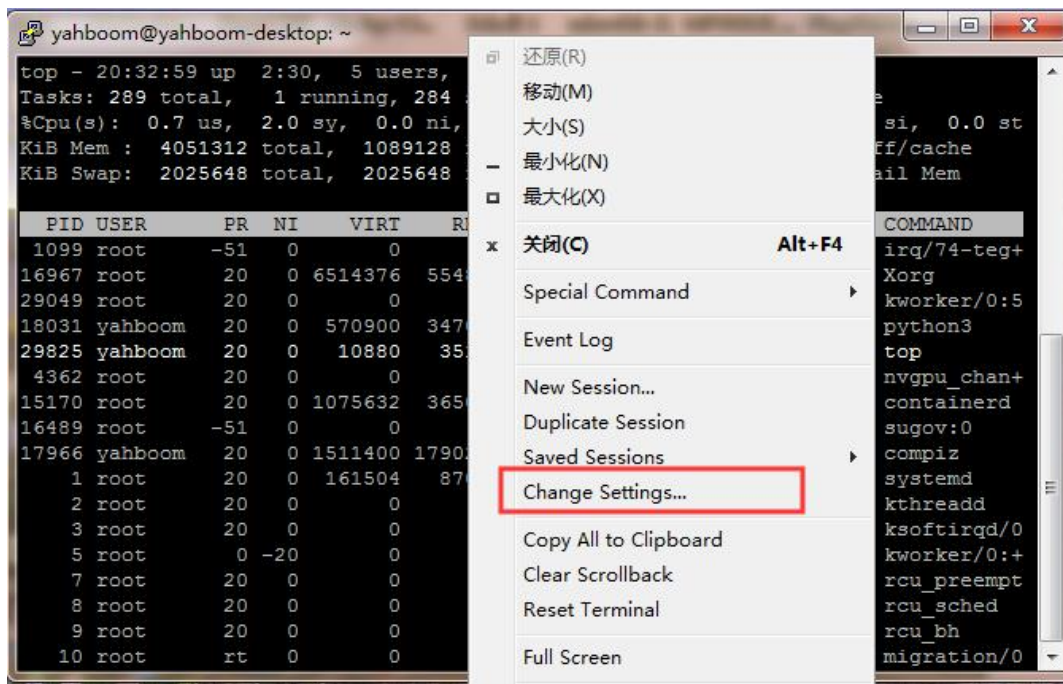


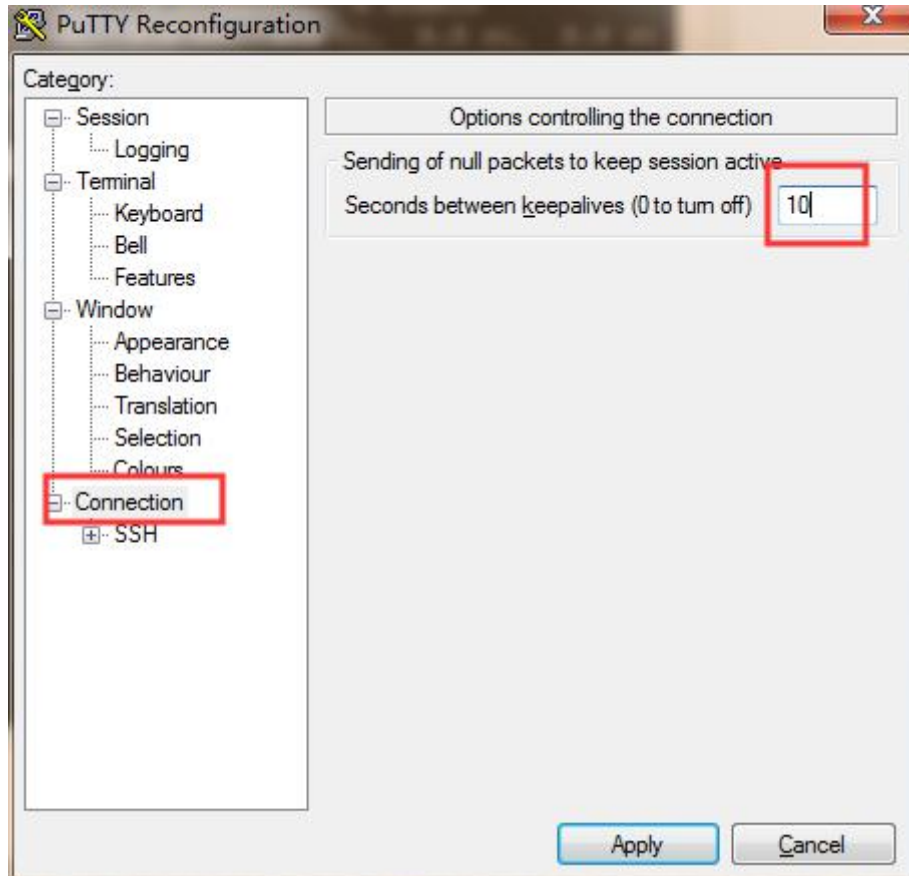
Jetson Xavier NX 开始玩转之前准备工作

1、远程登录。

根据自己喜好选择 PuTTY、SSH、Xshell 等工具远程登录。以下以 PuTTY 工具举例。注意：如果发现电脑无法远程，可以尝试双方互相 ping 一下，NX 上查看 ip 地址命令：ifconfig。Windows 下查看本地 ip 地址 cmd 命令：ipconfig。知道对方 ip 地址后，ping 192.168.1.xx 后面 ip 地址根据实际命令得出来的修改。

如果发现 putty 经常会自动掉线，可以尝试以下方法：





- A.进入 putty, 选择左侧的 Connection
B.在右侧有 Sending of null packets to keep session active
设为 10 即可
(意思每十秒发送一次空包用来保持连接)

2、关于更新源。

一般来说, 安装完系统后应当更新源, 但是由于 Jetson Xavier NX 采用的是 aarch64 架构的 Ubuntu 18.04.2 LTS 系统, 与 AMD 架构的 Ubuntu 系统不同, 而我没有找到完美的国内源, 所以不推荐大家换源。

这里并没有换源, 还是使用 Jetson Nano 的默认源进行更新。更新过程很漫长, 大家可以执行完命令, 做其他事吧。以下两个操作建议一定在做 AI 的项目前执行, 否则安装一些库会找不到安装地址, 导致后面频繁出错。

sudo apt-get update

```

nx@nx-desktop:~$ sudo apt-get update
[sudo] password for nx:
Get:1 file:/var/cuda-repo-10-2-local-10.2.89 InRelease
Ign:1 file:/var/cuda-repo-10-2-local-10.2.89 InRelease
Get:2 file:/var/visionworks-repo InRelease
Ign:2 file:/var/visionworks-repo InRelease
Get:3 file:/var/visionworks-sfm-repo InRelease
Ign:3 file:/var/visionworks-sfm-repo InRelease
Get:4 file:/var/visionworks-tracking-repo InRelease
Ign:4 file:/var/visionworks-tracking-repo InRelease
Get:5 file:/var/cuda-repo-10-2-local-10.2.89 Release [574 B]
Get:5 file:/var/cuda-repo-10-2-local-10.2.89 Release [574 B]
Get:6 file:/var/visionworks-repo Release [2,001 B]
Get:6 file:/var/visionworks-repo Release [2,001 B]
Get:7 file:/var/visionworks-sfm-repo Release [2,005 B]
Get:7 file:/var/visionworks-sfm-repo Release [2,005 B]
Get:8 file:/var/visionworks-tracking-repo Release [2,010 B]
Get:8 file:/var/visionworks-tracking-repo Release [2,010 B]
0% [Release.gpg gpgv 2,005 B] [Waiting for headers] [Connecting to repo.download.nvidia.com]

```

sudo apt-get full-upgrade

```

nx@nx-desktop:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done

```

在过程中输入 Y 确认更新。第二个过程时间根据网络情况可能会花 2 个小时左右，请耐心等待。完成后如下图。

```

正在设置 python3-distupgrader (1:18.04.32) ...
正在设置 libreoffice-common (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 libreoffice-core (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 ubuntu-release-upgrader-core (1:18.04.32) ...
正在设置 python3-uno (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 libreoffice-gtk3 (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 libreoffice-style-breeze (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 libreoffice-gnome (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 libreoffice-pdfimport (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 libreoffice-draw (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 libreoffice-avmedia-backend-gstreamer (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 ubuntu-release-upgrader-gtk (1:18.04.32) ...
正在设置 update-manager-core (1:18.04.11.10) ...
正在设置 libreoffice-impress (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 libreoffice-math (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 libreoffice-base-core (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 libreoffice-calc (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 update-manager (1:18.04.11.10) ...
正在设置 libreoffice-ogltrans (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 libreoffice-writer (1:6.0.7-0ubuntu0.18.04.5) ...
正在设置 ubuntu-desktop (1.417.1) ...
正在处理用于 libc-bin (2.27-3ubuntu1) 的触发器 ...
正在处理用于 resolvconf (1.79ubuntu10.18.04.3) 的触发器 ...
正在处理用于 initramfs-tools (0.130ubuntu3.7) 的触发器 ...
update-initramfs: Generating /boot/initrd.img-4.9.140-tegra
Warning: couldn't identify filesystem type for fsck hook, ignoring.
/sbin/ldconfig.real: Warning: ignoring configuration file that cannot be opened
/etc/ld.so.conf.d/aarch64-linux-gnu_EGL.conf: No such file or directory
/sbin/ldconfig.real: Warning: ignoring configuration file that cannot be opened
/etc/ld.so.conf.d/aarch64-linux-gnu_GL.conf: No such file or directory

```

3、检查已经安装的系统组件

Jetson Xavier NX 的 OS 镜像已经自带了 JetPack, cuda, cudnn, opencv 等都已经安装好，并

有例子，这些例子安装路径如下所示

```
TensorRT /usr/src/tensorrt/samples/
CUDA /usr/local/cuda-/samples/
cuDNN /usr/src/cudnn_samples_v7/
Multimedia API /usr/src/tegra_multimedia_api/
VisionWorks /usr/share/visionworks/sources/samples/
/usr/share/visionworks-tracking/sources/samples/ /usr/share/visionworks-sfm/sources/samples/
OpenCV /usr/share/OpenCV/samples/
```

(1) 检查 CUDA

Jetson Xavier NX 中已经安装了 CUDA10.2 版本，但是此时你如果运行 `nvcc -V` 是不会成功的，需要你吧 CUDA 的路径写入环境变量中。OS 中自带 Vim 工具，所以运行下面的命令编辑环境变量

```
sudo vim ~/.bashrc
```

在最后添加下面三行：（英文输入法下按下 i，进入插入模式，上下键让光标移动到最下面一行，然后复制以下三行，在光标处按下鼠标右键就会自动复制进去，然后按住 `esc`，输入冒号放开 `esc`，在输入 `wq!` 强制保存退出）

```
export CUDA_HOME=/usr/local/cuda-10.0
export LD_LIBRARY_PATH=/usr/local/cuda-10.0/lib64:$LD_LIBRARY_PATH
export PATH=/usr/local/cuda-10.0/bin:$PATH
```

注意:如果添加上面三句还是显示 `nvcc not found`，首先，查看 `cuda` 的 `bin` 目录下是否有 `nvcc`：

```
cd /usr/local/cuda/bin
```

如果存在，把上面两句减少为下面两句

```
vim ~/.bashrc 进入配置文件； 添加以下两行：
```

```
export PATH=/usr/local/cuda/bin:$PATH
```

```
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
```

然后保存退出

然后需要 `source` 下生效下。

```
source ~/.bashrc
```

`source` 后，此时再执行 `nvcc -V` 执行结果如下

```
beckhans@Jetson:~$ nvcc -V
```

```
nx@nx-desktop:~$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Wed Oct 23 21:14:42 PDT 2019
Cuda compilation tools, release 10.2, V10.2.89
```

(2) 检查 OpenCV

Jetson Xavier NX 中已经安装了 OpenCV4.1.1 版本，可以使用命令检查 OpenCV 是否安装就绪

```
pkg-config opencv --modversion
```

如果 OpenCv 安装就绪，会显示版本号

注意：如果此条命令无法查看，可以在终端输入 **python** 进入 **python** 编程环境，然后添加输入一下两段代码极客查看 **opencv** 版本

```
import cv2
```

```
cv2.__version__
```

```
>>> cv2.__version__
'4.1.1'
```

(3) 检查 cuDNN

Jetson-nano 中已经安装好了 cuDNN，并有例子可供运行，我们运行一下例子，也正好验证上面的 CUDA

```
cd /usr/src/cudnn_samples_v8/mnistCUDNN/ #进入例子目录
```

```
sudo make #编译一下例子
```

```
./mnistCUDNN # 执行
```

如果以上无法运行可以添加权限如下方法：

```
sudo chmod a+x mnistCUDNN # 为可执行文件添加执行权限
```

如果成功，如下所示

```
nx@nx-desktop:/usr/src/cudnn_samples_v8/mnistCUDNN$ ./mnistCUDNN
Executing: mnistCUDNN
cudnnGetVersion() : 8000 , CUDNN_VERSION from cudnn.h : 8000 (8.0.0)
Host compiler version : GCC 7.5.0

There are 1 CUDA capable devices on your machine :
device 0 : sms 6 Capabilities 7.2, SmClock 1109.0 Mhz, MemSize (Mb) 7763, MemClock 1109.0 Mhz, Ecc=0, boardGroupID=0
Using device 0

Testing single precision
Loading binary file data/conv1.bin
Loading binary file data/conv1.bias.bin
Loading binary file data/conv2.bin
Loading binary file data/conv2.bias.bin
Loading binary file data/ipl1.bin
Loading binary file data/ipl1.bias.bin
Loading binary file data/ip2.bin
Loading binary file data/ip2.bias.bin
Loading image data/one_28x28.pgm
Performing forward propagation ...
Testing cudnnGetConvolutionForwardAlgorithm_v7 ...
~~~~ CUDNN_STATUS_SUCCESS for Algo 1: -1.000000 time requiring 0 memory
~~~~ CUDNN_STATUS_SUCCESS for Algo 0: -1.000000 time requiring 0 memory
~~~~ CUDNN_STATUS_SUCCESS for Algo 2: -1.000000 time requiring 0 memory
~~~~ CUDNN_STATUS_SUCCESS for Algo 7: -1.000000 time requiring 2057744 memory
```

```
Testing cudnnFindConvolutionForwardAlgorithm ...
~~~~ CUDNN_STATUS_SUCCESS for Algo 2: 0.173056 time requiring 0 memory
~~~~ CUDNN_STATUS_SUCCESS for Algo 0: 0.173120 time requiring 0 memory
~~~~ CUDNN_STATUS_SUCCESS for Algo 4: 0.181248 time requiring 2450080 memory
~~~~ CUDNN_STATUS_SUCCESS for Algo 1: 0.219232 time requiring 0 memory
~~~~ CUDNN_STATUS_SUCCESS for Algo 7: 0.393216 time requiring 1433120 memory
~~~~ CUDNN_STATUS_SUCCESS for Algo 5: 0.474112 time requiring 4656640 memory
~~~~ CUDNN_STATUS_NOT_SUPPORTED for Algo 6: -1.000000 time requiring 0 memory
~~~~ CUDNN_STATUS_NOT_SUPPORTED for Algo 3: -1.000000 time requiring 0 memory
Resulting weights from Softmax:
0.0000000 0.0000000 0.0000000 1.0000000 0.0000000 0.0000714 0.0000000 0.0000000 0.0000000 0.0000000
Loading image data/five_28x28.pgm
Performing forward propagation ...
Resulting weights from Softmax:
0.0000000 0.0000008 0.0000000 0.0000002 0.0000000 1.0000000 0.0000154 0.0000000 0.0000012 0.0000006

Result of classification: 1 3 5

Test passed!
nx@nx-desktop:/usr/src/cudnn_samples_v8/mnistCUDNN$
```