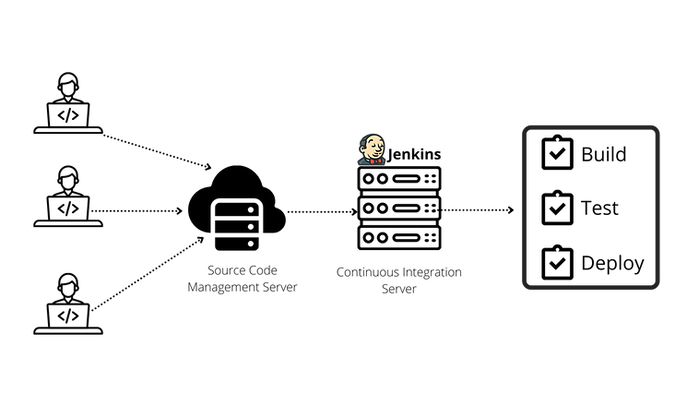
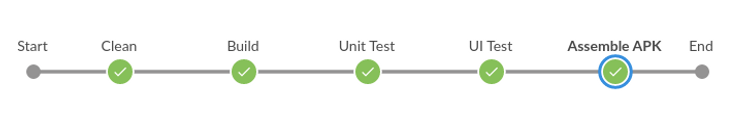
Jenkins CI/CD for Android App

Jenkins is an open source automation tool that allows continuous development, test and delivery of newly implemented code. The tool makes it easier for developer to integrate changes to the project. Jenkins provides both **Continuous Integration(CI)** and **Continuous Delivery(CD)** features.



**Jenkins Pipeline**

Jenkins pipeline is a collection of tasks that are responsible to maintain workflow. It can holds multiple steps for performing one task. Usually Jenkins executes tasks serially. We can control workflow of pipeline from code by Jenkinsfile.



Usually there are 4 tasks for building android apps in a pipeline:

1. **Clean** - It deletes the build directory and cleans the whole project
2. **Build** - It builds the project.
3. **Test** - It runs unit and UI test.
4. **Assemble Apk** - It prepares APK and uploads the APK to artifact or other cloud app store.

If any of the tasks gets error, all other tasks will not be executed. The following Jenkinsfile code will be worked only for building debug APK.

pipeline {

agent **any**

stages {

stage('Clean') {

steps {

sh './gradlew clean'

}

}

stage('Build') {

steps {

sh './gradlew buildDebug'

}

}

stage('Unit Test') {

steps {

sh './gradlew test'

}

}

stage('UI Test') {

steps {

sh './gradlew connectedAndroidTest'

}

}

stage('Assemble Apk') {

steps {

sh './gradlew assembleDebug'

}

}

}

post {

always{

archiveArtifacts artifacts: '\*\*/\*-debug.apk',

onlyIfSuccessful: **true**

}

}

}

The following Jenkinsfile code works for both debug and release APK.

class Constants {

static final String *DEBUG\_BUILD* = 'Debug'

static final String *RELEASE\_BUILD* = 'Release'

}

def isReleaseCandidate() {

*TAG* = "tag: ${env.TAG\_NAME}"

return *TAG* ==~ /tag: release-v.\*/

}

def getBuildType() {

if(!isReleaseCandidate()) {

return Constants.*DEBUG\_BUILD*

}

return Constants.*RELEASE\_BUILD*

}

pipeline {

agent any

environment {

*RELEASE\_KEY\_PASS* = credentials('releaseKeyPass')

*VARIANT* = getBuildType()

}

stages {

stage('Clean') {

steps {

script {

sh './gradlew clean'

}

}

}

stage('Build') {

steps {

script {

sh './gradlew build${VARIANT}'

}

}

}

stage('Unit Test') {

steps {

script {

sh "./gradlew test"

}

}

}

stage('UI Test') {

steps {

script {

sh './gradlew connectedAndroidTest'

}

}

}

stage('Assemble APK') {

steps {

script {

sh './gradlew assemble${VARIANT} -P releaseKeyPass=$RELEASE\_KEY\_PASS'

}

}

}

}

post {

always{

script {

archiveArtifacts *artifacts*: "\*\*/\*-${VARIANT.toLowerCase()}.apk",

*onlyIfSuccessful*: *true*

}

}

}

}

* [CI/CD](https://www.jndiary.com/my-blog/tags/ci-cd)
* [Jenkins](https://www.jndiary.com/my-blog/tags/jenkins)
* [Pipeline](https://www.jndiary.com/my-blog/tags/pipeline)
* [Jenkinsfile](https://www.jndiary.com/my-blog/tags/jenkinsfile)
* [Android Apk](https://www.jndiary.com/my-blog/tags/android-apk)
* [Android Development](https://www.jndiary.com/my-blog/categories/android-development)