https://www.tensorflow.org/install/

10.20.210.55服务器上

**Anaconda安装指南**

1.可以到清华大学开源软件镜像站下载指定的安装包<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>

(anaconda官网上的是最新版本的，python都是3.7的，有些包不太支持)

2. 进行安装

打开终端——进入下载安装包所在的文件夹（cd 文件名）——输入命令：bash + 下载包文件名 ——进行安装，根据提示选择yes——安装成功；

例如：打开terminal ———》

cd Downloads/ ———》

bash Anaconda3-5.2.0-Linux-x86\_64.sh ———》 一路yes

提示信息“Do you wish to proceed with the installation of Microsoft VSCode? [yes|no]”，输入no；

3.验证安装是否成功：

方法一：启动新终端；查看conda版本：$ conda –version 如正常显示则证明安装成功；

方法二：启动新终端；打开Python；$ python –V 输入import scipy，未报错即已成功。

4.配置环境变量

sudo gedit ~/.bashrc ——》

export PATH="/home/xupp/anaconda3/bin:$PATH" ——》

source ~/.bashrc ——》

或者

$ vi .bash\_profile

在最后一行加上如下配置:

export PATH=/anaconda/anaconda3/bin:$PATH

如果是默认安装路径的话则为:

export PATH=/home/py/anaconda3/bin:$PATH

然后使其生效:

$source .bash\_profile

或者

在终端输入$sudo gedit /etc/profile，打开profile文件。

在文件末尾添加一行：export PATH=/home/grant/anaconda2/bin:$PATH，其中，将 “/home/grant/anaconda2/bin”替换为你实际的安装路径。保存。

* 添加anaconda环境变量   
  输入命令：vi /etc/profile 进入 /etc/profile 文件，在文件末尾加上如下两行代码

PATH=$PATH:/root/anaconda3/bin #路径名跟自己实际情况而定

export PATH

5. 在ubuntu上卸载anaconda的步骤 ：

1）删除整个anaconda目录：

    由于Anaconda的安装文件都包含在一个目录中，所以直接将该目录删除即可。到包含整个anaconda目录的文件夹下，删除整个Anaconda目录：

    rm -rf anaconda文件夹名

2）建议——清理下.bashrc中的Anaconda路径：

   1.到根目录下，打开终端并输入：

      sudo gedit ~/.bashrc

   2.在.bashrc文件末尾用#号注释掉之前添加的路径(或直接删除)：

      #export PATH=/home/lq/anaconda3/bin:$PATH

      保存并关闭文件

   3.使其立即生效，在终端执行：

      source ~/.bashrc

   4.关闭终端，然后再重启一个新的终端，这一步很重要，不然在原终端上还是绑定有anaconda.

**bash配置文件/etc/profile，/etc/bashrc，~/.bash\_profile，~/.bashrc详解及修改PATH环境变量**

1、bash 的配置文件

1）、全局配置和个人配置

全局配置

/etc/profile ,/rtc/profile.d/\*.sh,/etc/bashrc

个人配置

~/.bash\_profile,~/.bashrc

2)、各配置文件解析：

/etc/profile:此文件为系统的每个用户设置环境信息，当用户第一次登录时，该文件被执行，并从/etc/profile.d目录的配置文件中搜集shell的设置；

/etc/bashrc:为每一个运行bash shell的用户执行此文件，当bash shell被打开时，该文件被读取。

~/.bash\_profile:每个用户都可以使用该文件输入专用于自己使用的shell信息，当用户登录时，该文件仅仅执行一次默认情况下，其他设置一些环境变量，执行用户的.bashrc文件。

~/.bashrc:该文件包含用于你的bash shell 的bash信息，当登录时以及每次打开新的shell时，该文件被读取。

3）profile 类文件和bashrc类文件的区别：

profile类的文件：

设定环境变量

运行命令或脚本

bashrc类的文件：

设定本地变量

定义命令别名

4)shell如何读取配置文件

登录式shell如何读取配置文件

/etc/profile -->/etc/profile.d/\*.sh

-->~/.bash\_profile-->~/.bashrc-->/etc/bashrc

非登录式shell如何配置文件

~/.bashrc-->/etc/bashrc-->/etc/profile.d/\*.sh

2、export 临时导入环境变量

$PATH:决定了shell将到哪些目录中寻找命令程序，PATH的值是一系列目录，当您运行一个程序时，Linux在这些目录下进行搜寻编译链接

**安装TensorFlow的GPU版本指南**

GPU 版本则要求计算机有一个 CUDA compute capability 3.0 及以上的 NVDIA GPU 显卡（对于台式机而言最低配置为 NVDIA GTX 650）

对于 TensorFlow GPU 版本，需要先安装 CUDA toolkit 7.0 及以上版本、NVDIA【R】驱动程序和 cuDNN v3 或以上版本

**查看GPU版本信息**  
nvidia-smi

**查看GPU型号**

lspci | grep -i nvidia

**查看NVIDIA驱动版本**

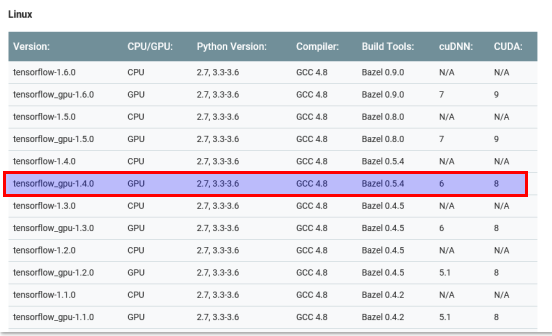
sudo dpkg --list | grep nvidia-\*

lspci | grep VGA # 查看集成显卡

lspci | grep NVIDIA # 查看NVIDIA显卡

**注意：** 安装之前一定要先把CUDA和cuDNN的版本搞清楚了，因为CUDA与cuDNN的版本必须要和Tensorflow的版本对应上，否则即使安装成功，最后在python环境里导入tensorflow时也会报错。如下图所示。

<https://www.tensorflow.org/install/source>此网站可以查看Tensorflow，cuda，cudnn对应版本，也可下载最新tensorflow



此网站是TensorFlow各版本自主下载地址：<https://pypi.python.org/pypi/tensorflow-gpu/1.5.0>

Tensorflow版本1.5.0

SHA256：57b1ebfaba9fa2d9834c943ce00f6f6f886b3e2a9fe60eb5551cfdec78c4c780

MD5：470ea7ec62a189424ed742f7743e761a

BLAKE2-256: d58b094add4d2d667ddfef867285e2c16ac4fee6a1d51cece87d6d490b6e5571

得到本文选择的版本是Tensorflow1.5，cuda9，cudnn7。分别去安装相应的版本即可

**第一步：安装NVIDIA驱动**

**去英伟达官网下载驱动程序（<http://www.nvidia.cn/Download/index.aspx?lang=cn>）**

**查看GPU版本信息**  
nvidia-smi

知道centos7的GPU版本信息 Tesla V100-PCIE系列 CUDA选择9.1 Driver Version:390.46

**禁用nouveau第三方驱动**

【centos 下，如果默认安装了nouveau，一个第三方开源的nvidia驱动，必须将其禁用掉，如果没安装，忽略此步】

打开编辑配置文件： sudo gedit /etc/modprobe.d/blacklist.conf

在最后一行添加：blacklist nouveau

改好后执行命令：sudo update-initramfs –u #centos7是 sudo yum update-initramfs –u

重启使之生效：reboot

注意：

重启之后，可以查看nouveau有没有运行:

lsmod | grep nouveau # 没输出代表禁用生效

sudo vim /etc/modprobe.d/blacklist.conf //修改这个文件，在文件的后面添加第2行的内容

2 blacklist nouveau

3 sudo update-initramfs -u //执行这条语句之后重启系统

4 lsmod | grep no //如果没有任何输出证明禁止nouveau驱动成功

**安装驱动**

执行命令：lsmod | grep nouveau

禁用X服务：sudo /etc/init.d/lightdm stop (或者：sudo service lightdm stop)

【如果此步执行不成功：

在终端输入init 3 切换到超级管理员用户登陆：

输入 root

输入登录密码 111111

输入命令：sudo service lightdm stop

sudo chmod a+x NVIDIA-Linux-x86\_64-390.116.run

sudo sh ./NVIDIA-Linux-x86\_64-390.116.run -no-opengl-files弹框 一路接受 同意即可

这里可以参考<https://www.flagxue.cn/259.html>的步骤】

给驱动文件增加可执行权限：sudo chmod a+x NVIDIA-Linux-x86\_64-390.116.run

然后执行安装：sudo sh ./NVIDIA-Linux-x86\_64-390.116.run -no-opengl-files

参数介绍：

–no-opengl-files 只安装驱动文件，不安装OpenGL文件。这个参数最重要

–no-x-check 安装驱动时不检查X服务

–no-nouveau-check 安装驱动时不检查nouveau  
后面两个参数可不加。

检查GPU版本信息：

输入nvidia-smi命令，显示安装的版本，就表示安装成功！

**第二步：安装CUDA**

**（cuda 9.1）**

各个版本下载：

https://developer.nvidia.com/cuda-toolkit-archive

注意：下载runfile（local）文件

安装前的准备

**确认有可以执行cuda程序的gpu**

$ lspci | grep -i nvidia

**确认linux版本正确**

$ uname -m && cat /etc/\*release

**确认安装了gcc ，cuda工具箱需要gcc编译器，但是运行cuda应用不需要gcc模块**

$ gcc --version

**如果没有安装则安装gcc、g++编译器**

sudo yum install gcc gcc-c++

**查看系统的kernel版本**

$ uname -r

**安装kernel headers和开发包**

$ sudo yum install kernel-devel-$(uname -r) kernel-headers-$(uname -r)

**禁止Nouveau显卡驱动 （看上文）**

**开始安装**

下载完成以后，找到CUDA  
所在位置（我的是放在Downloads目录下），终端输入：.

            sudo chmod +x cuda\_8.0.61\_375.26\_linux.run

sudo ./cuda\_8.0.61\_375.26\_linux.run 或者sudo sh cuda\_8.0.61\_375.26\_linux.run

安装过程中会有几个选项需要确认：

【注意：不要安装cuda自带的显卡驱动，其他的直接选y，路径输入默认值即可，即出现Install NVIDIA Accelerated Graphics Driver for Linux-x86\_64 367.48?时 选择no，因为已经安装驱动了。】

**安装cuda时可能有下面的信息**

**Installing the CUDA Toolkit in /usr/local/cuda-8.0 …**

**Missing recommended library: libGLU.so**

**Missing recommended library: libX11.so**

**Missing recommended library: libXi.so**

**Missing recommended library: libXmu.so**

**……**

Driver: Not Selected

Toolkit: Installed in /usr/local/cuda-8.0

Samples: Installed in /home/qjk, but missing recommended libraries

原因是缺少相关的依赖库：

sudo apt-get install freeglut3-dev build-essential libx11-dev libxmu-dev libxi-dev libg libgl1-mesa-glx libglu1-mesa libglu1-mesa-dev

**设置环境变量**

**sudo gedit ~/.bashrc**

export PATH=/usr/local/cuda-8.0/bin:$PATH

export LD\_LIBRARY\_PATH=/usr/local/cuda-8.0/lib64:$LD\_LIBRARY\_PATH

export CUDA\_HOME=/usr/local/cuda

【前 2 个（PATH, LD\_LIBRARY\_PATH) 是 CUDA 官网安装文档中建议的变量。

第 3 个（CUDA\_HOME）是 tensorflow-GPU 版本要求的变量】

保存退出

$ source ~/.bashrc（立即生效文件）

**验证:**

终端输入： nvcc -V

可以看到cuda的版本信息：

**cuda的卸载：**

sudo /usr/local/cuda-9.0/bin/uninstall\_cuda\_9.0.pl

卸载之后，还有一些残留的文件夹，之前安装的是CUDA 9.0。可以一并删除：

sudo rm -rf /usr/local/cuda-9.0/

**第三步：安装深度学习库cuDNN**

到此网站<https://developer.nvidia.com/rdp/cudnn-archive>

需要注册为开发者才能下载，而且直接下载可能速度非常慢

名文件名为：cudnn-9.1-linux-x64-v7.1.tgz

**解压**

$ tar xvzf cudnn-8.0-linux-x64-v5.1.tgz

**然后将库和头文件copy到cuda目录**

（一定是你自己安装的目录如/usr/local/cuda-8.0）,不过正确安装的话，ubuntu一般就会有软链接/usr/local/cuda -> /usr/local/cuda-8.0/

$ sudo cp cuda/include/cudnn.h /usr/local/cuda/include

$ sudo cp cuda/lib64/libcudnn\* /usr/local/cuda/lib64

**接下来就是修改文件访问权限：**

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

**第四步：安装tensorflow的gpu版本**

**建立环境**

conda create -n tensorflow python=3.6 #环境名直接为tensorflow

如果是在离线状态下则执行此命令：conda config --set offline true

**激活环境**

source activate tensorflow

**安装**

**注意要安装在tensorflow环境下**

cd /home/wudl/.conda/envs/tensorflow #进入环境目录

**pip install tensorflow**

**退出环境**

source deactivate

**检查tensorflow安装情况**

**activate tensorflow**

python

**import tensorflow as tf**

**hello = tf.constant(‘hello,tensorflow!’)**

**sess = tf.Session()**

**print(sess.run(hello))**

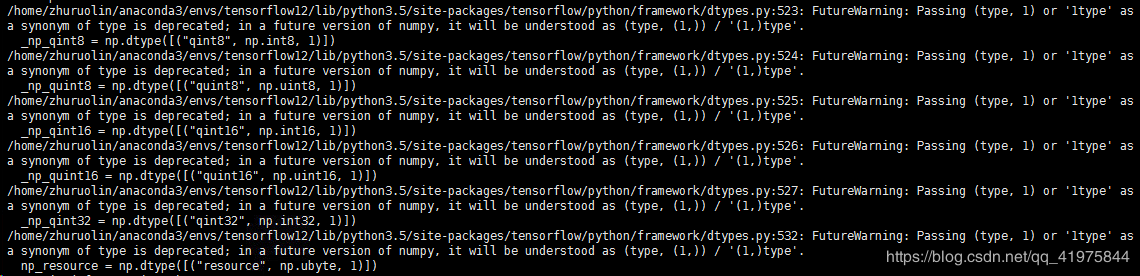
**b’hello,tensorflow!’**

**a=tf.constant(10)**

**b=tf.constant(32)**

**print(sess.run(a+b))**

**补充：安装完成，import tensorflow时可能出现如下警告**

****

解决方法：修改警告提示的

**解决方案，找到/home/hitwh/anaconda3/envs/tensorflow/lib/python3.5/site-packages/tensorflow/python/framework目录下的dtypes.py**

**相应行修改为例如将\_np\_quint8 = np.dtype([(“quint8”, np.uint8, 1)])改为\_np\_quint8 = np.dtype([("quint8", np.uint8, (1,))])**

**Centos7的tensorflow-gpu环境：**

**----- Anaconda3（5.2.0）**

**----- python（3.6.9）**

**-----cuda9（cuda\_9.0.176\_384.81\_linux.run）**

**-----cudnn7（cudnn-9.0-linux-x64-v7.6.4.38）**

**-----tensorflow\_gpu（tensorflow\_gpu-1.12.0-cp36-cp36m-linux\_x86\_64.whl）**

**https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow\_gpu-1.12.0-cp36-cp36m-linux\_x86\_64.whl**

**1.更新pip**

**pip3 install --upgrade pip**

**不更新的话,安装tensorflow可能会报错Could not find a version that satisfies the requirement tensorflow-gpu==1.4 (from versions: ) No matching distribution found for tensorflow-gpu==1.4**

**测试**

**输入python3进入交互命令**

**import tensorflow as tf没有报错就安装成功**

**测试问题：**

**有时候出现了错误ImportError: libcublas.so.8.0: cannot open shared object file: No such file or directory**

**1.检查/usr/local/cuda-8.0/lib64目录，看看libcudart.so.8.0是否存在，如果不存在，则先正确安装cuda。如果确定cuda已经存在，这个问题的出现于Linux的ldconfig命令有关。**