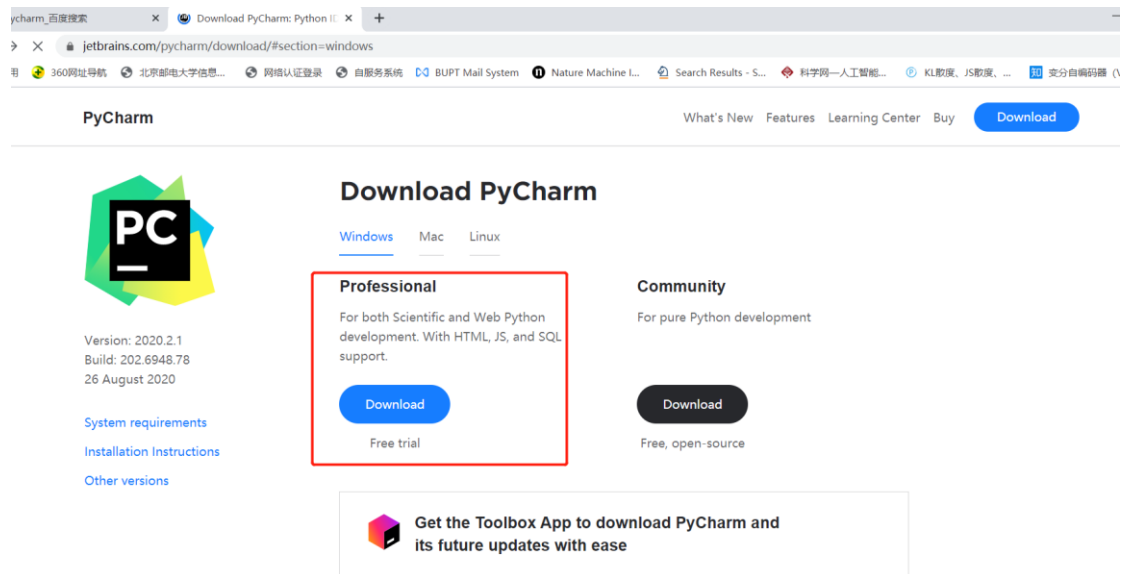


一、使用服务器上的 Python 环境跑代码的教程

- (1) 首先安装 Pycharm。去 Pycharm 官网 (<https://www.jetbrains.com/>) 下载 Pycharm Professional 版本。



虽然专业版收费，但是咱们有学校的教育邮箱 (xxx@bupt.edu.cn)，使用教育邮箱注册即可免费使用（一年评估一次，只要教育邮箱有效，就一直能使用）。

- (2) 申请免费使用 Pycharm

登录 pycharm 官网，点击右上角的 buy。



点击下图红框中的内容下面的 learn more。

Toolbox Subscription

Choose your country/region

China

**中文服务**
支付宝

JetBrains 致力于为开发者打造最高效智能的开发工具，总部以及主要研发中心位于欧洲的捷克。JetBrains 针对个人开发者及企业组织提供不同的授权方式。我们提供多种付款方式，包含银联卡、各种国际信用卡、PayPal、支付宝和微信支付。企业用户也可通过银行汇款的方式完成付款。若您有任何购买或授权上的疑问，欢迎联系我们的中文销售代表为您服务。
[立即联系中文销售代表](#)

✕

[For Organizations](#) [For Individual Use](#) [Special Offers](#) [FAQ](#)**For students and teachers**

FREE

Students and academic staff members are eligible to use all JetBrains tools free, upon verification of their

For classroom assistance

FREE

Universities, colleges, schools, and non-commercial educational organizations are eligible for free

For Open Source projects

FREE

Non-commercial open source projects can qualify for free licenses to all JetBrains tools if they meet the

点击 [Apply Now](#)

Free License Programs

[Academic Licensing](#) [Open Source](#)

teachers

Get free access to all JetBrains IDEs for personal use at school or at home.

Who can get free individual licenses for education

Students and faculty from accredited educational institutions (high schools, colleges, and universities) are welcome to apply.

Students need to be enrolled in an accredited educational program that takes one or more years of full-time study to complete.

Not sure about the license terms? [Check out the FAQ](#) or read the full terms [here](#).

[Apply now](#)

然后填写表单信息，在红框内填写学校的邮箱

Apply with: UNIVERSITY EMAIL ADDRESS ISIC/ITIC MEMBERSHIP OFFICIAL DOCUMENT GITHUB

Status: ☒ I'm a student ☐ I'm a teacher

Level of study:

Is Computer Science or Engineering your major field of study?
☒ Yes ☐ No

Graduation date:
Choose expected graduation date.

Email address:
I certify that the university email address provided above is valid and belongs to me.

Name:
Your name as it appears in your passport, driver's license, or other legal documents.

注册完之后会在邮箱收到确认邮件，然后确认完邮件后就注册成功。

(3) 最后在安装注册完成之后打开 pycharm，在如下界面登录刚刚注册的账户就行了。

License Activation

PyCharm
Activate

☒ Activate PyCharm ☐ Evaluate for free [Buy license ↗](#)

Get license from: ☒ JB Account ☐ Activation code ☐ License server

Username / email:

Password: [Forgot? ↗](#)

(4) 在连接服务器之前，先在服务器上创建属于自己的 anaconda (python) 环境

创建环境的命令为 `conda create --name <环境名> <package_name>`

显示已创建环境的命令为 `conda info --envs`

删除环境的命令为 `conda remove --name <环境名> --all`

切换到指定环境的命令为 `conda activate <环境名>`

退出当前环境的命令为 `conda deactivate`

比如现在创建一个 python 版本 3.7, gpu 版本的 tensorflow2.0 的环境，操作如下：

(a) 先使用 Xshell 连接到服务器，然后输入如下指令：

```

declare -x QTINC="/usr/lib64/qt-3.3/include"
declare -x QTLIB="/usr/lib64/qt-3.3/lib"
declare -x QT_GRAPHICSSYSTEM_CHECKED="1"
declare -x SELINUX_LEVEL_REQUESTED=""
declare -x SELINUX_ROLE_REQUESTED=""
declare -x SELINUX_USE_CURRENT_RANGE=""
declare -x SHELL="/bin/bash"
declare -x SHLVL="1"
declare -x SSH_CLIENT="10.28.192.84 12830 22"
declare -x SSH_CONNECTION="10.28.192.84 12830 10.112.3.105 22"
declare -x SSH_TTY="/dev/pts/1"
declare -x TERM="xterm"
declare -x USER="root"
declare -x XDG_DATA_DIRS="/root/.local/share/flatpak/exports/share:/var/lib/flatpak/exports/share:/usr/local/share:/usr/share"
[root@calhost ~]# conda create --name test python=3.7 anaconda tensorflow-gpu=2.0

```

然后会询问是否安装上面这些包，输入 y，按回车键即可。

```

tensorflow-gpu      anaconda/pkgs/main/linux-64::tensorflow-gpu-2.0.0-h0d30ee_
termcolor            anaconda/pkgs/main/linux-64::termcolor-1.1.0-py37_1
terminado            anaconda/pkgs/main/linux-64::terminado-0.8.3-py37_0
testpath             anaconda/pkgs/main/noarch::testpath-0.4.4-py_0
threadpoolctl        anaconda/pkgs/main/noarch::threadpoolctl-2.1.0-pyh5ca1d4c_
tk                   anaconda/pkgs/main/linux-64::tk-8.6.10-hbc83047_0
toml                 anaconda/pkgs/main/noarch::toml-0.10.1-py_0
toolz                anaconda/pkgs/main/noarch::toolz-0.10.0-py_0
tornado             anaconda/pkgs/main/linux-64::tornado-6.0.4-py37h7b6447c_1
tqdm                anaconda/pkgs/main/noarch::tqdm-4.48.2-py_0
traitlets            anaconda/pkgs/main/linux-64::traitlets-4.3.3-py37_0
typed-ast            anaconda/pkgs/main/linux-64::typed-ast-1.4.1-py37h7b6447c_
typing_extensions    anaconda/pkgs/main/noarch::typing_extensions-3.7.4.3-py_0
ujson                anaconda/pkgs/main/linux-64::ujson-1.35-py37h14c3975_0
unicodescv           anaconda/pkgs/main/linux-64::unicodescv-0.14.1-py37_0
unixodbc             anaconda/pkgs/main/linux-64::unixodbc-2.3.7-h14c3975_0
urllib3              anaconda/pkgs/main/noarch::urllib3-1.25.10-py_0
watchdog             anaconda/pkgs/main/linux-64::watchdog-0.10.3-py37_0
wcwidth             anaconda/pkgs/main/noarch::wcwidth-0.2.5-py_0
webencodings         anaconda/pkgs/main/linux-64::webencodings-0.5.1-py37_1
werkzeug             anaconda/pkgs/main/noarch::werkzeug-0.16.1-py_0
wheel                anaconda/pkgs/main/noarch::wheel-0.35.1-py_0
widgetsnbextension   anaconda/pkgs/main/linux-64::widgetsnbextension-3.5.1-py3
wrapt                anaconda/pkgs/main/linux-64::wrapt-1.11.2-py37h7b6447c_0
wurlitzer            anaconda/pkgs/main/linux-64::wurlitzer-2.0.1-py37_0
xlrd                 anaconda/pkgs/main/linux-64::xlrd-1.2.0-py37_0
xlsxwriter           anaconda/pkgs/main/noarch::xlsxwriter-1.3.3-py_0
xlwt                 anaconda/pkgs/main/linux-64::xlwt-1.3.0-py37_0
xz                   anaconda/pkgs/main/linux-64::xz-5.2.5-h7b6447c_0
yaml                 anaconda/pkgs/main/linux-64::yaml-0.2.5-h7b6447c_0
yapf                 anaconda/pkgs/main/noarch::yapf-0.30.0-py_0
zeromq               anaconda/pkgs/main/linux-64::zeromq-4.3.2-he6710b0_3
zict                 anaconda/pkgs/main/noarch::zict-2.0.0-py_0
zipp                 anaconda/pkgs/main/noarch::zipp-3.1.0-py_0
zlib                 anaconda/pkgs/main/linux-64::zlib-1.2.11-h7b6447c_3
zope                 anaconda/pkgs/main/linux-64::zope-1.0-py37_1
zope.event           anaconda/pkgs/main/linux-64::zope.event-4.4-py37_0
zope.interface       anaconda/pkgs/main/linux-64::zope.interface-5.1.0-py37h7b
zstd                 anaconda/pkgs/main/linux-64::zstd-1.4.5-h9ceee32_0

Proceed ([y]/n)? y

```

稍等片刻即可安装好相应的环境。

(b) 常见命令演示

输入 `conda info --envs` 即可查看安装好的 anaconda 环境的路径，如下图红框中的内容所示：

```
[root@localhost ~]# conda info --envs
# conda environments:
#
base
py_env1
test
tf2.0_env1
tf_1.4
tf_env1
yzw
```

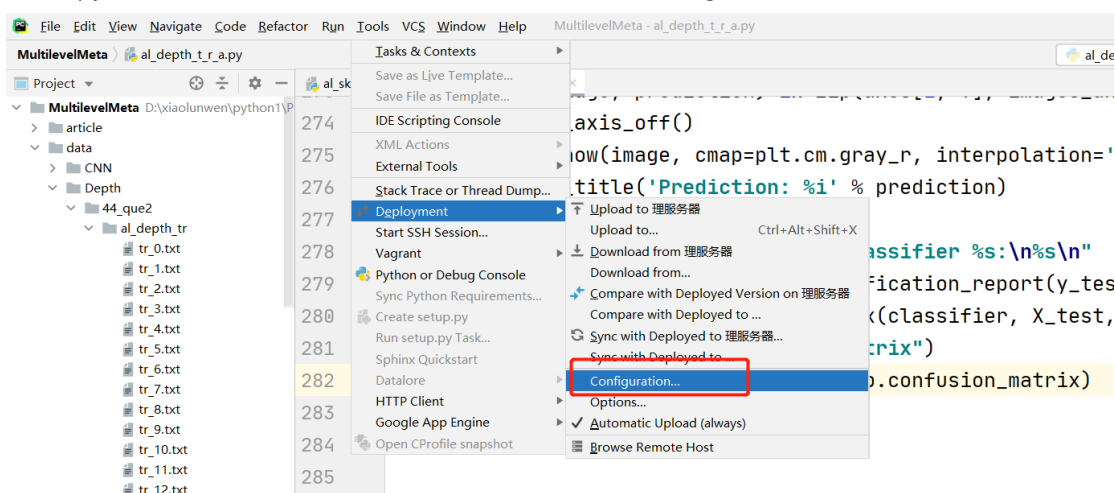
激活指定环境：

```
[root@localhost ~]# conda activate test
You have new mail in /var/spool/mail/root
(test) [root@localhost ~]#
```

退出当前环境：

```
(test) [root@localhost ~]# conda deactivate
[root@localhost ~]#
```

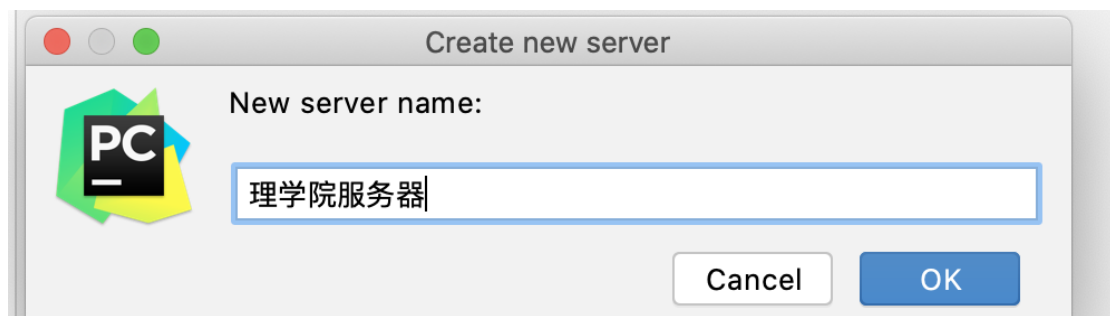
(5) pycharm 配置服务器环境，点击如下图所示的 configuration 选项



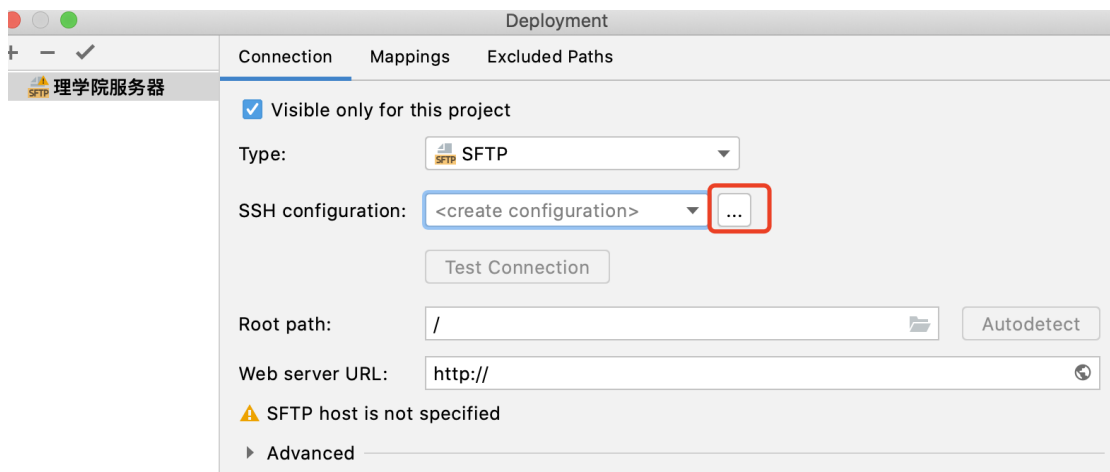
在打开的界面中点击左上角的加号，如下图所示：



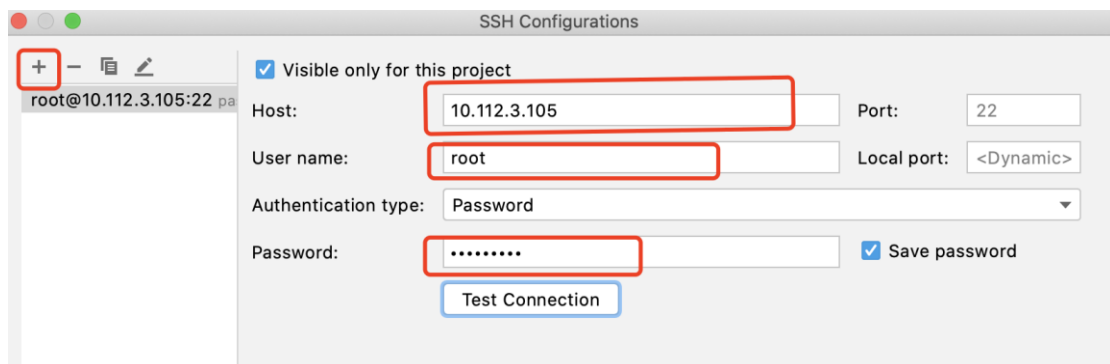
在新打开的对话框内输入服务器的名字，并点击确定：



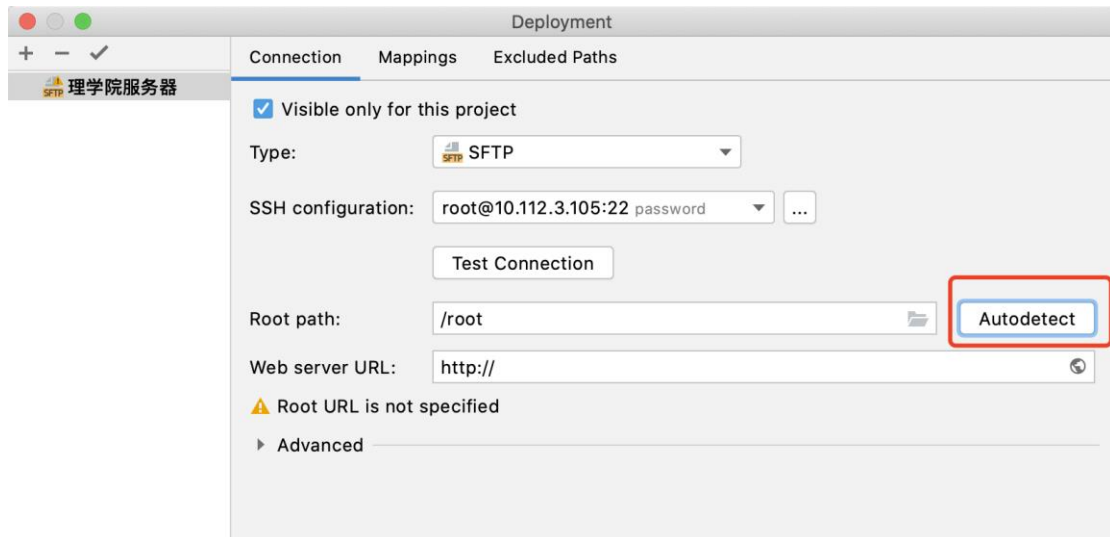
然后点击下图中所示的三个点：



在新弹出的界面中点击左上角的加号，然后输入服务器的 ip，用户名和密码，点击 ok。



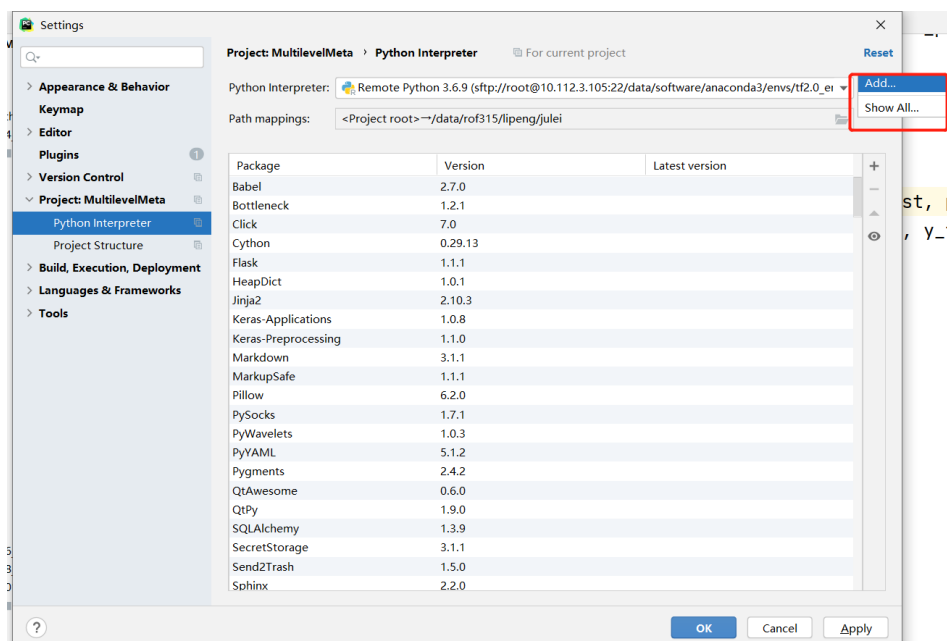
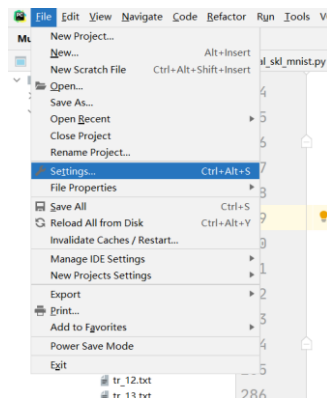
然后点击 Autodetect，最后点击 ok 即可。



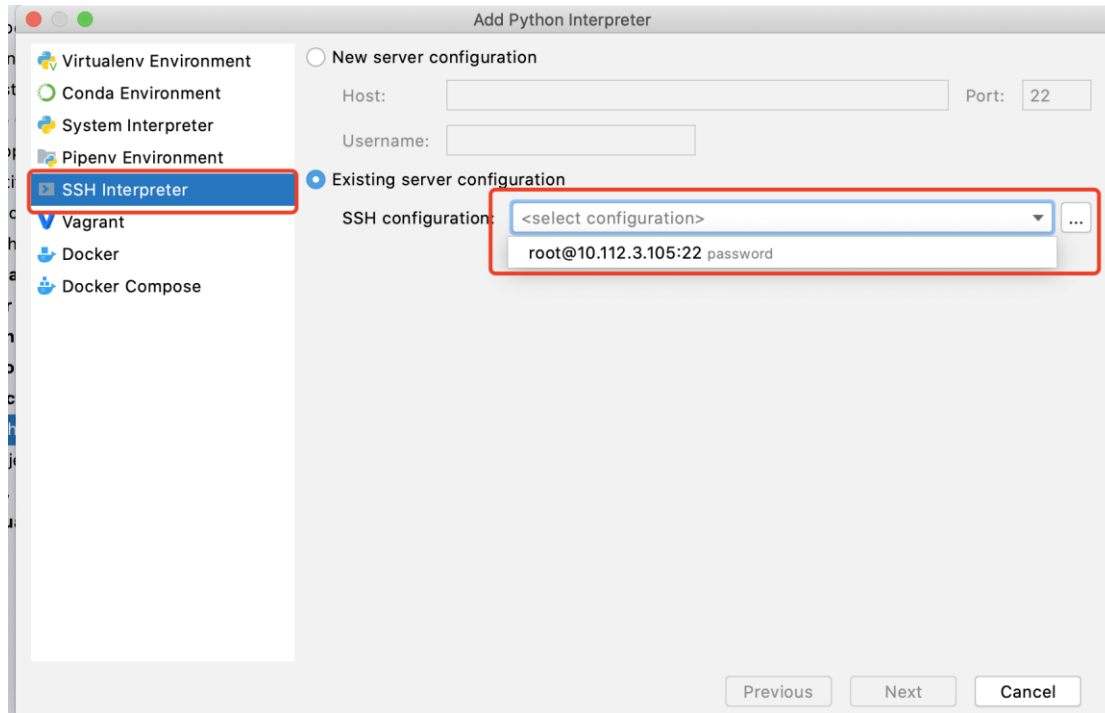
以上就把服务器的环境配置好了。

下面将当前 pycharm 的解释器设置为服务器端的：

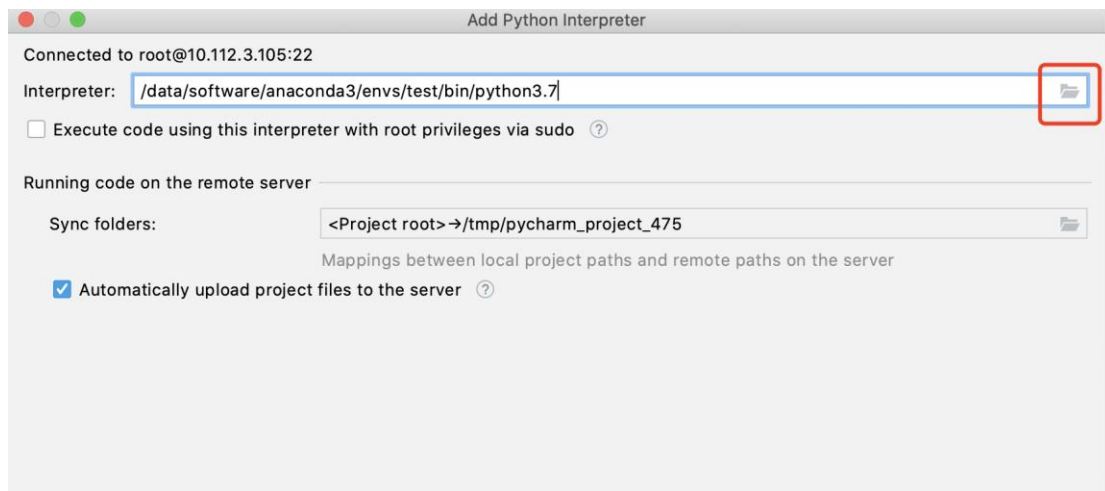
(a) 打开 Settings 界面，点击右上角的齿轮符号中的 add 选项：



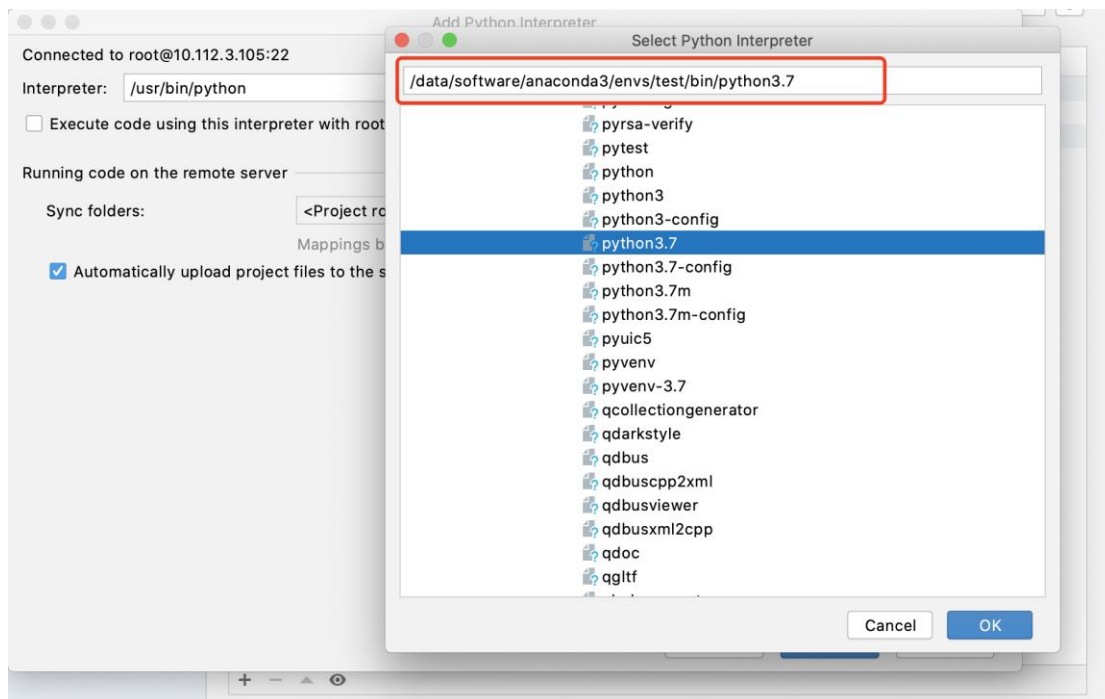
按如下图所示操作，选择下图下拉菜单中的服务器，并点击下一步：



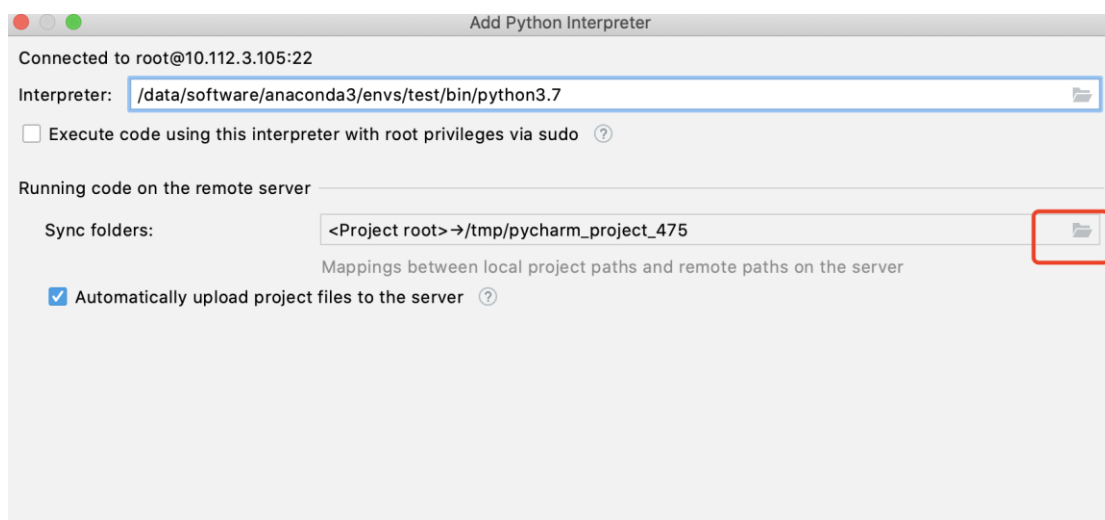
点击如下图所示的文件夹选项，选中服务器上对应 anaconda 环境的 python 程序，以 test 环境为例。上面讲过可以通过输入 `conda info --envs` 找到环境的位置。

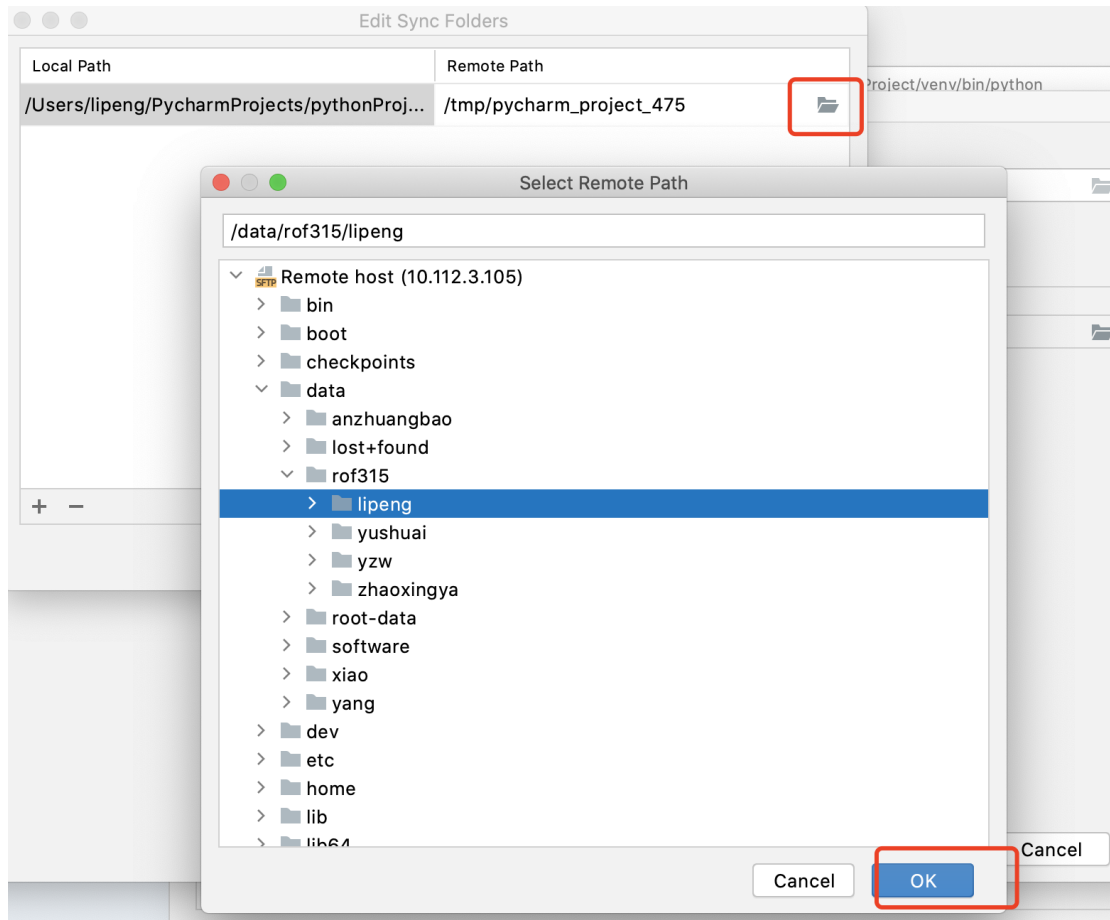


```
[root@localhost ~]# conda info --envs
# conda environments:
#
base                * /data/software/anaconda3
py_env1             /data/software/anaconda3/envs/py_env1
test                /data/software/anaconda3/envs/test
tf2.0_env1          /data/software/anaconda3/envs/tf2.0_env1
tf_1.4              /data/software/anaconda3/envs/tf_1.4
tf_env1             /data/software/anaconda3/envs/tf_env1
yzw                 /data/software/anaconda3/envs/yzw
```

最后选择把当前项目存储在服务器的哪个位置，如下图所示：





选择好后点 ok 即可，然后会花一点时间将当前项目上传到服务器，上传完后，即可像在本地上运行 python 代码一样使用。