

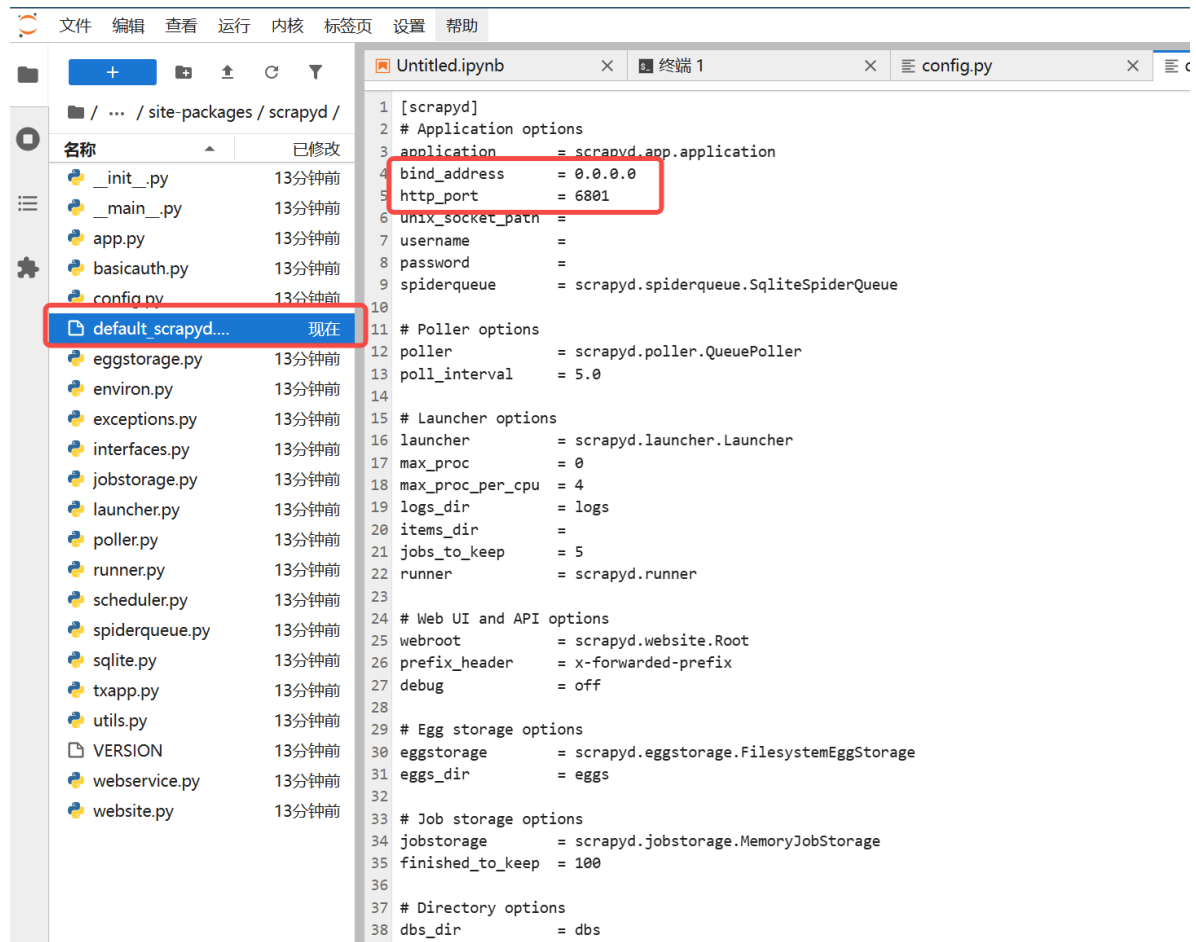
先装包

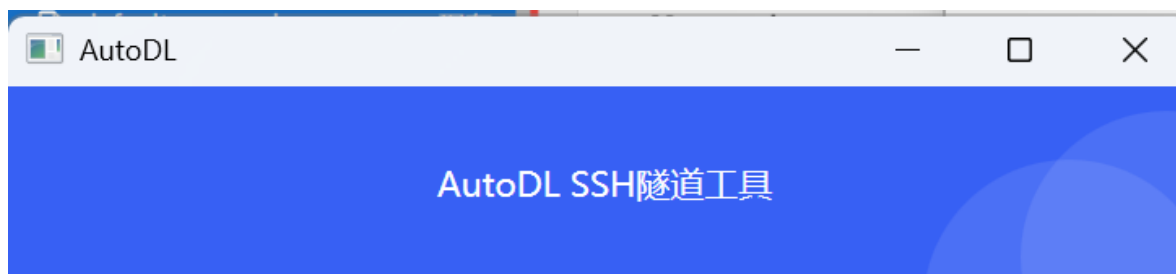
```
pip install scrapy
#scrapy部署工具
pip install scrapyd
#这个是把scrapyd接口封装了，方便使用
pip install python-scrapyd-api
#这个用来打包程序方便scrapyd部署
pip install scrapyd-client
pip install fake_headers
pip install scrapy_redis
pip install pymysql
```

该装的包没装会出现部署失败的可能

找到scrapyd的配置，改IP和端口

```
miniconda3/lib/python3.12/site-packages/scrapyd/default_scrapyd.conf
```





ssh指令:

`ssh -p 29428 root@connect.westc.gpuhub.com`

ssh密码:

.....

代理到本地端口:

6801

将远程端口代理到本地, 多个端口使用逗号分隔

代理到远程端口:

9000

将本地端口代理到远程, 多个端口使用逗号分隔

停止代理

在本地访问:

<http://127.0.0.1:6801>

在实例中执行:

`export https_proxy=http://127.0.0.1:9000`

在本地可以访问到服务器的scrapyd

Scrapyd

- [Jobs](#)
- [Logs](#)
- [Documentation](#)

No Scrapyd projects yet.


This web UI is for monitoring only. To upload projects and schedule crawls, use the API. For example, using [curl](#):

```
curl http://localhost:6800/schedule.json -d project=default -d spider=somespider
```

See the [Scrapyd documentation](#) for details.

主机二就可以添加了

127.0.0.1:8008/#/client/2/edit

 GERAPY

- 主机管理
- 项目管理
- 任务管理

编辑主机

* 名称

* IP

端口

认证 ☐

以上为该服务器配置方法

搞个阿里云服务器

超值优选 新人限时抢购

轻量云服务器 2核2G 38元/1年起

"99计划"助力开发者无忧上云

低价长效云服务器、云数据库皆99元/1年

开箱即用的云服务器和大模型服务平台百炼，轻松

每天两场（上午10:00，下午15:00），上云优选云服务器限量抢购，助力低成本上云 [活动规则](#)

轻量应用服务器 2核2G

上云优选

200M峰值带宽，适用于网站搭建、Web应用、容器环境、电商独立站等

购买时长

1年

轻量应用服务器 2核2G

200M峰值带宽，适用于网站搭建、Web应用、容器环境、电商独...

应用镜像 购买时长

宝塔Linux面板

▼

 1年

新人专享

¥68.00/1年起 [了解优惠](#)

官网折扣价: ¥459.00/1年

立即购买

轻量应用服务器 2核0.5G

200M峰值带宽，适用于账号安全管理、游戏加速

应用镜像 购买时长

宝塔Linux面板 1年

¥357.00/1年 [了解优惠](#)

官网折扣价: ¥357.00/1年

立即购买

防火墙规则开启，让服务器上的数据库外部可以访问到

阿里云

工作台

华北2 (北京)

Q 搜索

费用 ICP 备案 企业 支持 工单

81504

轻量应用服务器

服务器 / Docker-qdiy 运行中

远程连接 重置密码 续费 重启 关机 升级 重置系统 制作自定义镜像 退订

服务器概览 应用详情 域名 防火墙 安全防护 命令助手 快照 磁盘 监控 自助诊断 操作记录

1. 创建或修改防火墙规则时，限制IP来源可填写单个IP、CIDR段或者0.0.0.0/0(代表所有IPv4地址)，不填写默认为0.0.0.0/0，对所有IPv4地址都开放

2. 如果临时关闭某个端口，请使用禁用功能，免于开放对端口时再新建规则

3. 单个服务器最多可创建10条防火墙规则

4. 根据《中华人民共和国电信条例》及《电信业务经营许可管理办法》，开展递归解析服务需要具备相应的增值电信业务经营许可证 [查看详情](#)

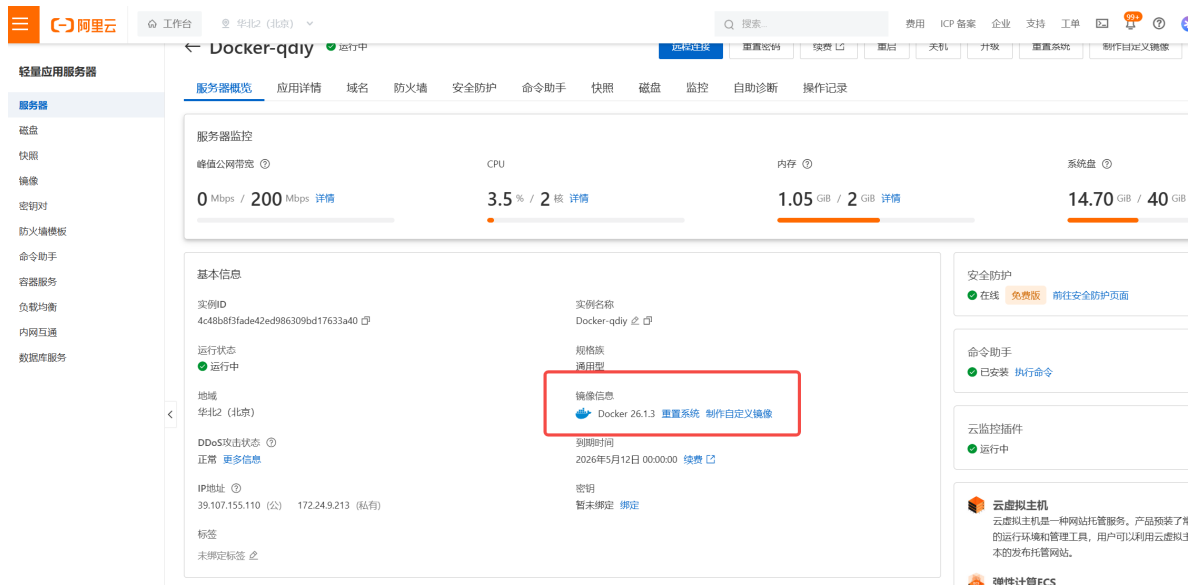
添加规则 应用防火墙模板

按启用状态排序

协议	端口范围	来源IP	备注	操作
<input type="checkbox"/> TCP	80	0.0.0.0/0		禁用 修改 删除
<input type="checkbox"/> TCP	443	0.0.0.0/0		禁用 修改 删除
<input type="checkbox"/> TCP	22	0.0.0.0/0		禁用 修改 删除
<input type="checkbox"/> ICMP	-1	0.0.0.0/0		禁用 修改 删除
<input type="checkbox"/> TCP	6802	0.0.0.0/0		禁用 修改 删除
<input type="checkbox"/> TCP	3306	0.0.0.0/0		禁用 修改 删除
<input type="checkbox"/> TCP	27017	0.0.0.0/0		禁用 修改 删除
<input type="checkbox"/> TCP	6379	0.0.0.0/0		禁用 修改 删除

共 8 条 < 1 > 10 条/页

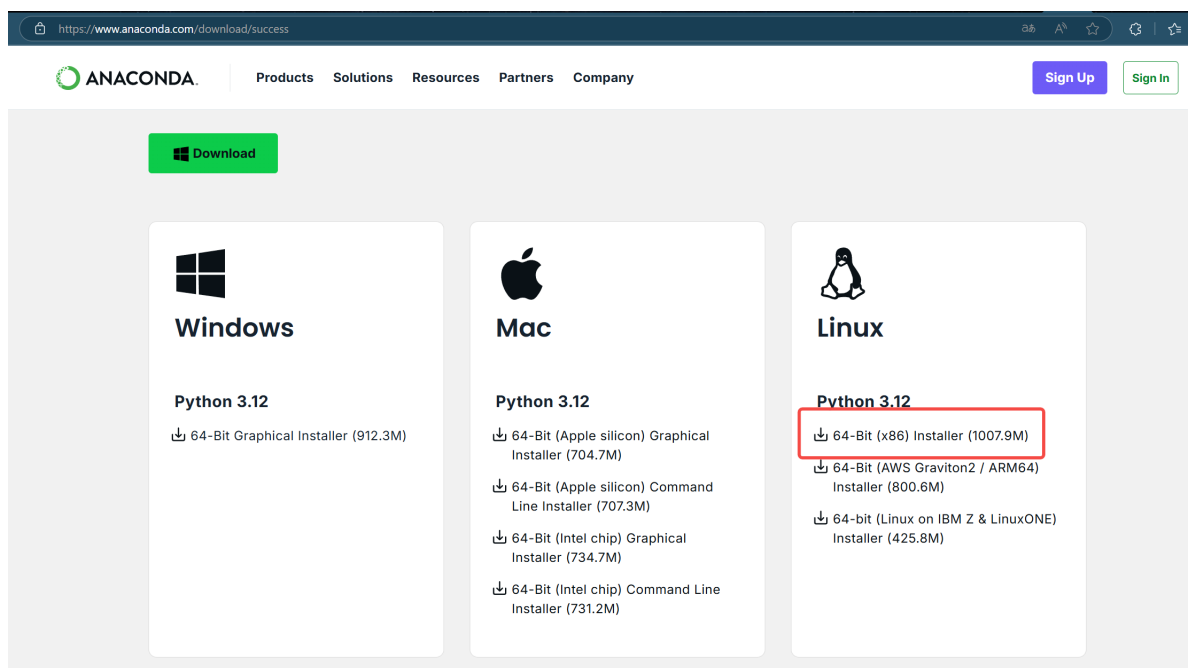
这里为了之后方便使用docker 干脆选择docker镜像



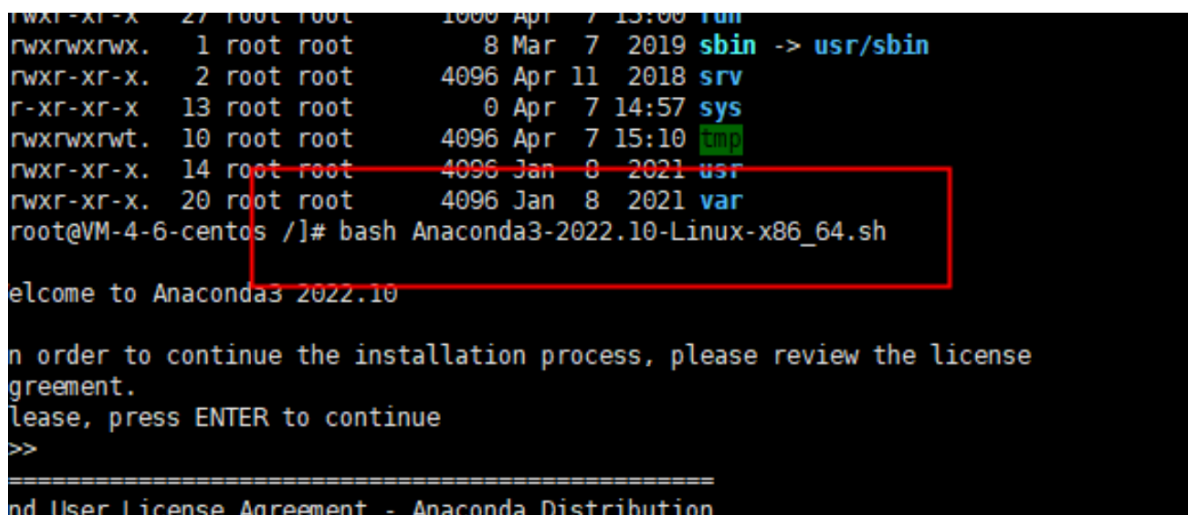
这里有个蛋疼的事，他上面只有python3.6

所以我们得自己整个python12

[Download Now | Anaconda](#)



然后把这玩意用fz传到服务器上，之后安装



按住enter 之后打个yes 如果停住继续按enter,等就行了

```
>>>
Please answer 'yes' or 'no':
>>>
Please answer 'yes' or 'no':
>>>
Please answer 'yes' or 'no':
>>>
Please answer 'yes' or 'no':
>>> yes

Anaconda3 will now be installed into this location:
/root/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/root/anaconda3] >>>
PREFIX=/root/anaconda3
Unpacking payload ...
Extracting : colorama-0.4.5-py39h06a4308_0.conda: 15%| 63/433 [00:05<00:46, 7.95it/s]
```

```
export PATH=/root/anaconda3/bin:$PATH
```

还要配置环境变量

```
[root@iZ2ze2ww4eolqooiamcdvsZ ~]# whereis bashrc
bashrc: /etc/bashrc
[root@iZ2ze2ww4eolqooiamcdvsZ ~]#
```

```

# if [ "$PS1" ]; then
#   PS1="\u@\h:\l \W\\$ "
# fi
# to your custom modification shell script in /etc/profile.d/ directory
fi

if ! shopt -q login_shell ; then # We're not a login shell
# Need to redefine pathmunge, it gets undefined at the end of /etc/profile
pathmunge () {
  case "${PATH}:" in
    *:"$1":*)
      ;;
    *)
      if [ "$2" = "after" ] ; then
        PATH=$PATH:$1
      else
        PATH=$1:$PATH
      fi
    esac
  }

# By default, we want umask to get set. This sets it for non-login shell.
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
  umask 002
else
  umask 022
fi

SHELL=/bin/bash
# Only display echos from profile.d scripts if we are no login shell
# and interactive - otherwise just process them to set envvars
for i in /etc/profile.d/*.sh; do
  if [ -r "$i" ]; then
    if [ "$PS1" ]; then
      . "$i"
    else
      . "$i" >/dev/null
    fi
  fi
done

unset i
unset -f pathmunge
fi

fi
# vim:ts=4:sw=4
export PATH=/root/anaconda3/bin:$PATH

```

可能需要重启下服务器

```

[root@iZ2ze2vw4eolqooiamcdvsZ ~]# python
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:27:36) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>

```

然后我们还要装docker-compose

#清除之前的残余

```
sudo rm /usr/local/bin/docker-compose
```

#下载安装

```
sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

#加上权限

```
sudo chmod +x /usr/local/bin/docker-compose
```

多个服务器自动化管理脚本

```
version: '3.8'
```

```
services:
  mongo:
    image: mongo:latest
    container_name: mongo
    environment:
      MONGO_INITDB_ROOT_USERNAME: root
      MONGO_INITDB_ROOT_PASSWORD: "123456"
    ports:
      - "27017:27017"
    volumes:
      - ./mongo_data:/data/db
    restart: always

  redis:
    image: redis:latest
    container_name: redis
    command: redis-server --requirepass "123456"
    ports:
      - "6379:6379"
    volumes:
      - ./redis_data:/data
    restart: always

  mysql:
    image: mysql:latest
    container_name: mysql
    environment:
      MYSQL_ROOT_PASSWORD: "123456"
    ports:
      - "3306:3306"
    volumes:
      - ./mysql_data:/var/lib/mysql
    restart: always
```

启动命令

```
docker-compose up -d
```


-d表示放到后台启动，可以先不要加，直接启动即可，如果想退出就ctrl+c就可以比较容易的退出

这是启动

```
170ccf42c3b: Pull complete
Digest: sha256:f496c25da703053a6e0717f1d52092205775304ea57535cc9fcaa6f35867800b
Status: Downloaded newer image for mysql:8.0
Creating mysql-docker ... done
Creating default_redis_1 ... done
Creating mongo ... done
Creating crawlab_master ... done
Creating worker ... done
```

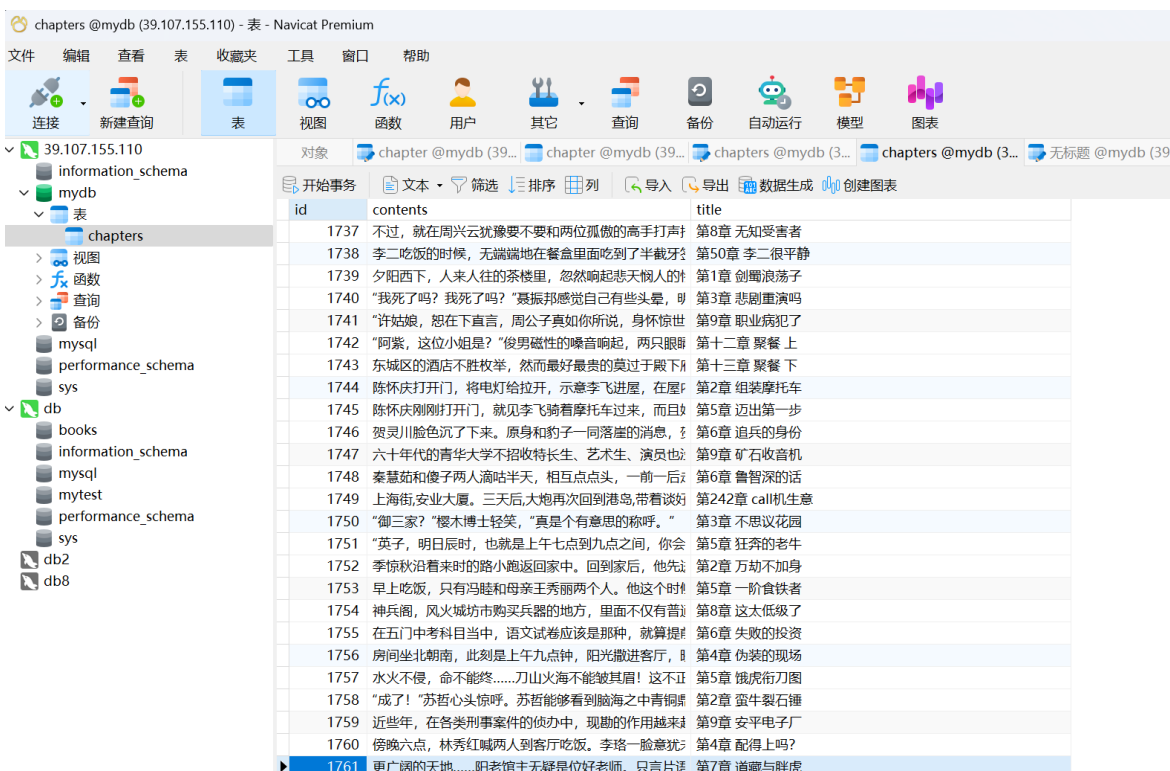
这是退出

```
Gracefully stopping... (press Ctrl+C a
Stopping worker ... done
Stopping crawlab_master ... done
Stopping mongo ... done
Stopping default_redis_1 ... done
Stopping mysql-docker ... done
```

启动状态下可以链接服务器上容器里的数据库了，用户名root密码是123456,这是再yaml文件里配置的

然后就可以远程连上了

之后就可以使用远程数据库了



如果要停止一切

可以用这个命令

docker-compose down

这里补充一个linux知识点

fg、bg、jobs、&、nohup、ctrl+z、ctrl+c 命令

一、&

加在一个命令的最后，可以把这个命令放到后台执行，如

watch -n 10 sh test.sh & #每10s在后台执行一次test.sh脚本

二、ctrl + z

可以将一个正在前台执行的命令放到后台，并且处于暂停状态。

三、jobs

查看当前有多少在后台运行的命令

jobs -l选项可显示所有任务的PID，jobs的状态可以是running, stopped, Terminated。但是如果任务被终

止了（kill），shell 从当前的shell环境已知的列表中删除任务的进程标识。

四、fg

将后台中的命令调至前台继续运行。如果后台中有多个命令，可以用fg %jobnumber（是命令编号，不是进程号）将选中的命令调出。

五、bg

将一个在后台暂停的命令，变成在后台继续执行。如果后台中有多个命令，可以用bg %jobnumber将选

中的命令调出。

六、kill

·法子1：通过jobs命令查看job号（假设为num），然后执行kill %num

·法子2：通过ps命令查看job的进程号（PID，假设为pid），然后执行kill pid

前台进程的终止：Ctrl+c

七、nohup如果让程序始终在后台执行，即使关闭当前的终端也执行（之前的&做不到），这时候需要nohup。该

命令可以在你退出帐户/关闭终端之后继续运行相应的进程。关闭中断后，在另一个终端jobs已经无法看

到后台跑得程序了，此时利用ps（进程查看命令）

查看进程是

ps -ef|grep 名字

a：显示当前终端下的所有进程信息，包括其他用户的进程。

u：使用以用户为主的格式输出进程信息。

x：显示当前用户在所有终端下的进程。

ps aux|grep 名字

a：显示当前终端下的所有进程信息，包括其他用户的进程。

u：使用以用户为主的格式输出进程信息。

x：显示当前用户在所有终端下的进程。

nohup scrapyd &