



链滴

记录 Linux 安装软件遇到的坑

作者: [zyk](#)

原文链接: <https://ld246.com/article/1583457656097>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



前言

个人还是喜欢在 Linux 环境中进行编程，因为在 Linux 中安装开发组件和配置环境比较方便，但有时经常会遇到一些坑，例如环境依赖问题等，所以特意写了这篇文章，把一些经历和心得记录下来，以不时之需。

常见问题

pip 安装 mysqlclient 报错

- 利用 pip 安装 MySQL 驱动时，会遇到以下的情况。

```

copying MySQLdb/constants/CR.py -> build/lib.linux-x86_64-3.6/MySQLdb/constants
copying MySQLdb/constants/ER.py -> build/lib.linux-x86_64-3.6/MySQLdb/constants
copying MySQLdb/constants/FIELD_TYPE.py -> build/lib.linux-x86_64-3.6/MySQLdb/constants
copying MySQLdb/constants/FLAG.py -> build/lib.linux-x86_64-3.6/MySQLdb/constants
running build_ext
building 'MySQLdb._mysql' extension
creating build/temp.linux-x86_64-3.6
creating build/temp.linux-x86_64-3.6/MySQLdb
x86_64-linux-gnu-gcc -pthread -DNDEBUG -g -fwrapv -O2 -Wall -g -fstack-protector-strong -Wformat -Werror=format-security -Wdate-time -D_FORTIFY_SOURCE=2 -fPIC -Dversion_info=(1,4,6,'final',0) -D__version__=1.4.6 -I/usr/include/mysql -I/usr/include/python3.6m -I/home/zyk/.virtualenvs/blog_api/include/python3.6m -c MySQLdb/_mysql.c -o build/temp.linux-x86_64-3.6/MySQLdb/_mysql.o
MySQLdb/_mysql.c:38:10: fatal error: Python.h: No such file or directory
   #include "Python.h"
          ^
compilation terminated.
error: command 'x86_64-linux-gnu-gcc' failed with exit status 1
-----
ERROR: Command errored out with exit status 1: /home/zyk/.virtualenvs/blog_api/bin/python3 -u -c 'import sys, setuptools, tokenize; sys.argv[0] = '"'"'/tmp/pip-install-f8ktvhma/mysqlclient/setup.py'"'"'; __file__ = '"'"'/tmp/pip-install-f8ktvhma/mysqlclient/setup.py'"'"'; f=getattr(tokenize, '"'"'open'"'"', open)(__file__); code=f.read().replace('"'"'\r\n'"'"', '"'"'\n'"'"'); f.close(); exec(compile(code, __file__, '"'"'exec'"'"'))' install --record /tmp/pip-record-nqrffzzi/install-record.txt --single-version-externally-managed --compile --install-headers /home/zyk/.virtualenvs/blog_api/include/site/python3.6/mysqlclient Check the logs for full command output.

```

- 可以通过以下命令行安装 dev 依赖来解决。

```
sudo apt-get install libmysqlclient-dev python3-dev
```

升级 pip 导致 pip 异常

- 升级 pip 以后出现 `ImportError:cannot import name 'main'` 错误。
- 可利用命令行 `sudo vim /usr/bin/pip3` 将原来的内容

```

from pip import main
if __name__ == '__main__':
    sys.exit(main())

```

- 修改为以下内容即可解决该问题：

```

from pip import __main__
if __name__ == '__main__':
    sys.exit(__main__.__main__())

```

npm 安装 node-sass 依赖报错

- npm 在安装 node-sass 时，会从 GitHub 上下载部分依赖，由于墙的存在，可能会导致下载超时。
- 可以通过设置淘宝镜像源来解决：

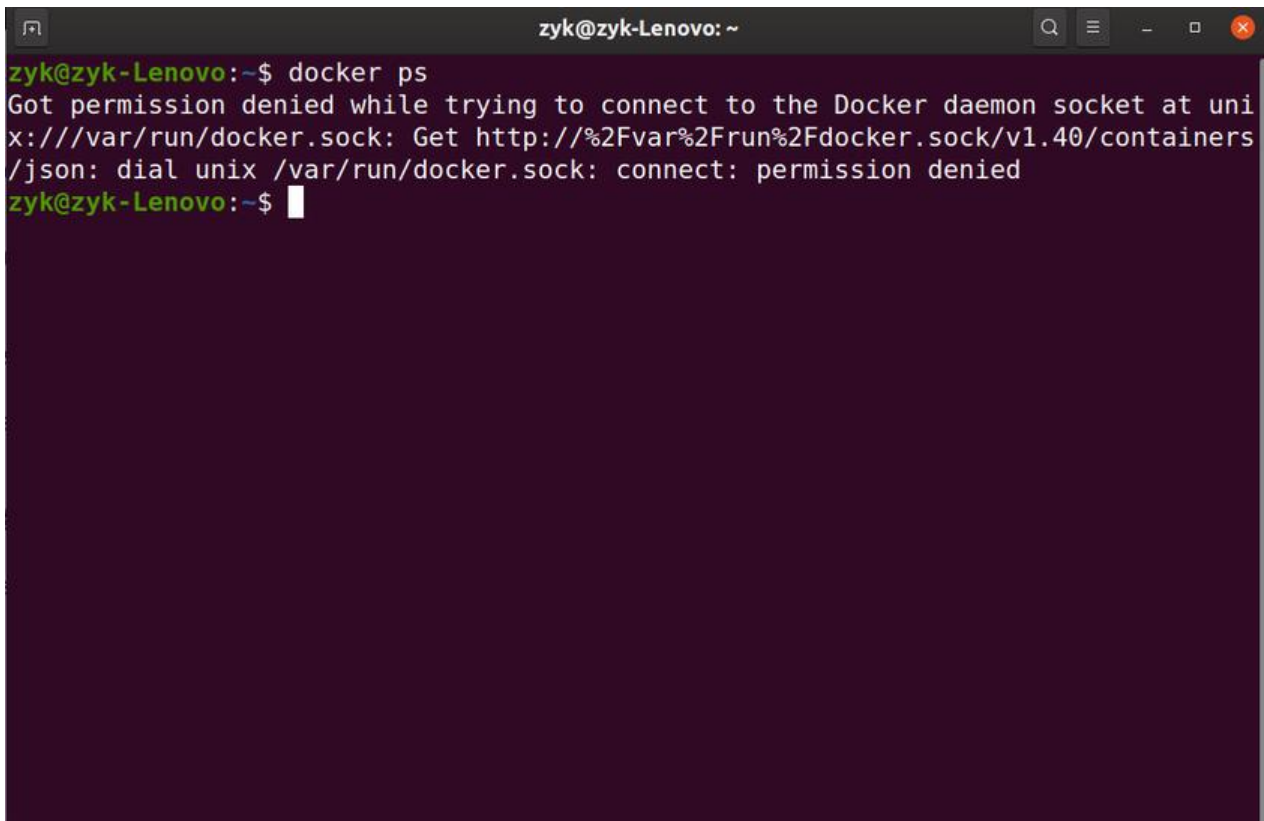
```
npm i node-sass --sass_binary_site=https://npm.taobao.org/mirrors/node-sass/
```

- 或者设置全局镜像源，再进行依赖安装：

```
npm config set sass_binary_site https://npm.taobao.org/mirrors/node-sass/
```

运行 docker 命令提示权限不足

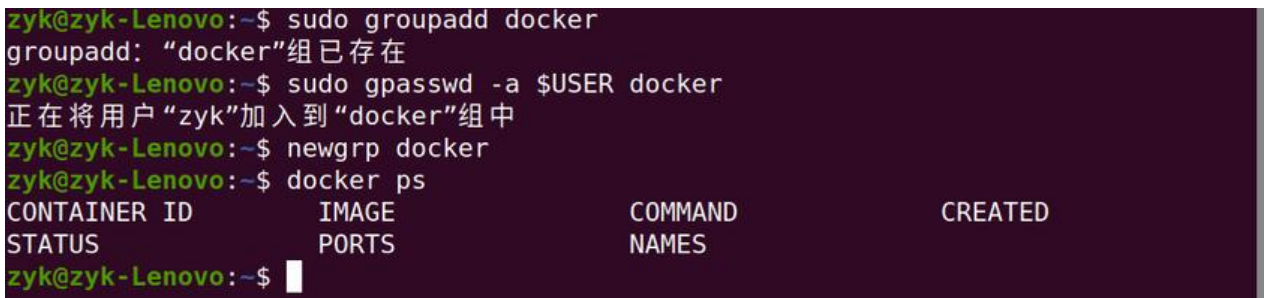
- 在安装 docker 之后运行 docker 命令，可能会遇到如下的情况，提示 Got permission denied:



```
zyk@zyk-Lenovo:~$ docker ps
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.40/containers/json: dial unix /var/run/docker.sock: connect: permission denied
zyk@zyk-Lenovo:~$
```

• 因为 docker 运行在 Unix Socket 端口上，默认只有 root 用户才有权限访问，因此需要将当前用户加入到 docker 用户组中，如此一来当前用户就有权访问 Unix Socket 端口了。

- 依次执行以下命令即可：



```
zyk@zyk-Lenovo:~$ sudo groupadd docker
groupadd: "docker"组已存在
zyk@zyk-Lenovo:~$ sudo gpasswd -a $USER docker
正在将用户“zyk”加入到“docker”组中
zyk@zyk-Lenovo:~$ newgrp docker
zyk@zyk-Lenovo:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS             PORTS              NAMES
```

```
sudo groupadd docker          #添加docker用户组
sudo gpasswd -a $USER docker  #将登陆用户加入到docker用户组中
newgrp docker                 #更新用户组
docker ps                     #测试docker命令是否可以使用sudo正常使用
```

docker pull 镜像速度太慢

- 用vim编辑 /etc/docker/daemon.json 文件:

```
sudo vim /etc/docker/daemon.json
```

- 向其中写入国内镜像源地址:

```
{
```

```
"registry-mirrors": [  
  "https://dockerhub.azk8s.cn",  
  "https://registry.docker-cn.com"  
]
```

- 保存退出后，重启docker服务：

```
sudo systemctl daemon-reload  
sudo systemctl restart docker
```