<https://zhuanlan.zhihu.com/p/32659977>

**TarvisCI 全流程使用实践（二）**

[史强](https://www.zhihu.com/people/shi-qiang-13)[史强](https://www.zhihu.com/people/shi-qiang-13)

2 个月前

tarvisCI既然是一个虚拟机，那么我们就要想办法尽可能的利用它。

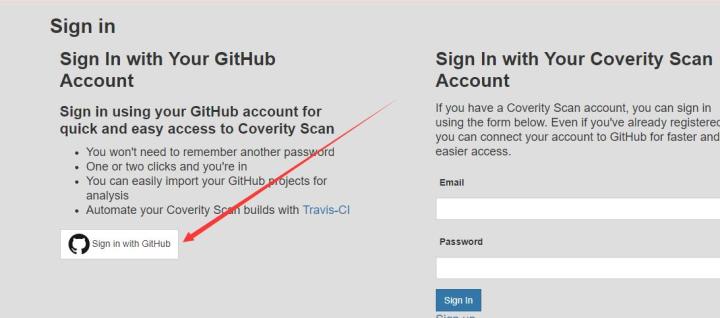
光编译还远远不能满足我的胃口。

那么，来点更棒的东西吧。

我有时候提交代码，经常有笔误的时候，这些小细节有时候不注意，往往累积到一定程度尅安定会爆发出来。于是，我想到了代码静态检查。

[https://scan.coverity.com​scan.coverity.comCoverity Scan - Static AnalysisCoverity Scan - Static Analysis​scan.coverity.com](http://link.zhihu.com/?target=https%3A//scan.coverity.com)

这个网站是和TarvisCI和github无缝集成的。它可以帮你审查你的代码中存在的问题。目前支持所有的主流语言。光C语法就支持规则上千种。

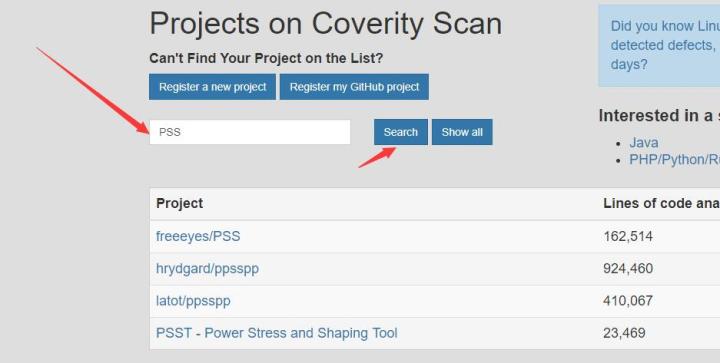


登录网站，在这里使用github账号绑定登录。

这样你可以看到你的github下的所有项目。选择添加工程

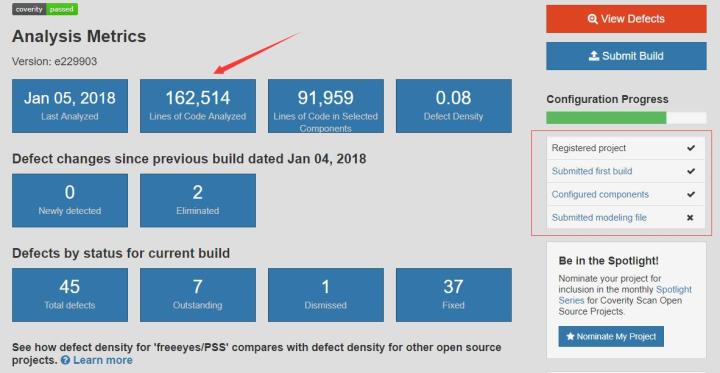


然后输出你的github项目地址后缀就行了。



这样，你的工程就建立了。

在这里，你需要做一些初始化工作。这个工具支持本地编译（下载它的一个巨大的工具包），你可以手动的在你的项目环境下安装测试。当然，这个有点low，既然我们有了免费的tarvis CI，那么就让它去帮我们搞好了。幸运的是，coverity本身就支持tarvisCI。



箭头是你当前项目的代码行数，这里包括了你所有的引用库的代码数量。也就是说，如果你不需要关注你使用的第三方库的代码缺陷，在这里你需要排除它们的路径。这样，代码缺陷检查就不会把扫描的这些缺陷记录在你的项目里面。

红框里的东西，我们必须设定一下，这里有一些要注意。

首先，要注意。如果选择tarvisCI的话，你一定要注意一个地方。



这里有一个secure:

这个一定要记录下来，因为tarvisCI需要这个TOKEN来授权coverity可以接收代码缺陷检查报告。coverity需要根据这个字段来定位，你的这个报告投递给谁。

这个TOKEN是一直有效的，如果你不喜欢，也不介意再生成一个新的。

然后我们来修改一下，tarvisCI的脚本。让它支持代码静态检查。

- os: linux

env:

- secure: "XXXXXXXXXXXXXX(这里是上面页面上显示的TOKEN)"

script:

- export

- echo -n | openssl s\_client -connect scan.coverity.com:443 | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' | sudo tee -a /etc/ssl/certs/ca-

- if [ "${COVERITY\_SCAN\_BRANCH}" != 1 ]; then

cd $TRAVIS\_BUILD\_DIR/purenessscopeserver/purenessscopeserver/PurenessScopeServer/Linux\_Bin;

$ACE\_ROOT/bin/mwc.pl -type gnuace pss.mwc;

make -j 4;

fi

addons:

coverity\_scan:

project:

name: "freeeyes/PSS"

description: "Build submitted via Travis CI"

notification\_email: XXXXX@163.com;XXXXXX@foxmail.com

build\_command\_prepend: "cd $TRAVIS\_BUILD\_DIR/purenessscopeserver/purenessscopeserver/PurenessScopeServer/Linux\_Bin;$ACE\_ROOT/bin/mwc.pl -type gnuace pss.mwc;make clean"

build\_command: "make -j 4"

branch\_pattern: master

这里添加一个addons:

name: 是你的项目在github上的位置。

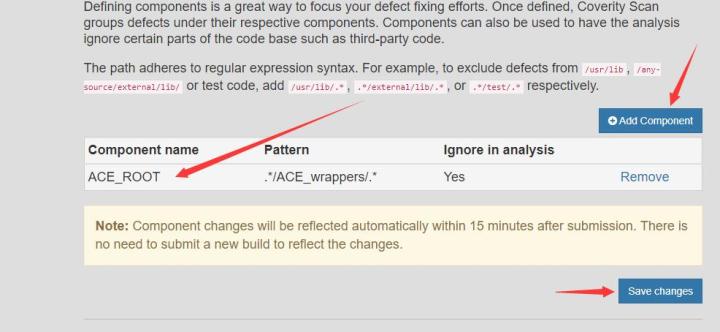
notification\_email: 是当代码缺陷检查完成后，会自动给你发邮件，这里记录你的接收邮箱地址。

branch\_pattern: 这里指定的是你分支。有时候我们不一定要此次都检测，可以在github上设置一个coverity\_scan分支。定期执行，不过我比较懒，喜欢每次提交都给我检测一下，所以直接绑定了master分支。

这里要特别说明一下，coverity免费版是有使用次数的，好像是一周28次，你用完了额度，就得等下周了。当然收费版没有这个限制。

有时候当你频繁提交的时候，会发现没有连续收到报告，当这种情况出现的时候，你去看一下tarvisCI的日志，有时候会有coverity提示。比如你提交的过于频繁啦，下一次执行在XXXX点之后执行等等。免费嘛，拿别人的手短，也就不用过于计较这些了。

最后，如何排除你代码中的第三方库代码呢？



在这个页面上，添加你的忽略路径，英文都很简单的，读一下就会用了。