

/Android App Deployment

Android app deployment
with Fastlane + Github
Actions CI/CD



fastlane

GitHub

/TABLE OF CONTENTS



/01 /CI/CD

- > A brief overview of mobile CI/CD and some of the tools you can use.

/02 /FASTLANE

- > The easiest way to build and release mobile apps.

/03 /GITHUB ACTIONS

- > A Continuous integration and continuous delivery (CI/CD) platform.

/04 /DEMO

- > Set up a CI/CD pipeline for the Android app using fastlane and Github Actions.



/01

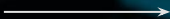
/CI/CD

A brief overview of mobile CI/CD considerations and share some of the tools you can use.





/WHAT IS (mobile) CI/CD?



<CI/CD>

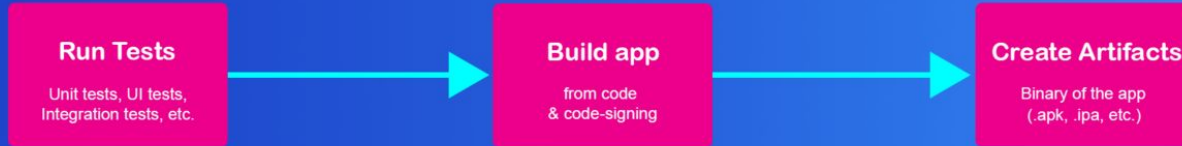
> CI/CD: One of the most important components of a modern DevOps workflow.

> CI/CD is a tool that helps to **automate** the tedious tasks, from building the app to deploying it into production.

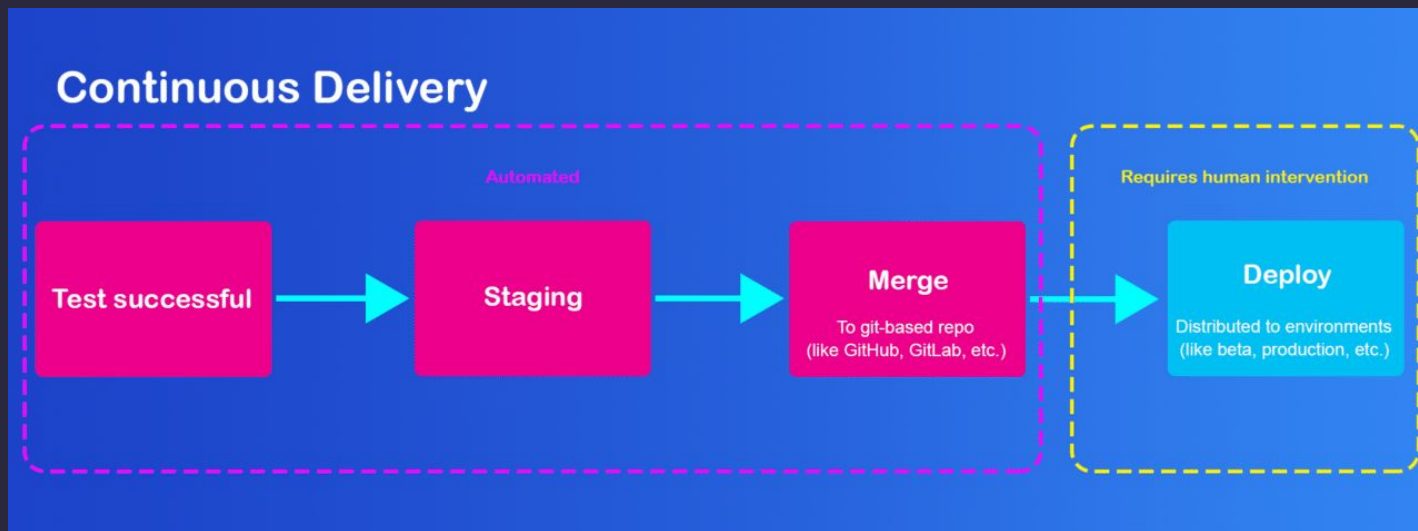


<Continuous Integration (CI)>

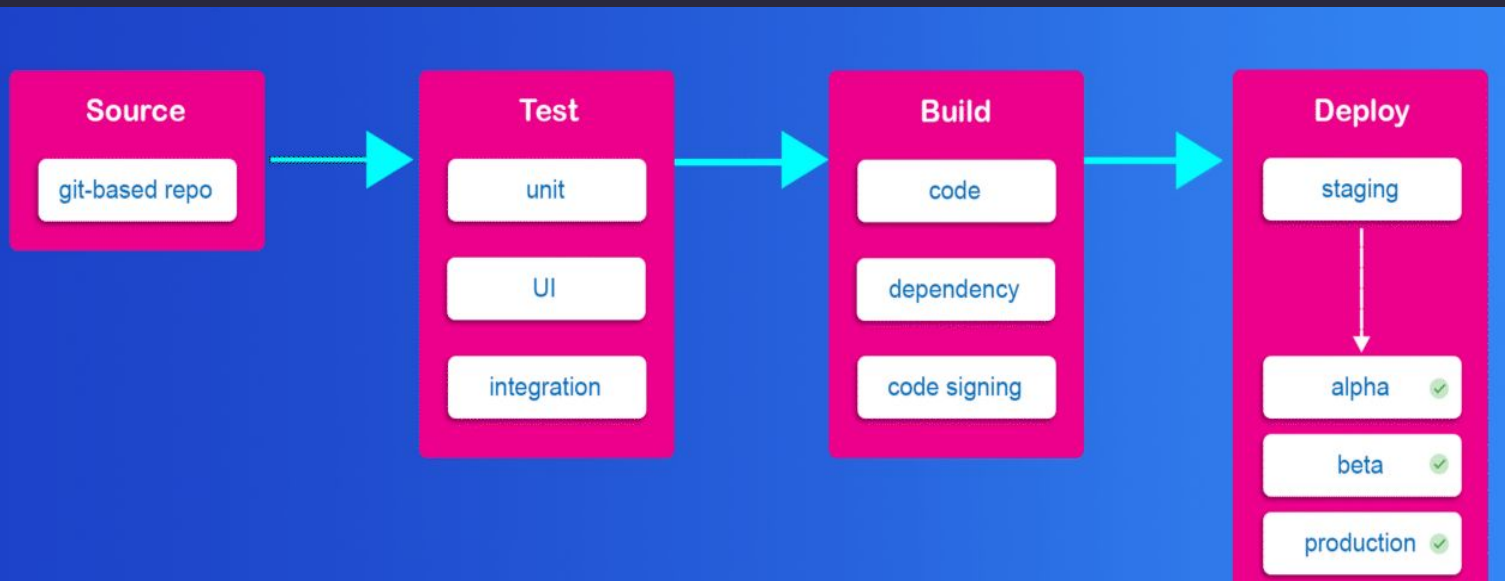
Continuous Integration

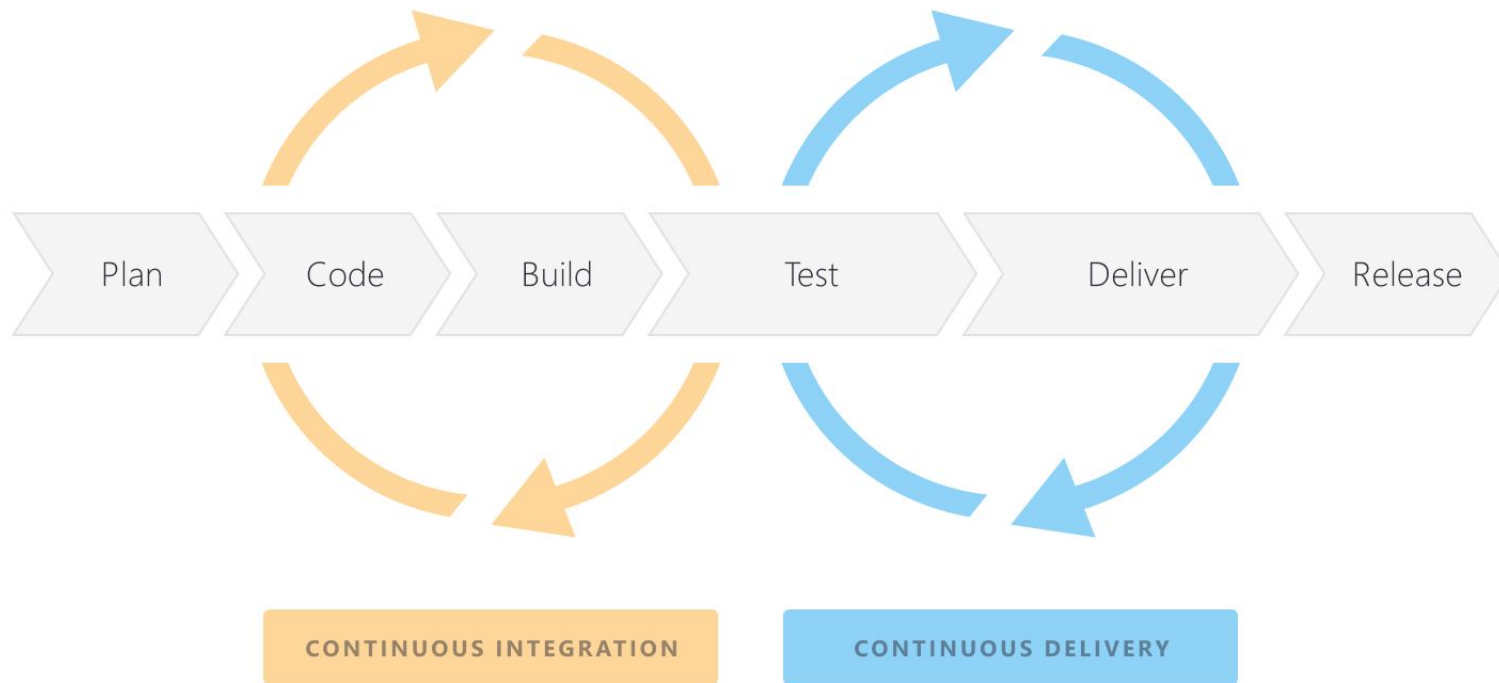


<Continuous Delivery (CD)>



<CI/CD Pipeline>







/WHY CI/CD is especially important for mobile development?



/Advantages of CI/CD



/SAVE TIME

Building & deploying for mobile → time-consuming process.

CI/CD: to reduce development and deployment time and effort → save up 20% time in a team.



/CONFIDENT IN YOUR BUILD

The code can be tested more regularly and robustly with CI/CD place.

→ fewer bugs and less time spent on bug fixes.

→ more feature updates and improvements.



Jenkins



CircleCI



Travis CI



Bitrise



Visual Studio App Center

/02

/FASTLANE

The easiest way to build and release mobile apps. Fastlane handles radius tasks so you don't have to.



fastlane



<Automate>



AUTOMATE SCREENSHOTS

Automatically generate
localized screenshots for
the app store

[LEARN MORE](#)

BETA DEPLOYMENT

Easily distribute beta builds
to testers

[LEARN MORE](#)

APP STORE DEPLOYMENT

Publish a new release to the
app store in seconds

[LEARN MORE](#)

CODE SIGNING

Reliably and consistently
code sign your app—no
more headaches

[LEARN MORE](#)

<Set up>

1. `fastlane init`
2. Enter your package name (`com.example.xyz`)
3. The prompt: Path to the JSON secret file
→ press Enter
4. The prompt: Do you plan on uploading ...
→ press n





```
default_platform(:android)

platform :android do
  desc "Runs all the tests"
  lane :test do
    gradle(task: "test")
  end

  desc "Submit a new Beta Build to Crashlytics Beta"
  lane :beta do
    gradle(task: "clean assembleRelease")
    crashlytics
  end

  desc "Deploy a new version to the Google Play"
  lane :deploy do
    gradle(task: "clean assembleRelease")
    upload_to_play_store
  end
end
```



◀Running Tests▶

```
desc "Runs all the tests"  
lane :test do  
  gradle(task: "test")  
end
```

Replace "test" with the gradle task name for running unit tests of your app.

fastlane tests



<Screenshots>

1. `sudo gem install screengrab`
2. `androidTestImplementation 'tools.fastlane:screengrab:x.x.x'`
3. Configuring your Manifest Permissions
4. Configuring your UI Tests for *screengrab*
5. Enter information in fastlane/screengrabfile
6. Generating Screenshots with Screengrab



```
desc "Build debug and test APK for screenshots. Then Capture Screenshots Automatically"
```

```
lane :grab_screens do
  gradle(
    task: "clean"
  )
  gradle(
    task: "assemble",
    build_type: "Debug"
  )
  gradle(
    task: "assemble",
    build_type: "AndroidTest"
  )
  screengrab
end
```



<Beta Deployment>

1. Beta Testing: a new feature ready → Sharing it with beta tester → **Firebase Distribution service.**
2. Play Store: Fastlane provides the upload_to_play_store action to upload metadata, screenshots and binaries to the Play Store.



<Firebase App Distribution>

1. Install the Firebase App Distribution plugin:

```
fastlane add_plugin firebase_app_distribution
```

2. Authenticate with Firebase:

```
fastlane run firebase_app_distribution_login
```


3. Add the firebase_app_distribution action to your lane:



<Firebase App Distribution>

```
lane :beta do
  # ...
  gradle(
    task: 'assemble',
    build_type: 'Release'
  )

  firebase_app_distribution(
    app: "1:123456789:android:abcd1234",
    groups: "qa-team, trusted-testers"
  )
  # ...
end
```



<Upload files to Google Drive>

1. Install the Google Drive plugin:

```
fastlane add_plugin google_drive
```

2. Add upload_to_google_drive to fastlane file:

```
upload_to_google_drive(  
  drive_keyfile: 'drive_key.json',  
  service_account: true,  
  folder_id: 'folder_id',  
  upload_files: ['file_to_upload', 'another_file_to_upload']  
)
```

<Upload files to Google Drive>

Reference:

https://github.com/huongntDev/fastlane-plugin-google_drive

<Google Play>

1. Setup Google credentials
2. Downloading/Uploading Metadata (if needed)
3. Add the `upload_to_play_store` action to your lane



<Google Play>

```
lane :beta do
  # ...
  gradle(
    task: 'assemble',
    build_type: 'Release'
  )
  upload_to_play_store(track: 'beta')
  # ...
end
```



<Download/Upload Metadata>

Metadata: including screenshots, descriptions and release notes

```
fastlane supply init
```

→ The **init** command downloads the existing metadata to **fastlane/metadata**.

```
fastlane supply --skip_upload_changelogs
```

→ This command without **init** to upload new **fastlane/metadata**.

<Release Deployment>

```
lane :playstore do
  # ...
  upload_to_play_store(
    track: 'rollout',
    rollout: '0.5'
  )
end
```



/03

/GITHUB ACTIONS

A Continuous integration and
continuous delivery (CI/CD)
platform.



<Overview>

1. Github Actions allows you to automate your build, test, and deployment pipeline.
2. The components of Github Actions are: **Workflows, Events, Jobs, Actions and Runners.**
3. Using: Create workflows that build and test every pull request to your repository, or deploy merged pull requests to production.



Event

Runner 1

Runner 2

Job 1

Step 1: Run action

Step 2: Run script

Step 3: Run script

Step 4: Run action

Job 2

Step 1: Run action

Step 2: Run script

Step 3: Run script



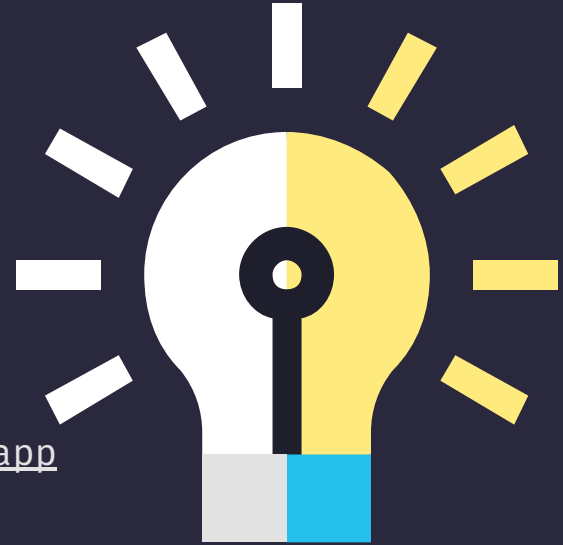
.github/workflows/learn-github-actions.yml

```
name: learn-github-actions
on: [push]
jobs:
  check-bats-version:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
        with:
          node-version: '14'
      - run: npm install -g bats
      - run: bats -v
```


/04

/DEMO

1. A perfect fastlane pipeline for Android
2. Set up a CI/CD pipeline for the Android app using fastlane and Github Actions



/RESOURCES

/DOCUMENTATIONS

- [CI/CD for mobile apps: The complete guide](#)
- [Fastlane tools](#)
- [Fastlane tutorial for Android](#)
- [How to build the perfect fastlane pipeline for Android](#)
- [How to setup a CI/CD pipeline for your Android app using fastlane and Github Actions.](#)

/PROJECTS

- [A perfect fastlane pipeline for Android](#)
- [Android App Deployment with Fastlane + Github Actions CI/CD](#)

/THANKS!

/DO YOU HAVE ANY QUESTIONS?

huongDeveloper2022@gmail.com

<https://twitter.com/HuongNguyenTha9>



CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**

> Please keep this slide for attribution

