## Anaconda安装及多内核环境搭建

### Anaconda是什么

Anaconda® 可以简单理解为，一个用于科学计算的Python的发行版，专注于数据分析。

**特点：**

* 提供包管理与虚拟环境管理的功能。
* 集成720+的开源软件包集合，很好地解决多版本Python并存、切换以及集成第三方问题。
* 开源、跨平台、易于安装，并且有社区支持。

其中，作为目前最流行的 Python 环境管理工具， Conda不仅可以进行安装、更新、卸载工具包，而且还能建立多个虚拟环境，用于隔离不同项目所需的不同版本的工具包，以防止版本的冲突。

因此，Conda这种基于将几乎所有的工具、第三方库都当做Package对待，甚至包括Python和Conda自身的设计理念，使Anaconda集成多种内核（如，Python2/3）、多种Package成为可能。由此，通过切换不同的内核，开发者多场景下开发的需求就能很好地实现。

### Anaconda搭建多内核环境实现方案

通过虚拟机进行实验与测试，提出如下低成本、易部署的搭建方式：

|  |
| --- |
| Anaconda3 + Anaconda2 + 简单配置 |

即，在目标服务器上先后安装Anaconda3/Anaconda2，然后通过命令设置Anaconda2中Python内核为可利用内核。从而实现Anaconda3环境下既能基于Python3进行开发，也能基于Python2进行开发。

最终实现效果如下图：

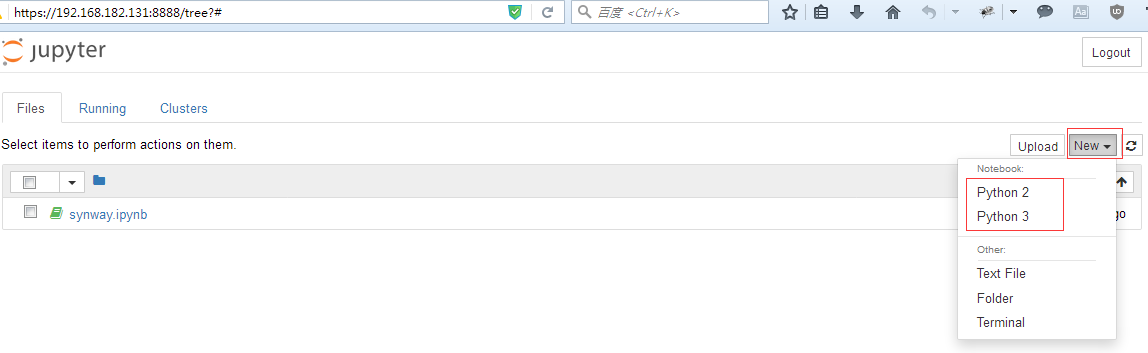


图 1 登录效果图

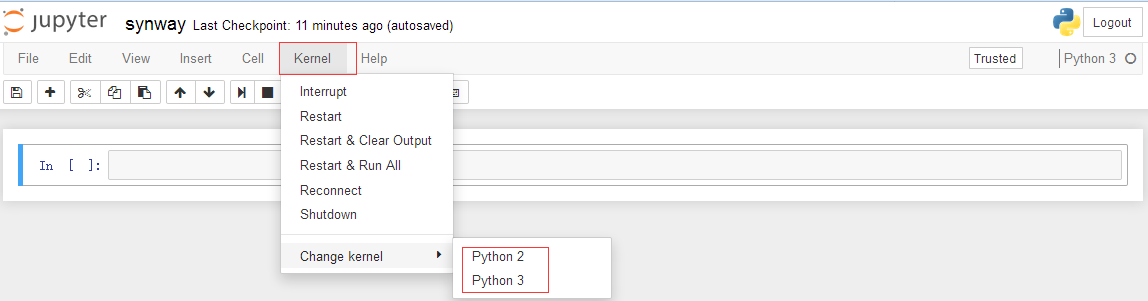


图 2开发效果图

### Anaconda集成多内核操作步骤

#### 确认系统版本、位数

在目标服务器（欲搭建Anaconda环境的服务器）中执行以下命令行，查看目标服务器系统的版本与位数信息。

|  |
| --- |
| cat /etc/redhat-release; getconf LONG\_BIT; |

从图3中可以看出，系统版本为CentOS 7.3,系统位数为64位，因此需要下载64位的Anaconda 3和Anaconda 2镜像，**值得注意的是，Anaconda 3和Anaconda 2版本号最好一致（例如，都是4.4.0）!**



图3 确认系统版本、位数命令执行结果

这里给出获取Anaconda镜像的途径如下表，链接1是官网镜像地址，链接2是清华开源软件镜像地址。如需下载，推荐使用链接2，其下载速度较快。

|  |
| --- |
| 1. https://www.continuum.io/downloads |
| 2 https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/ |

#### 上传镜像至服务器

使用SecureCRT或者FileZilla等工具远程连接服务器，将Anaconda镜像分别上传至服务器。如图4所示，表示将Anaconda镜像以二进制的方式上传至服务器的过程，图5是文件上传后的结果，由显示结果可知，Anaconda镜像上传成功。

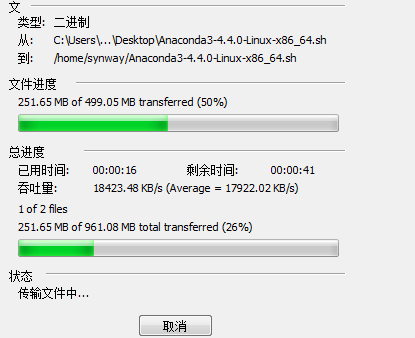


图4 SecureCRT上传Anaconda镜像过程



图5 文件上传结果

#### 安装 Anaconda 3

**注意：安装过程在管理员用户下完成!!!**

使用如下命令，进行Anaconda镜像的安装。

|  |
| --- |
| bash Anaconda3-4.4.0-Linux-x86\_64.sh |

如图6，执行命令后，进入安装阶段。首先我们根据提示，输入“**Enter**”查看许可证书。通过不断点击“Enter”键，直到许可证协议末尾，系统会自动给出如图6所示的提示。即，是否同意许可协议，输入“**yes**”。

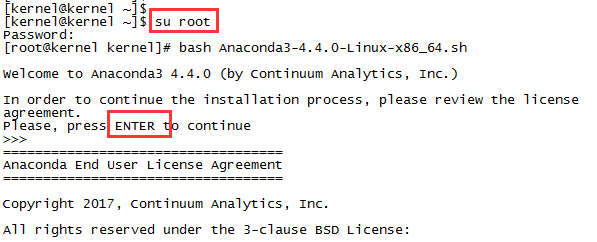


图6 输入 Enter 查看许可证书

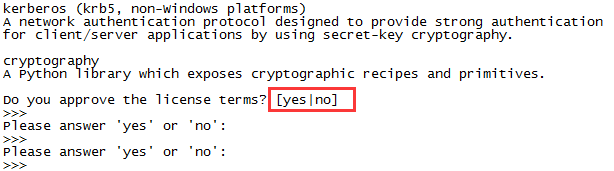


图7 输入 yes 同意许可协议

输入“yes”后，紧接着需要验证Anaconda 3的安装位置，如图8所示，默认安装位置为“/root/anaconda3”。

根据提示，输入“Enter”表示软件安装在默认位置；输入“CTRL-C”表示终止安装过程；如果需要自定义安装路径，在 “>>>”后输入即可。最后，点击“Enter”继续安装。

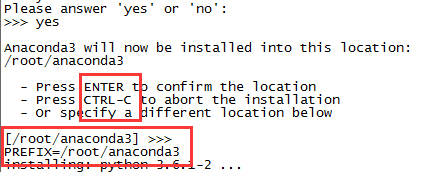


图8 Anaconda3默认安装位置

根据如图9所示的提示，“是否希望安装程序预置 Anaconda3 安装路径到/root/.bashrc文件中？[是 | 否]”，输入“**yes**”完成Anaconda3的安装。

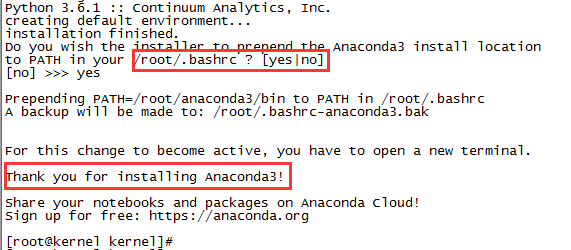


图9 是否预置安装路径到.bashrc文件

如果直接输入“Enter”，系统默认输入“no”，则需要手动编辑“/root/”路径下的.bashrc文件，将“**export PATH=/root/anaconda3/bin:$PATH**”路径信息添加到最后一行，最终结果如图10所示。

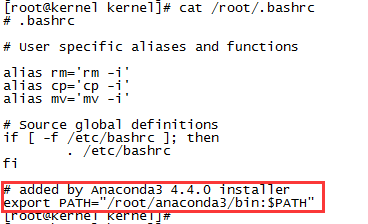


图10 编辑.bashrc添加路径信息

至此，Anaconda 3的安装已经完成。总结下来，安装过程中尽量输入“Enter”，有提示的尽量输入“Yes”即可。

#### 配置Jupyter-Notebook

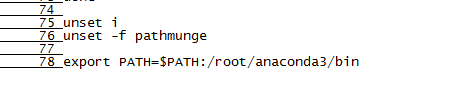
Anaconda 3安装完成后，还需要完成相关配置才能正常使用，配置过程大致分为以下几个步骤：

##### 生成配置文件

如图12所示，执行如下命令，将会在“/root/.jupyter/”路径下产生jupyter\_notebook\_config.py配置文件。

|  |
| --- |
| jupyter-notebook --generate-config –allow-config |

**如果提示“bash: jupyter-notebook: command not found...”，**则需要将Anaconda 3的安装路径添加到 “/etc/profile”文件中。如图12，第78便是Anaconda 3的路径信息。



**图 11 添加Anaconda 3路径信息后**

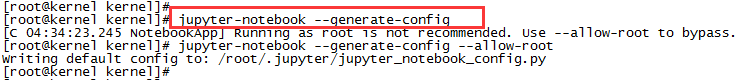


图12 生成jupyter\_notebook\_config.py配置文件

##### 生成sha1密文

在IPyhton环境下生成密文密码，密码用于验证是否具有访问Jupyter权限，执行语句如下：

|  |
| --- |
| # 进入IPython 环境  ipython  # 导入登录认证模块  from notebook.auth import passwd  # 调用生成密码方法  passwd() |

其中，在CentOS 7终端执行结果如图13所示，“Out[2]”位置是密码为“xxxxxx”后经过sha1加密的密文。

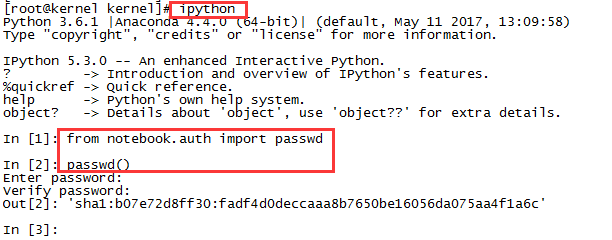


图13 利用IPython环境生成访问密码

##### 生成SSL服务器证书

|  |
| --- |
| openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout mycert.pem -out mycert.pem |

在控制台输入上述命令后，根据提示输入国家、省份、城市等信息，也可以直接点击“Enter”。如图14所示，在最后输入邮件信息后便生成了mycert.pem配置文件，其中包括密钥和证书2部份内容。

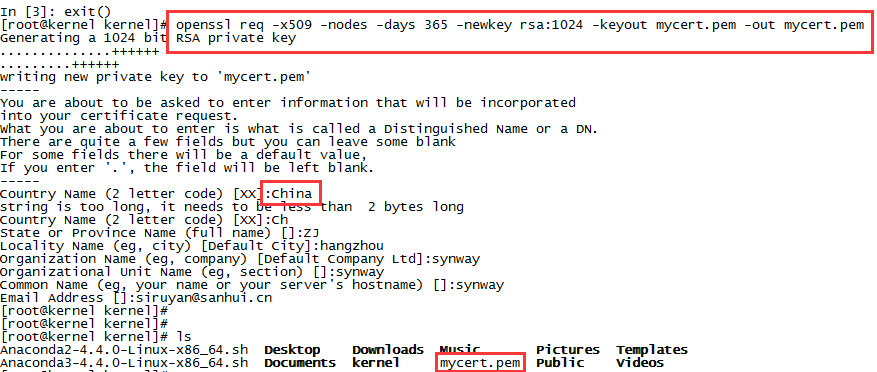


图14 生成服务器访问证书与密钥

##### NotebookApp服务器配置

修改下列代码将上述生成的信息配置到NotebookApp服务器中，从而完成配置。

|  |
| --- |
| c.NotebookApp.ip='\*' # 设置 ip  # 配置访问密码  c.NotebookApp.password =  u'sha1:626ceb2480e8:01ebe2f1fb582129ab1ae45aa6e715c5a5244e1c'  # 在服务端启动Jupyter服务器后，默认不自动打开浏览器  c.NotebookApp.open\_browser = False  c.NotebookApp. port =8888 # 设置 Jupyter服务器 访问端口  c.NotebookApp.keyfile = '/home/kernel/mycert.pem' # 配置密钥  c.NotebookApp.certfile = '/home/kernel/mycert.pem' # 配置证书 |

通过，vi/vim等命令编辑jupyter\_notebook\_config.py文件，将替换过password、keyfile和certfile值后的代码复制到jupyter\_notebook\_config.py文件中保存并退出。

配置完成后，jupyter\_notebook\_config.py文件末尾部分如图15所示，可以看出密码（password）、端口（port）、密钥及证书值等已经替换。

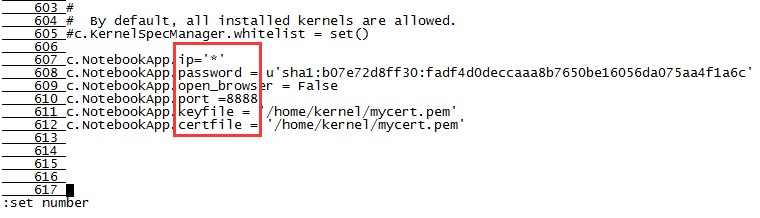


图15 jupyter\_notebook\_config.py文件配置后

##### 开启应用端口

jupyter\_notebook\_config.py文件配置完成后，开启应用端口用户可以访问了。如果端口未开启，则通过浏览器访问时会显示失败消息，用户不能使用Jupyter应用环境。

由于此次实验中jupyter\_notebook\_config.py文件配置的端口（port）为8888，因此，通过以下命令开启8888端口。

|  |
| --- |
| # 永久（permanent）开启 8888端口  firewall-cmd --zone=public --add-port=8888/tcp --permanent  # 重新载入防火墙以及状态信息  firewall-cmd --reload  # 查询 8888 端口是否开发  firewall-cmd --zone=public --query-port=8888/tcp |

执行结果如图16所示，值得注意的是，使用“firewall-cmd”命令开启8888端口后，需要执行“fiewwall-cmd reload”命令重载防火墙配置，才能使最新配置生效；否则，端口不能成功开启，用户仍旧不能成功访问Jupyter应用环境。

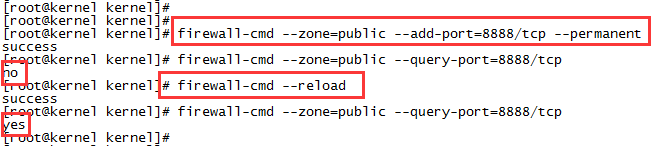


图16 开启端口命令执行过程

##### 启动服务器并访问

由于NotebookApp服务器的启动位置不同，其工作空间就不同。因此，为了便于管理，建议新建Jupyter工作空间，并在该目录下启动NotebookApp服务器，亦或者在jupyter\_notebook\_config.py文件中通过notebook\_dir项配置。

例如，将工作空间设置为“/home/kernel/synway”,只需在jupyter\_notebook\_config.py文件中添加下列代码即可。

|  |
| --- |
| c.NotebookApp.notebook\_dir = /home/kernel/synway |

在synway目录下，启动NotebookApp服务器。如图17所示，表示服务器启动成功。图18是通过火狐浏览器访问的结果，初次访问时需要添加信任，点击“添加例外”，弹出“添加安全例外”提示框，确认安全例外后进入Jupyter登录界面，输入密码后即可完成登录操作。

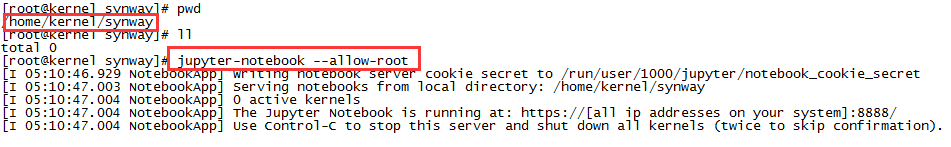


图17 启动NotebookApp服务器



图18 火狐浏览器访问结果

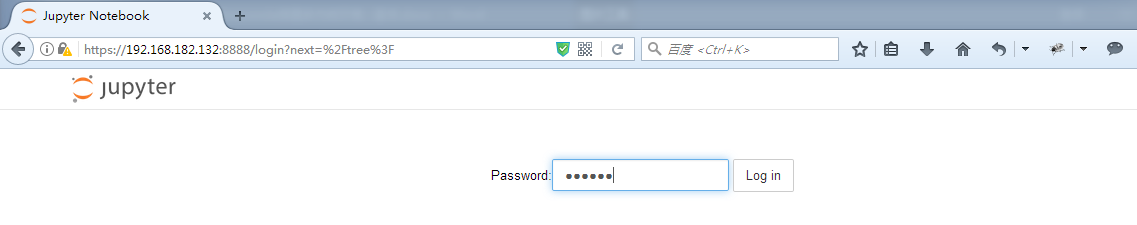


图19 Jupyter登录界面（密码：xxxxxx）

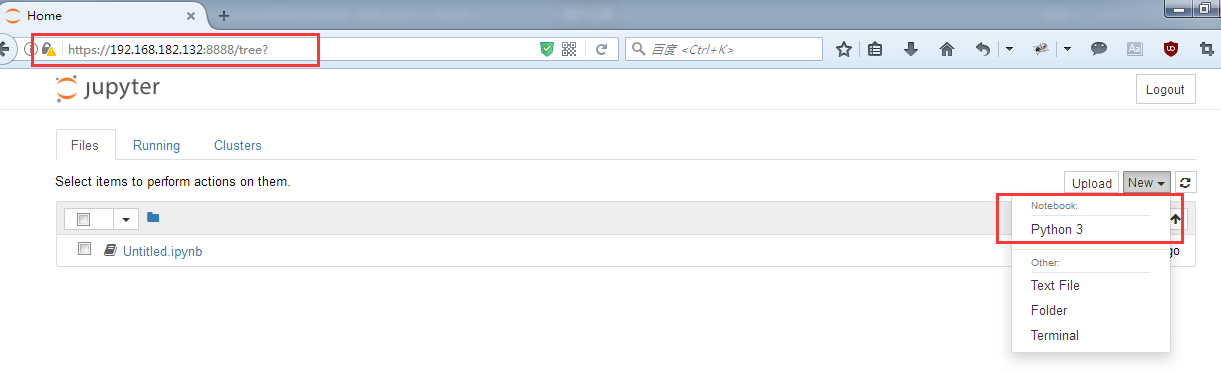


图20 Jupyter开发环境效果图

图20展示的是登录后的Jupyter开发环境，**至此Anaconda 3安装完毕。**

#### 安装 Anaconda 2

类似Anaconda 3的安装过程，通过bash命令安装Anaconda 2。**当执行到图21所示位置时，即是否Anaconda 2的安装路径预置入.bashrc文件，输入“no”或者按“Enter”键**，这里Anaconda 2安装路径不需要写入.bashrc文件中。

如果输入“yes”，则root路径下的.bashrc文件会被覆盖，之前安装写入的信息就会丢失，.bashrc中只有Anaconda 2的路径信息，**这样会导致环境配置失败！**

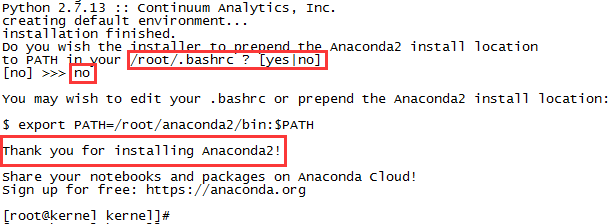


图 21 是否预置Anaconda 2安装路径到.bashrc文件

#### 配置可利用内核

在配置可利用内核前，首先学习一些Jupyter命令。如下表中，命令的顺序也是执行的顺序，图22通过命令执行过程演示了可利用内核配置的过程。

首先切换到已经安装过的anaconda2 bin目录下，然后执行表中第3部分命令，将Anaconda 2集成的Pyhton2配置为可利用内核，配置结果通过表中第4部分命令查看。

|  |
| --- |
| # 1 切换路径到anaconda2 bin目录下  cd /root/anaconda2/bin  # 2 查看当前路径下python\* 类型可执行文件  ll \*python\*  # 3 将Pyhton 2 设置为可利用内核  ./pip install ipykernel  ./python2.7 -m ipykernel install --name python2.7 --display-name "Python 2"  # 4 查看Jupyter可利用内核  jupyter-kernelspec list |

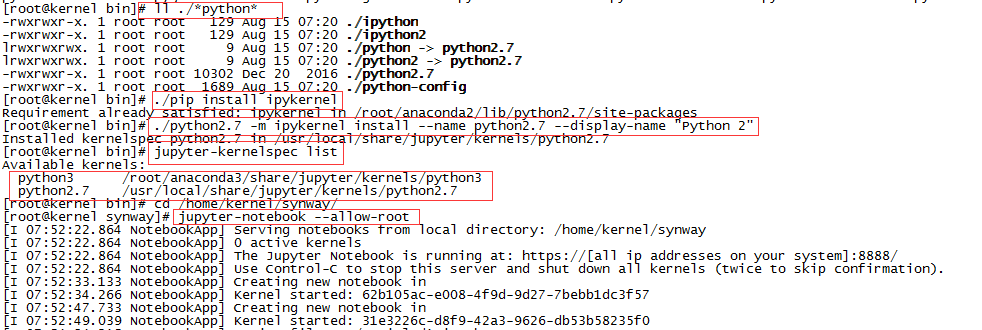


图 22 可利用内核配置过程

### 搭建是否成功

完成设置后，将路径切换到之前设定的Jupter工作空间，启动NotebookApp服务器验证配置是否成功。

如图23 所示是在“/home/kernel/synway”工作空间下启动NotebookApp服务器后的运行效果，红框部分为可利用内核，包括Python3/2。

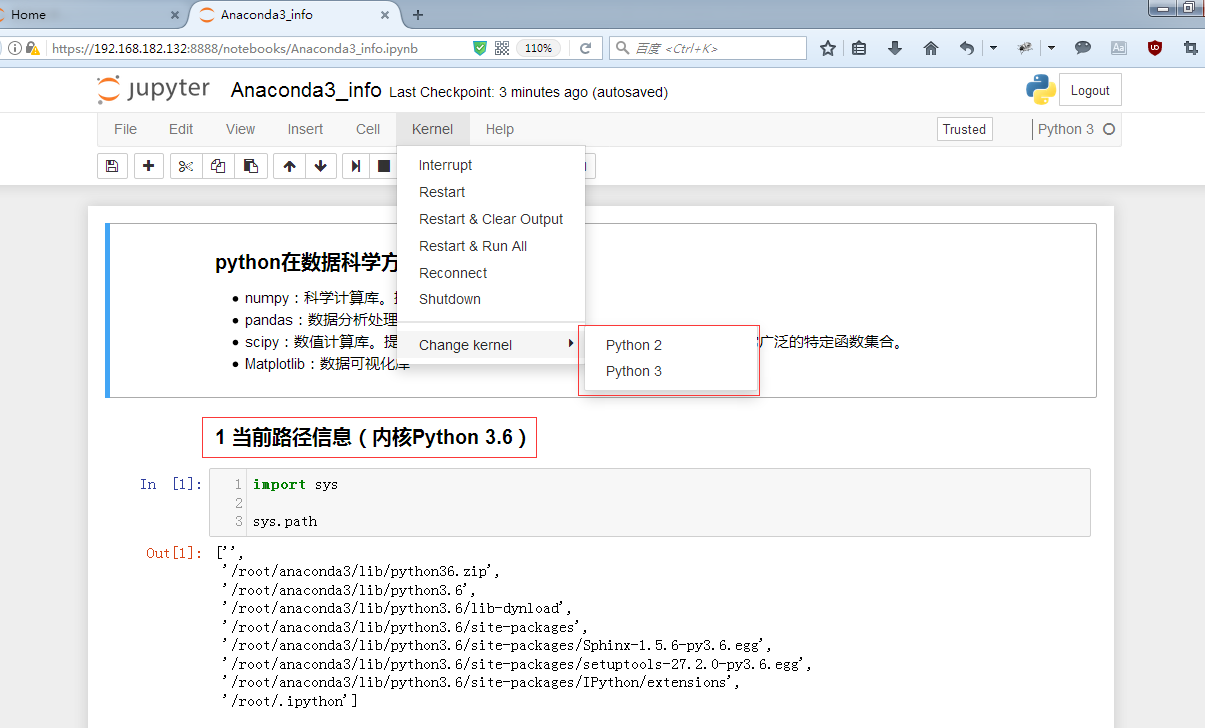


图 23 可利用内核配置过程

通过在Jupyter Cell中执行“**import sys; sys.path**”代码，打印出切换内核后相关引用包及内核的路径信息。从图23看出中，Anaconda环境引用内核Python3，而由路径信息得出引用包来自“/root/anaconda3”路径。

**由此，可以尝试切换内核Python 2进行校验，从而判断环境配置是否正确。**

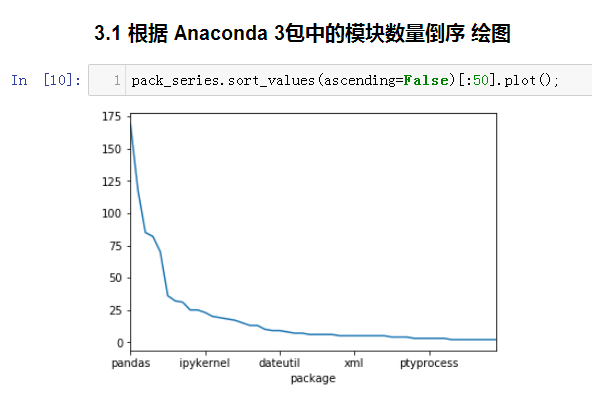
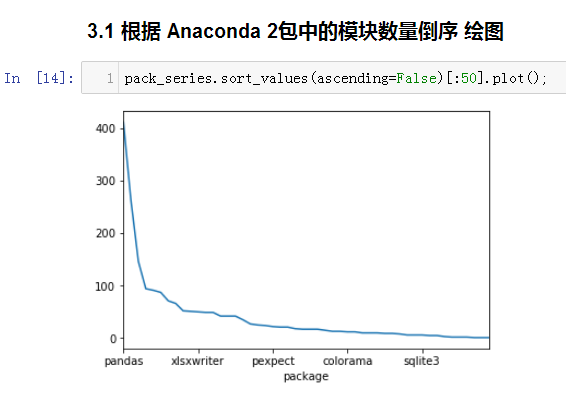
### 环境验证

通过以上步骤，初步认为环境搭建成功。下面通过执行一些脚本验证环境搭建的效果。





Python3/2 科学计算常用包版本及路径信息-对比

根据Anaconda3/2中各个包的模块数量倒序图-对比

**最后，还可以通过“ help(“modules”) ”查看环境中的模块信息进一步验证。**

**提醒：**

1. **安装Anaconda环境时，使用管理员用户进行安装。**
2. **Anaconda版本要一致！例如，下载Anaconda3/2镜像版本均为 4.4.0。**
3. 如果所有配置均正确，仍旧访问不到Jupyter应用，查看工作空间是否具有访问权限。