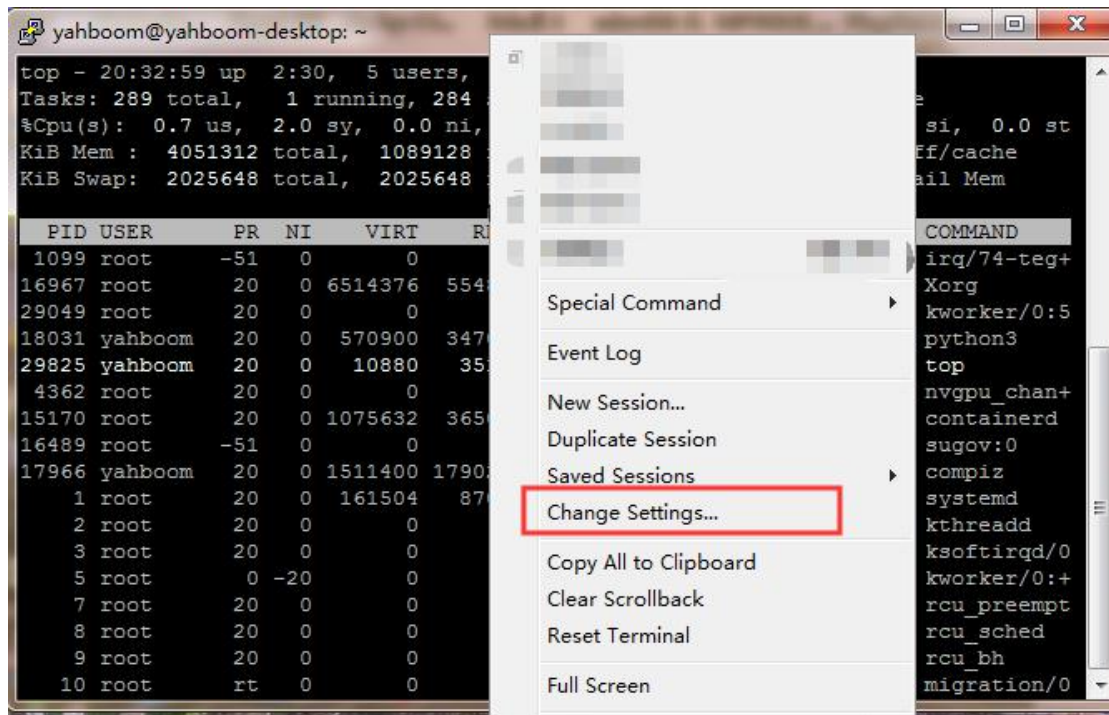


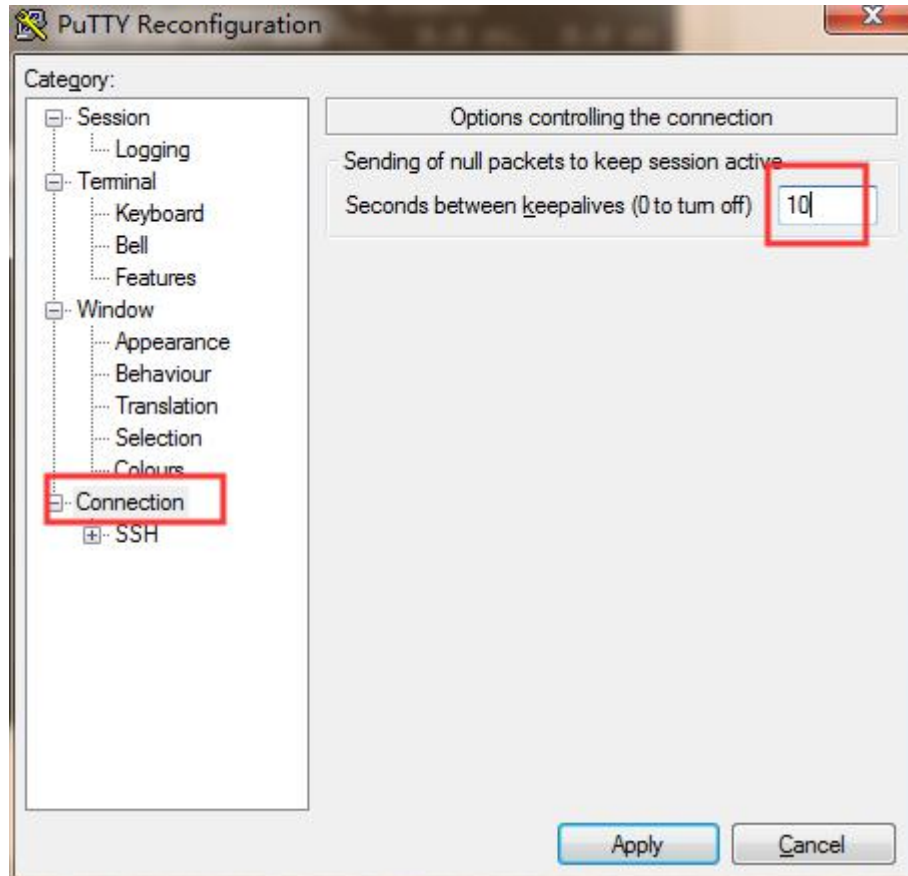
## 1.Preparation tutorial

### 1.Remote Log in

You can select PuTTY, SSH, Xshell and other tools to log in remotely. The following is an example of the PuTTY tool.

Note:If you find PuTTY will exit or offline, please refer to method as shown below.





## 2.About the update source

In general, the source should be updated after the system is installed, but because the Jetson Nano uses the AArch64-based Ubuntu 18.04.2 LTS system, it is different from the AMD-based Ubuntu system, and I have not found a perfect domestic source, **so i don't recommend change the source.**

We do not change source here, or update with the default source of Jetson Nano. The update process is very long. You can execute the commands and do other things. **The following two operations are recommended to be executed before the AI project,** otherwise some libraries will not find the installation address, resulting in frequent errors.

We need to input: **sudo apt-get update**

```

nano@nano-desktop: ~
]
获取:20 http://ports.ubuntu.com/ubuntu-ports bionic-security InRelease [88.7 kB]
获取:21 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 Packages [975 kB]
获取:22 http://ports.ubuntu.com/ubuntu-ports bionic/main Translation-en [516 kB]
获取:23 http://ports.ubuntu.com/ubuntu-ports bionic/main Translation-zh_CN [67.7
kB]
获取:24 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 DEP-11 Metadata [
472 kB]
获取:25 http://ports.ubuntu.com/ubuntu-ports bionic/main DEP-11 48x48 Icons [118
kB]
获取:26 http://ports.ubuntu.com/ubuntu-ports bionic/main DEP-11 64x64 Icons [245
kB]
获取:27 http://ports.ubuntu.com/ubuntu-ports bionic/restricted arm64 Packages [6
64 B]
获取:28 http://ports.ubuntu.com/ubuntu-ports bionic/restricted Translation-en [3
,584 B]
获取:29 http://ports.ubuntu.com/ubuntu-ports bionic/restricted Translation-zh_CN
[1,188 B]
获取:30 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 Packages [8,3
16 kB]
获取:31 http://ports.ubuntu.com/ubuntu-ports bionic/universe Translation-zh_CN [
174 kB]
获取:32 http://ports.ubuntu.com/ubuntu-ports bionic/universe Translation-en [4,9
41 kB]
获取:33 http://ports.ubuntu.com/ubuntu-ports bionic/universe arm64 DEP-11 Metada
ta [3,243 kB]
获取:34 http://ports.ubuntu.com/ubuntu-ports bionic/universe DEP-11 48x48 Icons
[2,151 kB]
获取:35 http://ports.ubuntu.com/ubuntu-ports bionic/universe DEP-11 64x64 Icons
[8,420 kB]
80% [35 icons-64x64 6,698 kB/8,420 kB 80%] 127 kB/s 50秒

```

```

nano@nano-desktop: ~
获取:67 http://ports.ubuntu.com/ubuntu-ports bionic-security/main arm64 DEP-11 M
etadata [14.9 kB]
获取:68 http://ports.ubuntu.com/ubuntu-ports bionic-security/main DEP-11 48x48 I
cons [10.4 kB]
获取:69 http://ports.ubuntu.com/ubuntu-ports bionic-security/main DEP-11 64x64 I
cons [20.9 kB]
获取:70 http://ports.ubuntu.com/ubuntu-ports bionic-security/restricted arm64 Pa
ckages [668 B]
获取:71 http://ports.ubuntu.com/ubuntu-ports bionic-security/restricted Translat
ion-en [2,192 B]
获取:72 http://ports.ubuntu.com/ubuntu-ports bionic-security/universe arm64 Pack
ages [245 kB]
获取:73 http://ports.ubuntu.com/ubuntu-ports bionic-security/universe Translatio
n-en [139 kB]
获取:74 http://ports.ubuntu.com/ubuntu-ports bionic-security/universe arm64 DEP-
11 Metadata [35.8 kB]
获取:75 http://ports.ubuntu.com/ubuntu-ports bionic-security/universe DEP-11 48x
48 Icons [16.4 kB]
获取:76 http://ports.ubuntu.com/ubuntu-ports bionic-security/universe DEP-11 64x
64 Icons [97.3 kB]
获取:77 http://ports.ubuntu.com/ubuntu-ports bionic-security/multiverse arm64 Pa
ckages [1,696 B]
获取:78 http://ports.ubuntu.com/ubuntu-ports bionic-security/multiverse Translat
ion-en [2,060 B]
获取:79 http://ports.ubuntu.com/ubuntu-ports bionic-security/multiverse DEP-11 4
8x48 Icons [29 B]
获取:80 http://ports.ubuntu.com/ubuntu-ports bionic-security/multiverse DEP-11 6
4x64 Icons [2,638 B]
已下载 34.8 MB, 耗时 5分 49秒 (99.5 kB/s)
正在读取软件包列表... 完成
nano@nano-desktop:~$

```



We need to input: **sudo apt-get full-upgrade**

```
nano@nano-desktop: ~
libreoffice-style-tango libreoffice-writer libseccomp2 libsmclient
libsnappy-glib1 libssl1.0.0 libstdc++-7-dev libstdc++6 libswresample2
libswscale4 libsystemd0 libtiff5 libtsan0 libubsan0 libudev1 libunistring2
libunity-core-6.0-9 libupower-glib3 libwavpack1 libwayland-egl1-mesa
libwbclient0 libwebkit2gtk-4.0-37 libx11-6 libx11-data libx11-dev libx11-doc
libx11-xcb-dev libx11-xcb1 libxcb-composite0 libxcb-damage0 libxcb-dri2-0
libxcb-dri2-0-dev libxcb-dri3-0 libxcb-dri3-dev libxcb-glx0 libxcb-glx0-dev
libxcb-present-dev libxcb-present0 libxcb-randr0 libxcb-randr0-dev
libxcb-render0 libxcb-render0-dev libxcb-res0 libxcb-shape0
libxcb-shape0-dev libxcb-shm0 libxcb-sync-dev libxcb-sync1 libxcb-xfixes0
libxcb-xfixes0-dev libxcb-xinerama0 libxcb-xkb1 libxcb-xv0 libxcb1
libxcb1-dev libxslt1.1 linux-firmware linux-libc-dev login mesa-common-dev
mesa-va-drivers mesa-vdpau-drivers mutter mutter-common nautilus
nautilus-data network-manager network-manager-config-connectivity-ubuntu
nfs-common ntfs-3g openssh-client openssh-server openssh-sftp-server
packagekit packagekit-tools parted passwd plymouth plymouth-label
plymouth-theme-ubuntu-logo policykit-1 poppler-utils pulseaudio
pulseaudio-utils python-gi python3-apport python3-distro-info
python3-distupgrade python3-gi python3-gi-cairo python3-httpplib2
python3-problem-report python3-uno python3-update-manager
qt5-gtk-platformtheme resolvconf samba-ls snapd systemd systemd-sysv
thunderbird thunderbird-gnome-support tzdata ubuntu-desktop
ubuntu-drivers-common ubuntu-release-upgrader-core
ubuntu-release-upgrader-gtk ubuntu-software udev unattended-upgrades unity
unity-schemas unity-services uno-libs3 update-manager update-manager-core
upower ure wget wpasupplicant
```

We need to input **Y** to confirm the update during the process.

The second process time may take about 2 hours depending on the network situation.

### 3. Check installed system components

Jetson-nano's OS image comes with JetPack, cuda, cudnn, opencv, etc., and there are examples.

The installation paths for these examples are as follows:

TensorRT: </usr/src/tensorrt/samples/>

CUDA: </usr/local/cuda/samples/>

cuDNN: [/usr/src/cudnn\\_samples\\_v7/](/usr/src/cudnn_samples_v7/)

Multimedia API: [/usr/src/tegra\\_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) Check CUDA

The CUDA10.0 version is already installed in Jetson-nano, but if you run `nvcc -V` at this time it will not succeed.

You need to write the path of CUDA to the environment variable. The OS comes with the Vim tool, so run the following command to edit the environment variables.

**sudo vim ~/.bashrc**

Add at the end:

```
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
```

Then, we need to save and quit.

Next, we need to input the following command to make the configuration take effect.

```
source ~/.bashrc
```

we need to input the following command:

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

As shown blew.

```
nano@nano-desktop:~$ nvcc -V  
nvcc: NVIDIA (R) Cuda compiler driver  
Copyright (c) 2005-2018 NVIDIA Corporation  
Built on Sun_Sep_30_21:09:22_CDT_2018  
Cuda compilation tools, release 10.0, V10.0.166
```

## (2) Check OpenCV

The OpenCV 3.3 version is already installed in Jetson-nano,we can check if OpenCV can work normally.

we can input the following command:

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

If OpenCV is already installed, system will appear number of version.(For example: 3.3.1)

```
nano@nano-desktop:~$ pkg-config opencv --modversion  
3.3.1
```

## (3) Check cuDNN

cuDNN has been installed in Jetson-nano and there are examples to run.

```
cd /usr/src/cudnn_samples_v7/mnistCUDNN #Go to the example directory  
sudo make #Compile the example  
sudo chmod a+x mnistCUDNN # Add execute permission to the executable  
./mnistCUDNN #Execut
```

If it runs successfully, the system will display the prompt shown below.

```

nano@nano-desktop:/usr/src/cudnn_samples_v7/mnistCUDNN$ ./mnistCUDNN
cudnnGetVersion() : 7301 , CUDNN_VERSION from cudnn.h : 7301 (7.3.1)
Host compiler version : GCC 7.4.0
There are 1 CUDA capable devices on your machine :
device 0 : sms 1 Capabilities 5.3, SmClock 921.6 Mhz, MemSize (Mb) 3956, MemClock 12.8 Mhz, Ecc=0, boardGroupID=0
Using device 0

Testing single precision
Loading image data/one_28x28.pgm
Performing forward propagation ...
Testing cudnnGetConvolutionForwardAlgorithm ...
Fastest algorithm is Algo 1
Testing cudnnFindConvolutionForwardAlgorithm ...
^^^^ CUDNN_STATUS_SUCCESS for Algo 1: 0.409948 time requiring 3464 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 0: 0.420104 time requiring 0 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 4: 4.925937 time requiring 207360 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 2: 14.162396 time requiring 57600 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 5: 24.979948 time requiring 203008 memory
Resulting weights from Softmax:
0.0000000 0.9999399 0.0000000 0.0000000 0.0000561 0.0000000 0.0000012 0.0000017

```

```

Resulting weights from Softmax:
0.0000001 1.0000000 0.0000001 0.0000000 0.0000563 0.0000001 0.0000012 0.0000017
0.0000010 0.0000001
Loading image data/three_28x28.pgm
Performing forward propagation ...
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!
nano@nano-desktop:/usr/src/cudnn_samples_v7/mnistCUDNN$ █

```