<https://github.com/nukc/how-to-use-travis-ci>

**如何简单入门使用Travis-CI持续集成**

不知道你们有没有看过这样一个标识， 只看文字就可以看出这个项目是否已经构建成功（让大家知道项目没有问题），如果不成功则会显示 Build failing。 如果你的项目还没有使用，那么赶快跟我一起来装13吧。233333

**Travis-CI 是什么？**

Travis-CI是一个开源的持续构建项目，能够测试和部署；Travis-CI会同步你在GitHub上托管的项目，每当你Commit Push之后，就会在几分钟内开始按照你的要求测试部署你的项目。

目前Travis-CI分<http://travis-ci.org/>（GitHub公开项目进这个）和<http://travis-ci.com/>（私有付费项目）

官方文档：<https://docs.travis-ci.com/>

**开始使用 Travis-CI**

1：用你的GitHub账号登录[Travis-CI](https://travis-ci.org/auth)，确认接受访问GitHub的权限。

2：登录之后，Travis-CI就会同步你GitHub账号的仓库。然后打开[个人页面](https://travis-ci.org/profile)并给你想要构建的项目启用Travis-CI。

就像这样：[https://raw.githubusercontent.com/nukc/how-to-use-travis-ci/master/images/enable.png](https://raw.githubusercontent.com/nukc/how-to-use-travis-ci/master/images/enable.png)

3：添加 .travis.yml 文件到你项目根目录下，Travis-CI会按照.travis.yml里的内容进行构建。

如下是一个Android项目配置例子：

language: android

android:

components:

# Uncomment the lines below if you want to

# use the latest revision of Android SDK Tools

# - platform-tools

# - tools

# The BuildTools version used by your project

- build-tools-19.1.0

# The SDK version used to compile your project

- android-19

# Additional components

- extra-google-google\_play\_services

- extra-google-m2repository

- extra-android-m2repository

- addon-google\_apis-google-19

# Specify at least one system image,

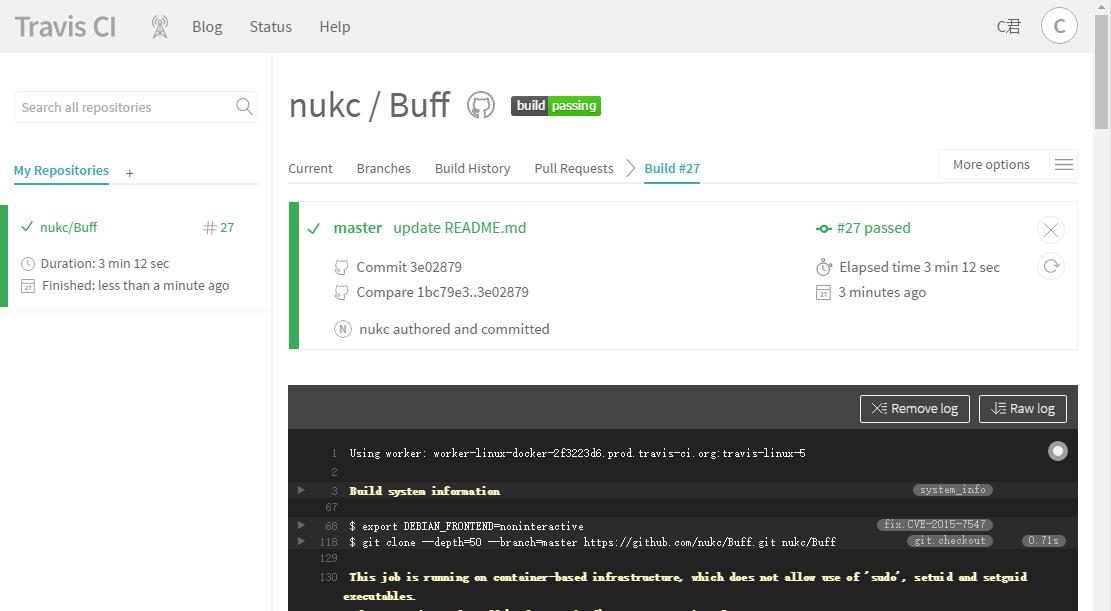
# if you need to run emulator(s) during your tests

- sys-img-armeabi-v7a-android-19

- sys-img-x86-android-17

4：把 .travis.yml push到你的GitHub上以触发Travis-CI进行构建。

5：最后你就可以到[构建状态页面](https://travis-ci.org/repositories)来查看你的项目是否构建成功。

[](https://raw.githubusercontent.com/nukc/how-to-use-travis-ci/master/images/build-status.png)

**配置构建脚本**

由于我也是刚入门不久，所以我只能讲解一些基础用法，如果有其他需求，大家可以自己慢慢探索。

先来看看下面这个我使用过的配置：

language: android # 声明构建语言环境

notifications: # 每次构建的时候是否通知，如果不想收到通知邮箱（个人感觉邮件贼烦），那就设置false吧

email: false

sudo: false # 开启基于容器的Travis CI任务，让编译效率更高。

android: # 配置信息

components:

- tools

- build-tools-23.0.2

- android-23

- extra-android-m2repository # Android Support Repository

- extra-android-support # Support Library

before\_install:

- chmod +x gradlew # 改变gradlew的访问权限

script: # 执行:下面的命令

- ./gradlew assembleRelease

before\_deploy: # 部署之前

# 使用 mv 命令进行修改apk文件的名字

- mv app/build/outputs/apk/app-release.apk app/build/outputs/apk/buff.apk

deploy: # 部署

provider: releases # 部署到GitHub Release，除此之外，Travis CI还支持发布到fir.im、AWS、Google App Engine等

api\_key: # 填写GitHub的token （Settings -> Personal access tokens -> Generate new token）

secure: 7f4dc45a19f742dce39cbe4d1e5852xxxxxxxxx

file: app/build/outputs/apk/buff.apk # 部署文件路径

skip\_cleanup: true # 设置为true以跳过清理,不然apk文件就会被清理

on: # 发布时机

tags: true # tags设置为true表示只有在有tag的情况下才部署

除了上面这些命令外，还有很多，比如：branches: （指定持续集成的分支）, install:（安装软件包）等待， 如果想进一步了解请到[Customizing Your Build](https://docs.travis-ci.com/user/customizing-the-build/)

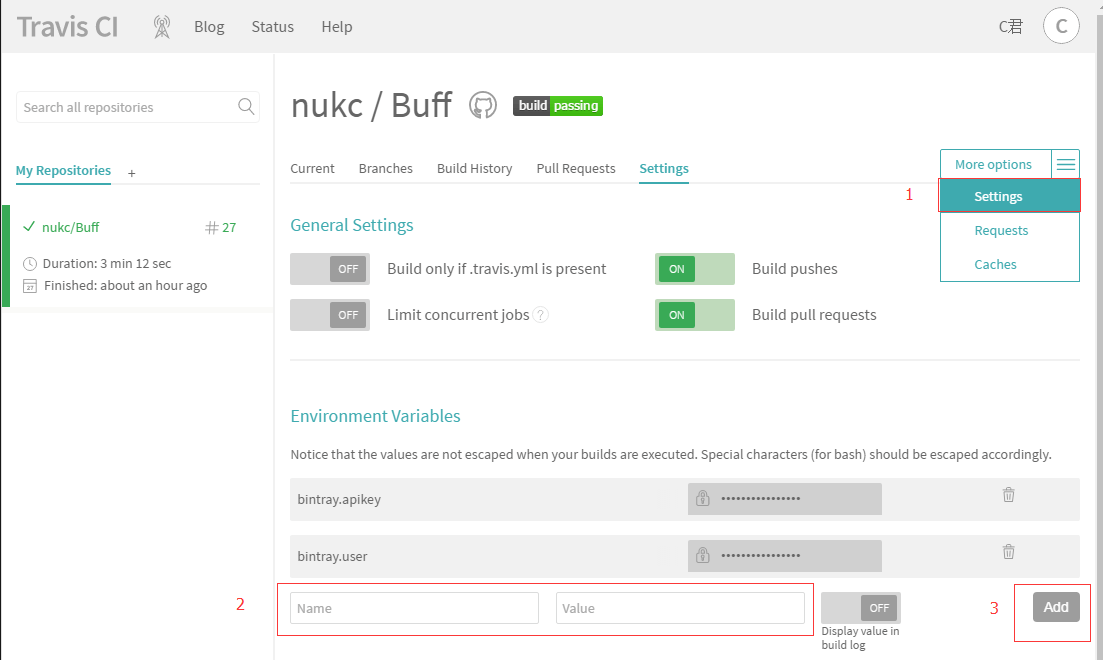
自动发布到GitHub Release：

[](https://raw.githubusercontent.com/nukc/how-to-use-travis-ci/master/images/releases.png)

**密码和证书安全**

对于密码等敏感信息，Travis CI提供了2种解决方案：

* 对密码等敏感信息进行加密，然后再构建环境时解密。
* 在Travis CI控制台设置环境变量，然后使用System.getenv()获取值。

[](https://raw.githubusercontent.com/nukc/how-to-use-travis-ci/master/images/var-setting.png)

考虑到安全问题，我们的签名设置可能是这样的:

release {

try {

storeFile file("nukc.jks")

storePassword KEYSTORE\_PASSWORD

keyAlias "C"

keyPassword KEY\_PASSWORD

} catch (ex) {

throw new Exception("You should define KEYSTORE\_PASSWORD and KEY\_PASSWORD in gradle.properties.")

}

}

KEYSTORE\_PASSWORD和KEY\_PASSWORD环境变量放在了gradle.properties文件，可以直接在build.gradle中使用 如果不知道是否能获取到gradle.properties里的环境变量，那就加个判断吧。（设置了会获取不到？比如.gitignore设置了忽略没有上传,判断一下也好） 然后最终就变成这样：

release {

try {

storeFile file("nukc.jks")

storePassword project.hasProperty("KEYSTORE\_PASSWORD") ? KEYSTORE\_PASSWORD : System.getenv("KEYSTORE\_PASSWORD")

keyAlias "C"

keyPassword project.hasProperty("KEY\_PASSWORD") ? KEY\_PASSWORD : System.getenv("KEY\_PASSWORD")

} catch (ex) {

throw new Exception("You should define KEYSTORE\_PASSWORD and KEY\_PASSWORD in gradle.properties.")

}

}

对于文件加密，Travis CI提供了一个基于ruby的CLI命令行工具，可以直接使用gem安装：

gem install travis

安装后进入安卓项目根目录，尝试对证书文件加密：

travis encrypt-file nukc.jks --add

如果首次运行，travis会提示需要登录，运行travis login --org并输入Github用户名密码即可。（付费版则为travis login --pro）

travis encrypt-file指令会做几件事情：

1. 在Travis CI控制台自动生成一对密钥: encrypted\_e6c55137b621\_key和encrypted\_e6c55137b621\_iv
2. 基于密钥通过openssl对文件进行加密，上例中会项目根目录生成xx.jks.enc文件
3. 在.travis.yml中自动生成Travis CI环境下解密文件的配置，上例运行后可以看到.travis.yml中多了几行：

before\_install:

- gem install fir-cli

- openssl aes-256-cbc -K $encrypted\_e6c55137b621\_key -iv $encrypted\_e6c55137b621\_iv

-in nukc.jks.enc -out nukc.jks -d

最后在.gitignore中忽略xx.jks以及gradle.properties

**项目使用了Bintray配置文件**

目前我的开源项目用不到证书签名，我还是比较关心Bintray apikey的安全，上传到JCenter还是比较实用的。 本来我把bintray.apikey和bintray.user设置在了local.properties中，但由于local.properties不会被上传， Travis CI无法获取到，那肯定是会build failing的。

之前：（报错，没有local.properties这样的文件）

Properties properties = new Properties()

properties.load(project.rootProject.file('local.properties').newDataInputStream())

bintray {

user = properties.getProperty("bintray.user")

key = properties.getProperty("bintray.apikey")

configurations = ['archives']

pkg {

repo = "maven"

name = "Buff"

websiteUrl = siteUrl

vcsUrl = gitUrl

licenses = ["MIT"]

publish = true

}

}

然后我把这2个变量添加到了Travis CI控制台，最后改一下。

Properties properties = new Properties()

boolean isHasFile = false

if (project.rootProject.findProject('local.properties') != null){

isHasFile = true

properties.load(project.rootProject.file('local.properties').newDataInputStream())

}

bintray {

user = isHasFile ? properties.getProperty("bintray.user") : System.getenv("bintray.user")

key = isHasFile ? properties.getProperty("bintray.apikey") : System.getenv("bintray.apikey")

configurations = ['archives']

pkg {

repo = "maven"

name = "Buff"

websiteUrl = siteUrl

vcsUrl = gitUrl

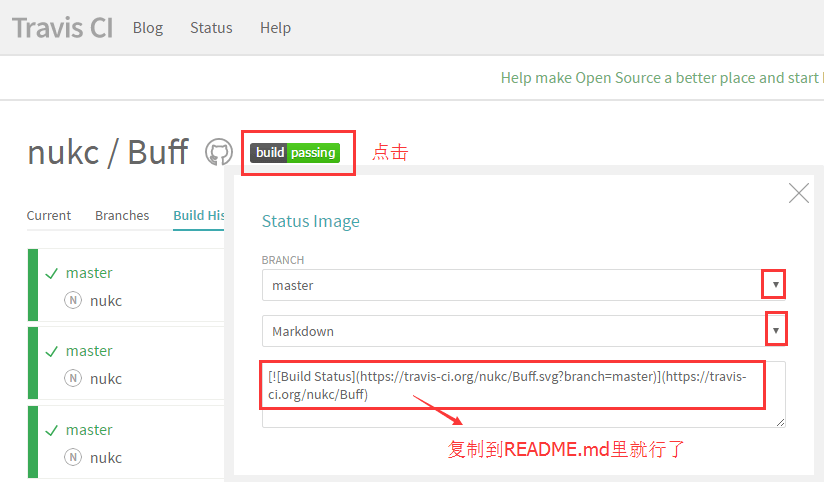
licenses = ["MIT"]

publish = true

}

}

**如何在自己的项目中显示Status Image**

[](https://raw.githubusercontent.com/nukc/how-to-use-travis-ci/master/images/image.png)

**其他技巧**

**跳过自动构建**

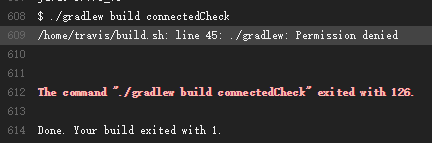
如果commit不想让travis ci构建，那么就在commit message里加上[ci skip]就行了。

git commit -m "[ci skip] commit message"

**爬过的坑**

如果你遇到了其他的问题，可以尝试到[travis-ci/issues](https://github.com/travis-ci/travis-ci/issues)里找找，或者Google / StackOverflow

/home/travis/build.sh: line 45: ./gradlew: Permission denied

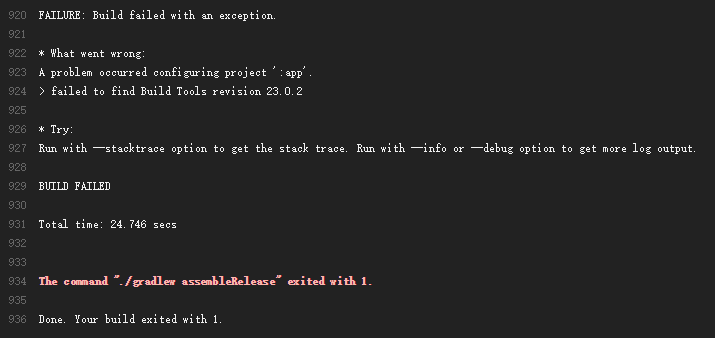
[](https://raw.githubusercontent.com/nukc/how-to-use-travis-ci/master/images/permission-denied.png)

gradlew的权限问题，修改gradlew的权限，在.travis.yml里加上：

before\_install:

- chmod +x gradlew

failed to find Build Tools revision 23.0.2

[](https://raw.githubusercontent.com/nukc/how-to-use-travis-ci/master/images/failed-to-find.png)

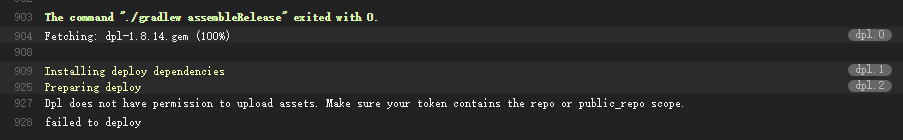
我是加上 - tools 解决的：

android:

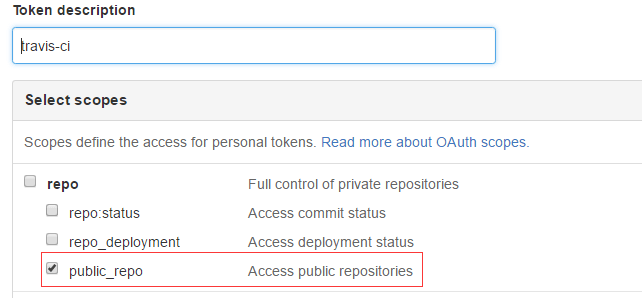
components:

- tools

failed to deploy

[](https://raw.githubusercontent.com/nukc/how-to-use-travis-ci/master/images/public_repo.png)

设置一下这个token的权限就好了

[](https://raw.githubusercontent.com/nukc/how-to-use-travis-ci/master/images/token-setting.png)

A problem occurred evaluating project ':app'. java.lang.UnsupportedClassVersionError: com/android/build/gradle/AppPlugin : Unsupported major.minor version 52.0

在 .travis.yml 里加上：

jdk:

- oraclejdk8

* What went wrong: A problem occurred evaluating project ':app'.

Failed to apply plugin [id 'com.android.application'] Could not create plugin of type 'AppPlugin'.

更新 gradle 版本就可以了

最后，希望大家都能顺顺利利的build passing。