

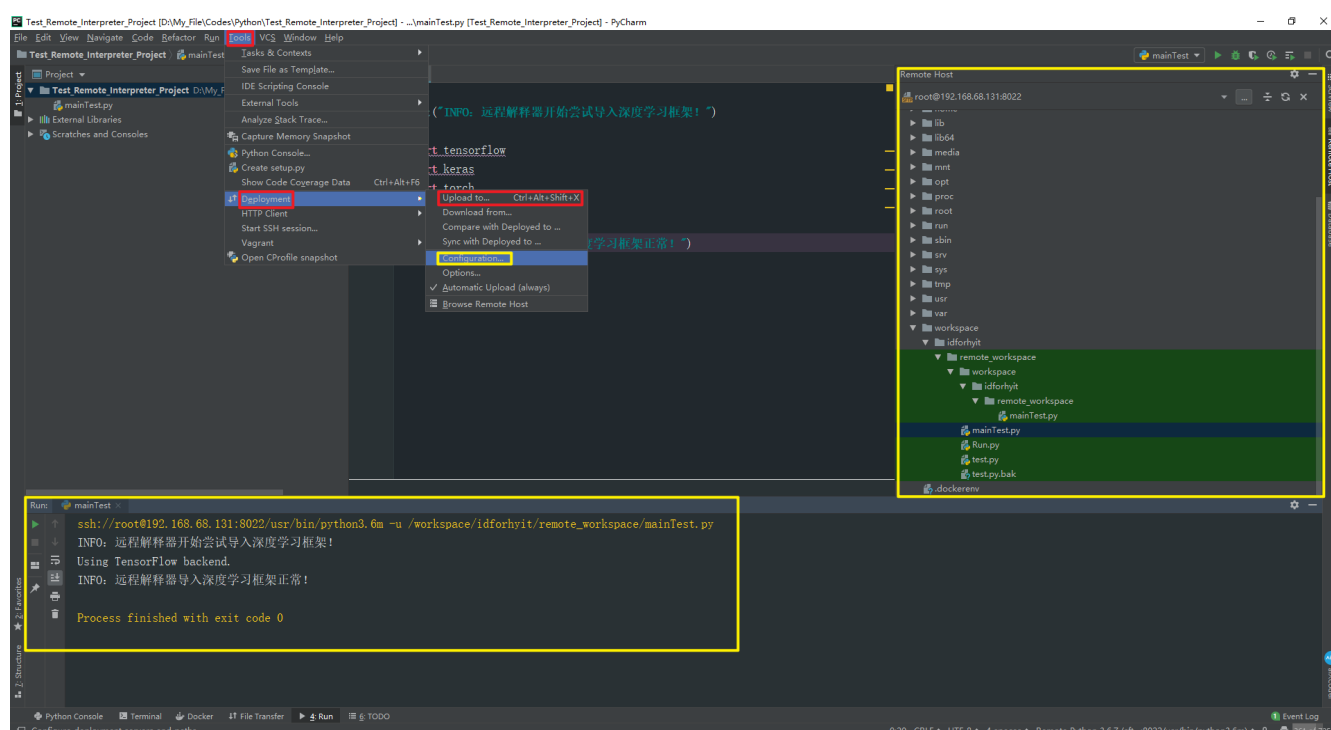
本文在线地址: <https://idforhyit.github.io/2019/06/01/Docker-GPU-RemoteInterpreter/>

# Pycharm+Docker配置备忘录

PyCharm+Docker: 打造最舒适的深度学习炼丹炉

当你配置完成之后, 你将能在本地PyCharm IDE上使用远程服务器的Python解释器和环境进行调试和运行, 同时能够很方便的将代码在本地和服务上进行同步。

本篇文章纯手码, 请勿转载。文章内容来自本菜鸟的亲自操作, 查阅了很多资料, 但作为菜鸟一枚, 肯定还存在许多错误, 大佬轻喷, 如见错误或操作不一致的地方, 烦请在[本篇博文地址](#)的文章底部评论指出, [戳我](#)直达。



PyCharm+Docker的远程解释器效果图

## 目录

### Pycharm+Docker配置备忘录

#### 目录

#### 流程命令总览

#### 详细步骤记录

##### 前期准备

以管理员权限运行

从DockerHub仓库下载一个深度学习作业系统的映像档

显示本机已有的映像档

查看目标端口占用 为下一步配置容器内的SSH作准备

##### 开始配置

新建docker container

守护态执行/后台运行方式新建容器

Docker Tips

配置SSH服务

先对目标容器环境进行更新

Docker换源三部曲

在容器内安装 `openssh-server` 并配置用户名和密码

验证配置

本地Pycharm配置

配置过程

大致配置过程如下：

具体过程截图说明：

从启动页进入设置（任何项目可用）

从项目页进入设置（默认仅配置项目可见）

如有错误或疑问，烦请评论指出

## 流程命令总览

```
1  # 以管理员权限运行
2  su root
3  [password]
4
5  # 从DockerHub仓库下载一个深度学习作业系统的映像档
6  docker pull hub-mirror.c.163.com/ufoym/deepo
7  ### 这里采用deepo，她几乎支持所有的深度学习框架，可以帮助俺们快速搭建一个包含TensorFlow、PyTorch、
   DarkNet等深度学习的研究环境。这里用的是CPU镜像，也可以采用GPU版本的 ###
8
9  # 显示本机已有的映像档
10 sudo docker images
11
12 # 查看目标端口占用 为下一步配置容器内的SSH作准备
13 lsof -i
14
15 # 新建docker container
16 # sudo nvidia-docker run -it -p 5592:5592 -p 5593:5593 -p 8022:22 --
   name="RemoteInterpreter_GPU" -v /host/data:/data -v /host/config:/config -v
   ~/workspace/idforhyit/remote_workspace:/workspace/idforhyit/remote_workspace
   ufoym/deepo:cpu /bin/bash
17 ### PS. 一定要留一个端口映射到容器22端口，因为SFTP默认使用22端口) ###
18
19 # 守护态执行/后台运行
20 sudo docker run -d -it -p 5592:5592 -p 5593:5593 -p 8022:22 --
   name="RemoteInterpreter_GPU_d" -v /host/data:/data -v /host/config:/config -v
   ~/workspace/idforhyit/remote_workspace:/workspace/idforhyit/remote_workspace
   ufoym/deepo:cpu /bin/bash
21 # 获取容器的输出信息
22 docker container logs RemoteInterpreter_GPU_d
23 # 进入容器 推荐用docker exec方式
24 docker exec -it RemoteInterpreter_GPU_d bash
25
26 # 查看已经安装包的命令以及版本
```

```

27 pip freeze
28 # 显示本机已有的容器（包括停止运行的）
29 docker ps -a
30 # 停用并删除全部容器：
31 docker stop $(docker ps -q) & docker rm $(docker ps -aq)
32
33 # 配置SSH服务
34 apt update
35 apt install -y openssh-server
36 mkdir /var/run/ssh
37 echo "user:password" | chpasswd # [用户名:密码] 按需设置并牢记
38 sed -i 's/PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config
39 sed 's@session\s*required\s*pam_loginuid.so@session optional pam_loginuid.so@g' -i
    /etc/pam.d/ssh
40 echo "export VISIBLE=now" >> /etc/profile
41 # 修改/etc/ssh/sshd-config以使可远程密码登录docker的root账户
42 ### 属性值
43 ### PermitRootLogin yes
44 ### PasswordAuthentication yes
45 service ssh restart # 重启SSH激活配置
46 exit #退出容器
47 sudo docker port RemoteInterpreter_GPU_d 22 # 0.0.0.0:8022 正确配置后的输出
48 ssh IDforHYIT@192.168.68.131 -p 8022 # 过程中确认的密码是你前面配置SSH设置的
49

```

```

1 # Docker换源
2
3 # 复制源文件备份，以防万一
4 cp /etc/apt/sources.list /etc/apt/sources.list.bak
5 # 查看新版本信息
6 lsb_release -c
7 ### 如果不是bionic 就将下列地址中的版本换为对应的 ###
8 ...
9 Ubuntu 12.04 (LTS)代号为precise, Ubuntu 14.04 (LTS)代号为trusty,
10 Ubuntu 15.04 代号为vivid, Ubuntu 15.10 代号为wily, Ubuntu 16.04 (LTS)代号为xenial
11 ...
12 # 编辑源列表文件
13 vim /etc/apt/sources.list
14 # 将原有的内容注释掉，添加以下内容（或者你把里面内容修改成下面的就可以，但是不能有除了以下内容的有效内容）
15 deb http://mirrors.aliyun.com/ubuntu/ bionic main restricted universe multiverse
16
17 deb-src http://mirrors.aliyun.com/ubuntu/ bionic main restricted universe multiverse
18
19 deb http://mirrors.aliyun.com/ubuntu/ bionic-security main restricted universe
    multiverse
20
21 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-security main restricted universe
    multiverse
22
23 deb http://mirrors.aliyun.com/ubuntu/ bionic-updates main restricted universe
    multiverse
24

```

```
25 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-updates main restricted universe
    multiverse
26
27 deb http://mirrors.aliyun.com/ubuntu/ bionic-backports main restricted universe
    multiverse
28
29 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-backports main restricted universe
    multiverse
30
31 deb http://mirrors.aliyun.com/ubuntu/ bionic-proposed main restricted universe
    multiverse
32
33 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-proposed main restricted universe
    multiverse
```

---

## 详细步骤记录

---

全文纯手码

## 前期准备

---

### 以管理员权限运行

```
1 su root
2 [password]
```

### 从DockerHub仓库下载一个深度学习作业系统的映像档

```
1 docker pull hub-mirror.c.163.com/ufoym/deepo
2 # 采用国内网易163镜像源 加快Pull速度
```

```
root@mu01:~  
[root@mu01 ~]# docker pull hub-mirror.c.163.com/ufoym/deepo  
Using default tag: latest  
latest: Pulling from ufoym/deepo  
898c46f3b1a1: Pull complete  
63366dfa0a50: Pull complete  
041d4cd74a92: Pull complete  
6e1bee0f8701: Pull complete  
112097260ef3: Pull complete  
30a67c795176: Pull complete  
0d286a7b6e12: Pull complete  
5d089cae17f6: Downloading [=====>  
55dd29113cda: Downloading [=====>  
5d089cae17f6: Downloading [=====>  
] 343.5MB/518.5MB=====>  
5d089cae17f6: Downloading [=====>  
] 344.1MB/518.5MB  
5d089cae17f6: Pull complete  
55dd29113cda: Pull complete  
987b8ae28a17: Pull complete  
0395fffb1efc: Downloading [=====>  
0395fffb1efc: Downloading [=====>  
0395fffb1efc: Downloading [=====>  
0395fffb1efc: Downloading [=====>  
0395fffb1efc: Downloading [=====>  
0395fffb1efc: Downloading [=====>  
0395fffb1efc: Pull complete  
Digest: sha256:b33a534b8b007e6f450c41caeb8dc74b339ec63fd8693f203
```

```
root@mu01:~  
[root@mu01 ~]# docker pull hub-mirror.c.163.com/ufoym/deepo  
Using default tag: latest  
latest: Pulling from ufoym/deepo  
Digest: sha256:b33a534b8b007e6f450c41caeb8dc74b339ec63fd8693f203b88558772009735  
Status: Image is up to date for hub-mirror.c.163.com/ufoym/deepo:latest  
[root@mu01 ~]#
```

**Note:** 这里采用 `deepo`，它几乎支持所有的深度学习框架，可以帮助俺们快速搭建一个包含 `TensorFlow`、`PyTorch`、`DarkNet` 等深度学习的研究环境。这里用的是GPU版本的Docker Image

## 显示本机已有的映像档

1 | `sudo docker images`

```
root@mu01:~# sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
tesnsorflow-zqy     latest             db2d943f813f       11 days ago        33.5GB
hub-mirror.c.163.com/ufoym/deepo latest             3256880e4c5b       5 weeks ago        10.6GB
caffe-dcx           latest             d13d93214ee3       6 weeks ago        712MB
tensorflow-huronglinpy27 latest             524ee2758bdd       11 months ago      3.89GB
tensorflow061202    latest             22b1dcae2f5        11 months ago      20.6GB
tensorflow0612      latest             0aaa68a3d052       11 months ago      16.4GB
tensorflow-huronglin latest             ecd115b00b26       11 months ago      3.85GB
<none>              <none>             c3d266586abe       11 months ago      13GB
tensorflow-xuwl3    latest             e2e80e90ac0e       11 months ago      16.4GB
tensorflow0604      latest             e2e80e90ac0e       11 months ago      16.4GB
tensorflow-0602     latest             e0ec40cf490b       12 months ago      12.2GB
<none>              <none>             038dadeefd44       12 months ago      3.87GB
tesnsorflow-zqy-bak latest             8b00efbae4be       12 months ago      16.6GB
12.12.12.100:5000/12.12.12.100 tensorflow2         195920e197c7       12 months ago      3.59GB
mxnet               latest             7ba1e22a795f       21 months ago      3.41GB
caffe2              latest             00a10af6cf18       21 months ago      3.59GB
cntk                latest             00a10af6cf18       21 months ago      3.59GB
tensorflow           latest             00a10af6cf18       21 months ago      3.59GB
theano              latest             00a10af6cf18       21 months ago      3.59GB
torch               latest             00a10af6cf18       21 months ago      3.59GB
12.12.12.100:5000/caffe latest             b3afe02fe6b6       21 months ago      3.87GB
caffe               latest             b3afe02fe6b6       21 months ago      3.87GB
registry            latest             c2a449c9f834       23 months ago      33.2MB
[root@mu01 ~]#
```

## 查看目标端口占用 为下一步配置容器内的SSH作准备

1 | `netstat -tunlp | grep 8022` # 8022是拟启用端口，用来转发docker容器SFTP的22端口

- Linux系统中，没返回信息就是最好的信息，2333

```
root@mu01:~  
[root@mu01 ~]# netstat -tunlp | grep 8022  
[root@mu01 ~]#
```

- 端口占用返回界面，以查看80端口使用情况为例，返回建立连接的程序及监听状态

```
root@mu01:~  
[root@mu01 ~]# netstat -ntulp | grep 80  
tcp        0      0 127.0.0.1:2380      0.0.0.0:*           LISTEN      2281/etcd  
tcp6       0      0 :::8009             :::*                LISTEN      6471/java  
tcp6       0      0 :::8080             :::*                LISTEN      6471/java  
tcp6       0      0 :::32980            :::*                LISTEN      8732/docker-proxy  
tcp6       0      0 :::32833            :::*                LISTEN      338980/docker-proxy  
tcp6       0      0 127.0.0.1:8005      :::*                LISTEN      6471/java  
[root@mu01 ~]#
```

## 开始配置

命令参考: <https://hub.docker.com/r/ufoym/deepo#GPU>

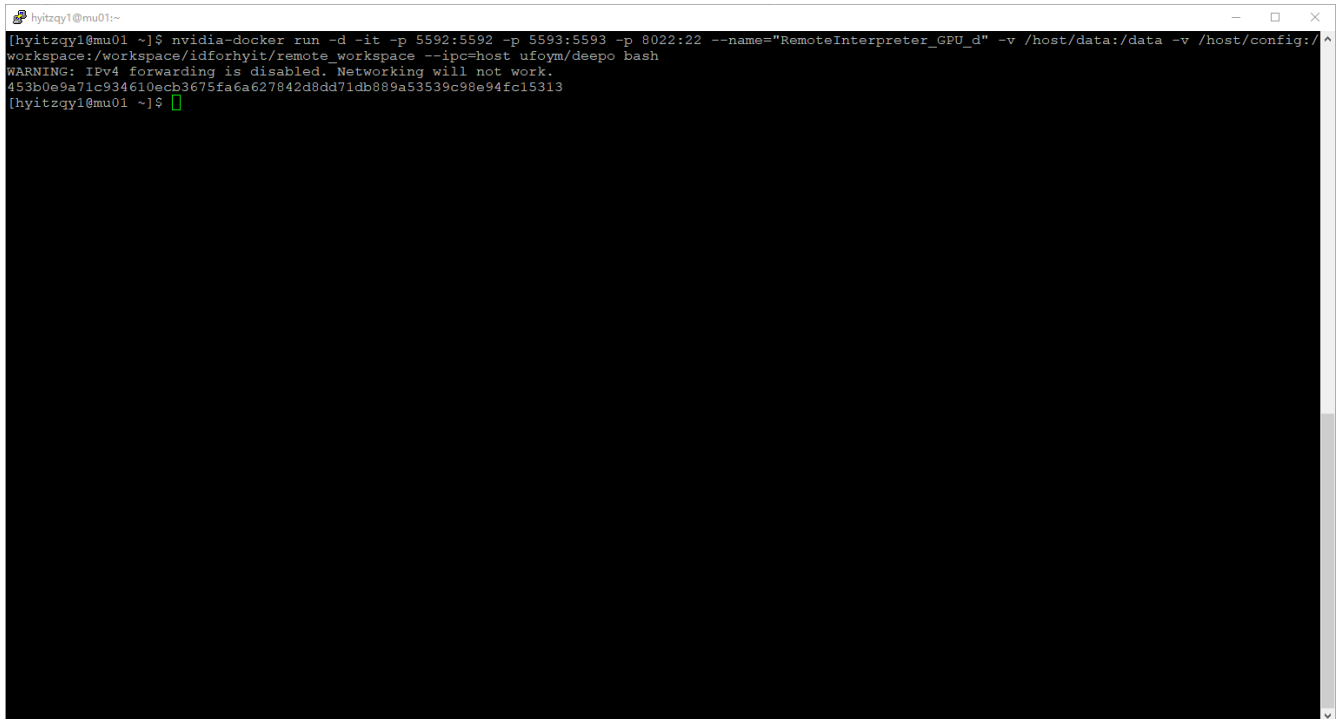
<https://www.runoob.com/docker/docker-command-manual.html>

## 新建docker container

1 | ### PS. 一定要留一个端口映射到容器22端口, 因为SFTP默认使用22端口) ###

### 守护态执行/后台运行方式新建容器

```
1 | sudo nvidia-docker run -d -it -p 5592:5592 -p 5593:5593 -p 8022:22 --  
   | name="RemoteInterpreter_GPU_d" -v /host/data:/data -v /host/config:/config -v  
   | ~/MaSiwei/workspace/idforhyit/remote_workspace:/workspace/idforhyit/remote_workspace --  
   | ipc=host ufoym/deepo bash  
2 | # or  
3 | nvidia-docker run -d -it -p 5592:5592 -p 5593:5593 -p 8022:22 --  
   | name="RemoteInterpreter_GPU_d" -v /host/data:/data -v /host/config:/config -v  
   | ~/workspace/idforhyit/remote_workspace:/workspace/idforhyit/remote_workspace  
   | ufoym/deepo bash
```



A terminal window showing the execution of the command: `nvidia-docker run -d -it -p 5592:5592 -p 5593:5593 -p 8022:22 --name="RemoteInterpreter_GPU_d" -v /host/data:/data -v /host/config:/config -v ~/workspace/idforhyit/remote_workspace:/workspace/idforhyit/remote_workspace --ipc=host ufoym/deepo bash`. The output shows the container starting successfully with a warning about IPv4 forwarding being disabled. The prompt changes to `[hyitzqy1@mu01 ~]$`.

新建的docker容器, 端口映射, 容器名, 镜像和路径映射这些换成你自己的就行, 但是一定要留一个端口映射到容器22端口, 因为SFTP默认使用22端口。

## Docker Tips

- 守护态执行新建后获取容器的输出信息

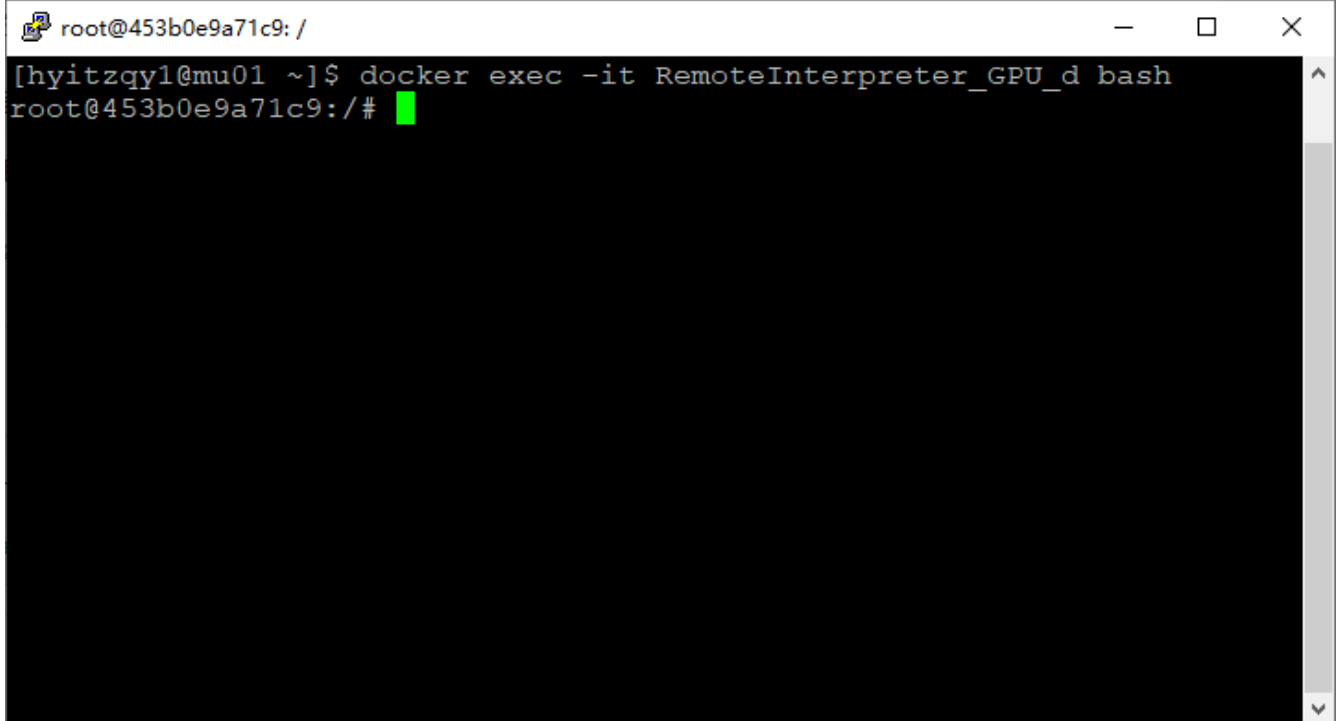
```
1 | docker container logs RemoteInterpreter_GPU_d
```

- 进入容器 (非启动)

推荐用docker exec方式



```
1 | docker exec -it RemoteInterpreter_GPU_d bash
```

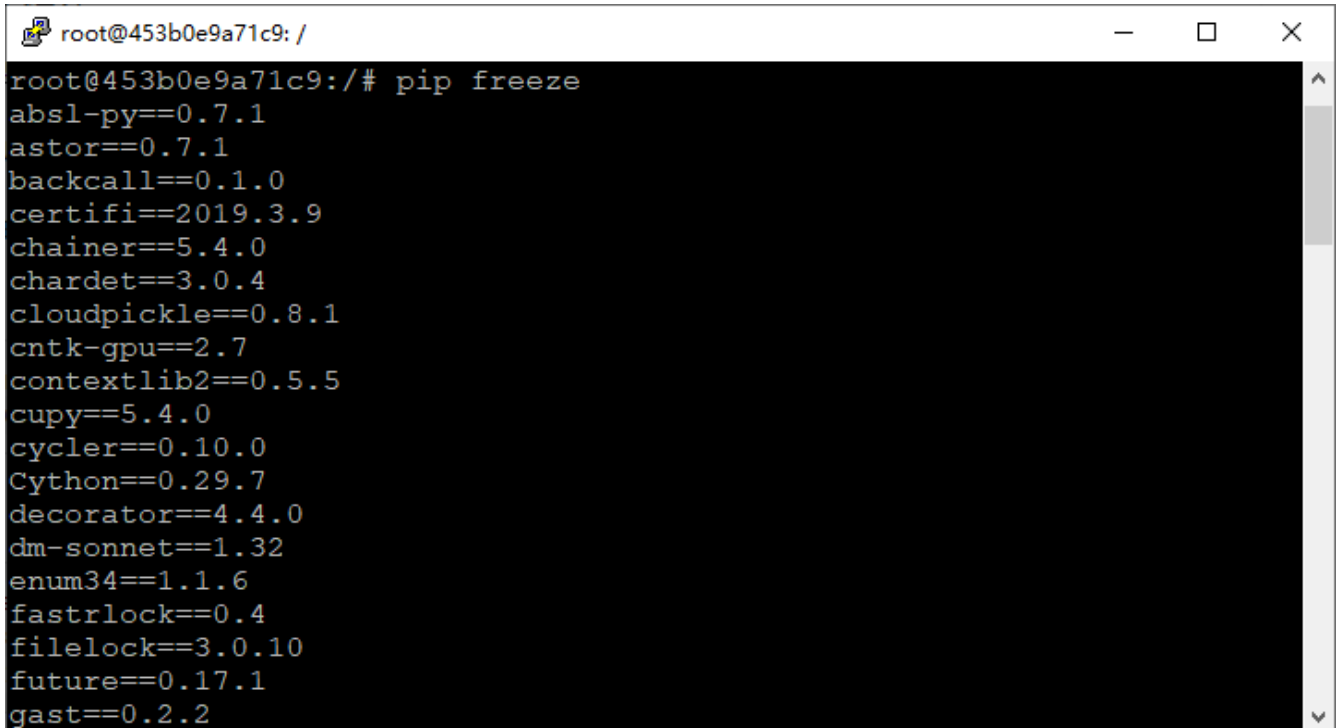


```
root@453b0e9a71c9: /  
[hyitzqy1@mu01 ~]$ docker exec -it RemoteInterpreter_GPU_d bash  
root@453b0e9a71c9:/#
```

- 查看已经安装包的命令以及版本

进入容器后使用，查看一下 `deepo` 的已有环境

```
1 | pip freeze
```



```
root@453b0e9a71c9:/# pip freeze  
absl-py==0.7.1  
astor==0.7.1  
backcall==0.1.0  
certifi==2019.3.9  
chainer==5.4.0  
chardet==3.0.4  
cloudpickle==0.8.1  
cntk-gpu==2.7  
contextlib2==0.5.5  
cupy==5.4.0  
cyclers==0.10.0  
Cython==0.29.7  
decorator==4.4.0  
dm-sonnet==1.32  
enum34==1.1.6  
fastrlock==0.4  
filelock==3.0.10  
future==0.17.1  
gast==0.2.2
```

下面为 DockerHub 中 ufoym/deepo 的已有环境内容，可以看到默认已经带了比较全面的机器学习的开发框架（TensorFlow，PyTorch，Keras，scikit-learn），这个容器对于深度学习科研实验来说，已经达到了开箱即用的程度，不然自己配置会很麻烦。另外，因为 DockerHub 开源，镜像安全性也是有一定保障的。

Docker，懒癌患者的福音。

```
1 root@453b0e9a71c9:/# pip freeze
2 absl-py==0.7.1
3 astor==0.7.1
4 backcall==0.1.0
5 certifi==2019.3.9
6 chainer==5.4.0
7 chardet==3.0.4
8 cloudpickle==0.8.1
9 cntk-gpu==2.7
10 contextlib2==0.5.5
11 cupy==5.4.0
12 cycler==0.10.0
13 Cython==0.29.7
14 decorator==4.4.0
15 dm-sonnet==1.32
16 enum34==1.1.6
17 fastrlock==0.4
18 filelock==3.0.10
19 future==0.17.1
20 gast==0.2.2
21 google-pasta==0.1.5
22 graphviz==0.10.1
23 grpcio==1.20.0
24 h5py==2.9.0
25 idna==2.8
26 imageio==2.5.0
27 ipython==7.4.0
28 ipython-genutils==0.2.0
29 jedi==0.13.3
30 Keras==2.2.4
31 Keras-Applications==1.0.7
32 Keras-Preprocessing==1.0.9
33 kiwisolver==1.0.1
34 Lasagne==0.2.dev1
35 leveldb==0.194
36 Mako==1.0.9
37 Markdown==3.1
38 MarkupSafe==1.1.1
39 matplotlib==3.0.3
40 mxnet-cu100==1.4.0.post0
41 networkx==2.3
42 nose==1.3.7
43 numpy==1.16.3
44 onnx==1.5.0
45 pandas==0.24.2
46 parso==0.4.0
47 pexpect==4.7.0
```

```
48 pickleshare==0.7.5
49 pillow==6.0.0
50 prompt-toolkit==2.0.9
51 protobuf==3.7.1
52 ptyprocess==0.6.0
53 Pygments==2.3.1
54 pygobject==3.26.1
55 pygpu==0.7.6
56 pyparsing==2.4.0
57 python-apt==1.6.3+ubuntu1
58 python-dateutil==2.8.0
59 python-distutils-extra==2.39
60 python-gflags==3.1.2
61 pytz==2019.1
62 PyWavelets==1.0.3
63 PyYAML==5.1
64 requests==2.21.0
65 scikit-image==0.15.0
66 scikit-learn==0.20.3
67 scipy==1.2.1
68 semantic-version==2.6.0
69 six==1.12.0
70 tb-nightly==1.14.0a20190424
71 tensorflow-estimator-2.0-preview==1.14.0.dev2019042300
72 tensorflow-probability==0.6.0
73 termcolor==1.1.0
74 tf-nightly-gpu-2.0-preview==2.0.0.dev20190424
75 Theano==1.0.4+unknown
76 torch-nightly==1.1.0.dev20190424
77 torchvision-nightly==0.2.3
78 tqdm==4.31.1
79 traitlets==4.3.2
80 typing==3.6.6
81 typing-extensions==3.7.2
82 urllib3==1.24.2
83 wcwidth==0.1.7
84 Werkzeug==0.15.2
85 wrapt==1.11.1
```

- 退出容器 (非终止)

```
1 exit
2 # or
3 Ctrl+D
```

- 显示本机已有的容器 (正在运行的)

```
1 docker ps
```

```
hyitzqy1@mu01:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
453b0e9a71c9   ufoym/deepo   "bash"                  16 minutes ago Up 16 minutes 0.0.0.0:5592-5593->5592-5593/tcp, 6006/tcp, 0.0.0.0:8022->22/tcp RemoteInterpreter_GFU_d
19d934dd8959   tensorflow-zqy:latest "/bin/bash"          4 hours ago   Up 4 hours    0.0.0.0:32985->6006/tcp             3142.mu01
92584da6712b   tensorflow-zqy:latest "/bin/bash"          4 hours ago   Up 4 hours    0.0.0.0:32980->6006/tcp             3139.mu01
67de8815b2e   tensorflow-zqy:latest "/bin/bash"          14 hours ago   Up 14 hours   0.0.0.0:32976->6006/tcp             3138.mu01
67d146c02192   tensorflow-zqy:latest "/bin/bash"          39 hours ago   Up 39 hours   0.0.0.0:32919->6006/tcp             3068.mu01
cf8b22d6e4c   tensorflow-zqy:latest "/bin/bash"          2 days ago     Up 2 days     0.0.0.0:32899->6006/tcp             3068.mu01
8228a8b51d05   tensorflow-zqy:latest "/bin/bash"          2 days ago     Up 2 days     0.0.0.0:32833->6006/tcp             3008.mu01
hyitzqy1@mu01:~$
```

- 显示本机已有的容器（包括停止运行的）

```
1 | docker ps -a
```

```
hyitzqy1@mu01:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
453b0e9a71c9   ufoym/deepo   "bash"                  16 minutes ago Up 16 minutes 0.0.0.0:5592-5593->5592-5593/tcp, 6006/tcp, 0.0.0.0:8022->22/tcp RemoteInterpreter_GFU_d
19d934dd8959   tensorflow-zqy:latest "/bin/bash"          4 hours ago   Up 4 hours    0.0.0.0:32985->6006/tcp             3142.mu01
92584da6712b   tensorflow-zqy:latest "/bin/bash"          4 hours ago   Up 4 hours    0.0.0.0:32980->6006/tcp             3139.mu01
67de8815b2e   tensorflow-zqy:latest "/bin/bash"          14 hours ago   Up 14 hours   0.0.0.0:32976->6006/tcp             3138.mu01
67d146c02192   tensorflow-zqy:latest "/bin/bash"          39 hours ago   Up 39 hours   0.0.0.0:32919->6006/tcp             3068.mu01
cf8b22d6e4c   tensorflow-zqy:latest "/bin/bash"          2 days ago     Up 2 days     0.0.0.0:32899->6006/tcp             3068.mu01
8228a8b51d05   tensorflow-zqy:latest "/bin/bash"          2 days ago     Up 2 days     0.0.0.0:32833->6006/tcp             3008.mu01
23d2287112db   tensorflow-zqy:latest "/bin/bash"          4 days ago     Exited (255) 4 days ago            2930.mu01
0759d095958e   tensorflow-zqy:latest "/bin/bash"          4 days ago     Exited (255) 4 days ago            2929.logContainer
e23d604224c2   tensorflow-zqy:latest "/bin/bash"          4 days ago     Exited (255) 4 days ago            2928.logContainer
5a33b1cda585   tensorflow-zqy:latest "/bin/bash"          4 days ago     Exited (255) 4 days ago            2927.logContainer
d7bd8da22e18a   tensorflow-zqy:latest "/bin/bash"          4 days ago     Exited (255) 4 days ago            2644.tensorboard
63b087dfb00cd   tensorflow-zqy:latest "/bin/bash"          6 days ago     Exited (255) 4 days ago            2644.logContainer
e6de0c1c6078a   tensorflow-zqy:latest "/bin/bash"          7 days ago     Exited (255) 6 days ago            2536.logContainer
d8cd217cedcb0   tensorflow-zqy:latest "/bin/bash"          7 days ago     Exited (255) 6 days ago            2765.tensorboard
f8f60a530794   tensorflow-zqy:latest "/bin/bash"          8 days ago     Exited (255) 6 days ago            2765.logContainer
b5b2815241ee   tensorflow-zqy:latest "/bin/bash"          8 days ago     Exited (255) 6 days ago            2876.mu01
b02e5c35f54b   tensorflow-zqy:latest "/bin/bash"          8 days ago     Exited (255) 6 days ago            2799.mu01
f4519cadb279   caffe-dnn:latest "/bin/bash"          10 days ago    Created                                2793.logContainer
c37f84c3d0eb3   tensorflow-zqy:latest "/bin/bash"          11 days ago    Exited (255) 6 days ago            silly_jennings
34f4e68501c9   Sb7637ecf405 "/bin/bash"          12 days ago    Exited (255) 6 days ago            2712.logContainer
674edf2c52fe   Sb7637ecf405 "/bin/bash"          12 days ago    Exited (255) 6 days ago            heuristic_franklin
f4d590a40c1c   tensorflow-xmjl:latest "/bin/bash"          12 days ago    Exited (255) 6 days ago            tender_euclid
decc3c063260   Sb7637ecf405 "/bin/bash"          2 weeks ago    Exited (255) 6 days ago            2272.logContainer
89e5879758e4   tensorflow-hurcomglinpy27:latest "/bin/bash"          3 weeks ago    Exited (255) 6 days ago            infallible_goodall
b997fa8d1ef3   Sb7637ecf405 "/bin/bash"          3 weeks ago    Exited (255) 6 days ago            vigilant_wilson
bc8cb1f75732   2ad1b1fecf2a "/bin/bash"          3 weeks ago    Exited (255) 6 days ago            2141.mu01.tensorboard
62fa5a63aac   2ad1b1fecf2a "/bin/bash"          3 weeks ago    Exited (255) 6 days ago            2138.logContainer
5e5e57d191fe   2ad1b1fecf2a "/bin/bash"          3 weeks ago    Exited (255) 6 days ago            2133.logContainer
c0c05f8a901a   2ad1b1fecf2a "/bin/bash"          3 weeks ago    Exited (255) 6 days ago            2128.logContainer
9121b467103f   2ad1b1fecf2a "/bin/bash"          4 weeks ago    Exited (255) 6 days ago            peaceful_pasteur
ba201955a4ba   fb7e566e3518 "/bin/bash"          5 weeks ago    Exited (255) 6 days ago            1863.mu01.tensorboard
4b0d48390e65   fb7e566e3518 "/bin/bash"          5 weeks ago    Exited (255) 6 days ago            1881.logContainer
1602da5e49a6   fb7e566e3518 "/bin/bash"          5 weeks ago    Exited (255) 6 days ago            determined_shirley
40aba365c6df   fb7e566e3518 "/bin/bash"          5 weeks ago    Exited (255) 6 days ago            frosty_shirley
95ae0a0d4573   fb7e566e3518 "/bin/bash"          5 weeks ago    Exited (255) 6 days ago            vigorous_turing
9640dcf01e7a   fb7e566e3518 "/bin/bash"          5 weeks ago    Exited (255) 6 days ago            1900.logContainer
dc3f49a0479b   222cb0d6970cc "/bin/bash"          5 weeks ago    Exited (255) 6 days ago            kind_bacon
ba83602f2b08   222cb0d6970cc "/bin/bash"          5 weeks ago    Exited (255) 6 days ago            1886.logContainer
426da1b3d729   222cb0d6970cc "/bin/bash"          5 weeks ago    Exited (255) 6 days ago            friendly_nobel
d2035d995f96   222cb0d6970cc "/bin/bash"          5 weeks ago    Exited (255) 6 days ago            stupified_lisakov
1b615a0f35ab   222cb0d6970cc "/bin/bash"          5 weeks ago    Exited (255) 6 days ago            1862a23b6d3bba0
f4319d7811be   222cb0d6970cc "/bin/bash"          6 weeks ago    Exited (255) 6 days ago            agitated_sammet
626df7e2f3a   tensorflow061202:latest "/bin/bash"          6 weeks ago    Exited (255) 6 days ago            pensive_payne
10a609424246   tensorflow061202:latest "/bin/bash"          6 weeks ago    Exited (255) 6 days ago            nifty_richie
49e0a515d7f1   181306cbe607 "/bin/bash"          6 weeks ago    Exited (255) 6 days ago            determined_lovelace
0f2e189f2688   181306cbe607 "/bin/bash"          6 weeks ago    Exited (255) 6 days ago            zealous_goodall
db1f573b5ea   4bb9a302ee50 "/bin/bash"          6 weeks ago    Exited (255) 6 days ago            youthful_bhaaskara
97b70ba8d82   4bb9a302ee50 "/bin/bash"          6 weeks ago    Exited (255) 6 days ago            clever_colden
e8b144bc9f0b   4bb9a302ee50 "/bin/bash"          6 weeks ago    Exited (255) 6 days ago            1864.mu01
e5d311e30d81   caffe-dnn:latest "/bin/bash"          6 weeks ago    Created                                18631f0f5f7d248c
ca8db98ed05   d5b06b143259 "/bin/bash"          6 weeks ago    Created                                quirky_lampost
76c24dbb70c0   d5b06b143259 "/bin/bash"          6 weeks ago    Created                                mystifying_jennings
97d93ccb94e8   d5b06b143259 "/bin/bash"          6 weeks ago    Created                                pedantic_colden
642321975927   d5b06b143259 "/bin/bash"          6 weeks ago    Created                                1862a23b6d3bba0
7d45419b435a   a7690abc7ca0 "/bin/bash"          6 weeks ago    Created                                priceless_benz
db7bf953ee0   a7690abc7ca0 "/bin/bash"          6 weeks ago    Created                                elated_einstein
009d9415b37   a7690abc7ca0 "/bin/bash"          6 weeks ago    Created                                keen_colicet
f0c1a17b0a07   a7690abc7ca0 "/bin/bash"          6 weeks ago    Created                                zen_perلمان
a112308fd29f
```

- 终止容器

使用 `docker stop $CONTAINER_ID` 来终止一个运行中的容器。并且可以使用 `docker ps -a` 来看终止状态的容器。

终止状态的容器，可以使用 `docker start` 来重新启动。

```
1 | docker stop [Name或者ID]
```

- 删除容器 (Container)

```
1 | docker rm [Name或者ID]
2
3 | #
4 | docker stop RemoteInterpreter_GPU_d
5 | docker rm RemoteInterpreter_GPU_d
```

- 删除镜像 (Image)

一般不删除镜像，毕竟十几个G，重下也麻烦。

**删除前务必保证该镜像的所有实例容器 (Container) 全部 stop 停止和删除**，否则会很麻烦。

需要注意删除镜像和容器的命令不一样。 `docker rmi ID` ,其中 容器(`rm`)和 镜像(`rmi`)。

```
1 | # 查询镜像，拿到欲删除的`IMAGE ID`
2 | docker images
3 | # 查询容器，拿到镜像实例容器的`CONTATNER ID`
4 | docker ps -a
5 | # 先删除容器，如果容器正在运行，先用`docker stop`停止
6 | docker rm [CONTATNER ID]
7 | docker ps -a    # 查看容器是否删除成功
8 | # 再删除镜像
9 | docker rmi [IMAGE ID]
```

## 配置SSH服务

配置部分命令在目标容器内进行

### 先对目标容器环境进行更新

```
1 | docker exec -it RemoteInterpreter_GPU_d bash    # 如已在容器内请忽略本条
2 | apt update
```

```
root@453b0e9a71c9: /
[hyitzqy1@mu01 ~]$ docker exec -it RemoteInterpreter_GPU_d bash
root@453b0e9a71c9:/# apt update
Err:1 http://archive.ubuntu.com/ubuntu bionic InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Err:2 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease
  Temporary failure resolving 'ppa.launchpad.net'
Err:3 http://security.ubuntu.com/ubuntu bionic-security InRelease
  Temporary failure resolving 'security.ubuntu.com'
Err:4 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Err:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/bionic/InRelease Temporary failure resolving 'archive.ubuntu.com'
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/bionic-updates/InRelease Temporary failure resolving 'archive.ubuntu.com'
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/bionic-backports/InRelease Temporary failure resolving 'archive.ubuntu.com'
W: Failed to fetch http://security.ubuntu.com/ubuntu/dists/bionic-security/InRelease Temporary failure resolving 'security.ubuntu.com'
W: Failed to fetch http://ppa.launchpad.net/deadsnakes/ppa/ubuntu/dists/bionic/InRelease Temporary failure resolving 'ppa.launchpad.net'
W: Some index files failed to download. They have been ignored, or old ones used instead.
root@453b0e9a71c9:/#
```

获更新失败解决参考

## Docker换源三部曲

- 官方源国内不好使，那咱就换源，盘他~
1. 复制源文件备份，以防万一

```
1 | cp /etc/apt/sources.list /etc/apt/sources.list.bak
```

2. 查看新版本信息

```
1 | lsb_release -c
```

```
root@453b0e9a71c9: /
root@453b0e9a71c9:/# cp /etc/apt/sources.list /etc/apt/sources.list.bak
root@453b0e9a71c9:/# lsb_release -c
Codename:         bionic
root@453b0e9a71c9:/#
```

○ 如果不是 `bionic` 就将之后镜像源地址中的版本代号部分换为对应的版本代号

- Ubuntu 12.04 (LTS)代号为 `precise`
- Ubuntu 14.04 (LTS)代号为 `trusty`
- Ubuntu 15.04 代号为 `vivid`
- Ubuntu 15.10 代号为 `wily`
- Ubuntu 16.04 (LTS)代号为 `xenial`
- Ubuntu 18.04 (LTS)代号为 `bionic`

### 3. 编辑源列表文件

```
1 | vim /etc/apt/sources.list
```

将原有的有效行前面添加 `#` 注释掉，添加以下内容（或者你把里面内容修改成下面的就可以，但是不能有除了以下内容的有效内容）

- Vim使用方法，防对命令行操作不熟悉的童鞋。笔记本键盘童鞋，如果操作失败，想想 `Fn` 键的作用。
  - `Insert`，进行插入方式的编辑文本文件
  - `Shift + Insert`，进行粘贴粘贴板中的文本内容
    - 如果和博主使用的Dock镜像不一样，注意将文本中 `bionic` 替换为对应版本的代号
  - `ESC ==> Shift+ ; (即 :) ==> wq`，保存修改并退出 编辑

```

1 deb http://mirrors.aliyun.com/ubuntu/ bionic main restricted universe multiverse
2 deb http://mirrors.aliyun.com/ubuntu/ bionic-security main restricted universe
  multiverse
3 deb http://mirrors.aliyun.com/ubuntu/ bionic-updates main restricted universe
  multiverse
4 deb http://mirrors.aliyun.com/ubuntu/ bionic-proposed main restricted universe
  multiverse
5 deb http://mirrors.aliyun.com/ubuntu/ bionic-backports main restricted universe
  multiverse
6 deb-src http://mirrors.aliyun.com/ubuntu/ bionic main restricted universe multiverse
7 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-security main restricted universe
  multiverse
8 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-updates main restricted universe
  multiverse
9 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-proposed main restricted universe
  multiverse
10 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-backports main restricted universe
    multiverse

```

```

root@453b0e9a71c9: /
1 deb http://mirrors.aliyun.com/ubuntu/ bionic main restricted universe multiverse
2
3 deb-src http://mirrors.aliyun.com/ubuntu/ bionic main restricted universe multiverse
4
5 deb http://mirrors.aliyun.com/ubuntu/ bionic-security main restricted universe multiverse
6
7 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-security main restricted universe multiverse
8
9 deb http://mirrors.aliyun.com/ubuntu/ bionic-updates main restricted universe multiverse
10
11 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-updates main restricted universe multiverse
12
13 deb http://mirrors.aliyun.com/ubuntu/ bionic-backports main restricted universe multiverse
14
15 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-backports main restricted universe multiverse
16
17 deb http://mirrors.aliyun.com/ubuntu/ bionic-proposed main restricted universe multiverse
18
19 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-proposed main restricted universe multiverse
20
21 # See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
22 # newer versions of the distribution.
23 #deb http://archive.ubuntu.com/ubuntu/ bionic main restricted
24 # deb-src http://archive.ubuntu.com/ubuntu/ bionic main restricted
25
26 ## Major bug fix updates produced after the final release of the
27 ## distribution.
28 #deb http://archive.ubuntu.com/ubuntu/ bionic-updates main restricted
29 # deb-src http://archive.ubuntu.com/ubuntu/ bionic-updates main restricted
30
31 ## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
32 ## team. Also, please note that software in universe WILL NOT receive any
33 ## review or updates from the Ubuntu security team.
34 #deb http://archive.ubuntu.com/ubuntu/ bionic universe
35 # deb-src http://archive.ubuntu.com/ubuntu/ bionic universe
36 #deb http://archive.ubuntu.com/ubuntu/ bionic-updates universe
37 # deb-src http://archive.ubuntu.com/ubuntu/ bionic-updates universe
38
39 ## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
40 ## team, and may not be under a free licence. Please satisfy yourself as to
41 ## your rights to use the software. Also, please note that software in
42 ## multiverse WILL NOT receive any review or updates from the Ubuntu
43 ## security team.
44 #deb http://archive.ubuntu.com/ubuntu/ bionic multiverse
45 # deb-src http://archive.ubuntu.com/ubuntu/ bionic multiverse
46 #deb http://archive.ubuntu.com/ubuntu/ bionic-updates multiverse
47 # deb-src http://archive.ubuntu.com/ubuntu/ bionic-updates multiverse
48
49 ## N.B. software from this repository may not have been tested as
50 ## extensively as that contained in the main release, although it includes
51 ## newer versions of some applications which may provide useful features.
52 ## Also, please note that software in backports WILL NOT receive any review
53 ## or updates from the Ubuntu security team.
54 #deb http://archive.ubuntu.com/ubuntu/ bionic-backports main restricted universe multiverse
55 # deb-src http://archive.ubuntu.com/ubuntu/ bionic-backports main restricted universe multiverse
56

```

开启行号: `:` + `set number`

更多vi/vim命令详见: <https://www.runoob.com/linux/linux-vim.html>

```

1 # 国内源
2 deb http://cn.archive.ubuntu.com/ubuntu bionic main multiverse restricted universe
3 deb http://cn.archive.ubuntu.com/ubuntu bionic-updates main multiverse restricted
  universe
4 deb http://cn.archive.ubuntu.com/ubuntu bionic-security main multiverse restricted
  universe
5 deb http://cn.archive.ubuntu.com/ubuntu bionic-proposed main multiverse restricted
  universe
6

```



```
7 #添加阿里源
8 deb http://mirrors.aliyun.com/ubuntu/ bionic main restricted universe multiverse
9 deb http://mirrors.aliyun.com/ubuntu/ bionic-security main restricted universe
  multiverse
10 deb http://mirrors.aliyun.com/ubuntu/ bionic-updates main restricted universe
  multiverse
11 deb http://mirrors.aliyun.com/ubuntu/ bionic-proposed main restricted universe
  multiverse
12 deb http://mirrors.aliyun.com/ubuntu/ bionic-backports main restricted universe
  multiverse
13 deb-src http://mirrors.aliyun.com/ubuntu/ bionic main restricted universe multiverse
14 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-security main restricted universe
  multiverse
15 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-updates main restricted universe
  multiverse
16 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-proposed main restricted universe
  multiverse
17 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-backports main restricted universe
  multiverse
18
19 ##中科大源
20 deb https://mirrors.ustc.edu.cn/ubuntu/ bionic main restricted universe multiverse
21 deb-src https://mirrors.ustc.edu.cn/ubuntu/ bionic main restricted universe multiverse
22 deb https://mirrors.ustc.edu.cn/ubuntu/ bionic-updates main restricted universe
  multiverse
23 deb-src https://mirrors.ustc.edu.cn/ubuntu/ bionic-updates main restricted universe
  multiverse
24 deb https://mirrors.ustc.edu.cn/ubuntu/ bionic-backports main restricted universe
  multiverse
25 deb-src https://mirrors.ustc.edu.cn/ubuntu/ bionic-backports main restricted universe
  multiverse
26 deb https://mirrors.ustc.edu.cn/ubuntu/ bionic-security main restricted universe
  multiverse
27 deb-src https://mirrors.ustc.edu.cn/ubuntu/ bionic-security main restricted universe
  multiverse
28 deb https://mirrors.ustc.edu.cn/ubuntu/ bionic-proposed main restricted universe
  multiverse
29 deb-src https://mirrors.ustc.edu.cn/ubuntu/ bionic-proposed main restricted universe
  multiverse
30
31 ##163源
32 deb http://mirrors.163.com/ubuntu/ bionic main restricted universe multiverse
33 deb http://mirrors.163.com/ubuntu/ bionic-security main restricted universe multiverse
34 deb http://mirrors.163.com/ubuntu/ bionic-updates main restricted universe multiverse
35 deb http://mirrors.163.com/ubuntu/ bionic-proposed main restricted universe multiverse
36 deb http://mirrors.163.com/ubuntu/ bionic-backports main restricted universe
  multiverse
37 deb-src http://mirrors.163.com/ubuntu/ bionic main restricted universe multiverse
38 deb-src http://mirrors.163.com/ubuntu/ bionic-security main restricted universe
  multiverse
39 deb-src http://mirrors.163.com/ubuntu/ bionic-updates main restricted universe
  multiverse
```

```
40 deb-src http://mirrors.163.com/ubuntu/ bionic-proposed main restricted universe
    multiverse
41 deb-src http://mirrors.163.com/ubuntu/ bionic-backports main restricted universe
    multiverse
42
43 ##清华源
44 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe
    multiverse
45 deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe
    multiverse
46 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted
    universe multiverse
47 deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted
    universe multiverse
48 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted
    universe multiverse
49 deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted
    universe multiverse
50 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted
    universe multiverse
51 deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted
    universe multiverse
52 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-proposed main restricted
    universe multiverse
53 deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-proposed main restricted
    universe multiverse
```

---

## 在容器内安装 openssh-server 并配置用户名和密码

```
1 apt update
2 apt install -y openssh-server
3 mkdir /var/run/ssh
4 echo "user:password" | chpasswd # [用户名:密码] 按需设置并牢记
5 sed -i 's/PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config
6 sed 's@session\s*required\s*pam_loginuid.so@session optional pam_loginuid.so@g' -i
    /etc/pam.d/ssh
7 echo "export VISIBLE=now" >> /etc/profile
```

- 修改 sshd-config 属性值

修改 /etc/ssh/sshd-config 以使可远程密码登录docker的root账户

```
1 | vi /etc/ssh/sshd-config
```

NAME	KEY
PermitRootLogin	yes
PasswordAuthentication	yes

### 验证配置

```
1 service ssh restart # 重启SSH激活配置
2 exit #退出容器
3 sudo docker port RemoteInterpreter_GPU_d 22 # 0.0.0.0:8022 正确配置后的输出
4 ssh IDforHYIT@172.20.1.100 -p 8022 # 过程中确认的密码是你前面配置SSH设置的
```

宿主机的IP地址可用 `ifconfig` 获得，按照博主几个虚拟机来瞎测试来总结的经验，一般是第一个以 `en` 开头的网卡就是本机对外IP地址。

## 本地Pycharm配置

社区版无远程解释器功能，**请用专业版**。激活教程请在本博内搜索 `Jetbrain`。

因上一步服务器Docker内的SSH配置中因服务器安全设置等原因，禁用了Docker的端口转发，Docker内无法连接外网，故本步骤用之前虚拟机搭建的IP地址进行配置，除了IP地址、账号密码等因配置原因不一样外，其他无异。

### 配置过程

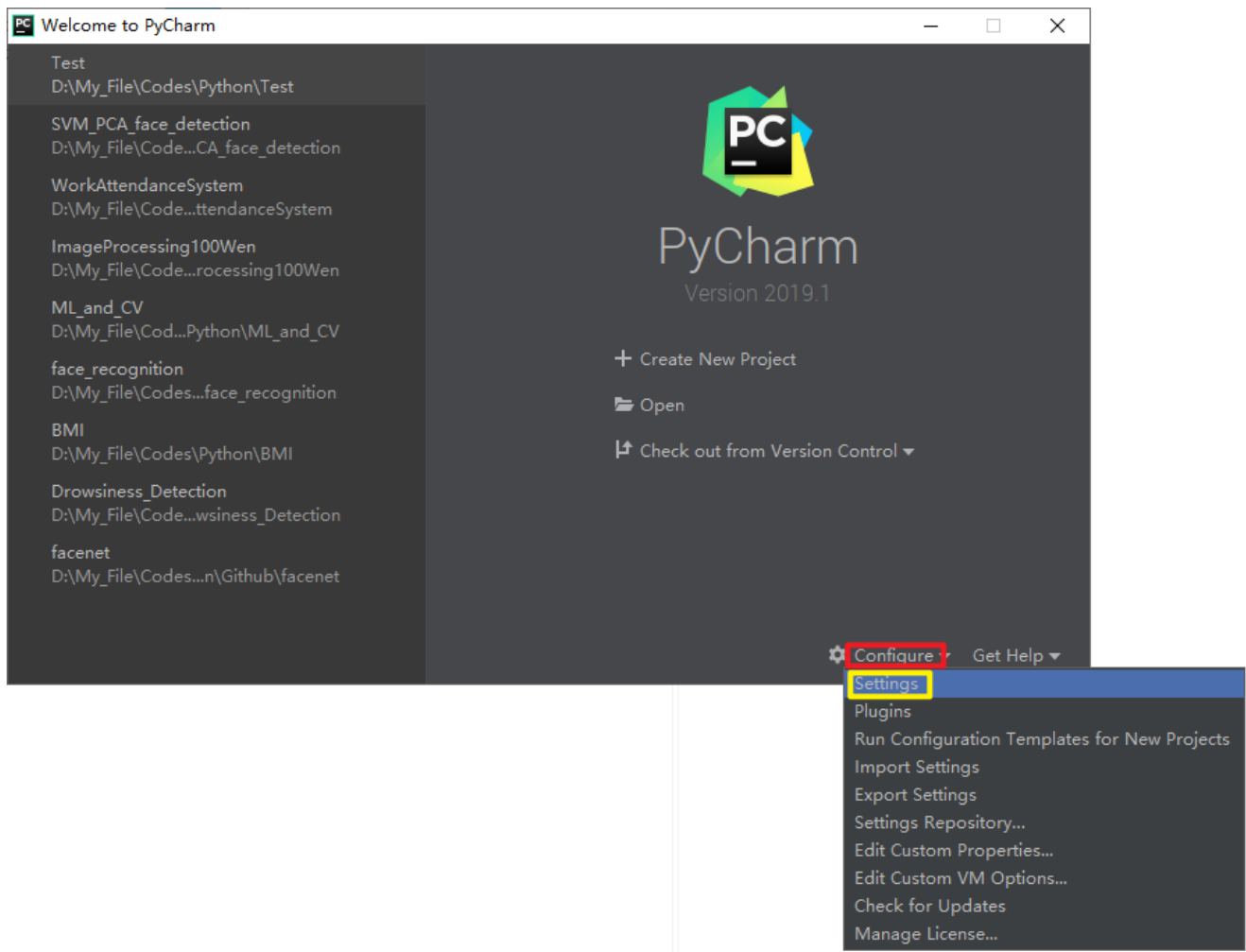
#### 大致配置过程如下：

- 从启动页进入设置（任何项目可用）  
Pycharm启动页 ---> `Configure` ---> `Setting` ---> `Project Interpreter` ---> 小齿轮按钮 ---> `Add...` ---> `SSH Interpreter` ---> `New server configuration` ---> `Host` ---> `Port` ---> `Username` ---> `Next` ---> `Password` ---> `Next` ---> `Interpreter` ---> `Finish`
- 从项目页进入设置（默认仅配置项目可见）  
菜单栏 ---> `File` ---> `Settings...` ---> `Project: (项目名)` ---> `Project Interpreter` ---> 小齿轮按钮 ---> `Add...` ---> `SSH Interpreter` ---> `New server configuration` ---> `Host` ---> `Port` ---> `Username` ---> `Next` ---> `Password` ---> `Next` ---> `Interpreter` ---> `Finish`

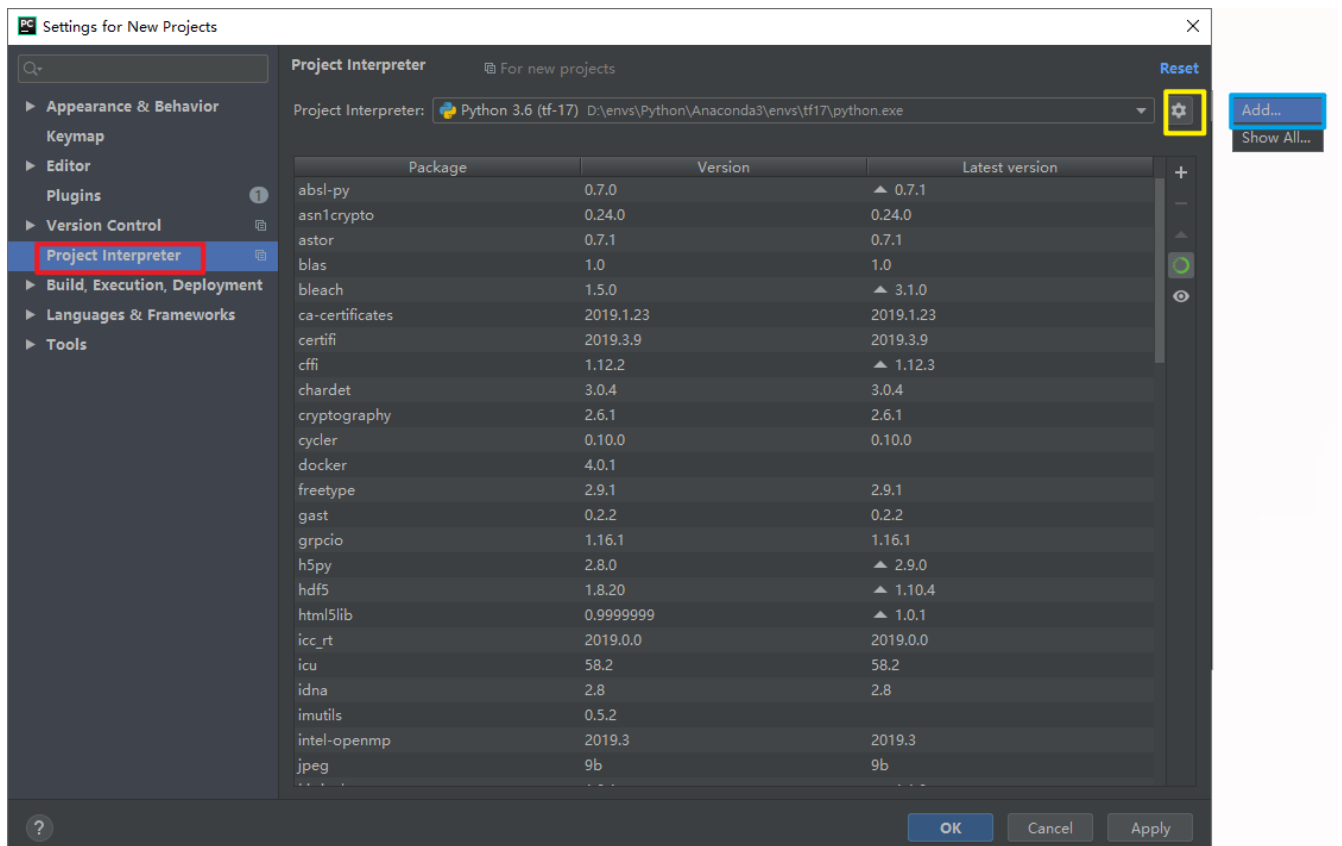
#### 具体过程截图说明：

##### 从启动页进入设置（任何项目可用）

1. Pycharm启动页 ---> `Configure` ---> `Setting`



2. Project Interpreter ---> 小齿轮按钮 ---> Add...



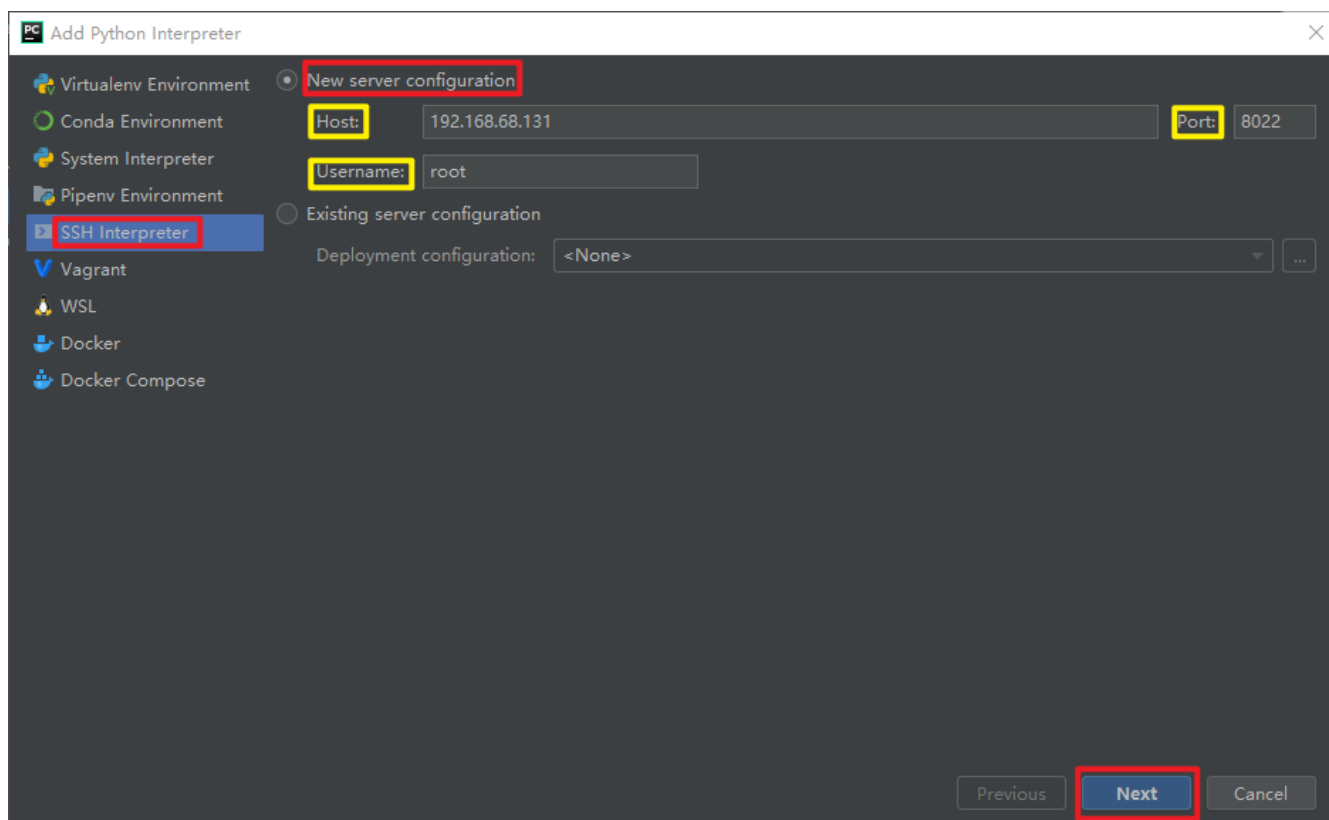
3. SSH Interpreter ---> New server configuration ---> 填入上一步SSH配置中获得的Host地址 ---> Port输入之前新建Docker时设置的22端口的转发端口 ---> Username ---> Next

Host 为宿主机的IP地址，可用 `ifconfig` 命令获得，之前说过了

Port 转发端口为之前 新建Docker 时设置的宿主机端口22的转发端口

`docker run ... -p xxx:22 ...` 命令中的 xxx 即是配置的转发端口

本步及下一步的账号密码请填入你在SSH配置中用 `echo "user:password" | chpasswd` 命令配置的 [用户名:密码]



PC Add Python Interpreter

Virtualenv Environment • New server configuration

Conda Environment

System Interpreter

Pipenv Environment

SSH Interpreter

Vagrant

WSL

Docker

Docker Compose

Host: 192.168.68.131 Port: 8022

Username: root

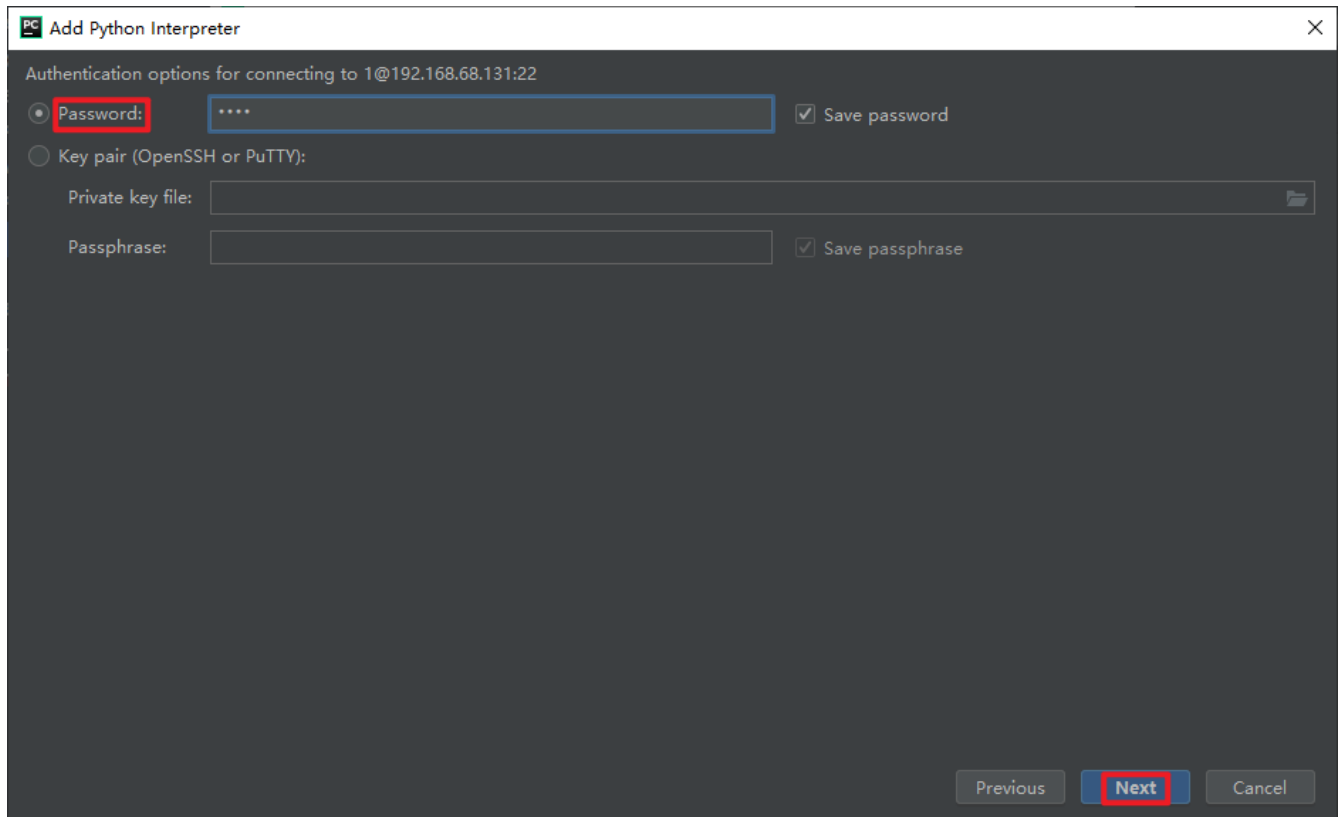
Existing server configuration

Deployment configuration: <None>

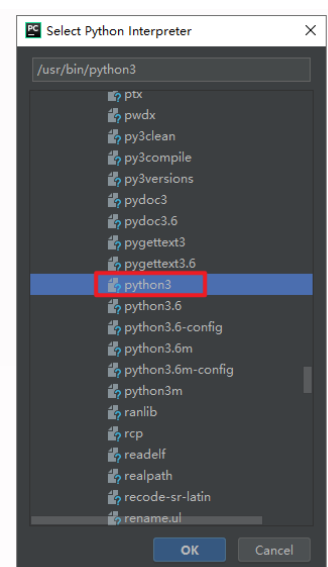
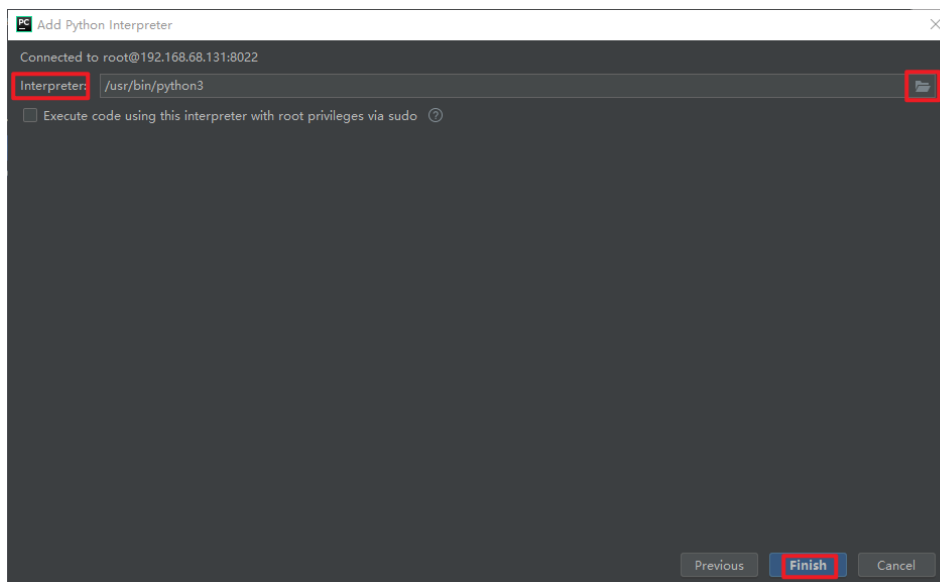
Previous Next Cancel

4. Password ---> Next

建议 Save password 复选框选中

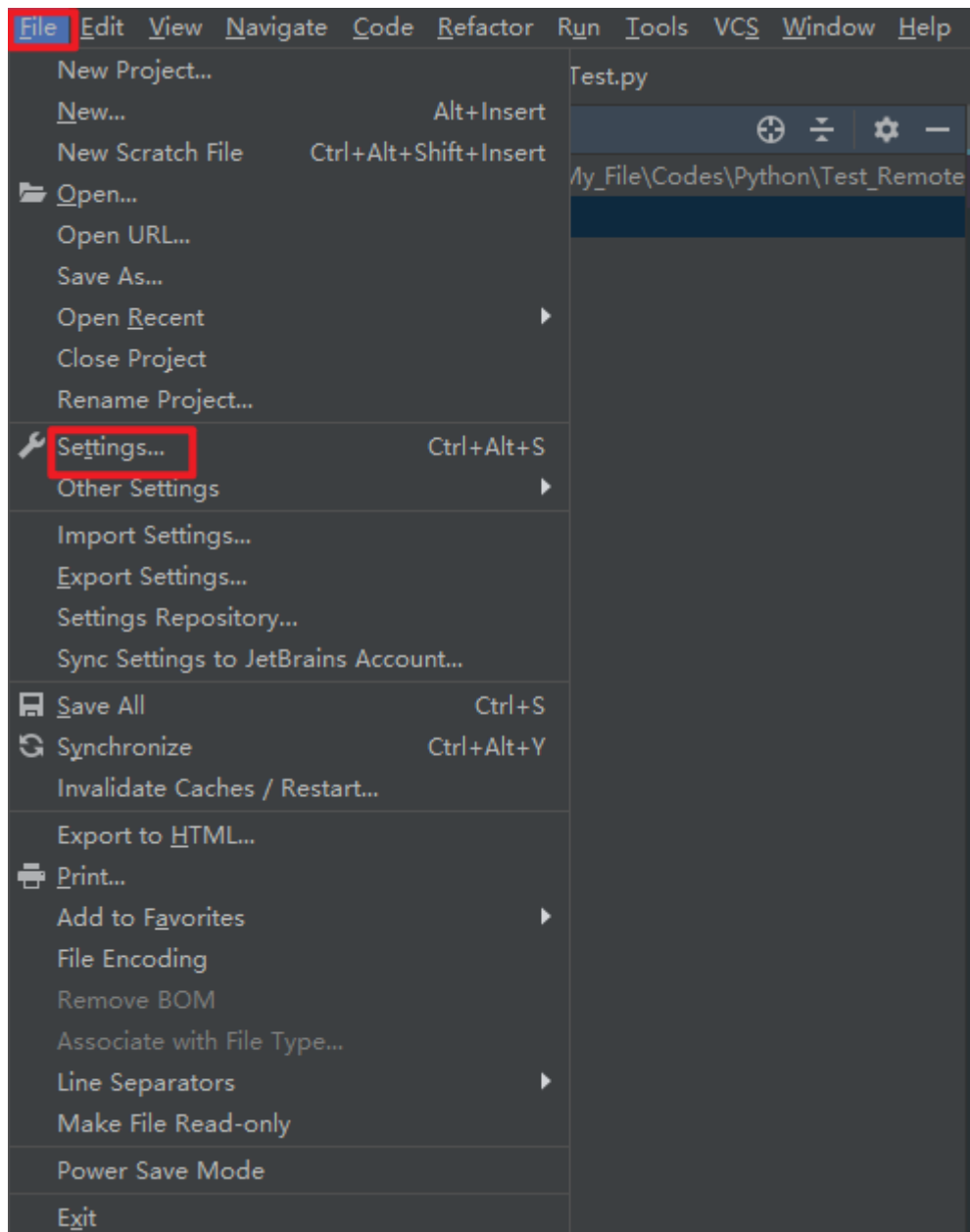


5. **Interpreter** 浏览你目标使用的远程解释器的位置，一般是 `/usr/bin/python` 或 `/usr/bin/python3` ---> **Finish**

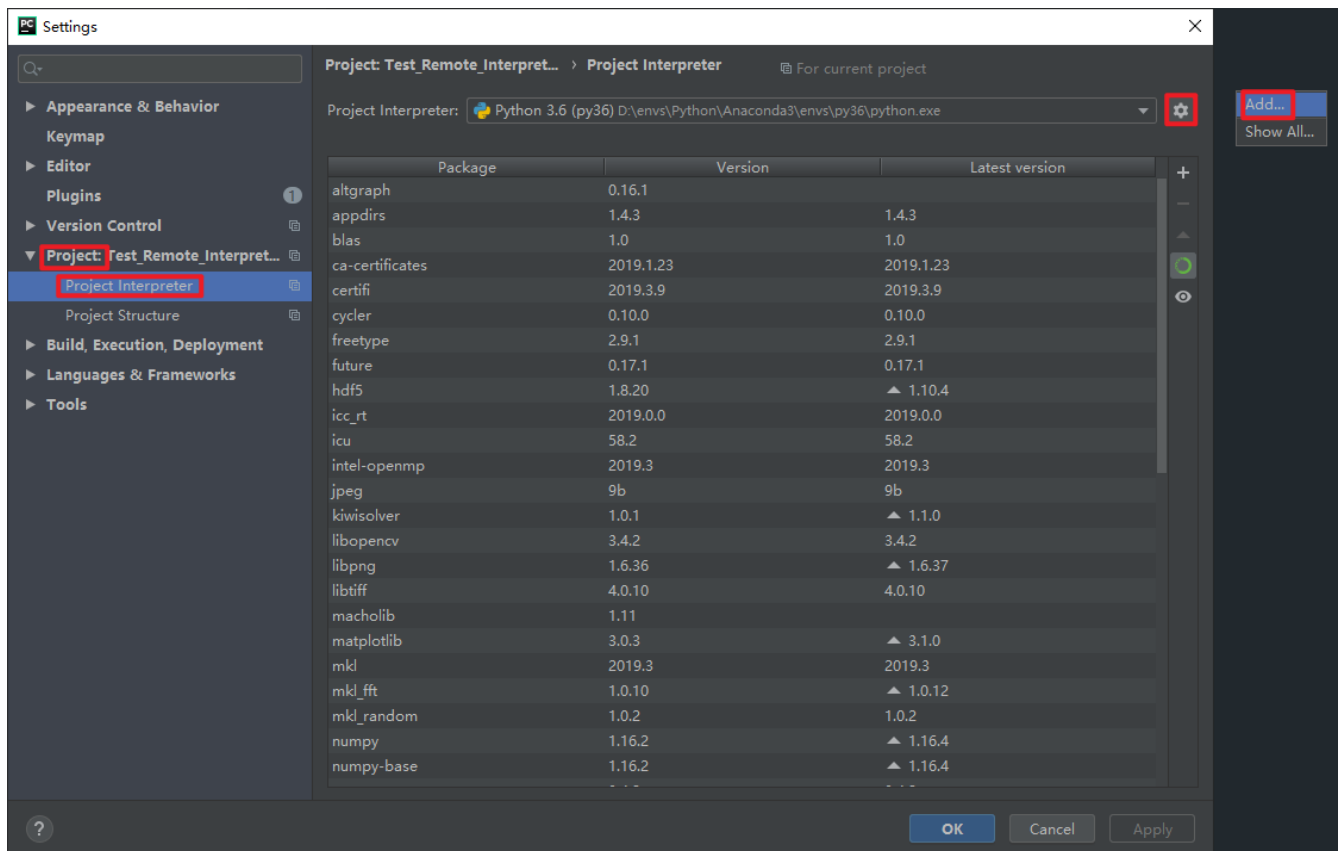


从项目页进入设置（默认仅配置项目可见）

1. 菜单栏 ---> **File** ---> **Settings...**



2. Project: (项目名) ---> Project Interpreter ---> 小齿轮按钮 ---> Add...



3. SSH Interpreter ---> New server configuration ---> 填入上一步SSH配置中获得的Host地址 ---> Port输入之前新建Docker时设置的22端口的转发端口 ---> Username ---> Next

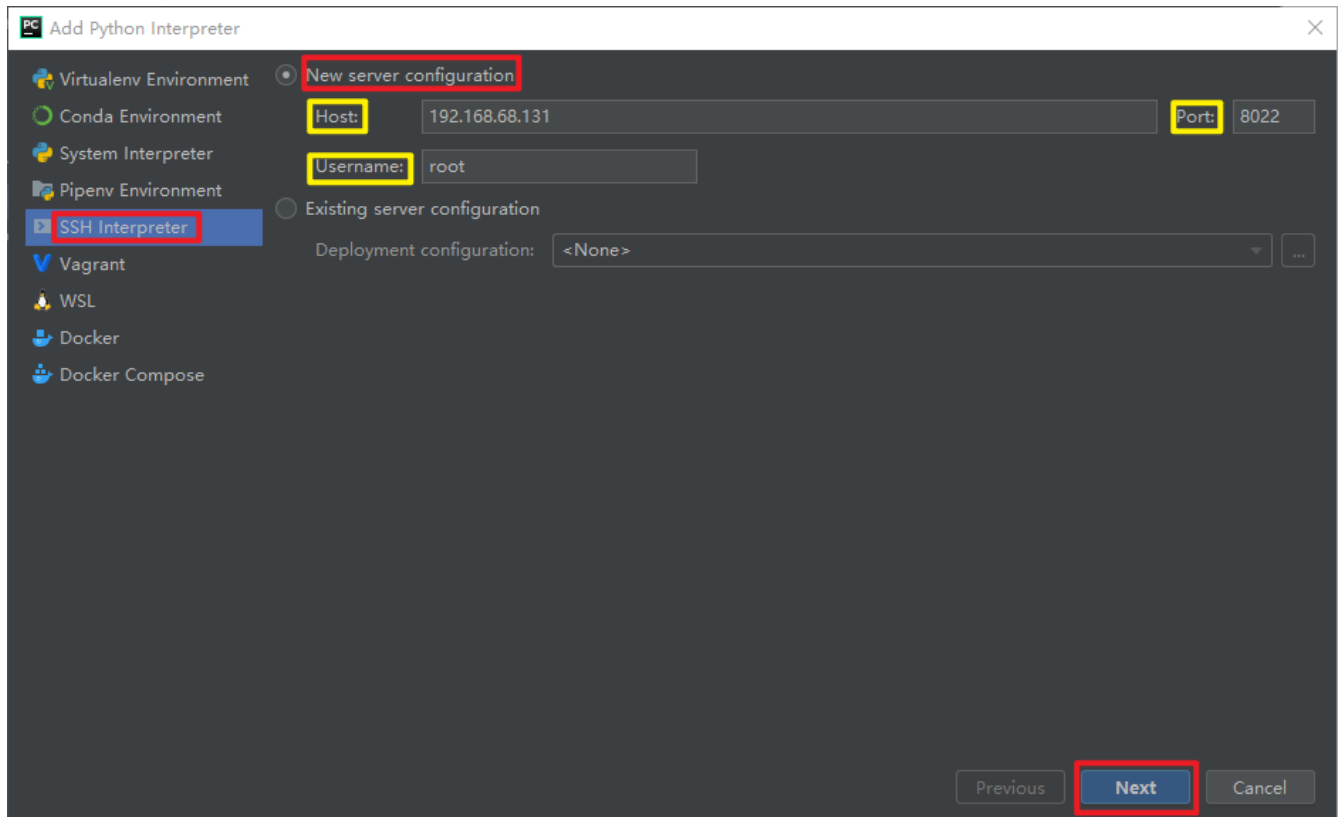
Host 为宿主机的IP地址，可用 `ifconfig` 命令获得，之前说过了

Port 转发端口为之前 新建Docker 时设置的宿主机端口22的转发端口

`docker run ... -p xxx:22 ...` 命令中的 xxx 即是配置的转发端口

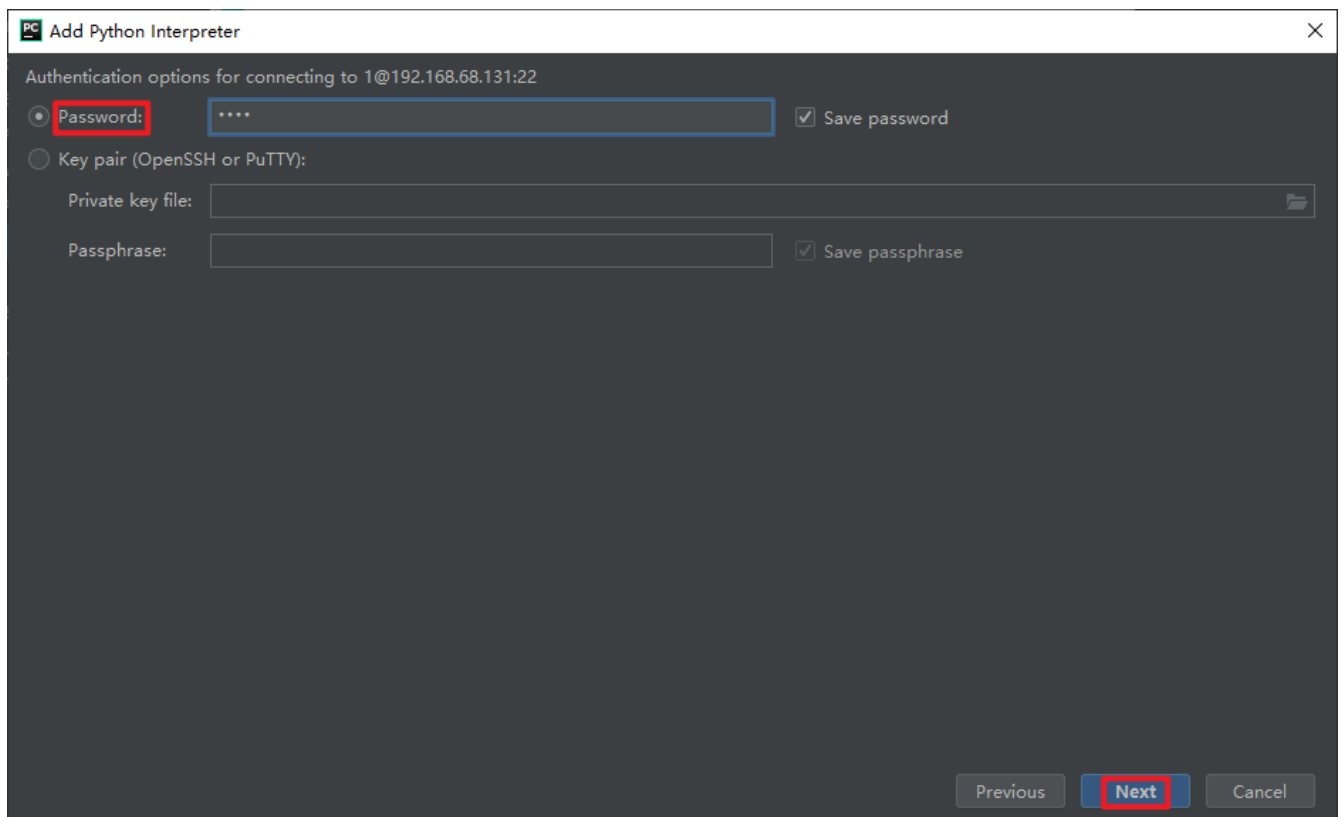
本步及下一步的账号密码请填写你在SSH配置中用 `echo "user:password" | chpasswd` 命令配置的 [用户名:密码]





4. Password ---> Next

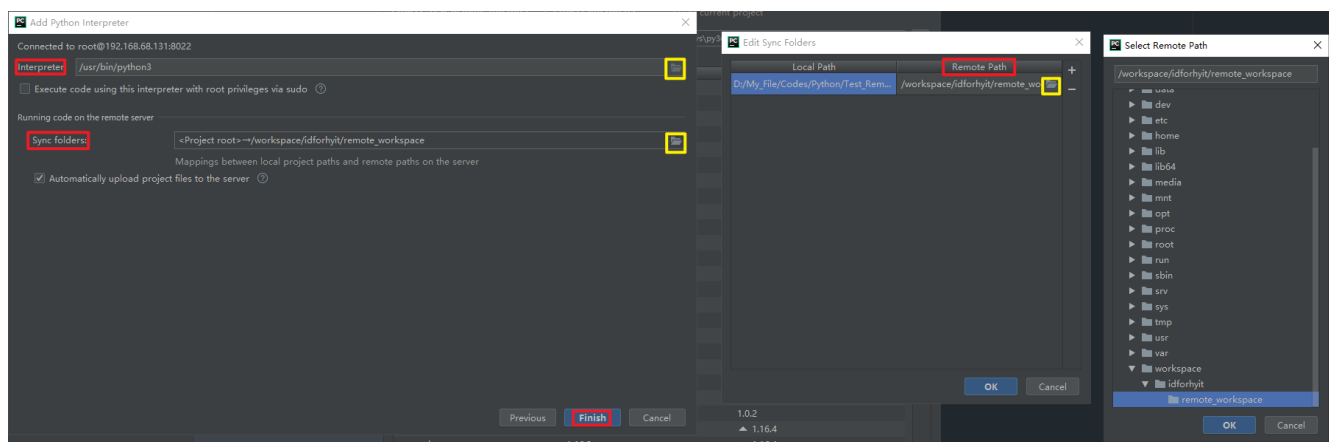
建议 Save password 复选框选中



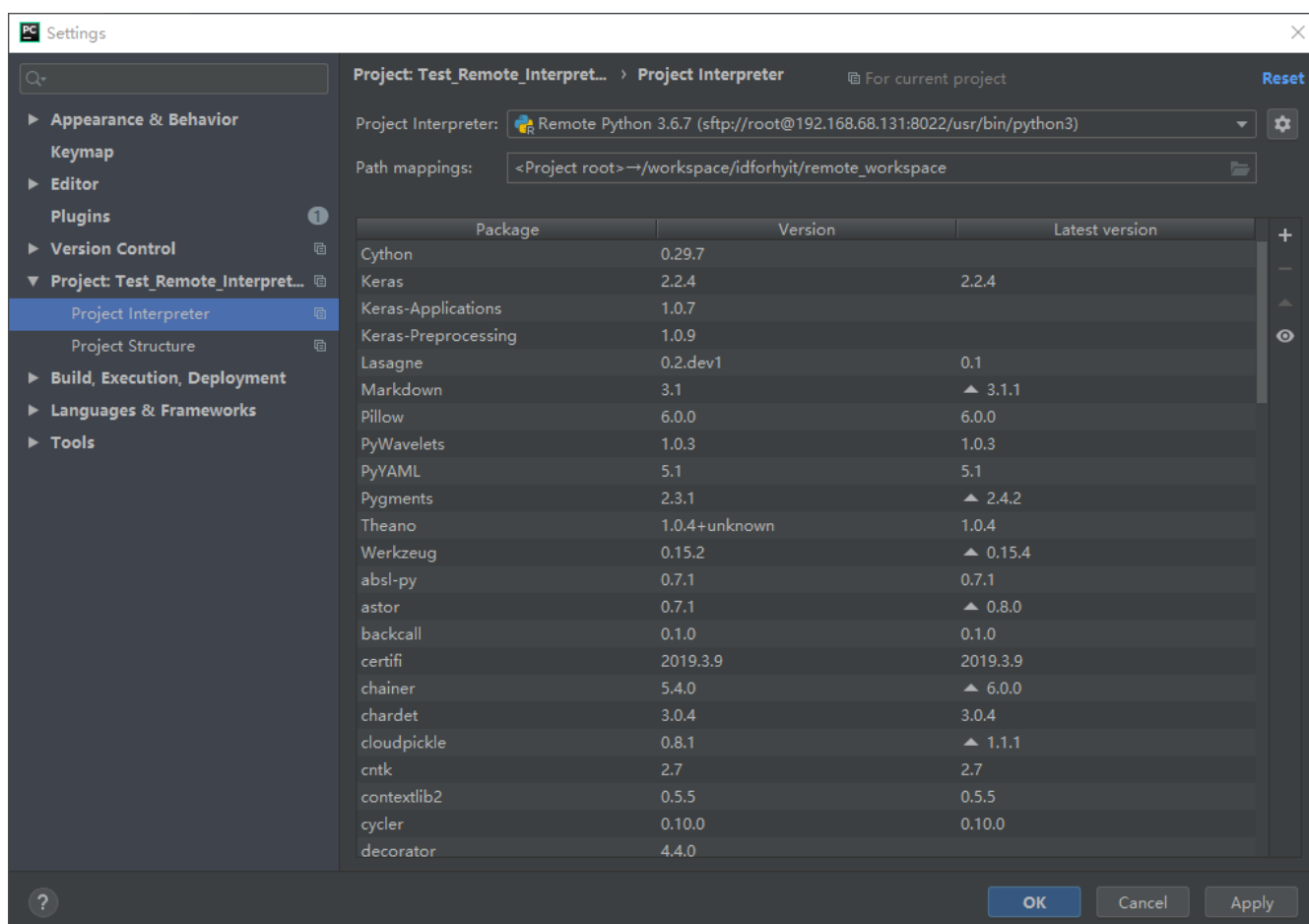
5. 配置Interpreter ---> 配置Sync folders ---> Finish

Interpreter 是远程解释器的地址，一般是 /usr/bin/python 或 /usr/bin/python3

**Sync folders** 是远程解释器的项目存放目录，一般与之前创建Docker时用到的命令地址一样就可，也可点击文件夹图标，选定远程服务器的目录地址。这个地址是你代码的远程同步地址，项目运行时会同步你的代码到此目录，然后用远程解释器的算力进行运算。



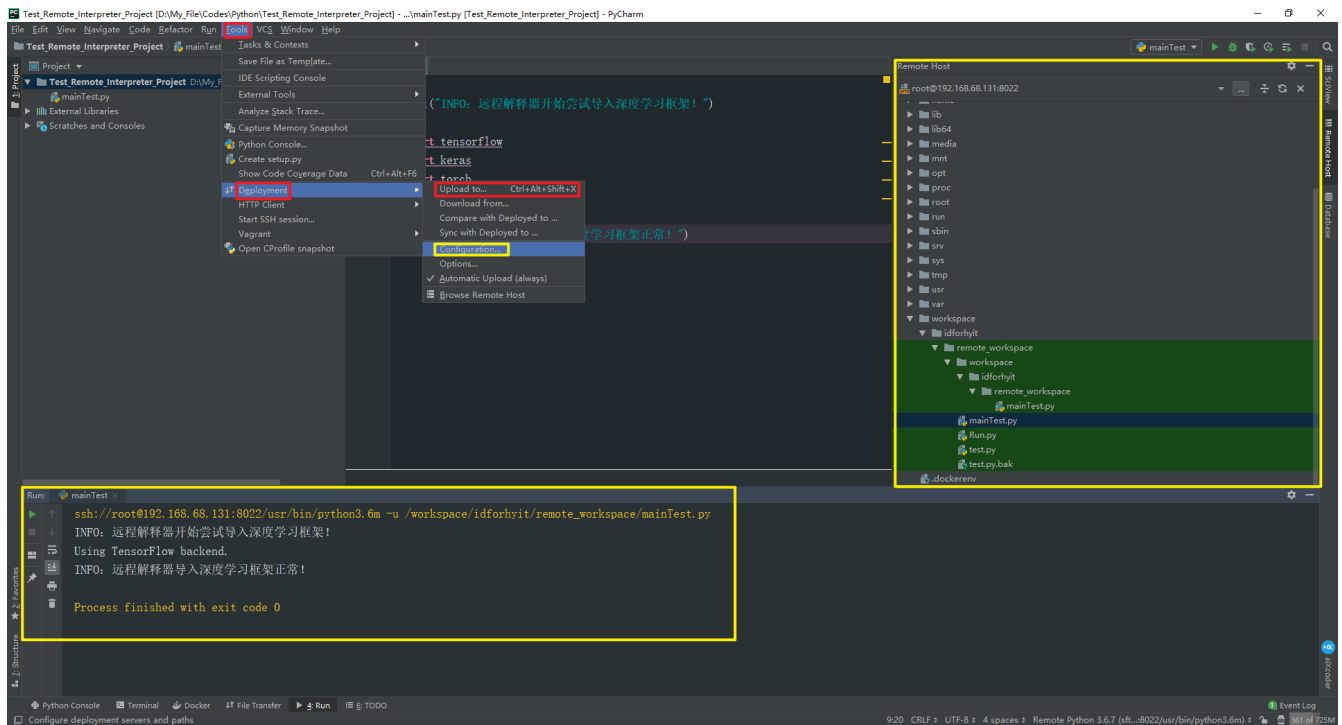
## 6. 配置完的效果及代码执行界面



点击 菜单栏 ---> **Tools** ---> **Deployment** ---> **Configuration..** 可进一步配置。

运行代码前，点击 菜单栏 ---> **Tools** ---> **Deployment** ---> **Upload to...** 或 **Ctrl+Alt+Shift+x** 将代码同步到远程服务器的配置目录。

最后，整个远程解释器执行代码时的界面如下图所示。代码执行不消耗本地资源，与本地孱弱的性能相比，远程服务器跑实验模型的时间大大缩短。如果学校或是有其他云平台的算力，能够拿到云平台的算力分配给俺们，还是比较美滋滋的，会节省很多时间。



## 如有错误或疑问，烦请评论指出

本篇文章纯手码，请勿转载。文章内容来自博主的亲自操作，查阅了很多资料，但作为菜鸟一枚，肯定还存在许多错误，大佬轻喷，如见错误或操作不一致的地方，烦请在[本篇博文地址](#)的文章底部评论指出，会及时于本博文地址进行更正。