Releasing App on Android

App Signing:

App Signing is like getting a product barcode for your app.

- It verifies your app's identity.
- Grant access to google play.
- Help store to prioritize security and user trust.
- Protects user from harmful software.
- Efficient key management is vital to ensure security of your app.

Two Methods for app signing:

1- **Play App Signing Service:**
   - Allows google to manage your app's key.
   - Prevents vital loss or compromise.

2- **Keystore files:**
   - This process provides you more control over the signing process.
   - Configures a keystore file in your app.
   - Safeguard your keystore file to prevent unauthorized access.

Generating a keystore

Run this command terminal:

```
keytool -genkey -v -keystore release-key.jks -keyalg RSA
-keysize 2048 -validity 100000 -alias key

android{
   signingConfigs{
        release{
           keyAlias keystoreProperties['keyAlias']
           keyPassword keystoreProperties['keyPassword']
           storeFile file(keystoreProperties['storeFile'])
           storePassword keystoreProperties['storePassword']
           }
    }
    buildTypes{
      release{
          signingConfig signingConfigs.release } }
flutter build appbundle –release
```

As a Best Practice make a backup of your keystore file.

AABs versus APKs:

Application App Bundle (AAB)	**Android Package Kit (APK)**
Distribute apps in google preferred format.	Represents the traditional app packaging format.
Helps Google Play generate APKs for your device.	Remains supported but is less practical.
Improves installation times.	Results in large file sizes.
Ensures a better user experience.	Impacts app performance.

Google Play Console:

- Helps Manage your App.
- Guide through the submission process.
- Sets up Meta data.
- Configures Distribution and Pricing.
- Offers Performance tracking and testing.
- Managing Updates, in-app purchases and user behavior.

CI/CD integration:

- Streamlines Development Workflow.
- Automates repetitive tasks.

CI integration :

- Merges updated code into a shared repository.
- Automates testing of each integration.
- CI tools like GitHub actions and Jenkins Run tests and Provides alerts.

CD integration:

- Automatically releases your app to users.

- Rolls out bug fixes and new features.
- App is deployed to google playstore.

Post-Release Management:

Actions:

- Monitor and optimize your app.
- Track app issues and feedback.
- Track user behavior, downloads and apps performance.
- Roll out new features and fix bugs.
- Make frequent app updates.