链滴

使用 pyenv 管理 Python 版本

作者：whitespur
原文链接：https://ld246.com/article/1540353206749
来源网站：链滴

参考：

http://einverne.github.io/post/2017/04/pyenv.html

Posted on 04/22/2017 by Ein Verne | View revision history

记录一下使用过程，留备以后使用。

pyenv 是 Python 版本管理工具。 pyenv 可以改变全局的 Python 版本，安装多个版本的 Python，设置目录级别的 Python 版本，还能创建和管理 virtual python environments 。所有的设置都是用级别的操作，不需要 sudo 命令。

pyenv 主要用来管理 Python 的版本，比如一个项目需要 Python 2.x ，一个项目需要 Python 3.x 。而 virtualenv 主要用来管理 Python 包的依赖，不同项目需要依赖的包版本不同，则需要使用虚拟环 。

pyenv 通过系统修改环境变量来实现 Python 不同版本的切换。而 virtualenv 通过将 Python 包安到一个目录来作为 Python 包虚拟环境，通过切换目录来实现不同包环境间的切换。

pyenv 的美好之处在于，它并没有使用将不同的 PATH植入不同的shell这种高耦合的工作方式，而是单地在PATH 的最前面插入了一个垫片路径（shims）：~/.pyenv/shims:/usr/local/bin:/usr/bin:/bi 。所有对 Python 可执行文件的查找都会首先被这个 shims 路径截获，从而使后方的系统路径失效。

安装之前

不同系统请参考 Common build problems，安装必须的工具。

pyenv 安装

根据官网的 安装说明 或者 自动安装 。 如果使用 Mac 直接使用 Homebrew。安装成功后记得在 .ba hrc 或者 .bash_profile 中添加三行来开启自动补全。

```
export PATH="$HOME/.pyenv/bin:$PATH"
eval "$(pyenv init -)"
eval "$(pyenv virtualenv-init -)"
```

根据自己的环境配置。

自动安装

pyenv 提供了自动安装的工具，执行命令安装即可：

```
curl -L https://raw.githubusercontent.com/yyuu/pyenv-installer/master/bin/pyenv-installer | b sh
```

保证系统有 git ，否则需要新安装 git。

手动安装

如果想要更加详细的了解安装过程，可以使用手动安装。将 pyenv 检出到你想安装的目录。建议路为：$HOME/.pyenv

```
cd ~
git clone git://github.com/yyuu/pyenv.git .pyenv
echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bashrc
echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bashrc
echo 'eval "$(pyenv init -)"' >> ~/.bashrc
source ~/.bashrc
```

添加环境变量。PYENV_ROOT 指向 pyenv 检出的根目录，并向 PATH 添加 `PYENV_ROOT/bin`以供访问 pyenv 命令的路径。

这里的 shell 配置文件（~/.bash_profile）依不同 Linux 而需作修改，如果使用 Zsh 则需要相应的置 ~/.zshrc

在使用 pyenv 之后使用 pip 安装的第三方模块会自动安装到当前使用 python 版本下，不会和系统块产生冲突。使用 pip 安装模块之后，如果没有生效，记得使用 pyenv rehash 来更新垫片路径。

pyenv 常用命令

使用 pyenv commands 显示所有可用命令

查看本机安装 Python 版本

使用如下命令查看本机安装版本

```
pyenv versions
```

星号表示当前正在使用的 Python 版本。使用 python -V 确认版本。

查看可安装 Python 版本

使用如下命令查看可安装版本

```
pyenv install -l
```

python 安装与卸载

```
$ pyenv install 2.7.3   # 安装 python
$ pyenv uninstall 2.7.3 # 卸载 python
```

python 切换

```
$ pyenv global 2.7.3  # 设置全局的 Python 版本，通过将版本号写入 ~/.pyenv/version 文件的方
。
$ pyenv local 2.7.3 # 设置 Python 本地版本，通过将版本号写入当前目录下的 .python-version 文
的方式。通过这种方式设置的 Python 版本优先级较 global 高。
```

python 优先级

shell > local > global

pyenv 会从当前目录开始向上逐级查找 .python-version 文件，直到根目录为止。若找不到，就用 gl
bal 版本。

原文链接：使用 pyenv 管理 Python 版本

```
$ pyenv shell 2.7.3 # 设置面向 shell 的 Python 版本，通过设置当前 shell 的 PYENV_VERSION 环变量的方式。这个版本的优先级比 local 和 global 都要高。–unset 参数可以用于取消当前 shell 设的版本。
$ pyenv shell --unset

$ pyenv rehash  # 创建垫片路径（为所有已安装的可执行文件创建 shims，如：~/.pyenv/versions/*/bin/*，因此，每当你增删了 Python 版本或带有可执行文件的包（如 pip）以后，都应该执行一次命令）
```

pyenv-virtualenv

pyenv 插件：pyenv-virtualenv

使用**自动安装 pyenv** 后，它会自动安装部分插件，通过 pyenv-virtualenv 插件可以很好的和 virtualenv 结合：

```
einverne@ev  ~  cd ~/.pyenv/plugins
einverne@ev  ~/.pyenv/plugins   master  ll
total 24K
drwxr-xr-x 4 einverne einverne 4.0K Apr 22 10:55 pyenv-doctor
drwxr-xr-x 5 einverne einverne 4.0K Apr 22 10:55 pyenv-installer
drwxr-xr-x 4 einverne einverne 4.0K Apr 22 10:55 pyenv-update
drwxr-xr-x 7 einverne einverne 4.0K Apr 22 10:55 pyenv-virtualenv
drwxr-xr-x 4 einverne einverne 4.0K Apr 22 10:55 pyenv-which-ext
drwxr-xr-x 5 einverne einverne 4.0K Apr 22 10:54 python-build
```

创建虚拟环境

```
$ pyenv virtualenv 2.7.10 env-2.7.10
```

若不指定 python 版本，会默认使用当前环境 python 版本。如果指定 Python 版本，则一定要是已安装过的版本，否则会出错。环境的真实目录位于 ~/.pyenv/versions 下

列出当前虚拟环境

```
pyenv virtualenvs
pyenv activate env-name  # 激活虚拟环境
pyenv deactivate #退出虚拟环境，回到系统环境
```

删除虚拟环境

```
pyenv uninstall my-virtual-env
rm -rf ~/.pyenv/versions/env-name  # 或者删除其真实目录
```

使用 pyenv 来管理 python，使用 pyenv-virtualenv 插件来管理多版本 python 包。此时，还需注，当我们将项目运行的 env 环境部署到生产环境时，由于我们的 python 包是依赖 python 的，需要意生产环境的 python 版本问题。

所有命令

```
$ pyenv commands
```

原文链接：使用 pyenv 管理 Python 版本

```
activate
commands
completions
deactivate
doctor
exec
global
help
hooks
init
install
installer
local
offline-installer
prefix
rehash
root
shell
shims
uninstall
update              # 更新 pyenv 及插件
version
--version
version-file
version-file-read
version-file-write
version-name
version-origin
versions
virtualenv
virtualenv-delete
virtualenv-init
virtualenv-prefix
virtualenvs
whence
which
```

PyCharm

PyCharm 中可以非常方便的切换 Python 环境非常方便。但是因为个人原因 Java 和 Python 同时用概率比较高，所以一直使用 Intellij IDEA 安装了 Python 插件。PyCharm 是为 Python 单独开发，以如果不需要在语言之间切换用 PyCharm 还是比较方便的。

Tips

更换 pip 源

因为国内网络环境，如果在局域网内下载 pip 慢，可以尝试使用 aliyun 提供的镜像，创建 vim ~/.pip pip.conf ，然后填入：

```
[global]
index-url = http://mirrors.aliyun.com/pypi/simple/
```

```
[install]
trusted-host=mirrors.aliyun.com
```

参考

- pyenv 下载地址　https://github.com/pyenv/pyenv
- virtualenv 中文文档地址　http://virtualenv-chinese-docs.readthedocs.io/en/latest/#
- http://my.oschina.net/lionets/blog/267469
- https://github.com/yyuu/pyenv-virtualenv
- http://seisman.info/python-pyenv.html