Ubuntu 使用指南

Q. 一些 Ubuntu 学习体会

1. 学习实践类型的技术，跟打篮球是一样的，一方面要注重理论知识的理解和体系话，另一方面更重要的是对技术的反复练习，反复试错，反复试新，才能达到熟悉/防错/创新的应用境界。

2. 学习

Q. 如何查看 ubuntu 的版本，以及本机其他重要软件的版本

ubuntu 的版本：cat /etc/issue，本机为 ubuntu 16.04.5
ubuntu 的系统位数：sodu uname –m，本机为 x86_64
python 的版本：python –version，本机有 2 个环境，base 为 python3.7，py35 为 python3.5

Q. 默认 ubuntu 系统不支持 exfat 格式的移动硬盘，为了跟 mac/windows 交互文件又只能用 exfat，怎么办?

安装一个东西即可：for ubuntu 14.04 or later version, run below: sudo apt-get install exfat-utils

Q. 如何在 Ubuntu 安装中文输入法?

在 Ubuntu 系统下安装一个中文输入法都能让我搞半小时，可见 ubuntu 系统跟 mac 和 windows 的差距

1.安装汉语语言包
sudo apt-get install language-pack-zh-hans
执行该命令后，系统就会自动安装所需要的汉语语言包
2.安装谷歌拼音输入法
sudo apt-get install fcitx-googlepinyin
打开 SystemSettings-->Language Support，选择“键盘输入方式系统”为：fcitx
重启电脑或者注销用户。
打开终端，执行“fcitx-configtool”命令，在如图界面进行输入法配置，增加谷歌拼音输入法。

Q. 在 ubuntu 如何安装类似 office 套件?

可以下载 WPS，选择 Deb 格式的安装包安装即可

Q. 到底安装那个版本的 Anaconda 以及那个版本的 python?

相关的限制条件是：

- opencv3 只支持
- chaincv 只支持
- pytorch 只支持

- python 3.6.0 可以支持 chenyun 的 object detectron with simple-faster-rcnn

Q. 在 ubuntu 上常用的文件夹操作命令?

包括：

- 启动终端：ctrl + alt + T，或者搜索 terminal
- 查看当前目录下所有子目录：ls

- 切换目录：
    cd /　　进入根目录
    cd ..　　进入上一级目录
    cd -　　进入上次访问的目录
    cd ~　　进入 home 目录
- 切换成 root 用户权限：sudo -i
- 显示当前目录地址：pwd (= print working dirctory)
-

Q. 在 ubuntu 安装应用程序的方法?

卸载 anaconda 的方法：
- 直接删除 anaconda 文件夹
- 清楚系统残留数据

安装 anaconda 的方法：
第一次安装是用 sudo 安装，发现安装好以后各种补丁不能做，说没有权限，怀疑是 anaconda 只能 root 用，所以打算卸载后重新用普通登录用户再安装一次。
卸载后重新下载重新按照 anaconda 官网的指示安装，却发现 bash shell 在 ubuntu 不能被识别，百度后知道 ubuntu 默认的 shell 是 dash 而不是 bash，所以手动把 dash 改为 bash，安装依然不识别我下载的 sh 文件，所以又该回来 dash，重新下载，重新关闭 terminal 重新打开，再次按官网指令安装：
dash Anaconda3-5.3.0-Linux-x86_64.sh (不过把官网的 bash 改成了 dash)
中间选择配置环境变量到：/home/ubuntu/.bashrc
这回好了，尝试启动 anaconda navigator，在 terminal 中直接输入：
anaconda-navigator 即可启动
这就是诡异的安装过程。这次安装的 anaconda 文件夹没有文件锁的标志了，所属 group 也变为普通用户而不是 root 了。

Q. 如何添加清华镜像?

在精华镜像站官网找到某个软件的镜像添加方式，比如 pytorch：
`conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/`
然后可以查看加入的镜像是否成功：
`conda config --set show_channel_urls yes`(#会生成一个`~/.condarc` 文件,运行 `cat` 命令查看)
`cat ~/.condarc` 即可显示已添加的仓库列表如下：

```
ssl_verify: true
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/cloud/pytorch/
  - defaults
show_channel_urls: true
```

Q. 如何安装 GPU 版本的 pytorch?

从官网上找到配置表，选好后自动生成安装语句（稳定版，CUDA9.0，linux）
然后在线安装：`conda install pytorch torchvision -c pytorch`
但这种在线安装方式太慢，下载可能需要 2-3 天
于是采用从清华大学的镜像网站上下载安装包：主要下载
`torchvision-0.2.1-py37_1.tar.bz2`
`pytorch-0.4.1-py37_cuda9.0.176_cudnn7.1.2_1.tar.bz2`
然后通过离线方式安装，但发现安装不了 `tar.bz2` 文件。

只好换方案：
先添加清华镜像 conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/

这次不知道是因为添加清华镜像的原因还是因为安装为早上，在线命令安装下载速度很快完成，然后安装采用源网站的说明：conda install pytorch torchvision -c pytorch
但发现安装报错说 anaconda 的权限不允许，查看 anaconda 的权限发现是 root 权限安装的 anaconda 即 sudo 方式安装，怀疑这是根源，所以重新卸载 anaconda，重新基于普通用户安装 anaconda 后，再运行 conda install pytorch torchvision -c pytorch
这次连下载都不用了，估计前次下载好了，直接秒速安装完成。

验证 pytorch 安装：import torch, import torchvision
验证 GPU 安装： print(torch.cuda.is_available())
Q. 如何升级 pip
默认 ubuntu 的 pip 是 pip 10.0.1，而最新的 pip 已经到 18.1 了。升级方式如下：
python -m pip install --upgrade pip
升级好以后通过 conda list 查看可以看到系统同时存在两个 pip 如下：

```
pip                       18.1                      <pip>
pip                       10.0.1                 py37_0    defaults
```

使用 pip(=pip2)命令就是用 10.0.1 的版本，而使用 pip3 就是用 18.1 的版本

Q. 如何查看当前用户和 root 用户？有哪些常用 ubuntu 命令？
（1）打开 terminal 就能看到用户情况：
ubuntu@ubun:$ 这说明当前用户就是 ubuntu，是普通用户的提示字符是$
root@ubun:~# 这说明当前用户就是 root，是 root 的提示字符是#
在@后边的 ubun 是主机名字
～代表家目录，其中 root 用户家目录是/root，普通用户家目录是/home/用户名

（2）默认是普通用户登录
切换到 root 用户：sudo -i，并能保持 root 用户 5 分钟然后自动切回普通用户权限
如果要退出 root 用户就是 exit 或者 logout

（3）普通用户
家目录显示为～，根目录显示为/
进入家目录：cd ~ (普通用户家目录为/home/用户名)
其他命令跟 root 一样（参考下面 root 用户的命令）

（4）root 用户
家目录显示为～，根目录显示为/
进入家目录：cd ~ (root 用户家目录为/root，该目录下默认没有文件，所以 ls 后什么也不显示)

进入根目录：cd /
显示目录内容：ls (文件夹和文件显示颜色不同，如果要区分可用 ls -F 则显示文件夹后带斜杠)
切换目录：cd 文件夹名
返回上一级目录：cd ..
删除文件夹及子文件：rm -rf 文件名
显示当前文件夹目录：pwd (=print working directory)

（5）基本指令形式： command [-option] parameter1 parameter2 …
比如：ls -F (ls 为显示目录指令，-F 为区分文件夹和文件的 option 选项)

（6）基础指令
- date：显示日期，也可以 date +%Y%m%d
- cal：显示日历，也可以 cal 2017
- bc：计算器，默认是整数，如果要小数，就用 scale=某数字。退出用 quit
- ctrl+c：停止运行程序
- commond+tab+tab：命令行加 2 个 tab 代表参数查询
- --help：用于调取每个命令的帮助参数

- man commond：用于调取每个命令的手册 manual
  - manual 中 1 代表指令
  - 2 代表函数与工具
  - 3 代表库
  - 4 代表装置档案…
- cat ~/.ssh/id_rsa.pub：其中 cat 用于打开文件
- mkdir 文件夹名：创建文件夹
- wget https://cn.wordpress.org/wordpress-3.1-zh_CN.zip　用于下载软件包
  - wget -c https://cn.wordpress.org/wordpress-3.1-zh_CN.zip 断点续传
-

（7）快捷键

ctrl + H：显示隐藏文件
shift + PgUp/PgDn：　翻页
ctrl + alt + C：区域截屏（在设置/keyboard/shortcuts/screeshots 中设置快捷键）
tab：补全文件名(这个非常有用)
ctrl + win + 上下左右：分屏幕显示
win + s：显示 4 分屏（爆方便，媲美 mac 的手指滑动！！！）
ctrl + alt + 上下左右：4 分屏切换（不过有了 win+s 以后，这个命令基本就没意义了）

Q．什么是 shell，什么是 bash?

（1）shell 是相对于 kernel 来说的：
首先 shell 是 linux 跟用户的交互程序，也叫外壳程序(shell)，用于接收用户指令
然后 shell 再把用户指令传递给内核程序(kernel)进行指令的执行
（2）shell 有很多种，包括 bash，csh 等。
linux 的所使用的 shell 叫做 Bourne Again Shell，他是由 Steven Bourne 发展出来的，所以叫 Bourne Shell，简称 bash
ubuntu 所使用的 shell 叫做 dash，其实是 bash 的变体，比 bash 更快，基本用法是一样的
查看该系统 shell：ls -al /bin/sh，显示如下：
lrwxrwxrwx 1 root root 4 10月 16 01:03 /bin/sh -> dash
（3）修改 dash 为原有的 bash：
sudo dpkg-reconfigure dash
选择 No
然后就通过 ls -al /bin/sh 查看 shell，会发现已变为 bash
（4）修改 bash 变回 dash
sudo dpkg-reconfigure dash

Q．如何安装 git?

（1）首先检查发现本机没有安装 git：git –version
（2）安装 git：sudo apt-get install git，但发现安装的版本比较旧(git version 2.7.4)
（3）升级 git：sudo add-apt-repository ppa:git-core/ppa （添加新的镜像源）
sudo apt-get update
sudo apt-get install -y git
此时再运行 git –version 就会发现版本变为 git version 2.19.1

Q．如何使用 git

（1）本地配置好用户名和密码：使用自己已申请的 github 帐号在本地配置好登录所用的用户名密码：

```
ubuntu@ubun:~$ git config --global user.name "ximitiejiang"
ubuntu@ubun:~$ git config --global user.email "ximitiejiang@163.com"
ubuntu@ubun:~$ git config –list    (这步是显示刚才输入的配置)
```

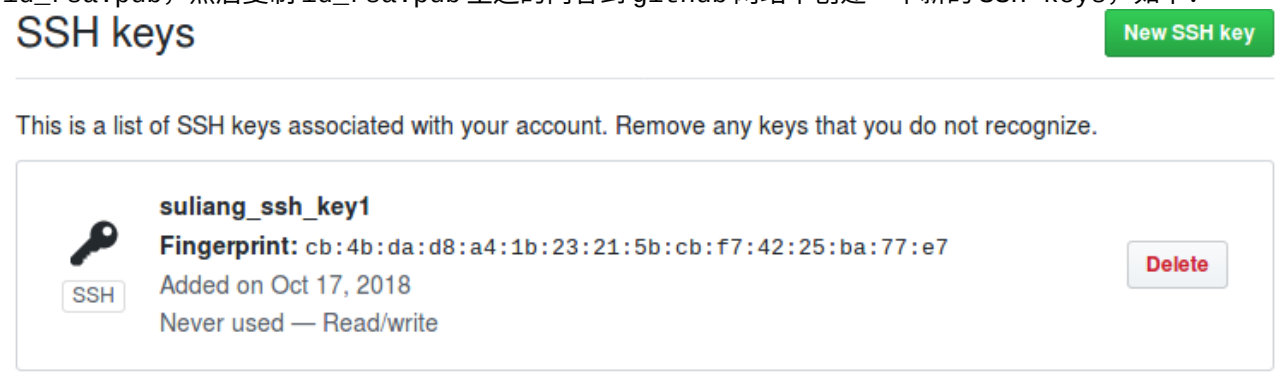
然后就会在 home 文件夹下产生一个隐藏文件.gitconfig 记录如下信息：

```
[user]
    name = ximitiejiang
    email = ximitiejiang@163.com
```

（2）创建 ssh key：SSH 是一个远程登录软件，github 和 ubuntu 都是支持 ssh 的，所以只要基于 ssh 配置了 key，就可以远程登录 github 而且不要每次输入密码。

```
ssh-keygen -t rsa -C "ximitiejiang@163.com"  (ssh-keygen 是 ssh 软件自带工具)
```

然后就会在 home 文件夹下产生一个隐藏文件夹.ssh，里边包含两个文件：私钥 id_rsa，公钥 id_rsa.pub，然后复制 id_rsa.pub 里边的内容到 github 网站中创建一个新的 SSH keys，如下：



（3）登录 github：

ssh -T git@github.com 成功登录后会显示：

```
ubuntu@ubun:~$ ssh -T git@github.com
Warning: Permanently added the RSA host key for IP address '192.30.253.113' to t
he list of known hosts.
Hi ximitiejiang! You've successfully authenticated, but GitHub does not provide
shell access.
```

（4）在本地创建文件夹：mkdir suliang_git

然后进入该文件夹：cd suliang_git

然后初始化 git 仓库：git suliang_git （使这个文件夹成为一个仓库属性，从而会在 home 文件夹下查收一个隐藏文件夹.git，用于存放所有 git 需要的数据）

然后似乎不需要把本地仓库与远程仓库关联？？？（直接 clone 就可以）

git remote add origin git@github.com:yourName/yourRepo.git

（5）查看文件状态：

常用 git status -s 其中 s 代表 short 简写状态

- D 代表本地删除，服务器还在
- A 代表本地新增
- M 代表本地修改
- ? 代表未被 git 进行管理，需要先 git.add

常用 git diff 查看文件具体区别

- git diff 工作区的变动
- git diff -cache 缓存区的变动
- git diff -HEAD 工作区和缓存区的所有变动

（5）git 基本结构

工作区：就是自己能看到的硬盘目录，

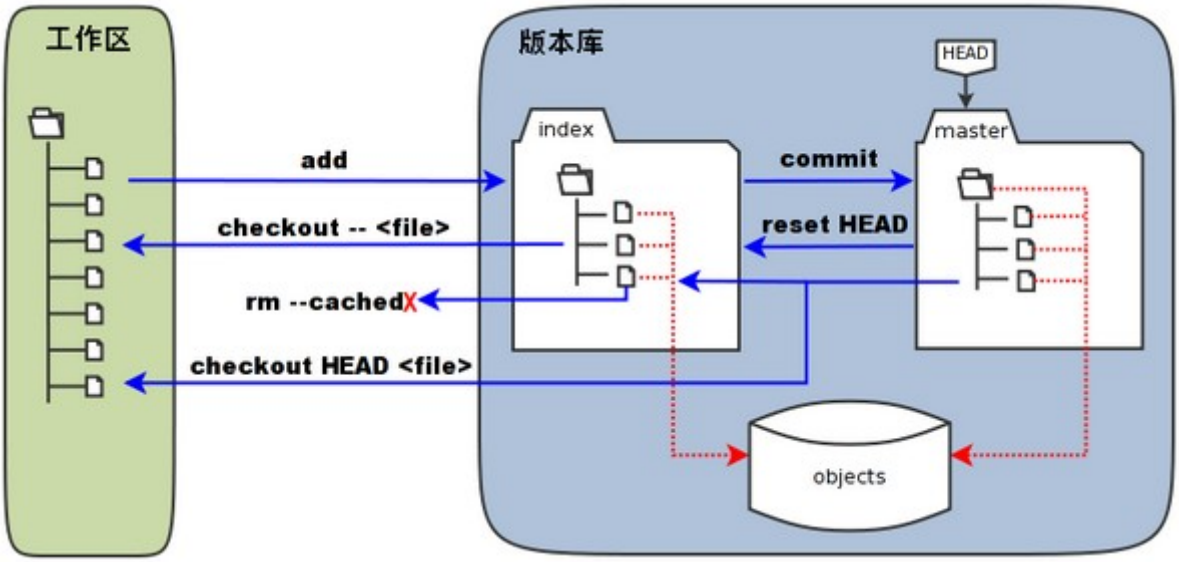
- git checkout 用暂存区内容替换工作区内容，可能导致工作区变更丢失
- git checkout -- <file> 用暂存区内容替换工作区内容，可能导致工作区变更丢失

暂存区：在.git 的 index 区

- git add 工作区变更更新到暂存区
- git reset HEAD 版本库覆盖暂存区的内容，但工作区变更不影响
- git rm --cached <file> 直接从暂存区删除文件，但工作区变更不影响

版本库：在.git 的 master 区，其中 HEAD 就是指向 master 的指针

- git commit 则加到版本库
- git checkout HEAD 用版本库内容同时替换暂存区和工作区的内容，可能导致全覆盖
- git checkout HEAD <file> 用版本库内容同时替换暂存区和工作区的内容，可能导致全覆盖



（5）从远程仓库克隆到本地(一般就在第一次使用克隆命令，后续都是 git pull)
git clone [github 远程仓库的 ssh 地址]，克隆结束会显示如下信息：

```
ubuntu@ubun:~/suliang_git$ git clone git@github.com:ximitiejiang/MyCodeForObject
Detection.git
Cloning into 'MyCodeForObjectDetection'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 24 (delta 3), reused 20 (delta 2), pack-reused 0
Receiving objects: 100% (24/24), 22.99 MiB | 603.00 KiB/s, done.
Resolving deltas: 100% (3/3), done.
```

（6）从远程更新到本地仓库

```
git pull
```

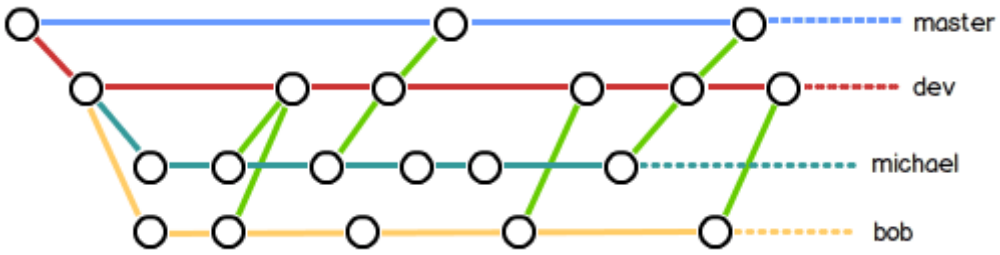
（7）从本地更新到远程仓库
注意：先进入该仓库目录下才能 git status

1. 进入本地某仓库：cd xxx
2. 检查文件状态：git status -s 此为显示 short 版文件区别
3. 把变更加入暂存区：git add . 此为添加该仓库内所有变更文件进暂存区
4. 把变更加入版本库：git commit -m ‘变更描述’ 如没有描述则会打开一个额外文件填写
5. 把变更推送到仓库：git push

（8）解决冲突：合并远程库的新变更和本地的新变更
由于远程与本地都有变更，无法直接合并，则无法直接 push 或者 pull，需要先解决冲突

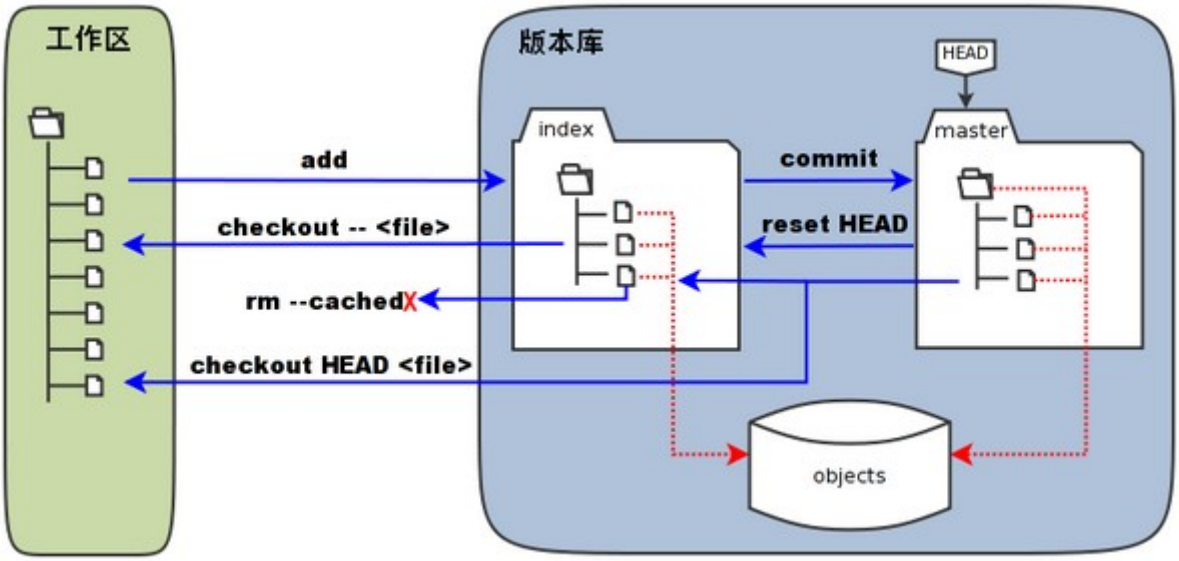
1. 进入本地某仓库：
   cd xxx
2. 先创建一个分支 dev，把本地相关更新 add 到这个分支：
   git checkout -b dev 创建新分支 dev，并切换到 dev 分支
   git add . 进入具体仓库目录，把更新 add 到当前的 dev 分支的暂存区(.表示全更新)
   git commit -m 把更新 commit 到当前分支的版本库
3. 切换到主分支 master，并更新主分支的内容
   git checkout master 切换到 master 主分支
   git pull 在主分支把远程更新拉下来到本地
4. 新建分支后需要建立与远程仓库的分支关联
   git push --set-upstream origin dev # 对新分支建立上传通道

（9）分支工作模式



`git status` 用于查看在暂存区有哪些变化了的文件(modified, deleted…)
`git diff 文件名` 用于查看本地文件变化的具体内容

- `git add 文件名` 用于增加要变更的文件到暂存区(在.git 文件夹的 stage 区域)
  - 暂存区内容取消：git
- `git commit -m “变更描述”` 用于正式提交变更到 master 分支(在.git 文件夹的 master 区域)
  - 回退方法
- `git push origin master` 用于把本地 master 推送到 github 中(合并到远程 github 仓库)
  - 回退方法：在网站上操作



（7）要回退过去或者返回未来的版本
HEAD 代表当前版本，HEAD^代表上一版本，HEAD^^代表上上版本，HEAD~100 代表上上 100 个版本
`git log`：可以显示我们的历史记录，也可 git log –pretty=online 更简洁
`git reset --hard HEAD^` ： 代表回退到上一版本
`git reset --hard 1049a` ： 代表回退到 commit id 以 1049a 开头的版本

`git reflog`：可以记录自己每一次命令，从而可以再从中找到对应 commit id 来恢复相关版本
`git reset` --hard xxx

（8）最重要的方法：创建分支，解决冲突

1. 创建分支：git checkout -b dev 代表先 branch 一个 dev，然后 checkout 到 dev

```
ubuntu@ubun:~$ git branch
* dev
  master
```

2. 查看分支：git branch 查看当前所有分支，且当前所在分支会标星号
3. 切换分支：git checkout master 切换到 master 主分支
4. 删除分支：git branch -d dev 删除 dev 分支
5. 合并分支：git

Q. 如何解決 Spyder 里边不能輸入中文?

参考：https://blog.csdn.net/weixin_41012049/article/details/82716554

参考：https://www.zhihu.com/question/56719140/answer/313083444

cd /usr/lib/x86_64-linux-gnu//qt5/plugins/platforminputcontexts/ (找到文件所在的文件夹)

cp libfcitxplatforminputcontextplugin.so /home/ubuntu/anaconda3/plugins/platforminputcontexts/ (将其复制到 anaconda 插件相应的文件下)

Q. 如何使用 conda 进行安装包的版本管理?

conda 是一个包管理软件，基本功能是自动搜索一条通道 channel，并自动下载软件

（1）基础操作

- 两个 dash 跟一个 dash 一样：[command --name] = [command -name] = [command --n] = [command -n]
- 当使用 conda，会已经默认使用一个环境，名叫 base
- 查看所有可用的环境：conda info –envs
- 创建一个新环境：conda create --name [your new env name] [your install software]
- 激活和使用新环境：source activate [your new env name]
- 返回缺省环境：source deactivate
- 实例创建一个新环境[py35]：conda create --name py35 python=3.5
- 查看指定环境中的软件清单：conda list

（2）搜索和安装新的软件

- conda search [software]：搜索软件
- conda install [software]：安装软件, conda install -e

Q. 如何安装 chainercv?

首先尝试用 conda 方式，这是 chainercv 推荐的，即 git clone chainercv 到本地，然后安装，但没有成功。然后尝试 pip 安装，发现 pip 版本太低(10.0.1)不能安装，只能先升级 pip 到最新版 pip3，但安装还是失败。

重启 terminal，先运行 pip3 install chainer，再运行 pip3 install chainercv，成功。

```
Successfully built chainercv
Installing collected packages: chainercv
Successfully installed chainercv-0.10.0
```

Q. 如何安装 opencv?

利用 conda 安装，conda install opencv，但失败了（估计跟默认环境-base 的 python 版本有关）

Q. 如何安装 CUDA?

CUDA 的介绍：是并行计算平台和编程模型，他通过驱动 GPU 快速提高运算能力，如果要使用 GPU，就必须用 CUDA 进行编程来驱动 GPU

（1）查看本机显卡：lspci | grep -VGA 说明本机一共 2 块显卡，都是 nvidia 公司的，后续如果安装 openGL 就不会冲突，因为 openGL 只支持 nvidia 的显卡，其他公司的会被 openGL 安装覆盖。

```
ubuntu@ubun:~/Downloads$ lspci | grep VGA
17:00.0 VGA compatible controller: NVIDIA Corporation Device 1b06 (rev a1)
65:00.0 VGA compatible controller: NVIDIA Corporation Device 1b06 (rev a1)
```

（2）查看本机 GPU：lspci | grep -i nvidia 此命令为显示(ls)所有 pci 设备，并只显示 nvidia 公司设备出现如下：

```
(base) ubuntu@ubun:~$ lspci | grep -i nvidia
17:00.0 VGA compatible controller: NVIDIA Corporation Device 1b06 (rev a1)
17:00.1 Audio device: NVIDIA Corporation Device 10ef (rev a1)
65:00.0 VGA compatible controller: NVIDIA Corporation Device 1b06 (rev a1)
65:00.1 Audio device: NVIDIA Corporation Device 10ef (rev a1)
```

（3）查看本机 gcc 版本：gcc --version 此 gcc 版本为 5.4.0，gcc 用于对 cuda 代码进行调试

从 nvidia 官网文件可以查到 ubuntu16.04 能支持的 GCC 是 5.4.0，这也是 CUDA10 的允许配置，跟我机器上的 ubuntu 版本及 GCC 版本是匹配的。如果安装 CUDA8 就有问题了，他只能支持 ubuntu16.04 里的 GCC5.3.1，就得对 GCC 进行降版本。

Table 1 Native Linux Distribution Support in CUDA 10.0

Distribution	Kernel*	GCC	GLIBC	ICC	PGI	XLC	CLANG
x86_64							
RHEL 7.5	3.10	4.8.5	2.17	18.0	18.x	NO	6.0.0
RHEL 6.10	2.6.32	4.4.7	2.12				
CentOS 7.5	3.10	4.8.5	2.17				
CentOS 6.10	2.6.32	4.4.7	2.12				
Fedora 27	4.13.9	7.3.1	2.26				
OpenSUSE Leap 15.0	4.12.14	7.3.1	2.26				
SLES 15.0	4.12.14	7.2.1	2.26				
Ubuntu 18.04.1 (**)	4.15.0	7.3.0	2.27				
Ubuntu 16.04.5 (**)	4.4	5.4.0	2.23				
Ubuntu 14.04.5 (**)	3.13	4.8.4	2.19				

```
(base) ubuntu@ubun:~$ gcc --version
gcc (Ubuntu 5.4.0-6ubuntu1~16.04.10) 5.4.0 20160609
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

（4）查看本机是否有正确的内核 Head 和安装包: CUDA 的安装要求 kernel 的版本跟 kernel head 以及跟开发包的版本必须匹配。因为 CUDA 在安装的时候如果检测到没有安装过 kernel head 和开发包会安装相应 kernel head 和开发包，但他是安装最新版，不一定跟本机的 kernel 匹配。所以最好手动先安装正确的版本的 kernel header 和开发包。

查看本机 kernel 版本：uname -r

```
ubuntu@ubun:~$ uname -r
4.15.0-36-generic
```

安装匹配的 kernel header 和开发包(ubuntu)：sudo apt-get install linux-headers-$(uname -r)

```
ubuntu@ubun:~$ sudo apt-get install linux-headers-$(uname -r)
[sudo] password for ubuntu:
Reading package lists... Done
Building dependency tree
Reading state information... Done
linux-headers-4.15.0-36-generic is already the newest version (4.15.0-36.39~16.04.1).
linux-headers-4.15.0-36-generic set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
```

（5）安装适合自己的 GPU 驱动：机器买的时候据说已经安装，保险起见，先查一下

cat /proc/driver/nvidia/version 查看显卡版本：显示为 NVRM 已安装，gcc 已安装

nvcc -V 查看 cuda 版本：显示为 cuda toolkit 还没有安装过

```
ubuntu@ubun:~$ cat /proc/driver/nvidia/version
NVRM version: NVIDIA UNIX x86_64 Kernel Module  384.130  Wed Mar 21 03:37:26 PDT
 2018
GCC version:  gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.10)
ubuntu@ubun:~$ nvcc -v
The program 'nvcc' is currently not installed. You can install it by typing:
sudo apt install nvidia-cuda-toolkit
```

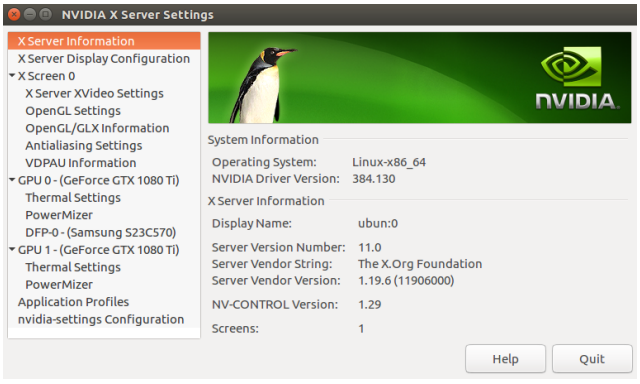
如下两个命令看看显卡驱动的信息：

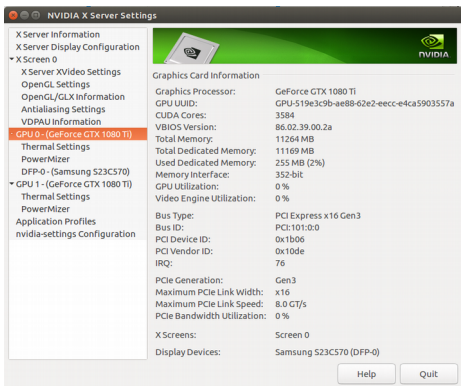
nvidia-smi 用于查看显卡状态，两块 GeForce GTX1080ti 显卡

```
ubuntu@ubun:~$ nvidia-smi
Fri Oct 19 10:06:33 2018
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 384.130                Driver Version: 384.130                   |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GeForce GTX 108...  Off  | 00000000:17:00.0 Off |                  N/A |
| 29%   26C    P8     7W / 250W |      1MiB / 11172MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  GeForce GTX 108...  Off  | 00000000:65:00.0  On |                  N/A |
| 29%   42C    P8    15W / 250W |    237MiB / 11169MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    1      1086      G   /usr/lib/xorg/Xorg                           121MiB |
|    1      1904      G   compiz                                        95MiB |
|    1      2124      G   fcitx-qimpanel                                10MiB |
|    1      2987      G   /usr/lib/firefox/firefox                       2MiB |
|    1      3105      G   /usr/lib/firefox/firefox                       2MiB |
|    1      3137      G   /usr/lib/firefox/firefox                       2MiB |
+-----------------------------------------------------------------------------+
```

nvidia-settings 用于查看设置

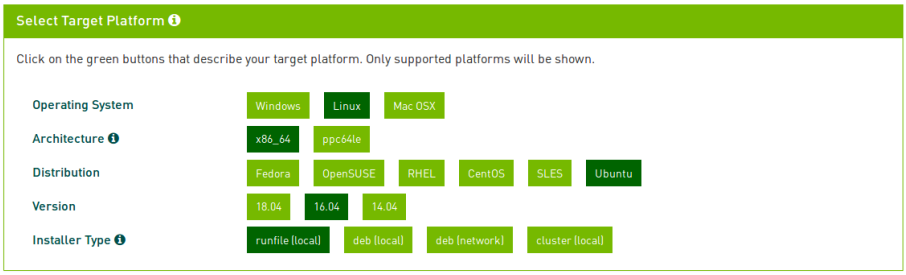


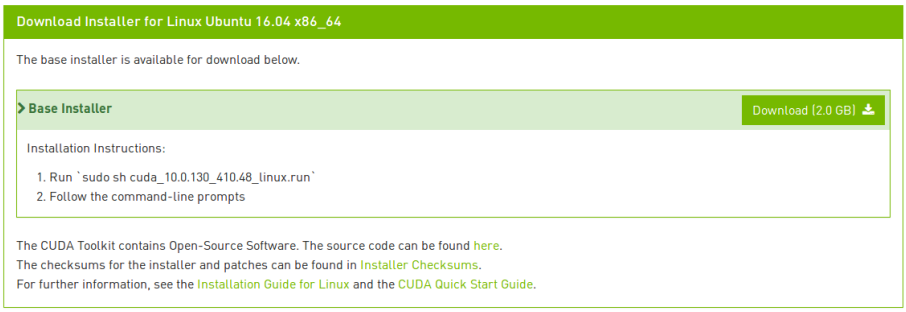


（6）下载 CUDA

从官网上选择安装方式：https://developer.nvidia.com/cuda-downloads

这里选择 runfile，不需要复杂的安装命令和后续一些补充包的安装，直接一部到位。带网线的下载速度不错，从 nvidia 网站下载 2G 的文件接近 1M/s 的速度。



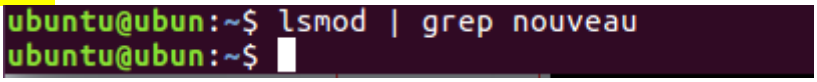


接下来基于 runfile installation 方式进行 NVIDIA 驱动/CUDA toolkit 的安装

（7）首先禁掉 nouveau driver：nouveau 是 ubuntu16.04 默认安装的第三方开源驱动，安装 cuda 会跟 nouveau 冲突，需要事先禁掉，运行命令后需要没有任何输出就代表禁掉了。

我的机器本来就没输出，不知道为什么。

lsmode | grep nouveau



（8）重启进入 text mode（runlevel 3）可以按 ctrl+alt+f1 进入文本模式

然后进入安装文件夹执行安装命令：sudo sh cuda_10.0.130_410.48_linux.run

第一次安装失败：提示说 X server is running

```
-----------------
Do you accept the previously read EULA?
accept/decline/quit: accept

Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 410.48?
(y)es/(n)o/(q)uit: y

Do you want to install the OpenGL libraries?
(y)es/(n)o/(q)uit [ default is yes ]: n

Do you want to run nvidia-xconfig?
This will update the system X configuration file so that the NVIDIA X driver
is used. The pre-existing X configuration file will be backed up.
This option should not be used on systems that require a custom
X configuration, such as systems with multiple GPU vendors.
(y)es/(n)o/(q)uit [ default is no ]: y

Install the CUDA 10.0 Toolkit?
(y)es/(n)o/(q)uit: y

Enter Toolkit Location
 [ default is /usr/local/cuda-10.0 ]:

Do you want to install a symbolic link at /usr/local/cuda?
(y)es/(n)o/(q)uit: y

Install the CUDA 10.0 Samples?
(y)es/(n)o/(q)uit: y

Enter CUDA Samples Location
 [ default is /home/ubuntu ]:

Installing the NVIDIA display driver...
It appears that an X server is running. Please exit X before installation. If you're s
ure that X is not running, but are getting this error, please delete any X lock files
in /tmp.

===========
= Summary =
===========

Driver:   Installation Failed
Toolkit:  Installation skipped
Samples:  Installation skipped
```

重新上网查了下网上的说，应该是我的 X server 没有关掉，解决办法是：

ctrl + alt + f1　进入文字界面

sudo service lightdm stop　关闭图形界面，此时如果 ctrl + alt + f7 是无法返回图形界面的

sudo sh cuda_10.0.130_410.48_linux.run　重新进入安装文件夹进行安装

- nvidia accelerated graphics driver ——不安装
- openGL——不安装
- 其他都 yes

最后安装完成，但提示 3 个 recommended library missing，以及一个 warning 需要安装高于 384.00 的驱动版本。我查了下之前安装的驱动版本是 384.130，如下，是满足要求的。

```
ubuntu@ubun:~$ nvidia-smi
Fri Oct 19 10:06:33 2018
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 384.130                Driver Version: 384.130                   |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GeForce GTX 108...  Off  | 00000000:17:00.0 Off |                  N/A |
| 29%   26C    P8     7W / 250W |      1MiB / 11172MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  GeForce GTX 108...  Off  | 00000000:65:00.0  On |                  N/A |
| 29%   42C    P8    15W / 250W |    237MiB / 11169MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    1      1086      G   /usr/lib/xorg/Xorg                           121MiB |
|    1      1904      G   compiz                                        95MiB |
|    1      2124      G   fcitx-qimpanel                                10MiB |
|    1      2987      G   /usr/lib/firefox/firefox                       2MiB |
|    1      3105      G   /usr/lib/firefox/firefox                       2MiB |
|    1      3137      G   /usr/lib/firefox/firefox                       2MiB |
+-----------------------------------------------------------------------------+
```

重启 reboot，回到图形界面

（9）环境设置：在/etc/profile 文件中进行设置，但由于该文件是只读的，直接打开后不能保存。需要用 sudo gedit 命令打开，这样就能保存了。然后用 source 命令(也叫点命令)重新执行刚修改的初始化文件，使之立刻生效而不必注销重新登录

`sudo gedit /etc/profile` 打开/etc/profile 文件

export PATH=/usr/local/cuda-10.0/bin${PATH:+:${PATH}} 把路径包含进 path 变量

export LD_LIBRARY_PATH=/usr/local/cuda-10.0/lib64\${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}} 修改 64x 系统环境变量

`source /etc/profile` 关闭/etc/profile 文件

（10）检查安装结果

cat /proc/driver/nvidia/version 查看 nvidia 驱动的版本（版本 340.130）

nvcc -V 查看 CUDA 的版本（注意是大写 V，版本 V10.0.130）

```
ubuntu@ubun:~$ cat /proc/driver/nvidia/version
NVRM version: NVIDIA UNIX x86_64 Kernel Module  384.130  Wed Mar 21 03:37:26 PDT
 2018
GCC version:  gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.10)
ubuntu@ubun:~$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2018 NVIDIA Corporation
Built on Sat_Aug_25_21:08:01_CDT_2018
Cuda compilation tools, release 10.0, V10.0.130
```

安装成功!

Q. 如何安装 cuDNN?

cuDNN 是 nvidia 开发的 deep neural network library(dnn)，即为深度神经网络 GPU 加速库。提供了前向/后向/卷积/池化/标准化/激活等层。

（1）注册

https://developer.nvidia.com/rdp/form/cudnn-download-survey

（2）选择 cuDNN 版本：需要跟 CUDA 的版本匹配，由于安装的是 CUDA10，需选择 cuDNN v7.3.1 这里选择了 cuDNN v7.3.1 library for linux。没有选择 runtime library forubuntu 16.04，安装说明里说，cuDNN v7.3.1 lib for linux 这种 tar 文件(下载的是 tgz 压缩文件)适用于所有 linux 系统，而 debian 文件适用于 ubuntu 系统。

（3）安装前提：

- GPU
- x86_64 – ubuntu 14.04 or ubuntu 16.04
- CUDA: 包括 nvidia 驱动

（4）进入下载目录，解压缩 cuDNN 安装文件

```
tar -xzvf cudnn-9.0-linux-x64-v7.tgz
```

```
ubuntu@ubun:~/Downloads$ tar -xzvf cudnn-10.0-linux-x64-v7.3.1.20.tgz
cuda/include/cudnn.h
cuda/NVIDIA_SLA_cuDNN_Support.txt
cuda/lib64/libcudnn.so
cuda/lib64/libcudnn.so.7
cuda/lib64/libcudnn.so.7.3.1
cuda/lib64/libcudnn_static.a
```

（5）拷贝如下解压缩文件到 CUDA toolkit 的目录中，然后变更文件允许

```
sudo cp cuda/include/cudnn.h /usr/local/cuda/include
sudo cp cuda/lib64/libcudnn* /usr/local/cuda/lib64
sudo chmod a+r /usr/local/cuda/include/cudnn.h /usr/local/cuda/lib64/libcudnn*
```

即完成安装了。

（6）查证是否安装成功

```
nvcc -V
```

Q. 如何安装 visdom?

visdom 是 facebook 开发的一款针对 pytorch 的可视化工具

（1）安装：

```
pip3 install visdom
```

之前安装的旧版会有打不开的问题，务必升级到最新版

```
pip3 install --upgrade visdom
```

（2）开启服务：

```
python -m visdom.server
```

这步需要的时间比较多，耐心等待 n 分钟就好。

```
ubuntu@ubun:~$ python -m visdom.server
/home/ubuntu/anaconda3/lib/python3.7/site-packages/visdom/server.py:30: Deprecat
ionWarning: zmq.eventloop.ioloop is deprecated in pyzmq 17. pyzmq now works with
 default tornado and asyncio eventloops.
  ioloop.install()  # Needs to happen before any tornado imports!
Downloading scripts. It might take a while.
It's Alive!
INFO:root:Application Started
You can navigate to http://localhost:8097
```

（3）运行代码

（4）打开 http://localhost:8097，查看输出

要求

1. 计算机专业硕士本科，研究方向图像处理和深度学习
2. 熟练 C/C++和 python，Linux 平台
3. 熟悉 tensorflow/pytorch 之一，熟悉 openCV，有 GPU 上模型调试经验
4. 了解数据库处理