[Travis CI 2](#_Toc432584700)

[Steps to use Travis CI: 2](#_Toc432584701)

[Writing .travis.yml file: 2](#_Toc432584702)

[.travis.yml in detail: 3](#_Toc432584703)

[Configuring .travis.yml 4](#_Toc432584704)

## Travis CI

Travis CI is a distributed build system for the open source community. It is a CI service used to build and test projects hosted on Github.

Travis is configured by adding a file named .travis.yml which is a YAML(a human-readable data serialization format) text file. It automatically detects when a commit has been made and pushed to a GitHub repository that is using Travis CI, and each time this happens, it will try to build the project and run tests. It also builds and runs pull requests and once it is completed it notifies the developer in the way it is configured.

## Steps to use Travis CI:

1. Sign-in: To get started with Travis CI we should sign-in to our github account.
2. Activate Github webhook: Once the Signup process gets completed we need to enable the service hook in the github profile page.
3. Add .travis.yml: We should add the yml file to the project.

## Writing .travis.yml file:

In order for Travis CI to build our project we need to tell the system a little bit about it. Add a file named .travis.yml to the root of our repository. The basic options in the .travis.yml should contain are language key which tells which language environment to select for our project and other options include the version of the language and scripts to run the tests, etc.

PB is an android project following android as language. Android project builds either use Maven or Gradle, depending on which build file is present in the repository and so in the case of PB, its a gradle build .

Following is the .travis.yml build file for PB Android studio project:

language: android  
jdk: oraclejdk7  
  
android:  
 components:  
 - android-22  
 - build-tools-22.0.1  
 - extra-android-m2repository  
  
sudo: false  
notifications:  
 email: false  
  
script:  
 - ./gradlew clean build

## .travis.yml in detail:

In the first line we have specified android as the language environment; the next specifies the version of java. Travis will then make sure that the project runs well with all the java versions specified.

language: android

Then comes the android SDK components. You can define the list of SDK components to be installed.

android:  
 components:  
 - android-22  
 - build-tools-22.0.1  
 - extra-android-m2repository

Jobs running on Travis CI’s newer container-based infrastructure has following features:

1. start up faster
2. allow the use of [caches](http://docs.travis-ci.com/user/caching) for public repositories
3. disallow the use of sudo, setuid and setgid executables

If you prefer to explicitly send your builds to container-based infrastructure use the following.

sudo: false

Travis CI Android builder assumes that your project is built with a JVM build tool like Maven or Gradle that will automatically pull down project dependencies before running tests without any effort on your side.

If your project has build.gradle file in the repository root, Gradle will be used to build it. By default it will use to run your test suite.

gradle build connectedCheck

If your project also includes the gradlew wrapper script in the repository root, Travis Android builder will try to use it instead. The default command will become:

./gradlew build connectedCheck

## Configuring .travis.yml

Build only specific branches:

Per default travis will build once we push to a branch on Github. That behavior can be annoying if we are committing often to development branches. We can restrict which branches Travis is monitoring for changes in the .travis.yml. Either we can blacklist the branches or whitelist them. This means that any commit on the dev and master branches will trigger a build.

# Whitelisting example  
branches:  
  only:  
    - master

- dev  
  
# Blacklisting example  
branches:  
  except:  
    - feature

Push your commit to your repository. That should add a build into one of the queues on Travis CI and your build will start as soon as one worker for your language is available.