

0.1 准备工作

环境

云服务器&Linux

使用Linux的云服务器是一个简单方便的方式来学习Python，原因有下述几条：

1. 使用云服务器可以不用担心搞崩溃自己的电脑
2. 使用Jupyter Notebook等方式能够随时在多种环境下通过网页浏览器访问（例如随身带着的Surface等进行联网访问）
3. 能够稍微熟悉Linux环境（此条不够强烈，仅仅是个人习惯）

本系列基于Ubuntu18.04，使用Anaconda3来进行管理环境，Python主要使用3.6.8，其他版本选修

（当然Windows或者OS也可，只是后续内容中请自行适配修改部分细节，例如文件路径等等）

云服务器选择

- 学生党：
阿里云、腾讯云、华为云等多家云服务器都提供了针对学生的优惠，大体上都是10元每月，配置大约都为1核心的CPU，1-2GB内存，40-50GB存储空间，足够了。
- 已毕业：腾讯云初次使用也有类似的优惠
- 土豪玩家/专业人士：Linux系统的主机（当然苹果的OS也可，但是我不怎么熟悉。。）

本教程环境以及搭建方式

必备工具：

1. XShell:
远程连接服务器的工具

推荐工具:

1. 一个写Markdown的工具
Markdown可以理解作为一种用特定格式边写边排版的语言，我是直接使用的印象笔记
2. NotePad++
方便的记事本工具，但是非常适合程序工作

搭建方式

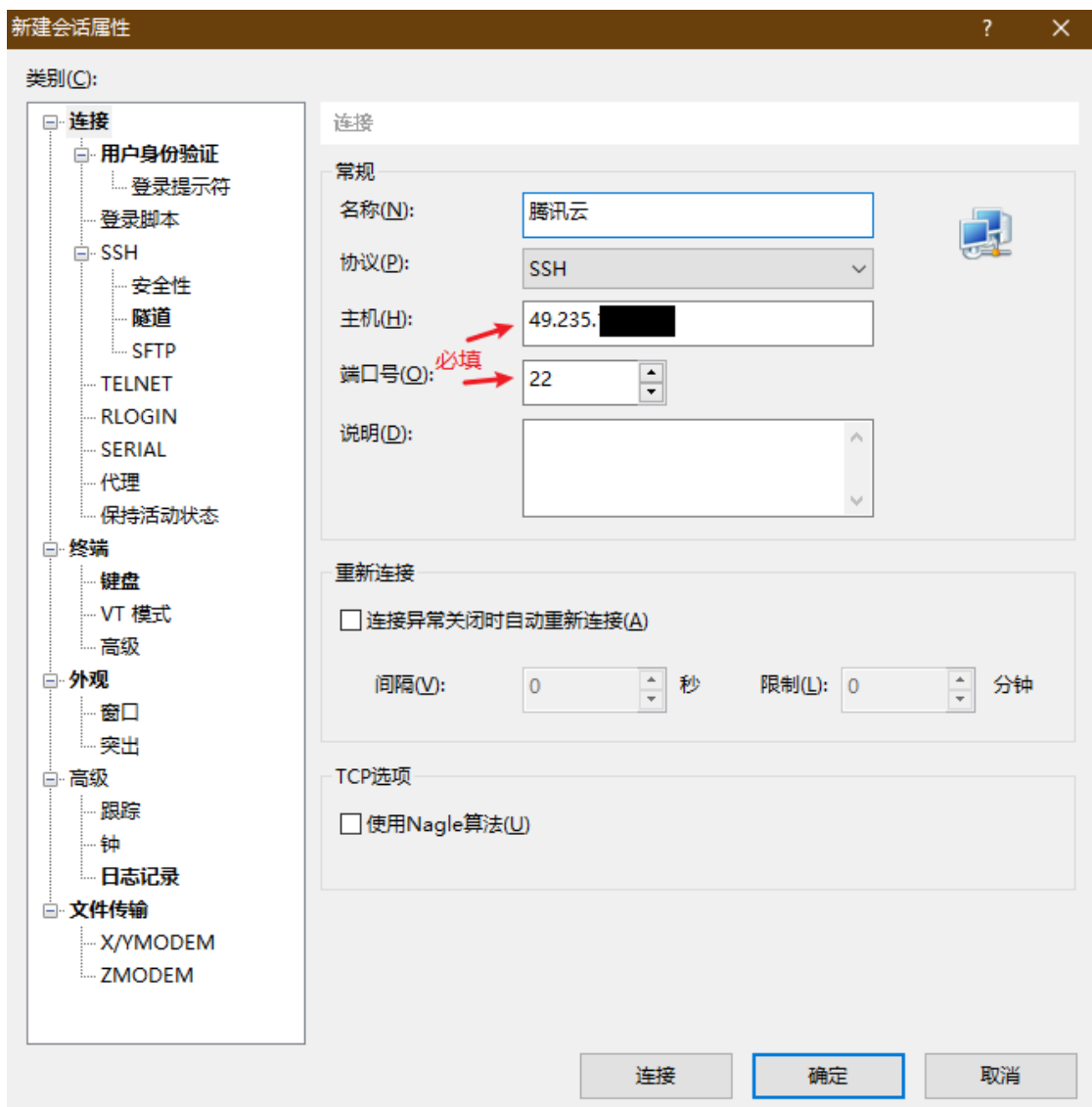
(以腾讯云Ubuntu18.04为例)

1. 首先购买服务器

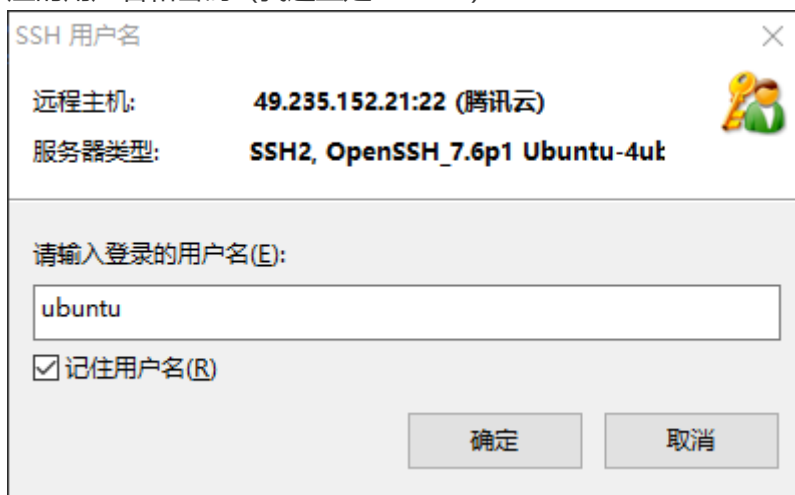
按照上一节说的，选择一个适合自己的服务器即可，系统本文是以Ubuntu18.04为例的，之后，进入到控制台页面，打开“实例”页面，能够在不同区域发现自己所购买的服务器，需要记住的是：

1. 公网IP
2. 登录方式（ssh的用户名和密码）[腾讯云指南](#)，其他平台可以对应参考各自的指南

2. 打开XShell



将服务器的IP粘贴如图，点击链接（如果弹窗提示，**接受并保存即可**），并输入指南中对应的用户名和密码（我这里是ubuntu）



如果是自己的电脑，可以选择记住

SSH用户身份验证

?

×

远程主机:

49.2 (腾讯云)

登录名:

ubuntu

服务器类型:

SSH2, OpenSSH_7.6p1 Ubuntu-4ubuntu1



请在下面选择恰当的身份验证方法并提供登录所需的信息。

☒ Password(P)

密码(W):

.....

☐ Public Key(U)

用户密钥(K):

浏览(B)...

密码(H):

☐ Keyboard Interactive(I)

使用键盘输入用户身份验证。

☒ 记住密码(R)

确定

取消

连接之后就会见到看起来非常酷炫的各种信息了：

```
Xshell 6 (Build 0184)
Copyright (c) 2002 NetSarang Computer, Inc. All rights reserved.

Type 'help' to learn how to use Xshell prompt.
[D:\~]$ 你的服务器IP地址
Connecting to 49.23[REDACTED] 22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+J'.

Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-54-generic x86_64)
系统版本

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Sat Feb  8 23:15:01 CST 2020

System load:  0.0                Processes:            90
Usage of /:   4.6% of 49.15GB    Users logged in:     1
Memory usage: 8%                IP address for eth0: 172.1[REDACTED]
Swap usage:   0%                内网IP地址, 不重要

* MicroK8s 1.15 is out! Thanks to all 40 contributors, you get the latest
greatest upstream Kubernetes in a single package.

https://github.com/ubuntu/microk8s

Last login: Sat Feb  8 23:09:50 2020 from 119.2[REDACTED] 上次登录信息
/usr/bin/xauth:  file /home/ubuntu/.Xauthority does not exist
ubuntu@VM-0-6-ubuntu:~$ █
↑
你的用户名
```

3. 搭建

接下来就是看起来很炫的敲命令操作了，选择XShell的原因是因为XShell非常方便支持复制粘贴等，右键单击任意一处空白就可以发现很多方便的操作。

```
clear # 清空屏幕, 井号代表注释, 不会实际执行
```

```
ls # 列出当前目录下的内容
```

```
ll # 实际上是 ls -l 命令的简写, 可以更详细的列举出内容
```

linux的命令有不懂的直接搜索,就会有很详细的教程.

接下来进入正式环境搭建过程.

```
clear # 清空屏幕
mkdir downloads # 在当前目录建立一个名为downloads的文件夹
cd downloads # 进入downloads文件夹
```

注意：云服务器为单核CPU的请使用本段：

由于Anaconda3的一个bug，我们使用的并非最新版本

```
wget https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2019.07-Linux-x86_64.sh # wget是一个下载用的命令
```

注意：下述内容使用于多核CPU

在浏览器中打开[Anaconda3](https://repo.anaconda.com/archive/Anaconda3-2019.10-Linux-x86_64.sh)下载页面,并选择Linux,可以看到是Python3.7版本(截止2020年2月8日),右键复制Download的链接地址,发现是一个结尾为.sh的文件.

https://repo.anaconda.com/archive/Anaconda3-2019.10-Linux-x86_64.sh

在XShell中:

```
wget https://repo.anaconda.com/archive/Anaconda3-2019.10-Linux-x86_64.sh #
wget是一个下载用的命令,后面的连接可以对应替换成你需要的版本也可以.
```

接下来就是一片漫长的等待.

如果嫌弃下载速度太慢,也可以在[清华镜像站](https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2019.10-Linux-x86_64.sh)进行下载对应的版本.

例如我这里就是

https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2019.10-Linux-x86_64.sh

所以对应的命令也变成了

```
wget https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2019.10-Linux-x86_64.sh
```

注意结束

下载结束后

```
bash ./Anaconda3-*Linux-x86_64.sh # 一个字一个字敲很麻烦对不对,其实输入个
Ana之类的然后按Tab键就会自动补全到无差异的地方,如果只有不重复的文件,比如今
在,就会自动补全了
```

然后根据提示,需要注意的是请务必先看完下述截图

首先是按了Enter之后会是一段很长的声明,按空格可以很快滚过去这一段。

然后需要输入yes

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

然后这里开始，各种按键都不要随便乱按了!!!

```
Anaconda3 will now be installed into this location:
/home/ubuntu/anaconda3

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

[/home/ubuntu/anaconda3] >>> 
```

只按一步回车然后等，等到安装结束之后会有这样一个填空

```
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>> 
```

这里需要输入yes(如果没有搞定的话...问题就有点麻烦了...参见下方的指导方案)

然后输入

```
source ~/.bashrc # 这一步是重新加载bash环境
```

如果说顺利的话,你的用户名前面会多出一个(base)

```
(base) ubuntu@VM-0-6-ubuntu:~/downloads$ 
```

然后

```
conda -V # 显示conda版本
```

```
(base) ubuntu@VM-0-6-ubuntu:~/downloads$ conda -V
conda 4.7.10
(base) ubuntu@VM-0-6-ubuntu:~/downloads$ 
```

如果失败了:

```
sudo vi ~/.bashrc
```

然后一直按键盘上的PageDown直到文件最末端,查看是否有

```
# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$(('/home/ubuntu/anaconda3/bin/conda' 'shell.bash' 'hook' 2>
/dev/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/home/ubuntu/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/home/ubuntu/anaconda3/etc/profile.d/conda.sh"
    else
```

```
export PATH="/home/ubuntu/anaconda3/bin:$PATH"
fi
fi
unset __conda_setup
# <<< conda initialize <<<
```

```
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup='${HOME}/anaconda3/bin/conda' 'shell.bash' 'hook' 2> /dev/null''
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/home/ubuntu/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/home/ubuntu/anaconda3/etc/profile.d/conda.sh"
    else
        export PATH="/home/ubuntu/anaconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<

~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

⋮wq

平台有,每个平台还有不同的镜像源,就像是不同货仓.清华源或者阿里云源以及中科大源等等都可以,这里使用隔壁的镜像是个人习惯原因.

apt

[清华源ubuntu](#)

选择对应版本后 (我们使用的是Ubuntu18.04)

```
sudo mv /etc/apt/sources.list /etc/apt/sources.list.bak # 将原本的源列表备份
(通过移动实现)
sudo vi /etc/apt/sources.list # 编辑源列表, 因为原来的源列表已经被移动(改名), 此时不存在这个文件, 所以会打开一个空文件
```

键盘输入 **i** 进入insert模式

鼠标右键粘贴对应清华源列出的源列表,此处是Ubuntu18.04对应的

```
# 默认注释了源码镜像以提高 apt update 速度, 如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted
universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted
universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main
restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main
restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main
restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main
restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main
restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main
restricted universe multiverse

# 预发布软件源, 不建议启用
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-proposed main
restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-proposed main
restricted universe multiverse
```

同样ESC退出insert模式之后,输入 **:wq** 退出

pip

首先测试是否有安装pip

```
pip -V
```

如果提示不存在,请参考其他介绍安装pip

换源:

```
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple pip -U  
pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
```

Anaconda

```
conda config --set show_channel_urls yes  
vi ~/.condarc
```

然后同样进入编辑模式,删除所有内容(可以适当使用[ggdG](#))

然后输入*i* 进入编辑模式,右键粘贴

```
channels:  
  - defaults  
show_channel_urls: true  
channel_alias: https://mirrors.tuna.tsinghua.edu.cn/anaconda  
default_channels:  
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main  
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free  
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r  
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/pro  
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/msys2  
custom_channels:  
  conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud  
  msys2: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud  
  bioconda: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud  
  menpo: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud  
  pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud  
  simpleitk: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

同样ESC退出insert,:wq 保存退出

搭建虚拟环境

虚拟环境是非常重要的一步,如果说把开发类比成为做饭,那么开发环境就可以理解成厨房.base这个环境可以理解成系统环境,可以当做你的卧室,如果在做饭的时候不小心把水洒了出来,洒到了床上或者电脑里后果是非常严重的(前者可能要挨一顿揍,后者可能挨更狠的揍).

那么虚拟环境可以理解为单独为了做饭生成一个厨房,这个厨房最开始的时候可能什么都没有,根据你的需求可以放烤箱放插座等等,如果你满意这个环境,还可以原封不动的复制一份,如果搞砸了什么,把厨房直接删掉就好了.不用担心会连带着把卧室也毁了.

而且对用不同的需求,比如你希望都用两孔插座(Python2),三孔插头(Python3)是没法用的,那么有两个不同的厨房(虚拟环境)就可以解决这个问题.

我们建立一个Python3.6.8的虚拟环境,这里我们命名为py36

```
conda create -n py36 ipykernel python==3.6.8 -y # 建立一个名为py36的环境,python版本为3.6.8,对于过程中的确认都选择yes
source activate py36
python -m ipykernel install --user --name=py36
conda deactivate # 安装成为ipython kernel
```

完成后,我们查看一下当前的环境信息

```
conda info --envs
```

发现有了py36环境,有了环境需要进入"厨房"才行.

```
source activate py36
```

退出虚拟环境时

```
conda deactivate
```

4. 启动Jupyter Notebook

Anaconda3是自带了Jupyter Notebook的,我们只需要进行一些配置并让它在后台运行即可

首先设置密码

```
ipython
```

进ipython界面后

```
from notebook.auth import passwd
```

```
passwd() # 注意,这里是看不到输入的,放心大胆输入完
```

输入两次密码后,会返回一串字符串,将这一串字符串完整复制到一个txt(建议使用NotePad++)中下面代码中的password部分

```
c.NotebookApp.ip='*' # 允许任何IP访问
c.NotebookApp.open_browser = False # 启动时不打开浏览器
c.NotebookApp.port =8888 # 监听端口
c.NotebookApp.password='' # 密码
c.NotebookApp.allow_remote_access = True # 是否允许远程访问
import os
c.NotebookApp.notebook_dir = os.path.expanduser('~') #工作目录直接设定在用户
目录
```

然后按Ctrl+Z退出ipython

```
jupyter notebook --generate-config
vi ~/.jupyter/jupyter_notebook_config.py
```

同样,进入insert模式后,将刚刚在txt界面中的一大段插入到最开头,如图

```
c.NotebookApp.ip='127.0.0.1' # 监听地址
c.NotebookApp.open_browser = False # 启动时不打开浏览器
c.NotebookApp.port =8888 # 监听端口
c.NotebookApp.password='sha1:594239262648:fb0b947c4f36df9935bb57af599d4cf0ac007ddf' # 密码
c.ContentsManager.root_dir = '/home/jupyter/Note' # 工作目录
c.NotebookApp.base_url = '/notebook/' # 访问基本路径
c.NotebookApp.allow_remote_access = True # 是否允许远程访问[]

# Configuration file for jupyter-notebook.

#-----
# Application(SingletonConfigurable) configuration
```

同样保存退出.

```
jupyter notebook # 启动jupyter notebook
```

如果发现顺利提示启动,我们搭建的环节就成功了.接下来要解决访问的问题(防火墙)根据不同平台的设定,增加TCP:8888端口的访问即可,可以参考[腾讯云对于Jupyter Notebook的指南](#)

我们在浏览器中输入你的ip:8888

然后回车,就可以访问到jupyter notebook了.

这样测试了一下可以运行之后,我们要将jupyter notebook转为后台运行

Ctrl+C停止运行后

```
nohup jupyter notebook &
```

之后就可以随时访问(只要服务器不挂掉)

我们可以首先尝试在工作目录下新建一个文件夹并改名为Projects

随后在Projects下建立一个新的Jupyter Notebook,选择py36虚拟环境,并单击Untitled部

分改名为0.1 Hello World

开始我们的第一段代码:

```
import sys
print("Hello World!\nin {}".format(sys.version))
```

```
In [1]: import sys
print("Hello World!\nin {}".format(sys.version))

Hello World!
in 3.6.8 |Anaconda, Inc.| (default, Dec 30 2018, 01:22:34)
[GCC 7.3.0]
```