链滴

使用 `pip` 将 Python module 安装到自定义目录

作者：StephenZhang
原文链接：https://ld246.com/article/1626454318326
来源网站：链滴

preface

偶尔在一些特殊的开发环境中，例如docker，会遇到通过pip3 install安装的python module无法持化的问题，如果这些module确实是项目所需，还可以联系docker的维护人员予以添加，否则可能要临重启一次docker就要重装一次的窘境

为了避免这种窘境，势必要找到一种方法将Python module安装到docker的持久化目录中，即docke启动时通过--volume或者-v选项挂载的目录，这样只需在重启docker后设置一下环境变量即可使用前安装的模块了

因此，这里介绍一种方法，和网上常见的修改site.py的方法不同，这种方法严格遵循Unix哲学，不会为修改了系统文件而引发各种问题

pip3 config

pip3命令原生支持通过pip.conf文件自定义用户设置，该文件具有类似.ini文件的结构，它的每个commands都可以是配置文件中的一个section，而section下每个field，都来自于pip3 command -h打出的选项，例如对于install命令而言，它的选项如下图：

```
Install Options:
  -r, --requirement <file>    Install from the given requirements
  -c, --constraint <file>     Constrain versions using the given
  --no-deps                   Don't install package dependencies.
  --pre                       Include pre-release and development
  -e, --editable <path/url>   Install a project in editable mode
  -t, --target <dir>          Install packages into <dir>. By de
                              existing packages in <dir> with new
  --platform <platform>       Only use wheels compatible with <pl
  --python-version <python_version>
                              The Python interpreter version to u
```

原文链接：使用 `pip` 将 Python module 安装到自定义目录

这里只是列举了一部分。这里的所有选项都可以去掉前面的--写入到配置文件中

继续向下查看，可以看到有一个选项叫做--prefix，如果是自己手动编译安装过软件的同学，看到这应该觉得比较熟悉吧，一般而言这个选项确实是用于配置安装路径的

接下来，创建属于当前用户的配置文件（均以linux为例）：

```
$ mkdir -p ~/.config/pip/pip.conf
```

然后，执行以下命令编辑配置文件：

```
$ pip3 config edit --editor vim
```

加上--editor的目的是为了指定编辑器，也可以去掉

将如下内容写入，指定~/local/作为安装路径：

```
[install]
prefix = ~/local/
```

保存退出后就可以使用pip3 install py_modules命令安装到自定义路径了

注意：

1. 如果要安装的Python module带有二进制可执行文件那么该文件会被放到 ~/local/bin/目录下，果此目录不在$PATH中，pip3将会在安装过程中弹出警告信息

2. Python module一般会被安装到 ~/local/lib/pythonX.Y/site-packages/目录下，你需要把这个录添加到环境变量$PYTHONPATH中，否则Python无法找到在此目录下的modules；此外，pythonXY代表你当前的Python的版本，例如python3.8.8，那么可得X=3, Y=8

docker的持久化目录

很多时候，开发用docker实例的持久化目录并不是用户的家目录，那么这种情况下，可以配置一个简的shell script，用于重启docker后配置环境

假设工作目录在docker中以及host中都被映射为/var/work/$USER

首先，可以把pip.conf的路径修改为/var/work/$USER/.config/pip/pip.conf，在docker启动后，使用n -s将其软链接到家目录中；同理，可以直接把local目录建立在工作目录中，例如/var/work/$USER/ocal/

接下来，我们希望这个脚本可以自动检查Python的版本号，Python的版本号可以通过python3 -V命获得，我们只需要写一个正则表达式，^Python (\d+\.\d+)\.\d+，并获取捕获到的Group 1即可

这样，准备工作已经完成了，下面开始写脚本，假设它的路径为/var/work/$USER/pyconfig.sh：

```
#!/bin/bash

python_version=$(python3 -V | grep -oP '\d+\.\d+\.\d+' | grep -oP '^\d+\.\d+')
user_python_path=/var/work/$USER/local/lib/"python$python_version"/site-packages

ln -sf /var/work/$USER/.config ~/.config
export PATH=$PATH:/var/work/$USER/local/bin
export PYTHONPATH=$user_python_path:$PYTHONPATH
```

这样在重启docker后，执行一下source pyconfig.sh即可重新使用之前配置好的Python modules

原文链接：使用 `pip` 将 Python module 安装到自定义目录