

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.

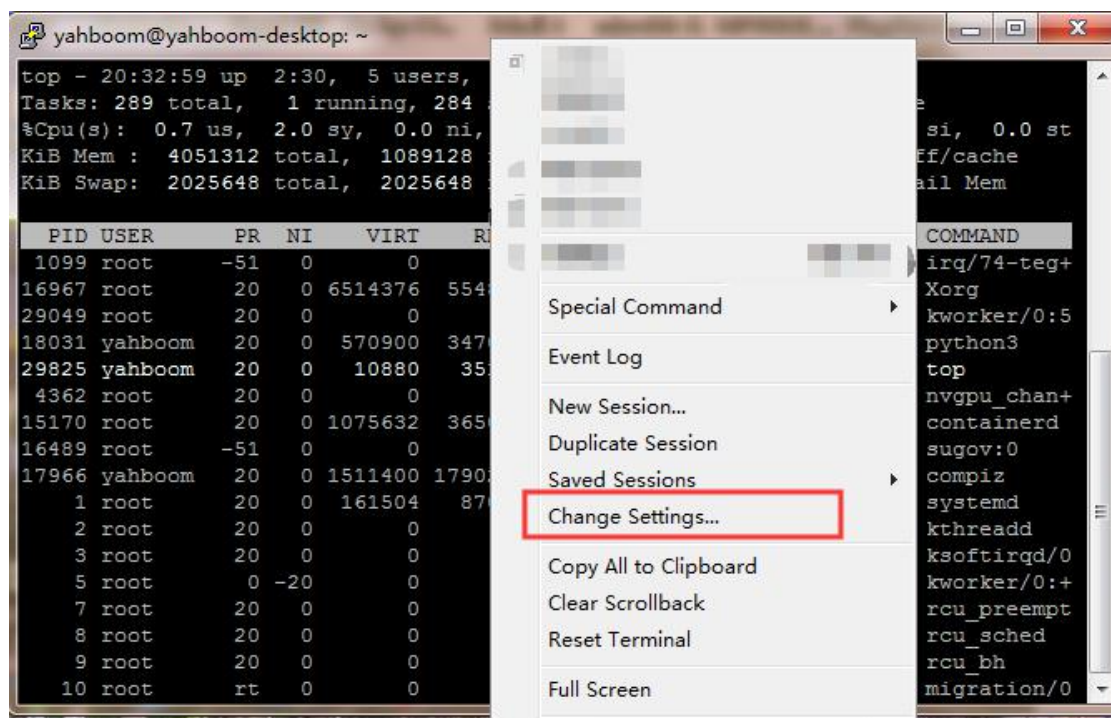
If you find that the computer can not log into Jetson NX, you can try to ping IP address of each other.

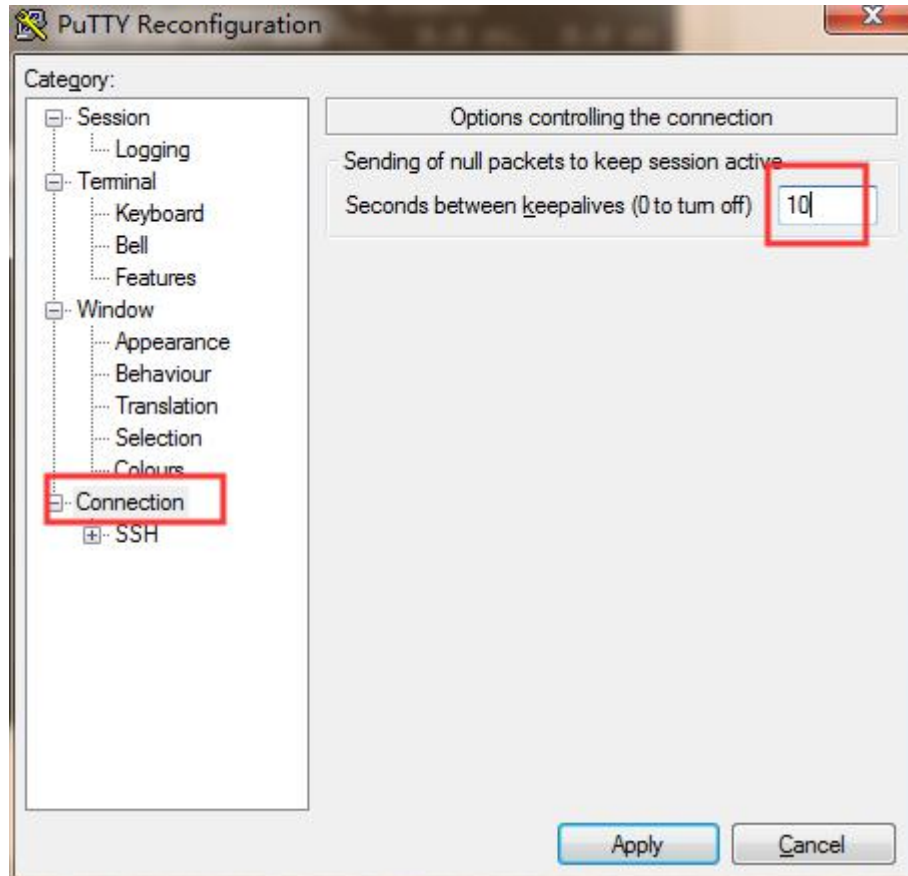
View the IP of NX. Input this command `ifconfig`.

View the IP address of your computer.

Input `cmd` in command run terminal, then input `ipconfig`.

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`

```

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]

```

We need to input: **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

```

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.

After completion, the following figure.

```

正在设置 python3-distupgrade (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. 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/
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 the following three lines 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.

Note: If adding the above three sentences, sytem still displays “nvcc not found”.
 First of all, we need to check if there is nvcc in the bin/cuda directory:

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

If nvcc exists, reduce the above sentences to the following two sentences.

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

Add the following two sentences.

```
export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_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:

nvcc -V

As shown below.

```
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) Check OpenCV

The OpenCV4.1.1 version is already installed in Jetson Xavier NX, 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.

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

Note: If this command cannot be viewed, you can enter python in the terminal to enter the python programming environment, and then add two pieces of code to view the opencv version

```
import cv2
cv2.__version__
```

(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 #Execute
```

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


```

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$ █

```