

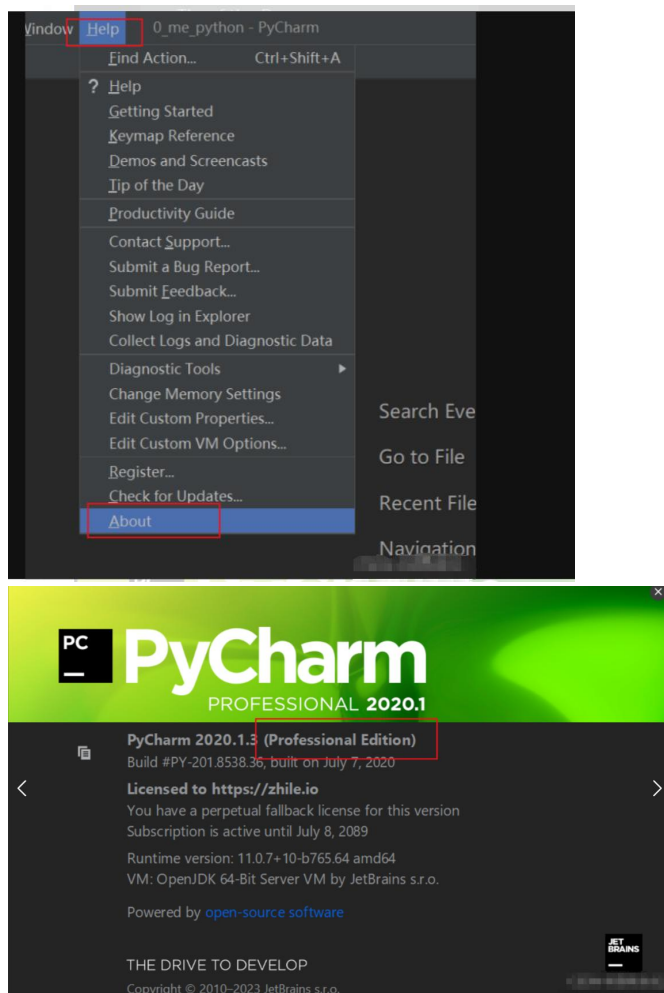
Pycharm 远程服务器手册【英文】

注：这个是针对 Pycharm 是英文的手册

一、建立远程服务器连接

说在前面：Pycharm 分为专业版和社区版，只有专业版才支持远程服务，所以大家可以先确认自己的

Pycharm 版本，可通过下图所示操作查看自己的版本



如果自己的不是专业版本

链接：<https://pan.baidu.com/s/1nb2RZbJpR9mf6zBwxwdCXQ>

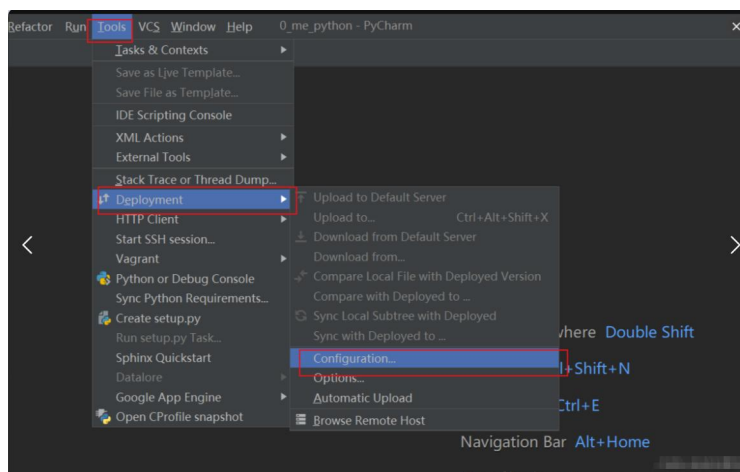
提取码：nxo0

然后开始进行本地与远程服务器之间的连接配置：

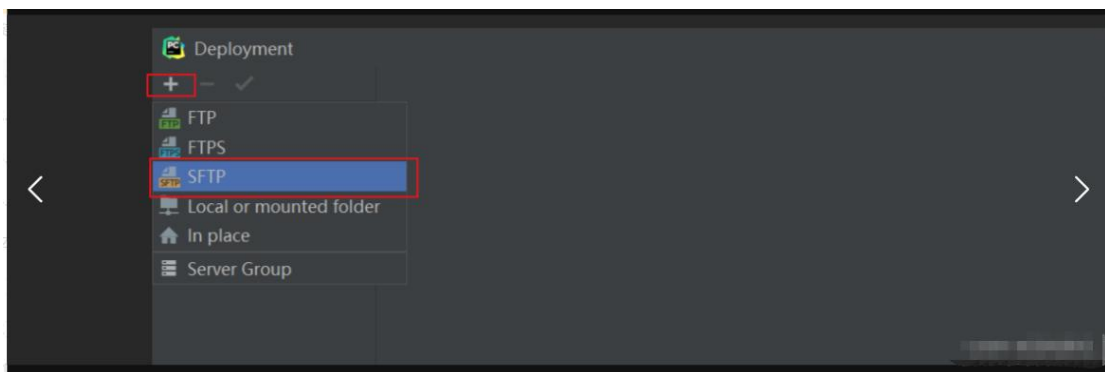
1. 创建远程连接

配置环境的第一部工作是建立本地与远程服务器之间的 SSH 连接，我们知道远程服务器上的任务拥有自己的 SSH 以及用户名，用户名通常为 root，SSH 由 ip:port 两部分组成，ip 为服务器 ip 地址，port 为端口号，我们需要获取这个 SSH 连接后续配置需要使用：

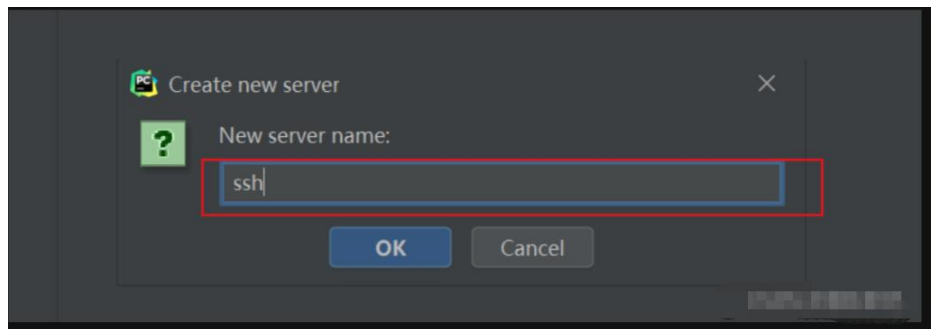
建立 SSH 连接



(1) 选择文件传输协议 SFTP

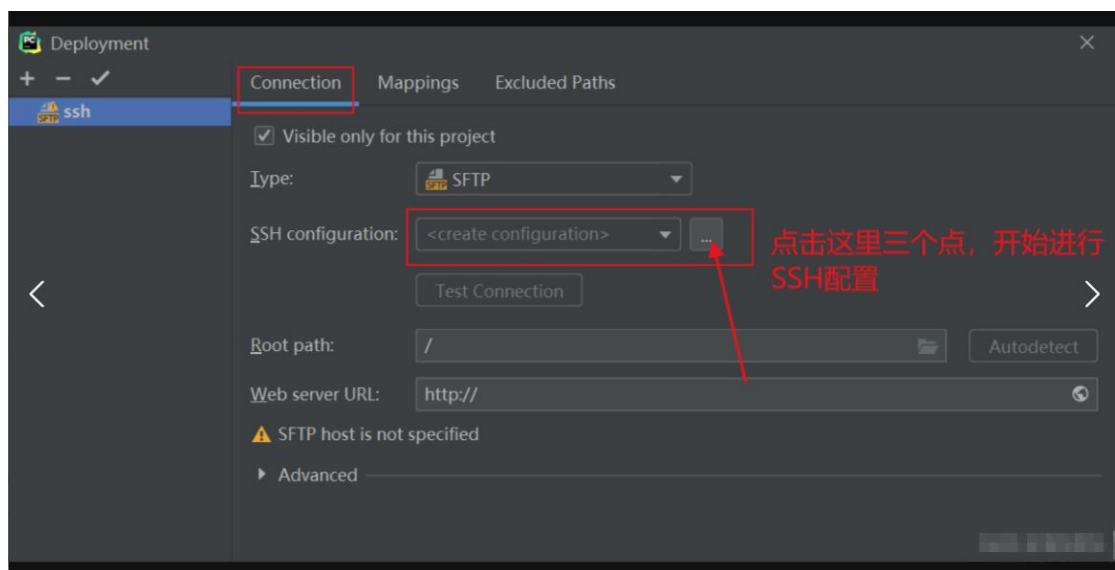


(2) 设置服务器名（这个可以随意命名），此处我命名为 ssh



3) 配置 SSH 连接

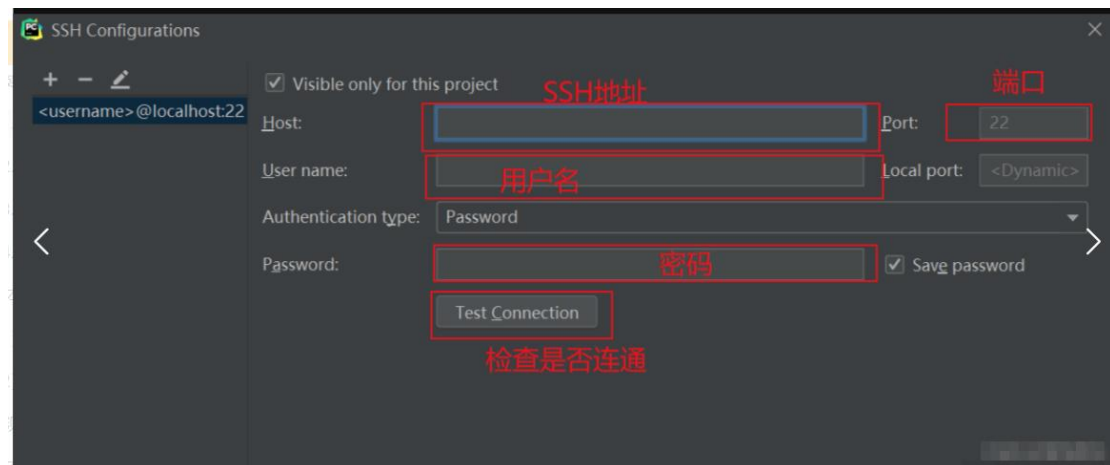
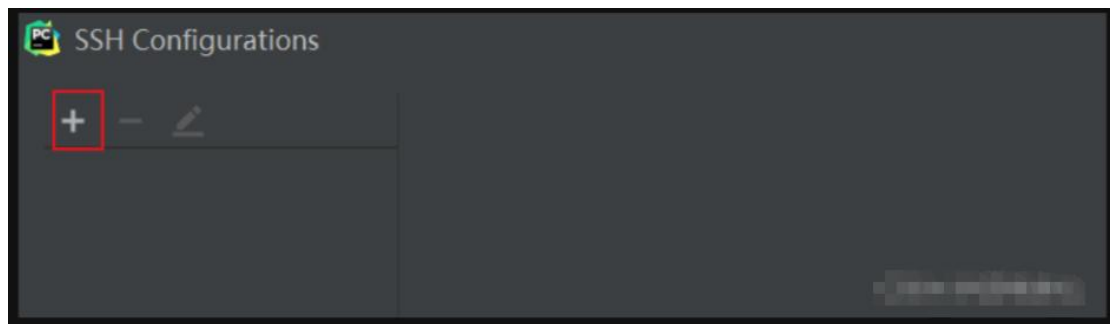
点击 SSH 配置右侧的 "...", 进行 SSH 内容设置:



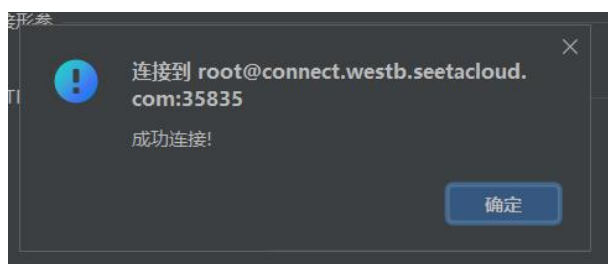
此处我们需要设置三个参数: Host 为前面提到的远程服务器 ip 地址 xxx.xxx.xxx.xxx; Port 为 SSH 连接后的端口号; User name 为远程服务器用户名, 通常为 root;



点击加号, 加载出来

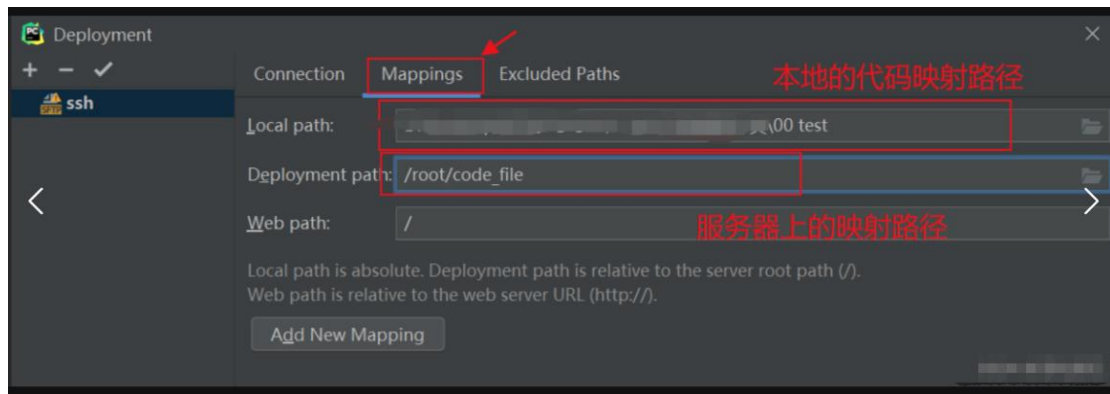


连接成功



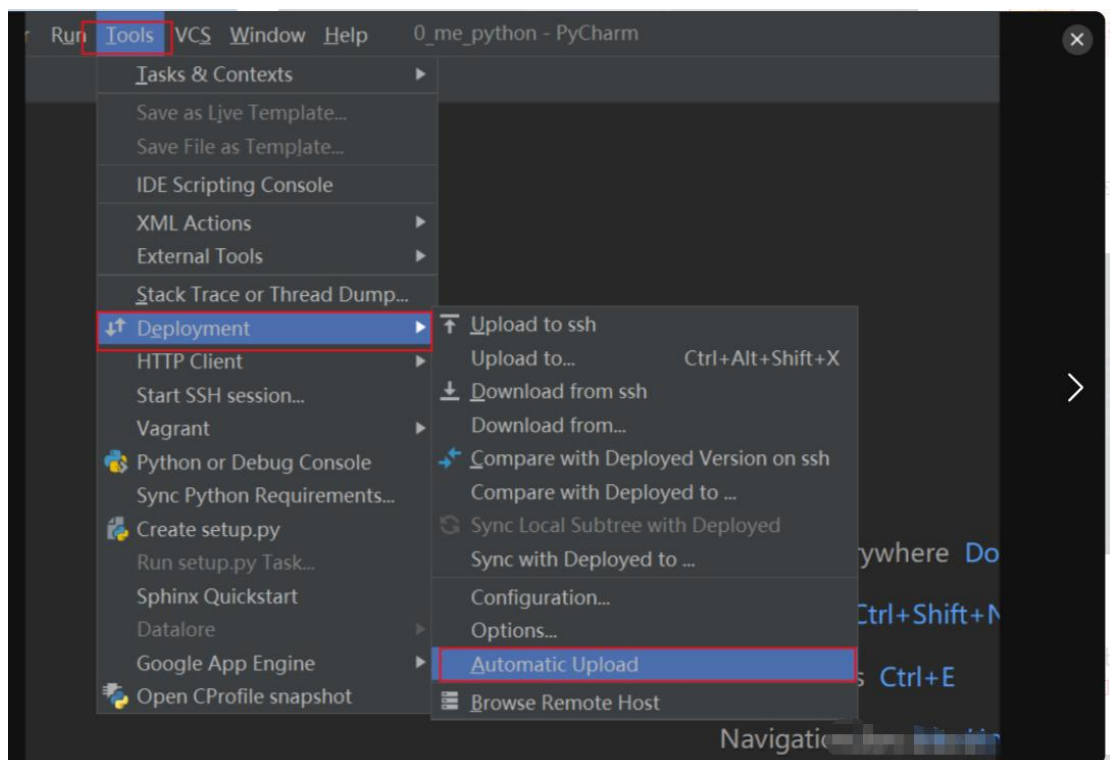
2. 进行本地项目与远程项目之间的文件夹路径映射

为了后续本地与服务器之间同一项目可以内容同步，我们还需要设置二者之间的项目路径映射，即本地项目在远程服务器中存放的位置：



3.设置自动上传项目（如有需要，可设置）

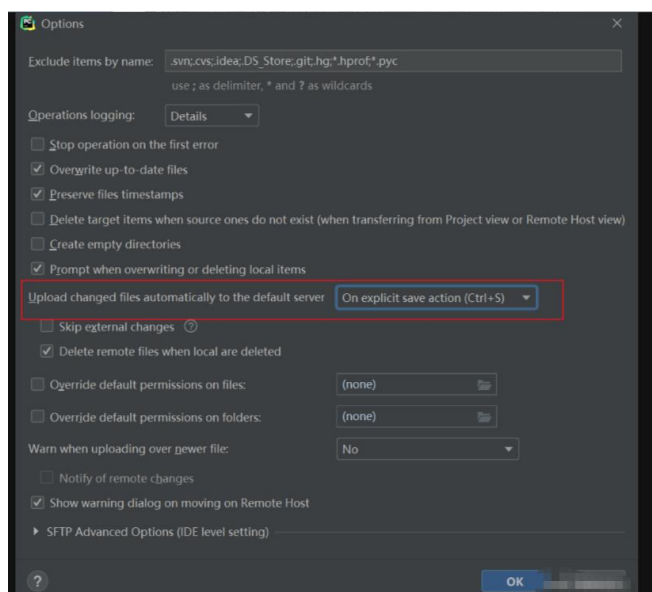
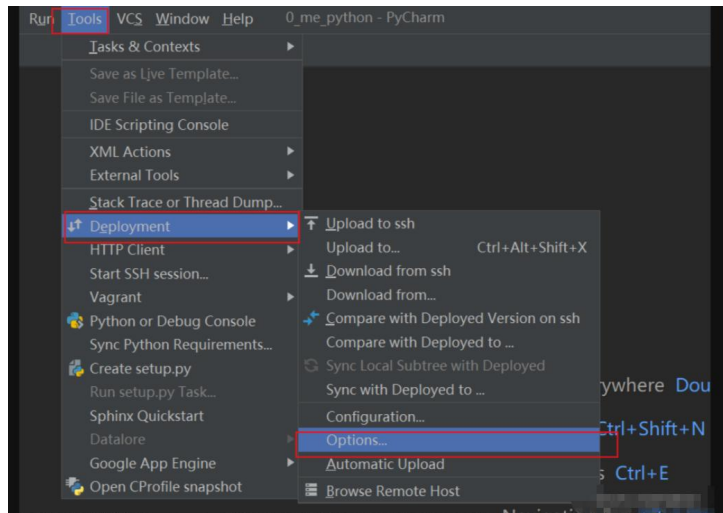
在连接建立并配置好本地与远程项目之间的路径映射后，我们再设置自动上传项目即可，如此本地与服务器之间就可以进行项目同步了，此时我们更新了本地代码之后只要使用 Ctrl+S 保存代码即可完成同步：（非必要，也可以自己手动更新上传）



此处需要注意如果自动上传无法成功，我们可以手动将本地项目上传到服务器（选择上面的 Upload to 即可），一定要在本地代码更改后及时上传同步到服务器！！反之，如果服务器对代码进行了修改，

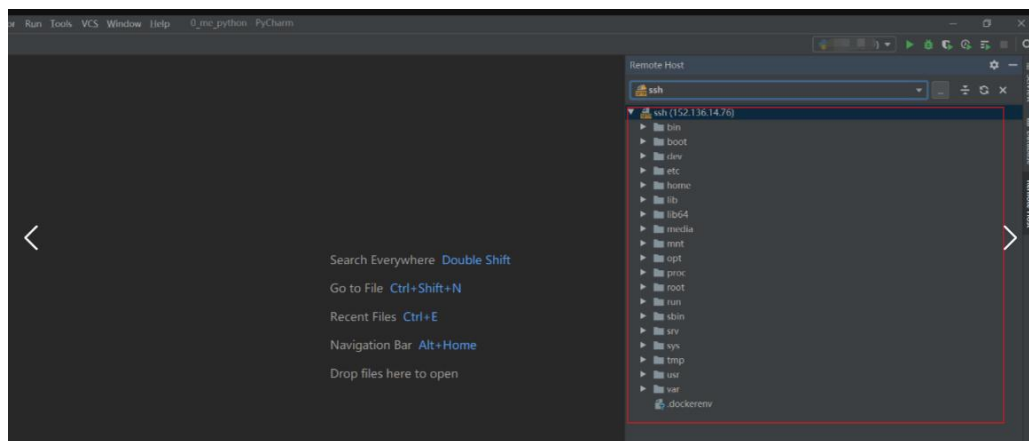
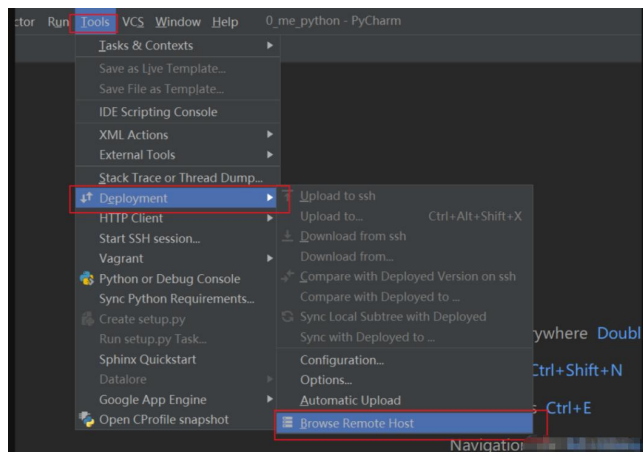
则通过本地的 Download from 拉取服务器中的代码同步本地。

设置 Ctrl+S 上传代码：Tools——>Deployment——>Options



4.验证是否连接成功（调出服务器的文件目录）

在进行 SSH 连接建立之后，通过下图所示方法可验证是否连接成功：



二、本地配置 Python 解释器

此时我们还不能在本地使用服务器的环境, 我们需要将本地 Pycharm 的 Python 解释器设置为远程服务器中的指定环境下的 Python 解释器:

1) 查询服务器中所使用的的 Python 解释器路径

进入服务器终端使用命令:

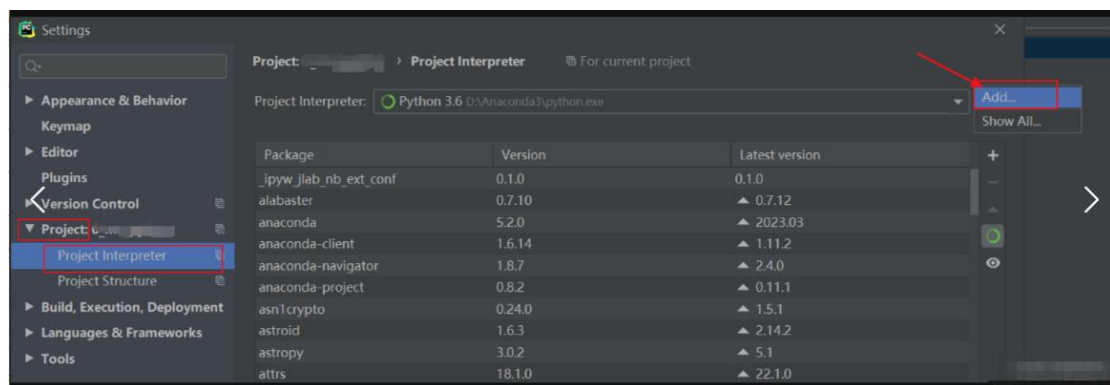
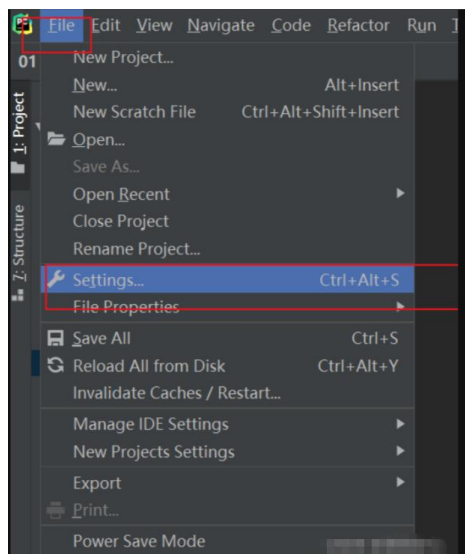
```
which python
```

即可查询 python 解释器的路径, 如果使用的是 Anaconda 中的虚拟环境, 需要先进入虚拟环境才可以使用该命令查询相应虚拟环境的 python 路径

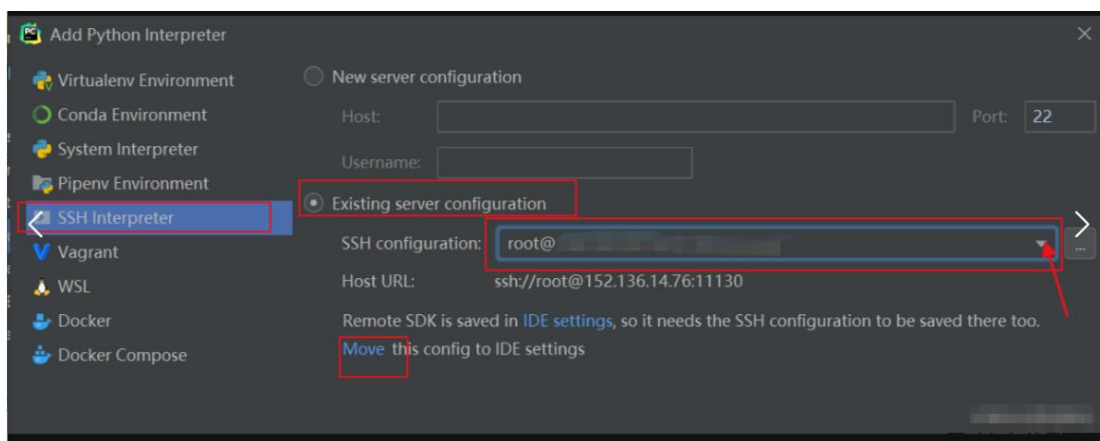
```
(base) ...:~# which python
/root/miniconda3/bin/python
(base) .#
```

2) 本地 Pycharm 中配置 Python 解释器

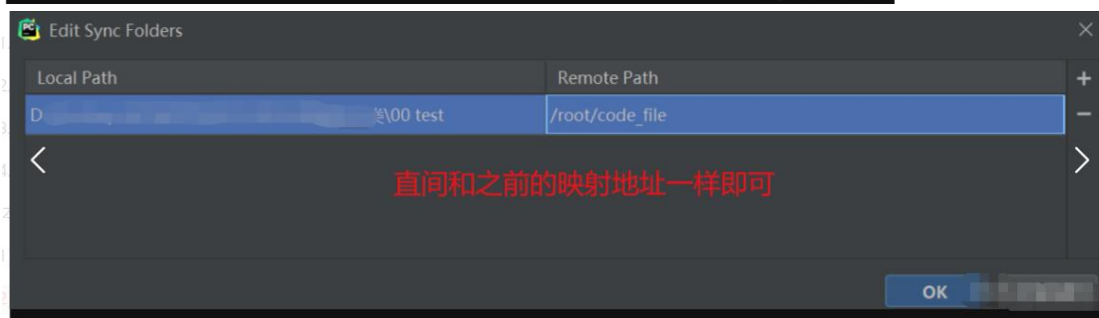
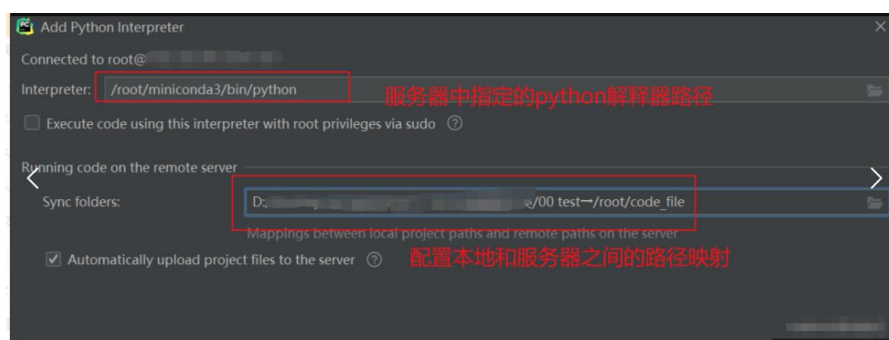
(1) 进入设置，将服务器中的 Python 解释器添加到 Pycharm 中：



(2) 选择 SSH 解释器，选择 SSH 配置，选中前面设置的服务器连接，然后点击 Move->Next 即可



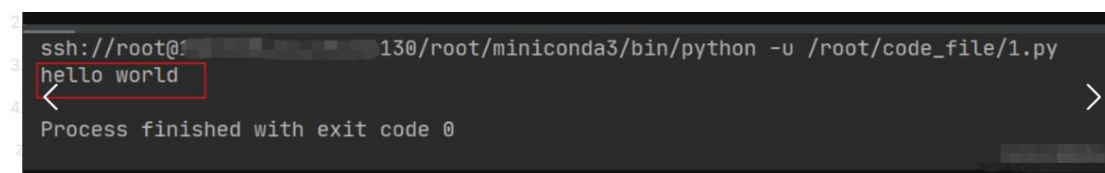
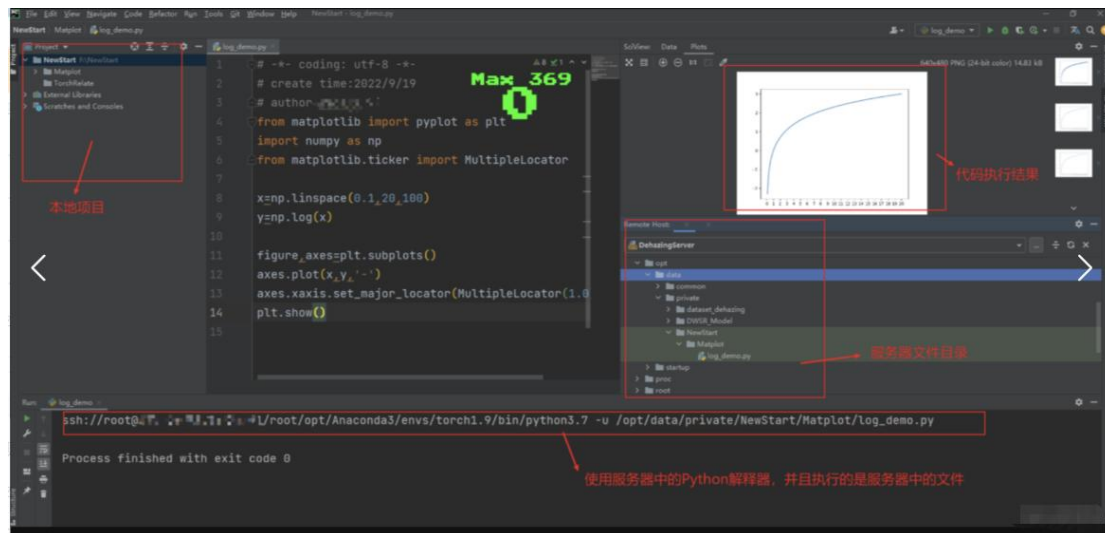
(3) 这里需要设置远程服务器中的 Python 解释器路径和项目之间的路径映射，如下图所示：



到此处我们的环境配置工作就完成了！！大家可以进行测试了！

三、测试案例

这里我配置好环境之后，将本地的一个绘图代码同步到了服务器，并在本地直接运行此代码文件，



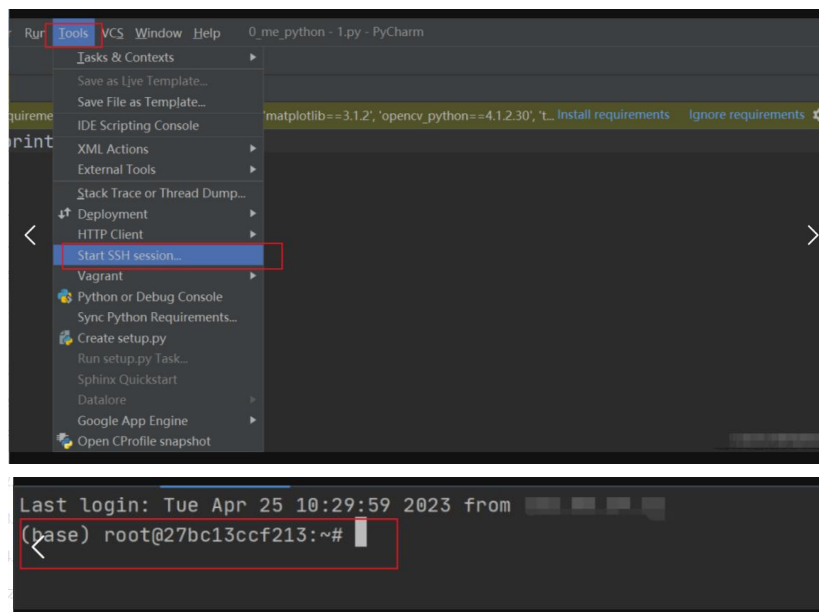
四、本地启动服务器终端

现在我们可以本地使用服务器的工作环境了，那如何在本地开启服务器终端呢，毕竟后面需要其他

第三方库还得装包，这里我也说一下：

工具栏中打开 Tools->Start SSH Session，然后选择我们前面配置的服务器，即可在本地 Terminal 处

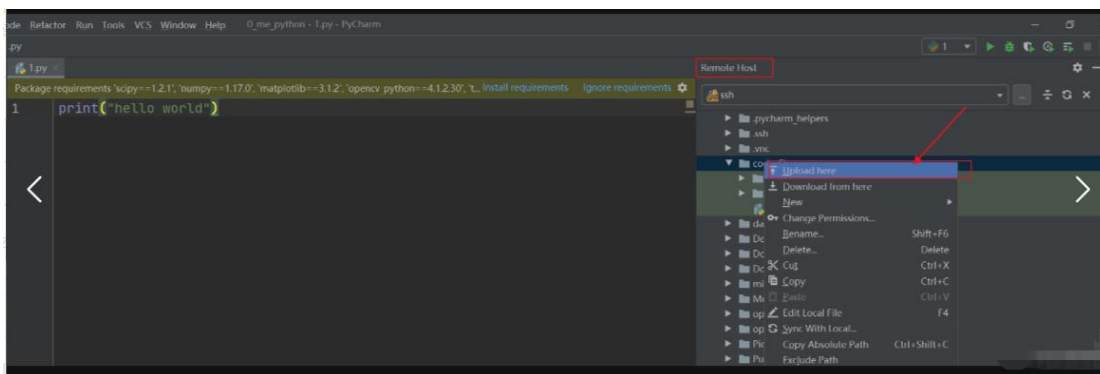
开启远程终端



五、PyCharm 与服务器中的文件手动更新和下载

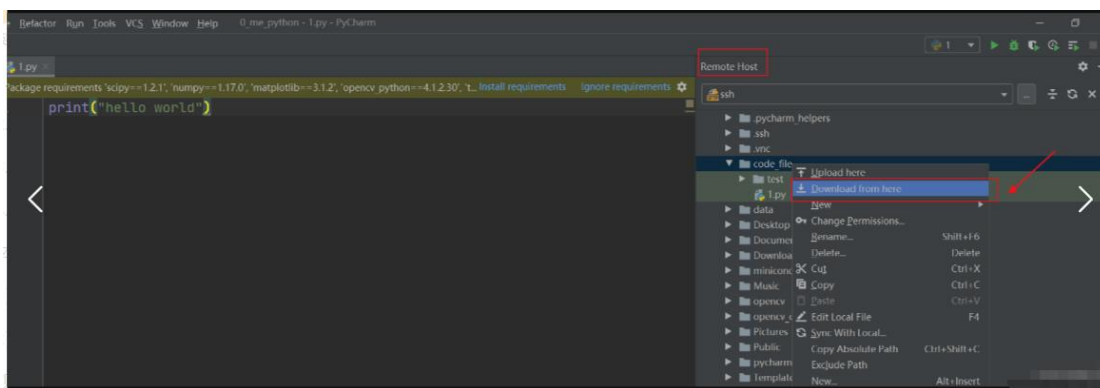
1) 本地 PyCharm 中的代码同步到云服务器

在 Remote Host 中点击想要上传的代码右键，选择 Upload here，就能实现将本地的代码同步到云服务器。其实在 PyCharm 中修改代码之后，可以直接通过 ctrl+s 保存文件，并实现文件同步到云服务器。



2) 从云服务器下载代码到本地 PyCharm

在 PyCharm 界面中找到右边的 Remote Host，这里查看到服务器端的全部文件信息。在 Remote Host 找到服务器中的工程文件夹，并右键选择 Download from here，便能实现将服务器中的文件下载到本地的 PyCharm 中，具体的文件对应关系，就由之前的 Mapping 页中的 Local path 和 Deployment path 所决定。



六、注意

后面如果要切换回本地工作环境，直接去设置里面切换 Python 解释器即可！！

上述中，配置 Pycharm 与服务器连接的步骤仅为一、二两个步骤。

Pycharm 与 Xshell 连接的区别

Pycharm 远程连接, 可以手动选择服务器上 Anaconda 或自带的 python 解释器。而 Xshell 连接, 只能用该登录账户下的 Anaconda 配的 python 解释器, 不能选择根路径下的 Anaconda 或 python 解释器。也就是说, Pycharm 远程连接可选的解释器更自由, 只要能选择到的都可用

Pycharm 远程连接和 Xshell 运行代码期间不能中断, 一旦关掉则远程服务器上也会停止, 不会继续跑。而 Xshell 连接可以通过指令: `nohup python xxx.py` 跑代码, 即使本地电脑关闭了 Xshell 或关机了都不会影响服务器跑代码, 服务器会继续运行代码 (除非远程服务器被关机了)