

# Android GitLabCi的配置

近期接到了这样一个小任务，由于在测试过程中我们需要经常打包apk，这个任务量说大不大说小也不小吧，然后决定配置一个ci，让gitlab自动来做这件事情。

## 服务器环境

服务器:centos7

## 任务列表

- 安装open JDK
- 安装sdk
- 安装gradle
- 安装gitlab-ci-multi-runner

### 安装jdk

```
yum search java-1.8.0
yum install java-1.8.0-openjdk.x86_64
```

查看安装结果：

```
# java -version

openjdk version "1.8.0_131"
OpenJDK Runtime Environment (build 1.8.0_131-b12)
OpenJDK 64-Bit Server VM (build 25.131-b12, mixed mode)
```

配置环境变量： 在当前用户的bash\_profile或者/etc/profile配置一下环境变量和JAVA\_HOME eg:

```
export JAVA_HOME=/opt/soft/java
export CLASSPATH=:/lib:/jre/lib
export PATH=$PATH:$JAVA_HOME/bin
```

source profile 让配置立即生效

### 安装sdk

现在sdk都是由apkmanager统一管理的

下载sdktools

```
cd /opt

mkdir androidSdk

wget https://dl.google.com/android/repository/sdk-tools-linux-3859397.zip

unzip sdk-tools-linux-3859397.zip
```

在/opt/androidsdk/tools/bin下面有sdkmanager的命令，我们可以利用 `sdkmanager --list` 查看我们已经安装的内容，还可以用例如这样的命令 `sdkmanager build-tools;19.1.0`，来安装我们需要的内容。装完之后我们还是要习惯性的 `sdkmanager --licenses` 看一下有没有没有同意的licenses.

配置sdk环境变量

```
PATH=$PATH:/opt/androidsdk/tools/bin
PATH=$PATH:/opt/androidsdk/platform-tools
export PATH
export ANDROID_HOME=/opt/androidsdk
export ANDROID_NDK_HOME=/opt/androidsdk/ndk-bundle
```

source profile 让配置立即生效

运行 adb version

```
Android Debug Bridge version 1.0.39
Revision 3db08f2c6889-android
Installed as /opt/androidsdk/platform-tools/adb
```

如果看到以上内容，就代表成功了。

## 安装gradle

创建gradle目录解压并安装

```
cd /opt
mkdir gradle
cd gradle
wget https://services.gradle.org/distributions/gradle-4.0.1-bin.zip
unzip gradle-4.0.1-bin.zip
```

配置环境变量(可以在/etc/profile下面配置，也可以在用户的bash\_profile下面配置)

```
PATH=$PATH:/opt/androidsdk/tools/bin
PATH=$PATH:/opt/gradle/gradle-4.0.1/bin
PATH=$PATH:/opt/androidsdk/platform-tools
export PATH

export ANDROID_HOME=/opt/androidsdk
export ANDROID_NDK_HOME=/opt/androidsdk/ndk-bundle
```

检查是否配置成功

```
gradle -version
```

## 安装gitlab-ci-multi-runner

```
yum install gitlab-ci-multi-runner
```

注册runner:

```
$ sudo gitlab-ci-multi-runner register

Please enter the gitlab-ci coordinator URL (e.g. https://gitlab.com )
https://mygitlab.com/ci
Please enter the gitlab-ci token for this runner
xxx-xxx-xxx
Please enter the gitlab-ci description for this runner
my-runner
INFO[0034] fcf5c619 Registering runner... succeeded
Please enter the executor: shell, docker, docker-ssh, ssh[?]
docker
Please enter the Docker image (eg. ruby:2.1):
node:4.5.0
INFO[0037] Runner registered successfully. Feel free to start it, but if it's
running already the config should be automatically reloaded!
```

更新runner

```
# For Debian/Ubuntu
sudo apt-get update
sudo apt-get install gitlab-ci-multi-runner

# For CentOS
sudo yum update
sudo yum install gitlab-ci-multi-runner
```

配置文件默认会在 `/etc/gitlab-runner/config.toml`

简单配置：

```
concurrent = 1
check_interval = 0
```

示例配置文件：

```
[[runners]]
  name = "test"
  url = "http://your-domain.com/ci"
  token = "your-token"
  executor = "docker"
  [runners.docker]
    tls_verify = false
    image = "node:4.5.0"
    privileged = false
    disable_cache = false
    volumes = ["/cache"]
  [runners.cache]
  [runners.kubernetes]
    host = ""
    cert_file = ""
    key_file = ""
    ca_file = ""
    image = ""
    namespace = ""
    privileged = false
    cpus = ""
    memory = ""
    service_cpus = ""
    service_memory = ""
```

## 配置构建任务

在项目的根目录下，添加 `.gitlab-ci.yml` 文件，

```
image: openjdk:8-jdk
stages:
  - build
  - test
build:
  tags:
    - test-env
  stage: build
  script:
    - ./gradlew assembleDebug
  artifacts:
    paths:
      - app/build/outputs/
```

这个就是简单的，**build**一下打包出一个**debug**版本的**apk**，当然我们还可以编辑出各种脚本，我们也可以在这里编写代码的测试的，这个都是可以的。

这里面有一个 `tags:- test-env` 的概念，这个就是决定我们要选择哪一个**runner**。

效果图：

- 
- 

最后补充一下，我们本次是在**centos7**上面配置嘛，遇到了很多权限方面的问题，简单补充一下：



这里是分成9位的，前三位是自己的权限，中间三位是同组人的权限，最后三位是其它人的权限。我们可以根据自己的实际情况进行分配权限，最大就是 `chmod -777 xxx` 这个就是最高级别的权限了，就是谁都可以动的了，大家要合理决定是否给出这么大权限，当然如果这是一个文件夹我们还可以加一个 `-r` 参数，这样就会递归分配它下面的目录。

我们也可以通过 `chown -R --recursive` 把文件夹的所有者分配给别人。

## 参考

- [使用Gitlab搭建Android和iOS的持续集成和持续发布环境（二）](#)
- [Setting up GitLab CI for Android projects](#)
- [劈荆斩棘：Gitlab 部署 CI 持续集成](#)
- [GitLab-CI安装教程](#)
- [centos7中安装JDK](#)
- [centos7中安装Android SDK](#)
- [centos7中安装gradle](#)
- [gradle build提示You have not accepted the license agreements of the following SDK components](#)