## CentOS下搭建JupyterHub  
  
### 基础环境准备 - python3安装  
CentOS下使用yum指令进行基础环境的安装。  
  
安装python3.7步骤如下：  
  
1. 下载python的源码包  
  
    wget https://www.python.org/ftp/python/3.7.4/Python-3.7.4.tgz  
  
2. 解压并使用config脚本来生成makefile文件，为安装准备  
  
    tar -zxvf Python-3.7.4.tgz  
  
3. 安装前执行下列指令，保证依赖服务可用  
  
    yum groupinstall "Development Tools"  
  
    yum -y install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel gdbm-devel db4-devel libpcap-devel xz-devel libffi-devel  
  
4. 完成解压后，进入解压目录  
  
     ./configure --prefix=/usr/local/python3  
  
5. make指令进行安装  
  
    make && make install  
  
6. 完成后建立软链并添加环境变量，PATH中添加路径/usr/local/python3/bin  
  
     ln -s /usr/local/python3/bin/python3 /usr/bin/python3  
  
    vim vim ~/.bash\_profile  
  
7. 上述完成后可输入指令查看python版本，可以看到python安装完成后会连pip3以及setuptool一起安装  
  
  
相关安装过程的错误解决：解决方案为安装步骤第三步  
  
错误1，configure指令出现没有C compiler错误，需要执行指令安装linux的基础开发工具。  
  
    yum groupinstall "Development Tools"  
  
错误2，出现zlib not available，需要执行指令，其中提示db4-devel没有安装包  
  
    yum -y install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel gdbm-devel db4-devel libpcap-devel xz-devel libffi-devel  
错误3，出现no module named '\_ctypes'，需要添加依赖libffi-devel，已在上述指令中添加。  
  
----------  
### 基础环境准备 - nodejs/npm安装  
nodejs的安装过程中会自带npm，因此仅需安装nodejs即可。  
  
安装nodejs10.16.0步骤如下：  
  
1. 下载源码包：  
  
    wget https://nodejs.org/dist/v10.16.0/node-v10.16.0-linux-x64.tar.xz  
2. 解压，需要二次解压：  
  
    xz -d node-v10.16.0-linux-x64.tar.xz  
  
    tar -xvf node-v10.16.0-linux-x64.tar  
  
3. 解压完成后直接添加软链  
  
    ln -s /path/to/node-v10.16.0-linux-x64/bin/node /usr/local/bin/node  
  
    ln -s /path/to/node-v10.16.0-linux-x64/bin/npm /usr/local/bin/npm  
  
4. 检查node及npm指令是否生效。  
  
    node -v  
   
    npm -v  
  
相关安装过程的问题：  
  
删除文件夹使用rm -rf package，rm是linux删除指令，r表示向下迭代，所有子文件夹都会被遍历，-f代表直接删除  
  
ln命令，创建链接，在安装过程的软链就是通过该指令创建。该指令创建的文件会保持原文件同步，软链不占空间，硬链需占空间，该文件在查看时文件名会带有@符号。  
  
    ln [参数] [原路径] [目标路径]  
  
其中：  
  
- 软链是路径的形式，可以跨文件系统，可对不存在的文件名进行连接，可对目录进行链接。  
- 硬链是文件副本的形式，不允许对目录创建，只有同一个文件系统才能创建。  
- 无论哪种形式，均会保持每处链接的同步性  
  
参数：  
  
- -b，删除，覆盖以前的链接  
- -d，运行超级用户制作目录硬链接  
- -f，强制执行  
- -i，交互模式，文件存在则提示是否覆盖  
- -n，把链接视为一般目录  
- -s，软链  
- -v，显示详细处理过程  
  
---  
### 基础环境准备 - jupyterhub安装  
  
执行指令：  
  
    pip3 install jupyterhub  
    npm install -g configurable-http-proxy  
    pip3 install notebook  
  
其中，由于node是在自定义路径安装的，在进行configurable-http-proxy的全局安装中，提示未找到命令。需要将PATH中添加路径。  
  
完成后启动jupyterhub，在本地登录localhost:8081可直接使用linux当前登录用户登录，但第一次尝试提示302及404  
  
从目前的反馈来看，是启动的单用户服务，代理将hub重定向到了8081。同时配置应该还存在一定问题，在启动single notebook的时候有异常发生。  
  
### 后记  
  
hub成功启动，单用户登录成功并启动notebook成功，这里记录下部分的坑：  
  
1. 在自定义路径的安装下（linux的安装似乎都属于自定义路径），按照上述流程配置好PATH，但jupyterhub默认使用的配置文件不可访问，其默认要求启动后生成的文件路径与jupyterhub-singleuser文件一致，否则会在启动中报错。  
2. 执行文件要求权限，至少要有当前目录的读写权限，测试时为了快速成功直接将普通用户加入了sudoer列表，具有root权限（因为安装时用的root）。  
3. 使用指令生成配置文件，可以放在任意路径下，启动时指定路径即可使用生成配置。  
  
    jupyterhub --generate-config  
      
    jupyterhub -f /path/to/jupyterhub\_config.py  
  
**\*\*Jupyterhub的组件启动顺序。\*\***  
  
读取Authenticator配置 -> 读取Spawner配置 -> 启动Hub -> 启动Proxy。  
  
登录hub后，会跳转到spawner等待页面，notebook启动完毕后会自动跳转到用户专属页面。  
  
若spawner生成有异常（有找不到路径，无权限的问题），会提示启动失败，或者有可能进入死循环（重定向循环，这个仅触发一次，原因未知）。  
  
若需要使用第三方验证器OAuth，需要安装oauthenticator  
  
    pip3 install oauthenticator