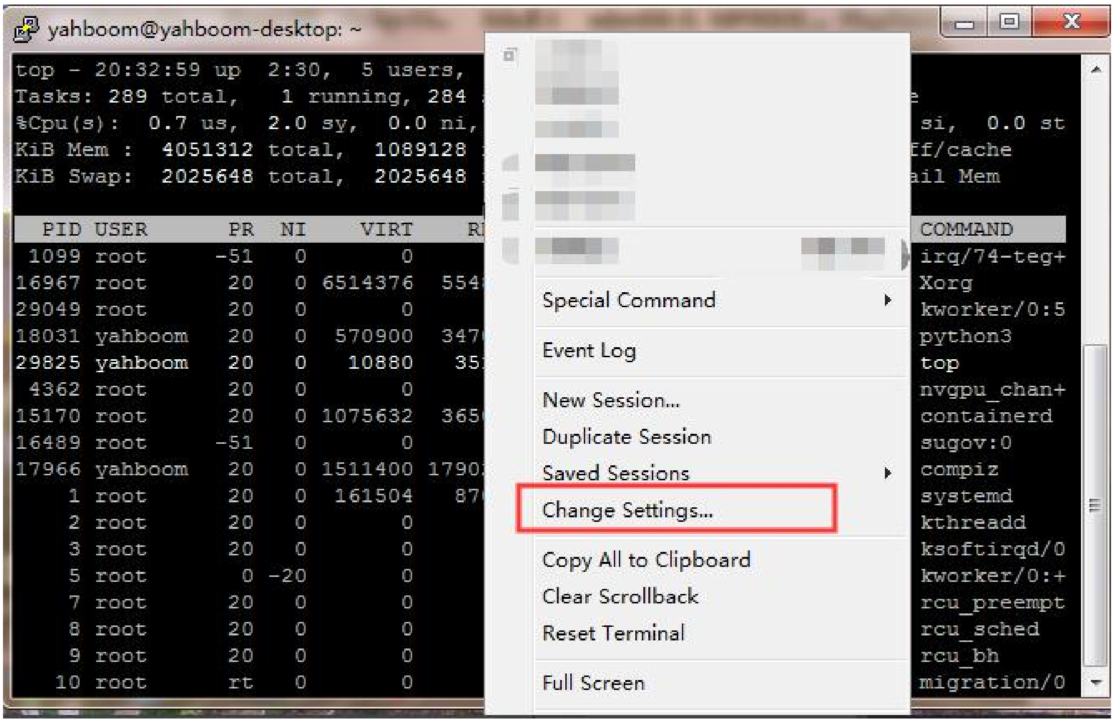
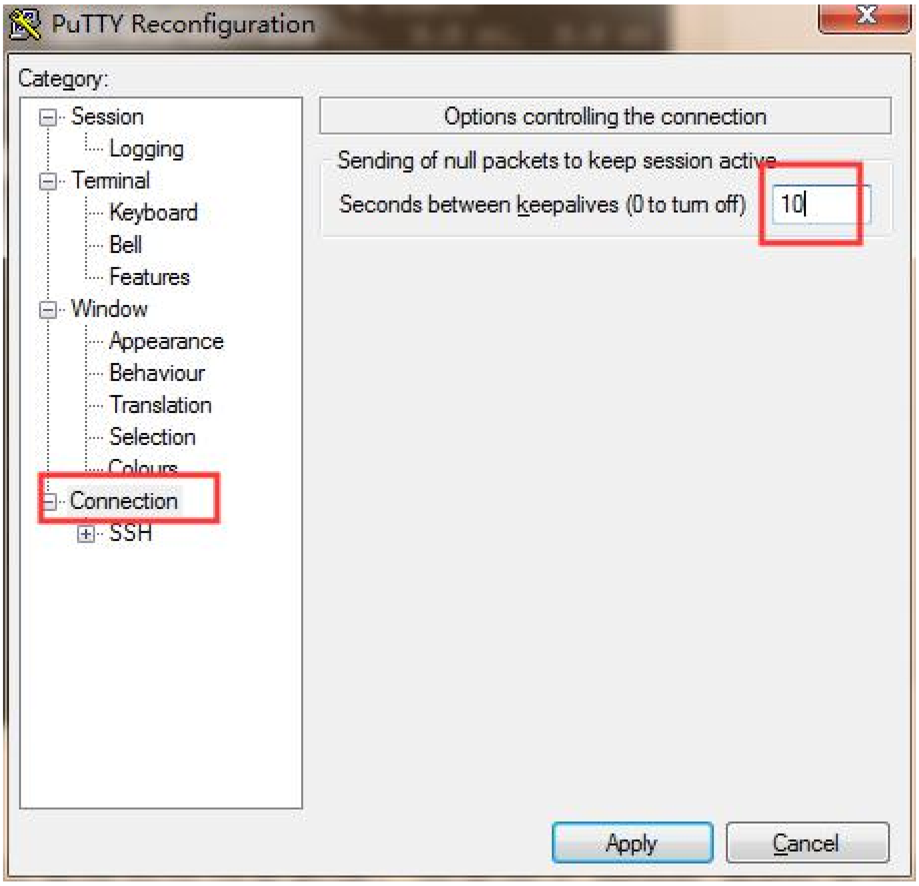
1.Remote Log in

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.



```
top - 20:32:59 up  2:30,  5 users,                                 e
Tasks: 289 total,   1 running, 284                                 si,  0.0 st
%Cpu(s):  0.7 us,  2.0 sy,  0.0 ni,                               ff/cache
KiB Mem :  4051312 total,  1089128                                ail Mem
KiB Swap:  2025648 total,  2025648

  PID USER      PR  NI    VIRT    RE                               COMMAND
 1099 root     -51   0       0                                     irq/74-teg+
16967 root      20   0 6514376  554                                Xorg
29049 root      20   0       0                                     kworker/0:5
18031 yahboom   20   0  570900  347                                python3
29825 yahboom   20   0   10880   35                                top
 4362 root      20   0       0                                     nvgpu_chan+
15170 root      20   0 1075632  365                                containerd
16489 root     -51   0       0                                     sugov:0
17966 yahboom   20   0 1511400 1790                                compiz
    1 root      20   0  161504   87                                systemd
    2 root      20   0       0                                     kthreadd
    3 root      20   0       0                                     ksoftirqd/0
    5 root       0 -20       0                                     kworker/0:+
    7 root      20   0       0                                     rcu_preempt
    8 root      20   0       0                                     rcu_sched
    9 root      20   0       0                                     rcu_bh
   10 root      rt   0       0                                     migration/0
```



Sending of null packets to keep session active

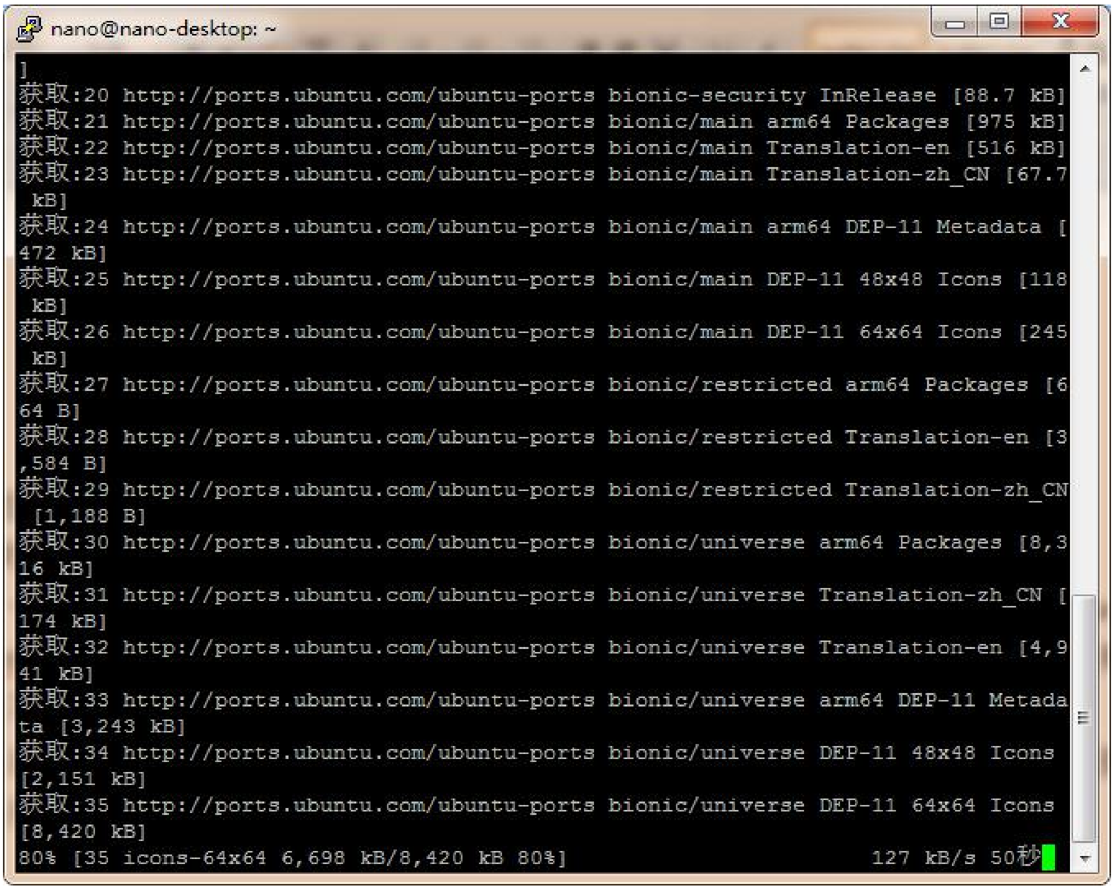
Seconds between keepalives (0 to turn off) 10

2.About the update source

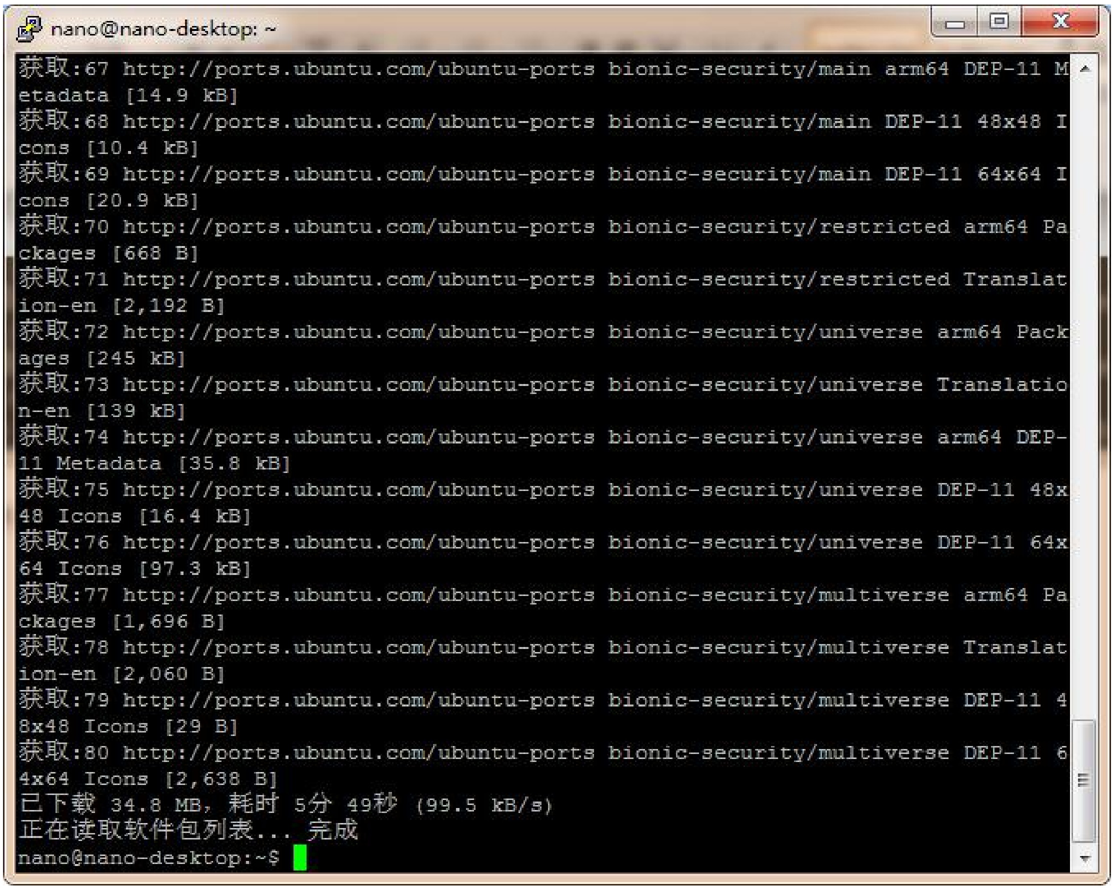
Input following command to update.

sudo apt-get update

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

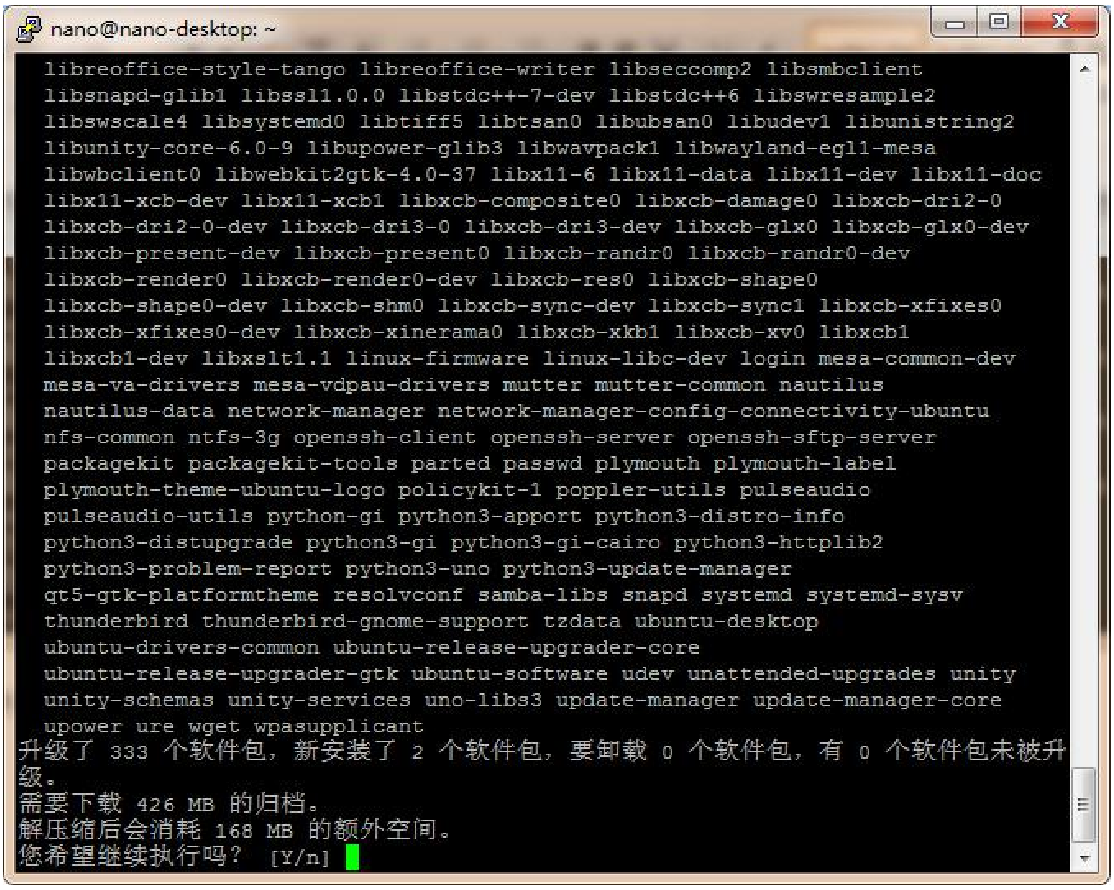


```
]
获取:20 http://ports.ubuntu.com/ubuntu-ports bionic-security InRelease [88.7 kB]
获取:21 http://ports.ubuntu.com/ubuntu-ports bionic/main arm64 Packages [975 kB]
获取:22 http://ports.ubuntu.com/ubuntu-ports bionic/main Translation-en [516 kB]
获取:23 http://ports.ubuntu.com/ubuntu-ports bionic/main Translation-zh_CN [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
```



```
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:~$ 
```

sudo apt-get full-upgrade



nano@nano-desktop: ~

```
  libreoffice-style-tango libreoffice-writer libseccomp2 libsmbclient
  libsnapd-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-httplib2
  python3-problem-report python3-uno python3-update-manager
  qt5-gtk-platformtheme resolvconf samba-libs 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
升级了 333 个软件包，新安装了 2 个软件包，要卸载 0 个软件包，有 0 个软件包未被升
级。
需要下载 426 MB 的归档。
解压缩后会消耗 168 MB 的额外空间。
您希望继续执行吗？ [Y/n]
```

This process takes about 2 hours, please be patient.

3. Check installed system components

Jetson-nano 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-10.2/samples/
cuDNN: /usr/src/cudnn_samples_v8/
VisionWorks:
/usr/share/visionworks/sources/samples/
/usr/share/visionworks-tracking/sources/samples/
/usr/share/visionworks-sfm/sources/samples/
OpenCV: /usr/share/opencv4/samples/

(1) Enter the following command to install JTOP, you can check the CPU usage.

```
sudo apt-get update
sudo apt-get full-upgrade
sudo apt install curl
```

sudo apt install nano

$ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py #Download and install script

$ sudo python3 get-pip.py #run install script

sudo pip3 install jetson-stats

jtop

(2) 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, we can run the following command to edit the environment variables.

First, we can check if there is nvcc in the bin directory of cuda.

```
ls /usr/local/cuda/bin
```

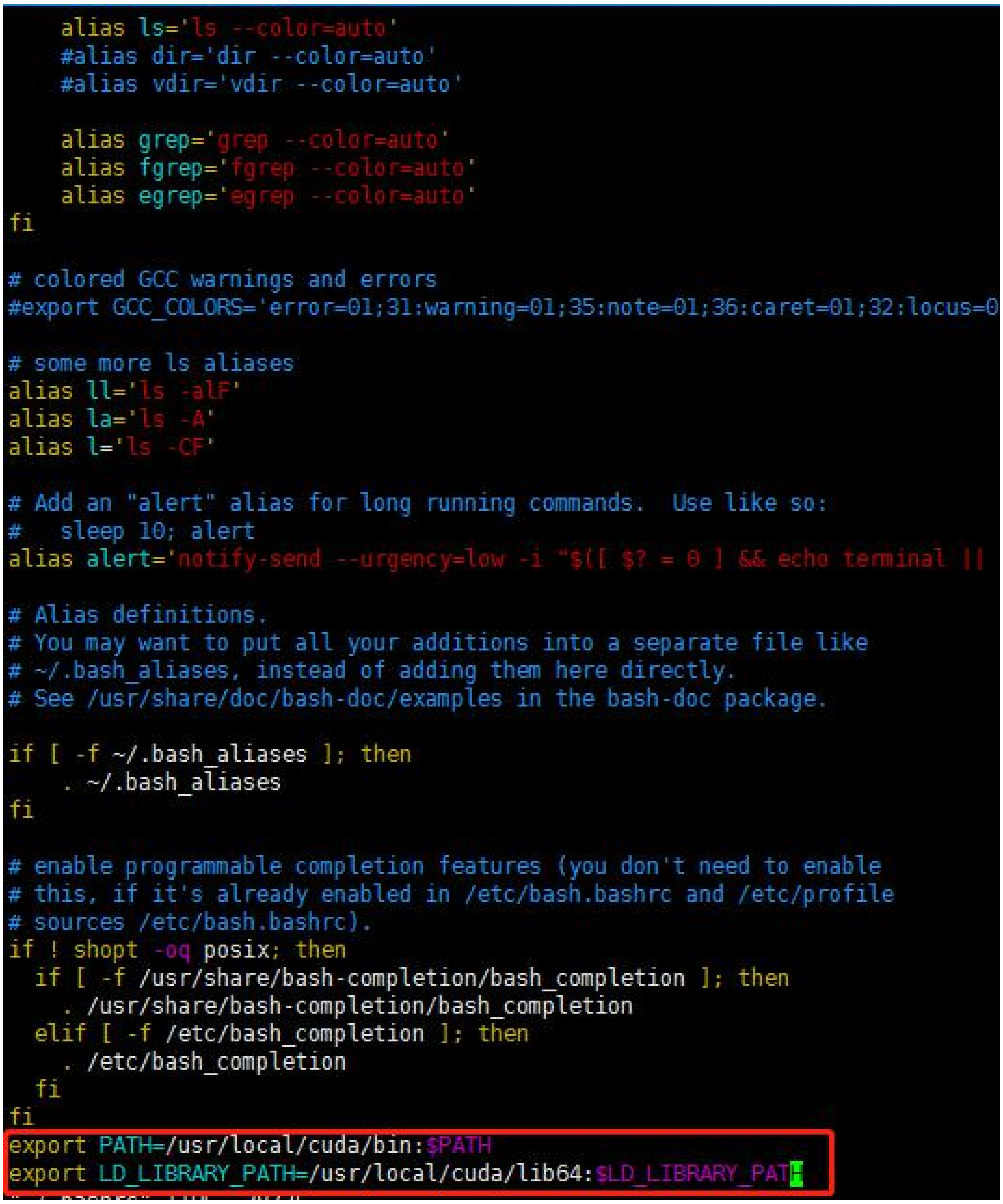
If it exists, enter the following command to enter the configuration file.

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

Add the following two lines at the end.

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

As shown below.



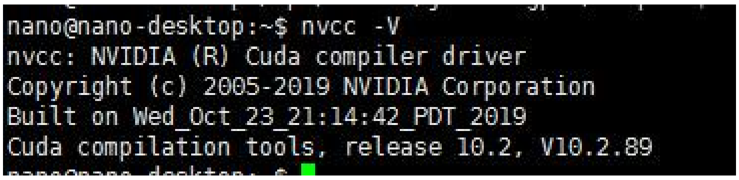
```
    alias ls='ls --color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=0

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal ||

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
```

Input following command.

source ~/.bashrc

nvcc -V

As shown below.

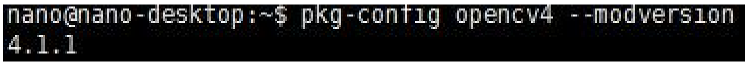


```
nano@nano-desktop:~$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Wed_Oct_23_21:14:42_PDT_2019
Cuda compilation tools, release 10.2, V10.2.89
```

(3) Check OpenCV

Input following command to check OpenCV version. As shown below.

```
pkg-config opencv4 --modversion
```



```
nano@nano-desktop:~$ pkg-config opencv4 --modversion
4.1.1
```

(4) Check cuDNN

CuDNN has been installed in Jetson-nano, and there are examples can be run. We run the example, and it will just verify the CUDA at the same time.

```
cd /usr/src/cudnn_samples_v8/mnistCUDNN/
```

#Enter the example directory

```
sudo make
```
#compile the example

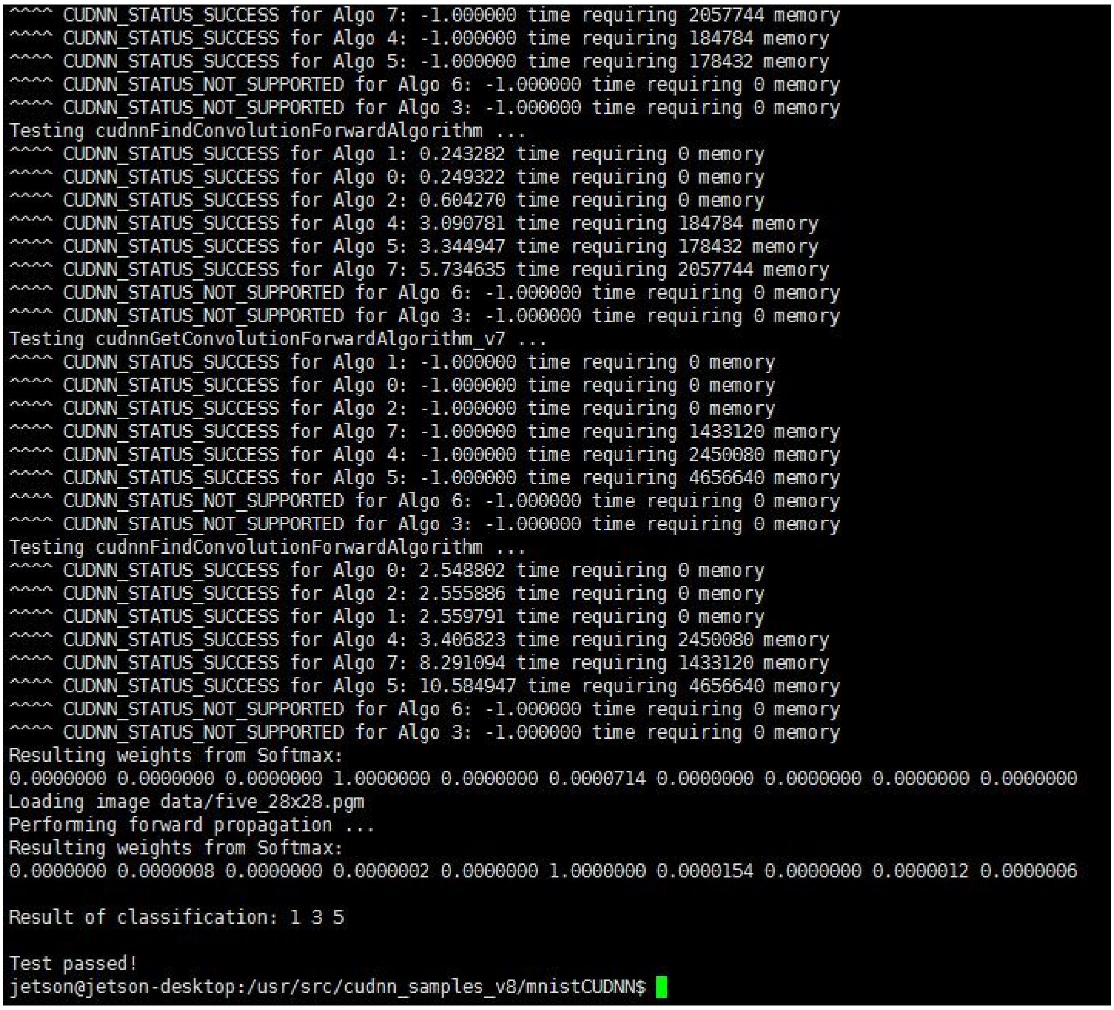
```
./mnistCUDNN
```
#Execute

If the above does not work, you can add permissions as follows:

```
sudo chmod a+x mnistCUDNN
```
Add execution permissions to executable files

If successful, as shown below

```
/usr/src/cudnn_samples_v8/mnistCUDNN$
```



```
^^^^ CUDNN_STATUS_SUCCESS for Algo 7: -1.000000 time requiring 2057744 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 4: -1.000000 time requiring 184784 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 5: -1.000000 time requiring 178432 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
Testing cudnnFindConvolutionForwardAlgorithm ...
^^^^ CUDNN_STATUS_SUCCESS for Algo 1: 0.243282 time requiring 0 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 0: 0.249322 time requiring 0 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 2: 0.604270 time requiring 0 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 4: 3.090781 time requiring 184784 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 5: 3.344947 time requiring 178432 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 7: 5.734635 time requiring 2057744 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
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 1433120 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 4: -1.000000 time requiring 2450080 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 5: -1.000000 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
Testing cudnnFindConvolutionForwardAlgorithm ...
^^^^ CUDNN_STATUS_SUCCESS for Algo 0: 2.548802 time requiring 0 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 2: 2.555886 time requiring 0 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 1: 2.559791 time requiring 0 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 4: 3.406823 time requiring 2450080 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 7: 8.291094 time requiring 1433120 memory
^^^^ CUDNN_STATUS_SUCCESS for Algo 5: 10.584947 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!
jetson@jetson-desktop:/usr/src/cudnn_samples_v8/mnistCUDNN$ 
```