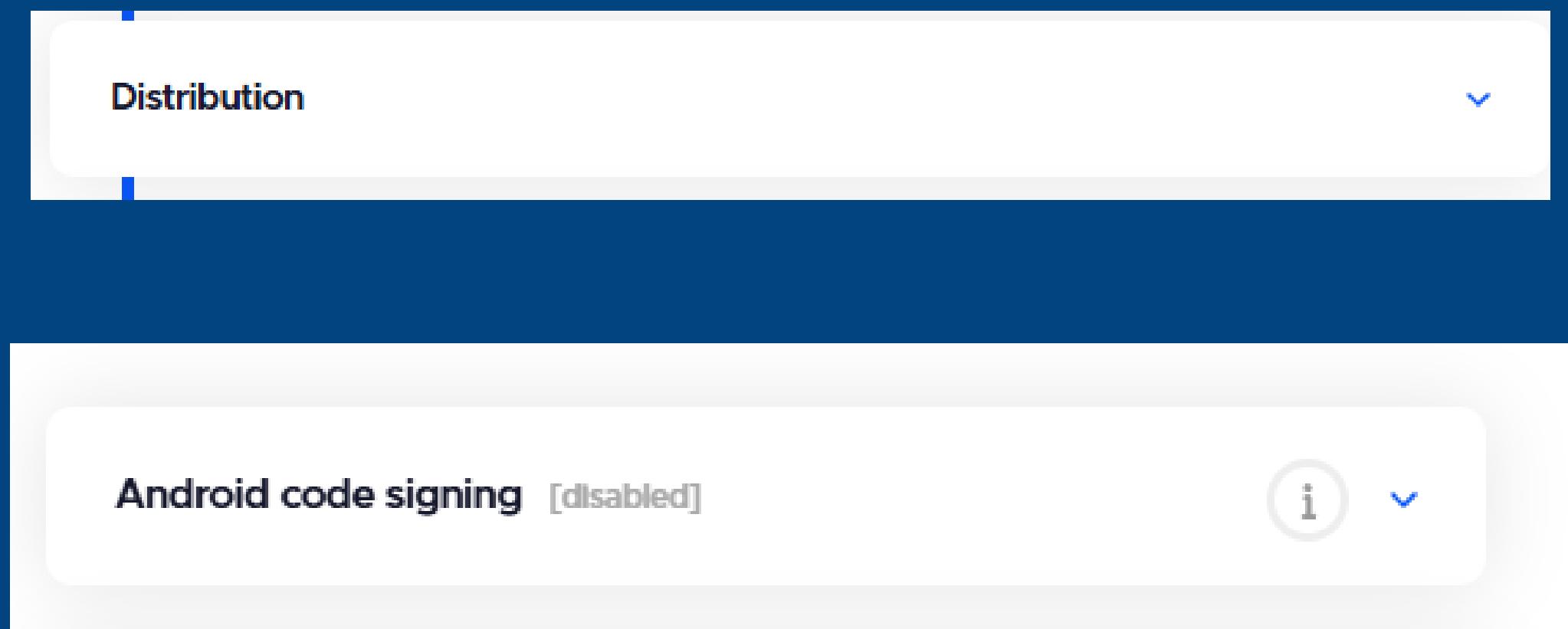
Automatic build triggering

Trigger on push Trigger on pull request update
 Trigger on tag creation Cancel outdated webhook builds

Step 3: (Build for Android)



Step 4: (Distribution for Android)



Set up Android code signing to enable installing your app on real devices and publishing it to Google Play.

Enable Android code signing

Keystore*

Choose a file or drag it here

Keystore password*

keystore password

Key alias*

key alias

Key password

key password

Step 5: (codemagic.yaml settings for Android)

codemagic.yaml settings

Code signing identities

Manage your personal account's [code signing identities](#) in one place and use them across all personal account applications.

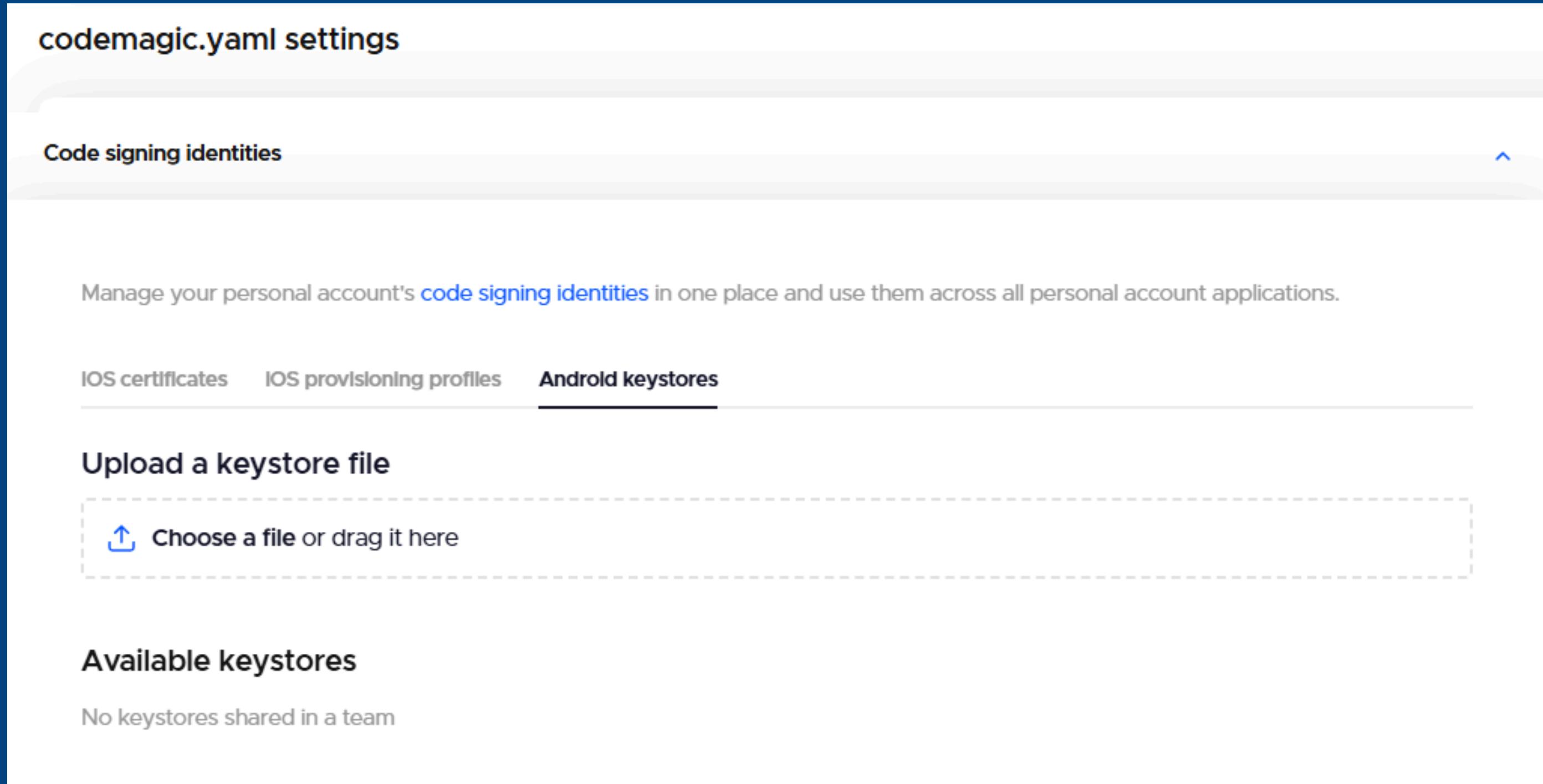
[iOS certificates](#) [iOS provisioning profiles](#) **Android keystores**

Upload a keystore file

 Choose a file or drag it here

Available keystores

No keystores shared in a team



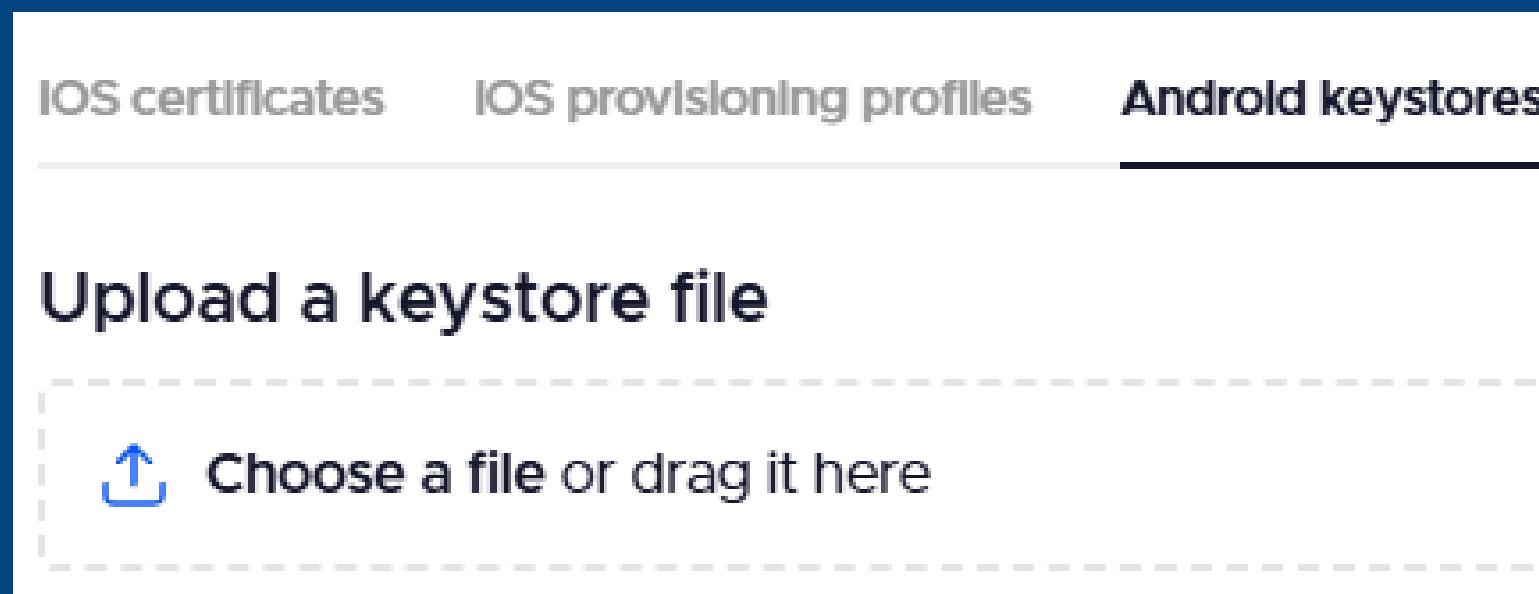
Step 6: (codemagic.yaml settings for Android)

Upload your keystore.jks

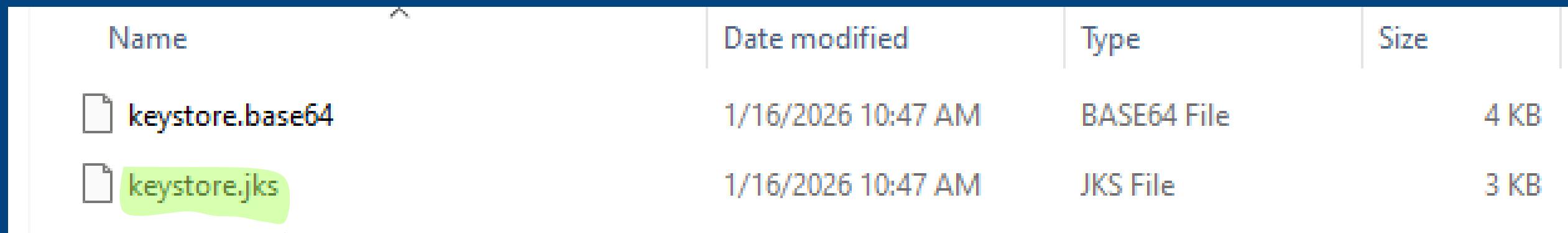
IOS certificates IOS provisioning profiles **Android keystores**

Upload a keystore file

Choose a file or drag it here



Name	Date modified	Type	Size
keystore.base64	1/16/2026 10:47 AM	BASE64 File	4 KB
keystore.jks	1/16/2026 10:47 AM	JKS File	3 KB



Step 7: (codemagic.yaml settings for Android)

Fill up the rest of your keystore data

The screenshot shows a user interface for managing Android keystores. At the top, there are three tabs: 'IOS certificates', 'IOS provisioning profiles', and 'Android keystores', with 'Android keystores' being the active tab. Below the tabs, there's a section titled 'Upload a keystore file' containing a text input field with the value 'keystore.jks'. To the right of this input field is a small 'X' button. The next section is 'Keystore password', which has a text input field with placeholder text 'Enter keystore password...'. Following that is 'Key alias', with a text input field placeholder 'Enter key alias...'. Below these is 'Key password' with an information icon (i) and a text input field placeholder 'Enter key password...'. Finally, there's 'Reference name' with an information icon (i) and a text input field placeholder 'Enter unique name...'. At the bottom left is a blue 'Add keystore' button, and at the bottom right is a grey 'Cancel' button.



IPT_PINTEREST_CLONE

Start your first build →

Step 9: (Start Build for Android)

Specify build configuration
Select which branch or tag to build

Build branch Build tag

Select branch
 ▼ ↻

Select workflow
 ▼

Enable SSH/VNC access i

Start new build → Cancel

Build Successful

The screenshot shows the codemagic app interface with a successful build summary.

Build Information:

- Project: **untitled1** (green checkmark)
- Repository: github.com/basillo-joseph-lee/untitled1
- Build ID: **6969a0d8a608241ea26fd002**
- Index: **3**
- Status: **finished**
- Workflow: **Default Workflow**
- Started by: manansalajosephlee@gmail.com
- Started: **3 hours ago**
- Duration: **4m 4s**
- Machine: **Mac mini M2**
- Branch: **main**
- Commit: [-o- adb6d28](#)

Artifacts:

- [**app-release.aab \[37.99 MB\]**](#)

Current configuration:

- Flutter channel: **stable**

Build Steps (Timeline):

Step	Time
Preparing build machine	22s
Fetching app sources	1s
Installing SDKs	46s
Installing dependencies	8s
Building Android	2m 41s
Publishing	3s
Cleaning up	< 1s

Left Sidebar:

- Applications**
- Builds**
- OTA Updates** (NEW)
- App Preview**
- Teams**
- Billing**

Bottom Left:

- All systems operational
- User preferences: manansalajosephlee@gmail.com
- Log out

Bottom Right:

- Feedback icon (blue speech bubble)



Thank you