Set up Environment for Linux

Tony Chen, Leo Wang

September 27, 2021

Abstract

This document gives an easy guideline on how to set up Linex. We will go through preparing image file, setting the basic information in Linex, and installing the mandatory package such as opencv2 and vnc for further bugging.

1 Hardware List

- Well Equipped PC
- USB or Disk
- Screen
- KeyBoard
- Mouse
- Mini HDMI to HDMI cable (connecting from PC to screen)
- power supply

2 Software List

- Linex Image (https://old-releases.ubuntu.com/releases/18.04.5/ubuntu-18.04-desktop-amd64.iso)
- Win32 Imnager
- Git Bash(https://gitforwindows.org/)(Optional)
- SD Card Formatter (https://www.sdcard.org/downloads/formatter/sd-memory-card-formatter-for-windows-download/)

3 Easy GuideLine

- If (SD Card or USB) is not formatted, please go through the process before installing image file
- Check SD Card Status
- Download 'Linex Image'
- Click on "OS system" and choose Full Version
- Click on "Disk" and choose proper (SD card or USB)
- Click on "Write" and wait for image to complete (Around 30 minutes depending on the download speed and write speed)
- Takeout (SD card or USB) and put into computer
- Turn on the power and forward the to the (screen)
- Press ESC or F2 or other key go to BIOS mode.
- Choose Boot or relevant button and change the boot disk to the USB and Reboot
- After the Reboot, it might have 4 options. Choose install Ubunte option.
- PRESS E BEFORE INSTALLING, CHANGE — -¿ nomodeset (This makes the computer to ignore GPU-Card.)
- Do not choose the 3rd party driver install in the Ubuntu.
- You will see the ubuntu Desktop chick on continue for all the setting.
- Becareful if you still need the disk information, do not choose format all disk.
- Enter other information and click on continue. The machine will reboot automatically.
- You will see ubuntu Desktop.

4 Walk Through to all the installation

4.1 Format SD Card

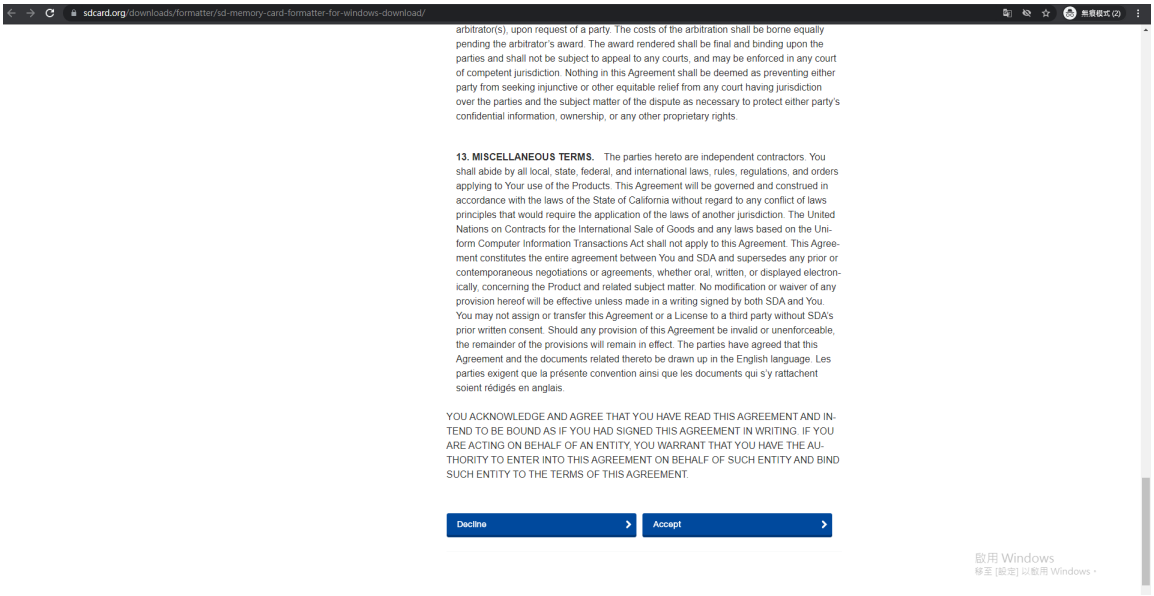

Figure 1: Download SD Card Formatter from the website

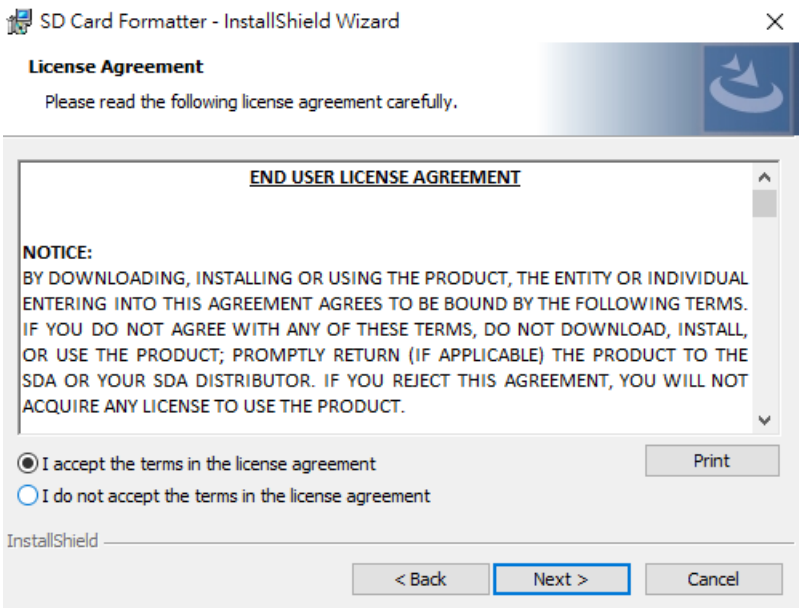

Figure 2: Click Accept and Click Next for all options

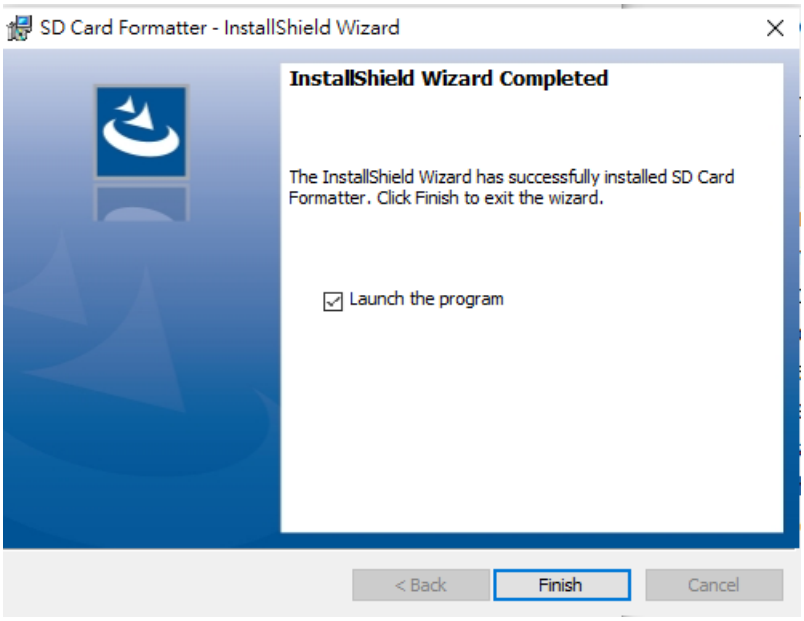

Figure 3: Click Finish and Launch the file

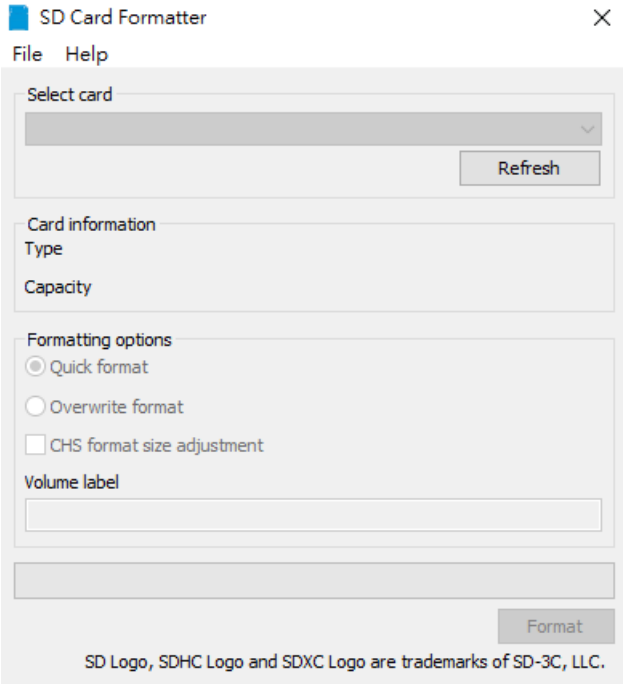

Figure 4: Select the card you would like to Format
Select Quick format
Click on Finish
SD card will be formatted and could be used for installing new image file

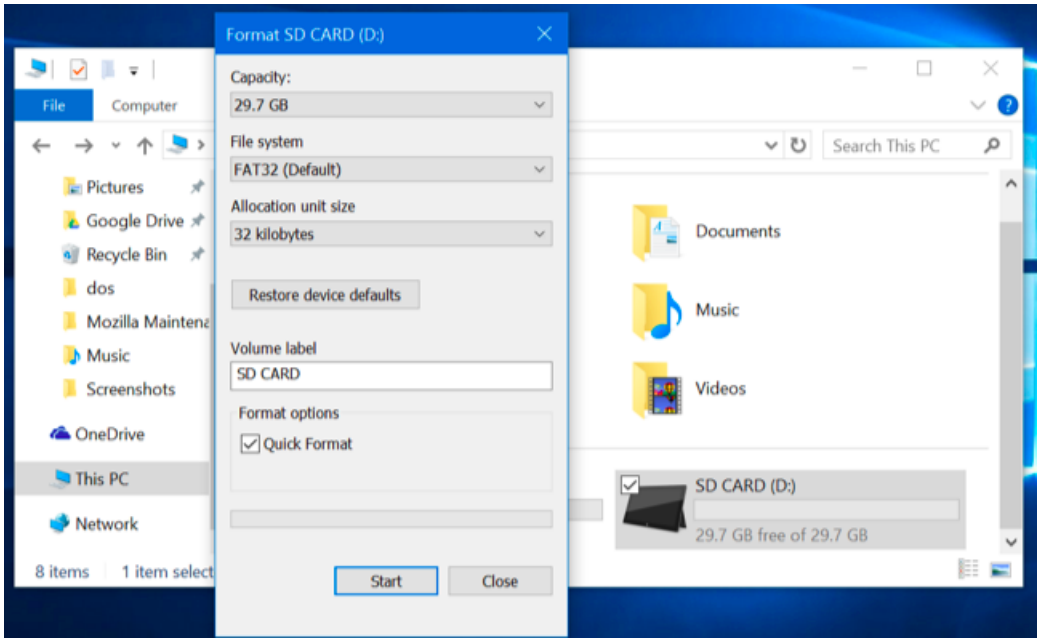

Figure 5: Check whether SD card is a disk in the computer

4.2 Download Linux Image

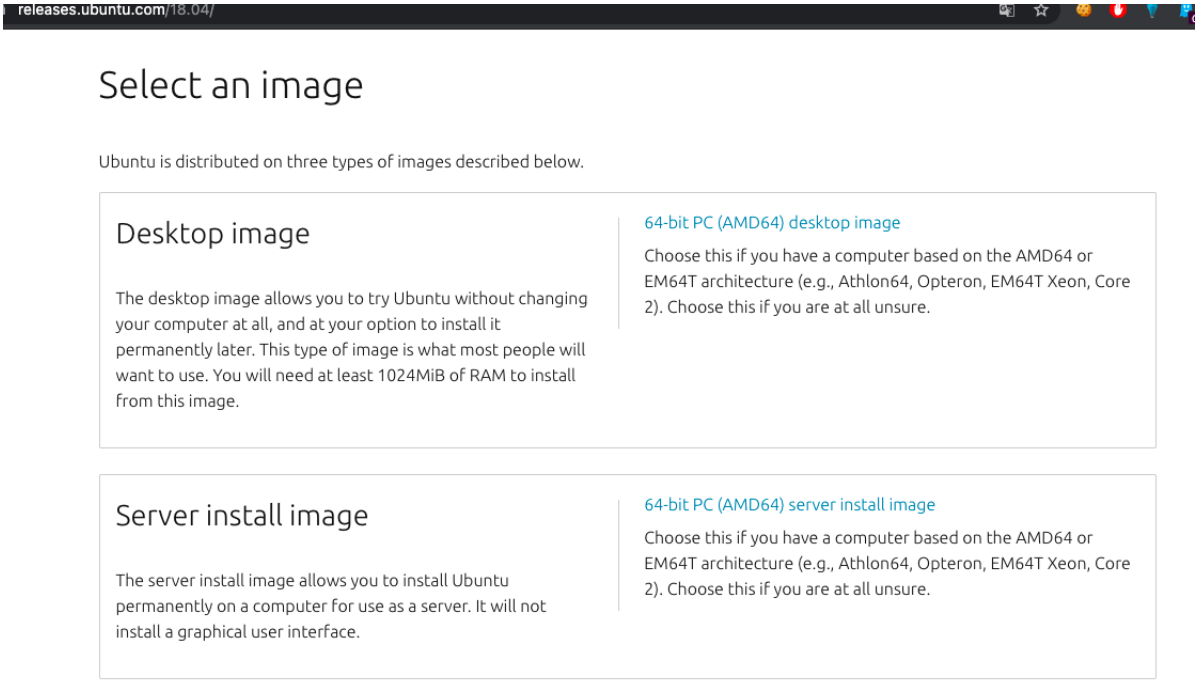

Figure 6: Click on the link provided in the quick guideline, choose the proper os system you are using

4.3 Open Linex

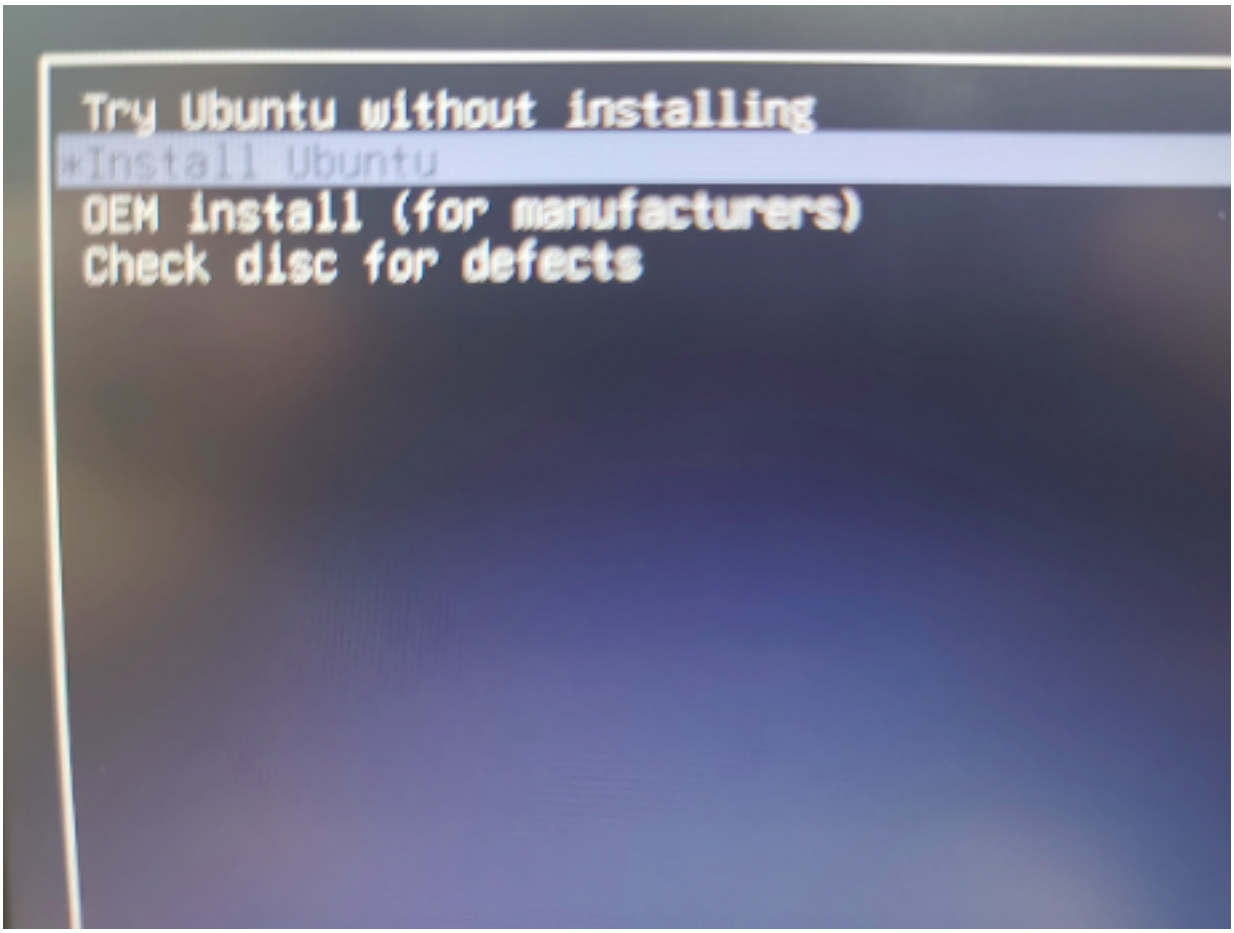

Figure 7: Click on Install Ubuntu. If there is graphic card issue, please change PRESS E BEFORE INSTALLING, CHANGE — to nomodeset (This makes the computer to ignore GPU-Card.

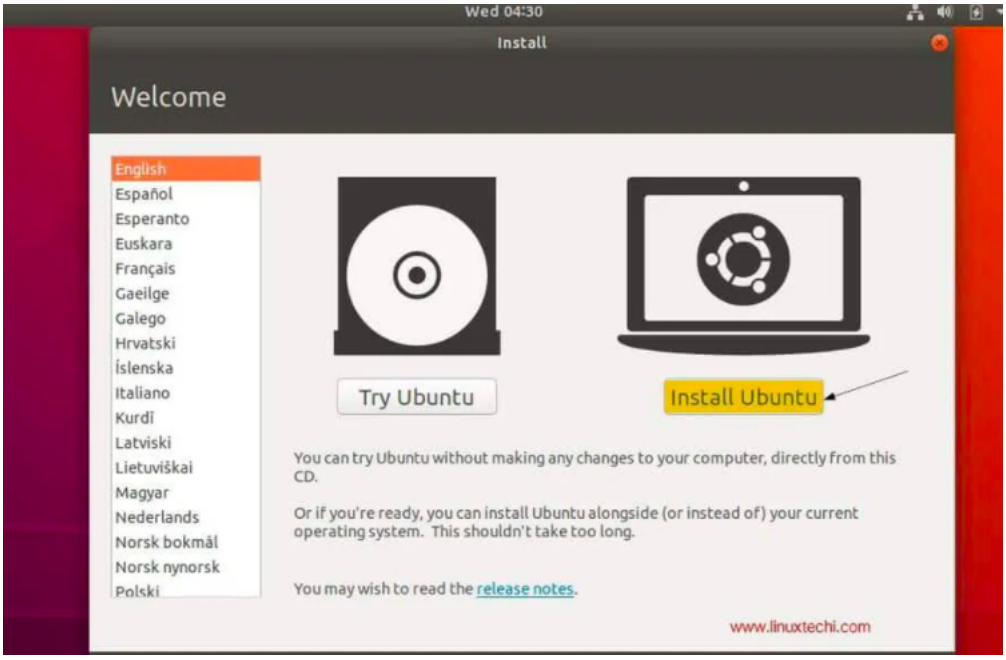

Figure 8: If you finish installation, it will show the following GUI, choose install Ubuntu)

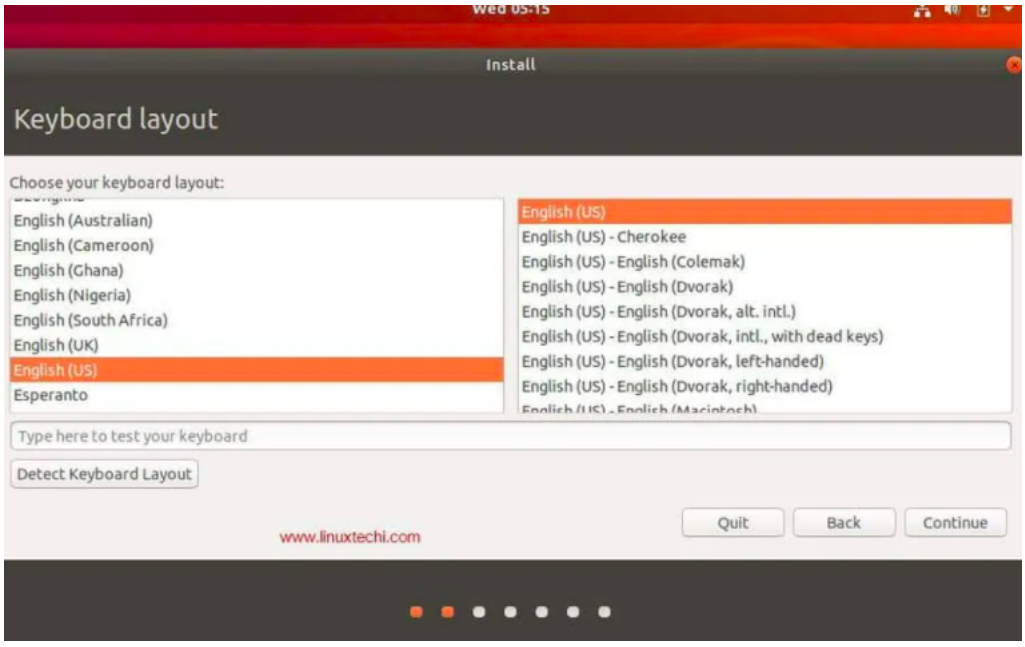

Figure 9: Choose on keyboard layout

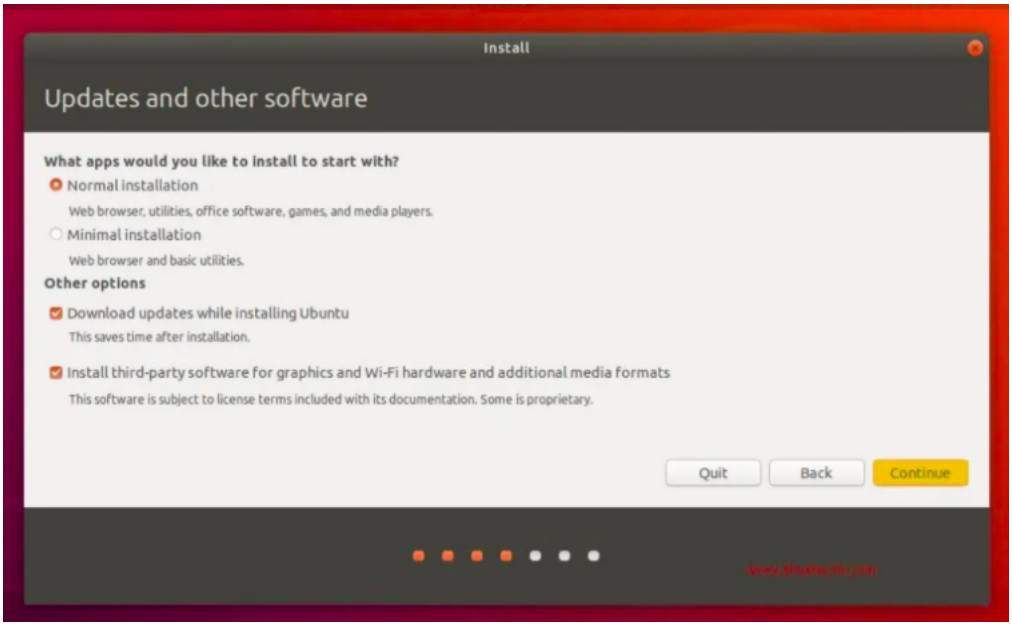

Figure 10: Choose normal installation, if there is a question on graphic card please unclick install third-party for graphic card.

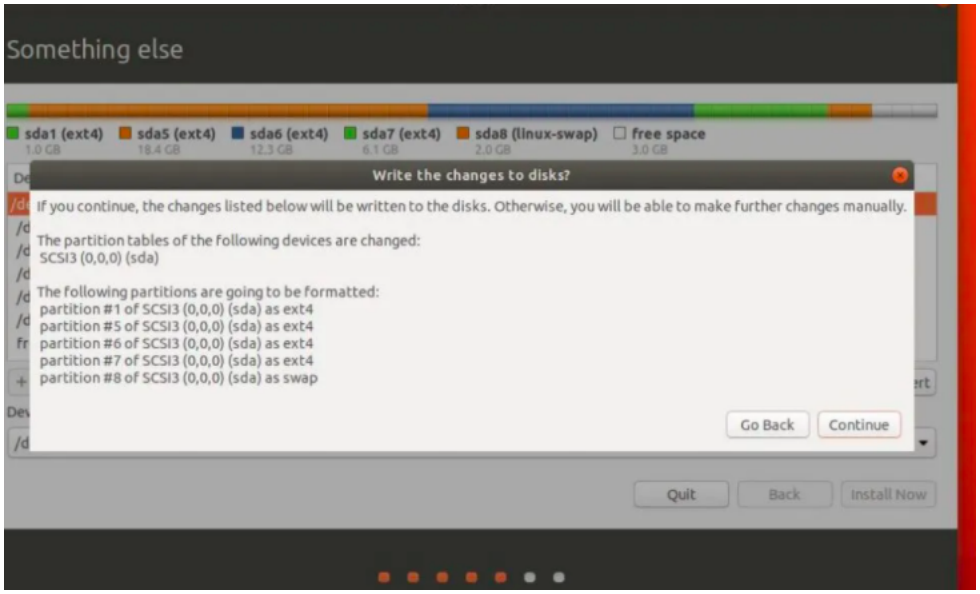

Figure 11: Press Continue

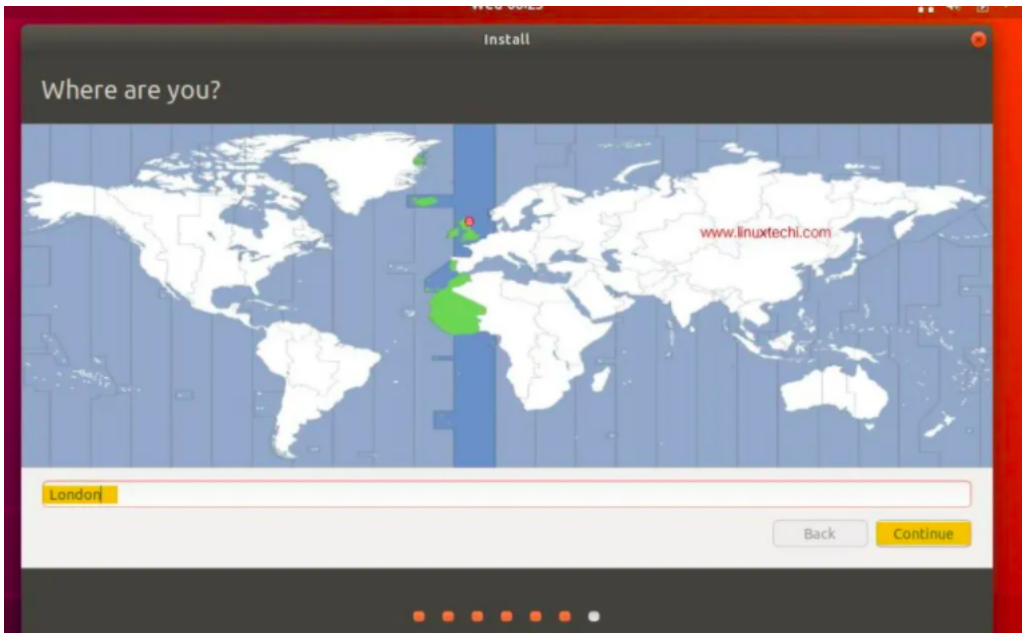

Figure 12: Choose Location

5 Mandatory Software Installation

5.1 Nvidia driver install

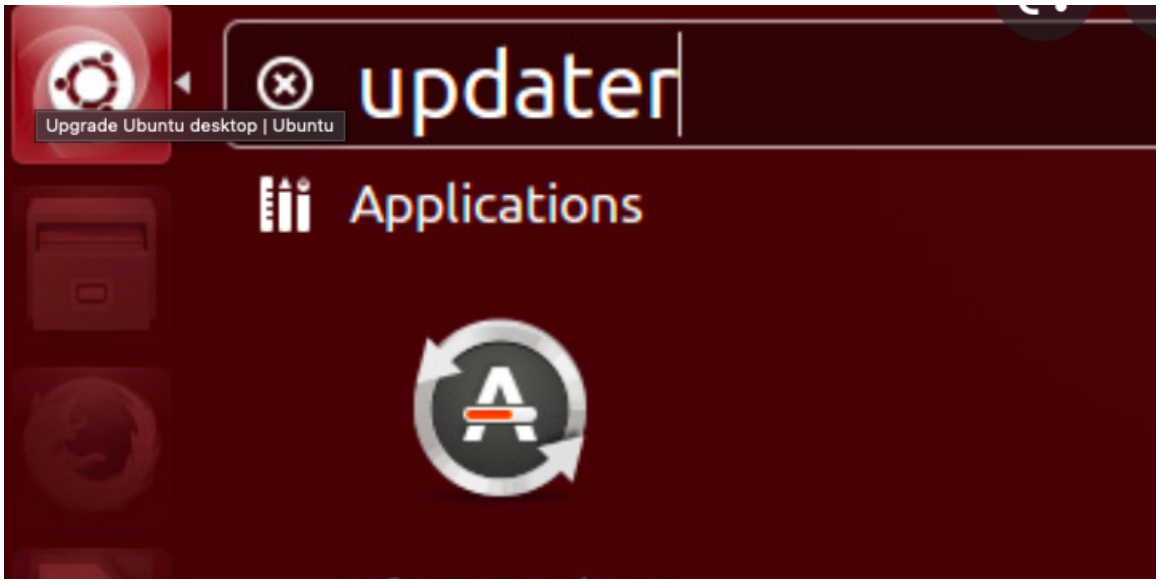

Figure 13: click the left top which enable searching, type in software update.

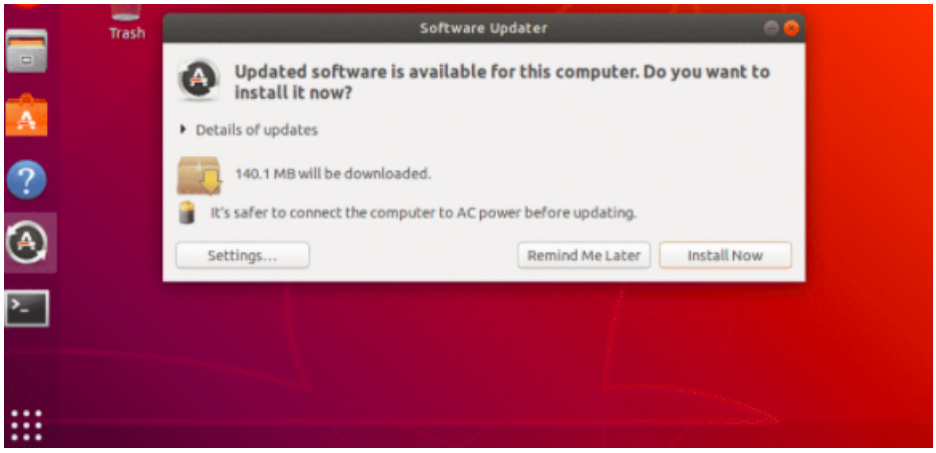

Figure 14: click on setting, and choose additional driver.

```
Sorting... Done
Full Text Search... Done
nvidia-384/bionic 390.48-0ubuntu3 amd64
  Transitional package for nvidia-driver-390

nvidia-384-dev/bionic 390.48-0ubuntu3 amd64
  Transitional package for nvidia-driver-390

nvidia-driver-390/bionic,now 390.48-0ubuntu3 amd64 [installed]
  NVIDIA driver metapackage

nvidia-headless-390/bionic 390.48-0ubuntu3 amd64
  NVIDIA headless metapackage

nvidia-headless-no-dkms-390/bionic 390.48-0ubuntu3 amd64
  NVIDIA headless metapackage - no DKMS

xserver-xorg-video-nvidia-390/bionic,now 390.48-0ubuntu3 amd64 [installed,automatic]
  NVIDIA binary Xorg driver
```

Figure 15: click the best suitable NVIDIA driver and press install.

Figure 16: follow the instruction.

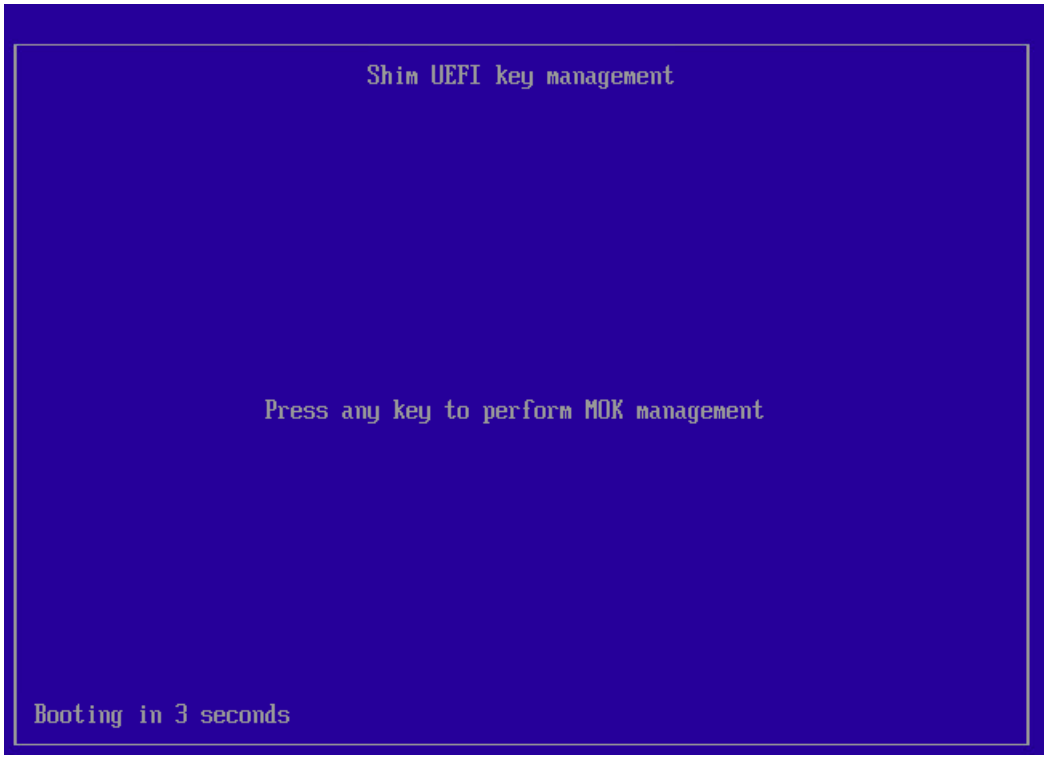

Figure 17: UEFI mode,MOK Image.

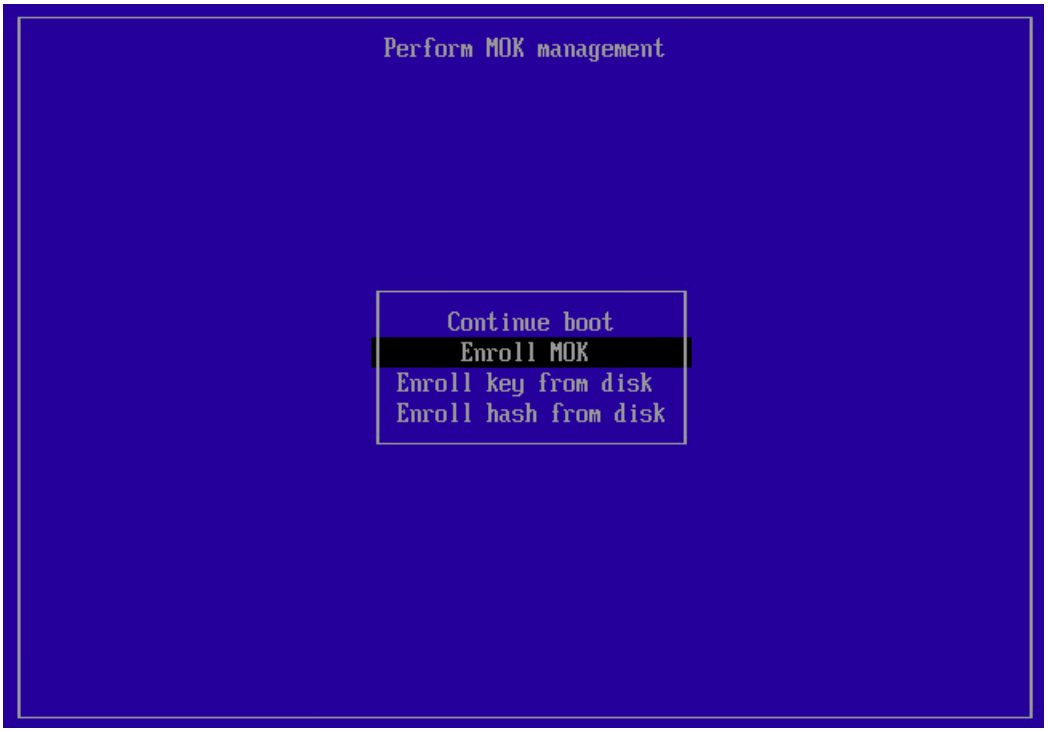

Figure 18: follow the instruction

Figure 19: follow the instruction

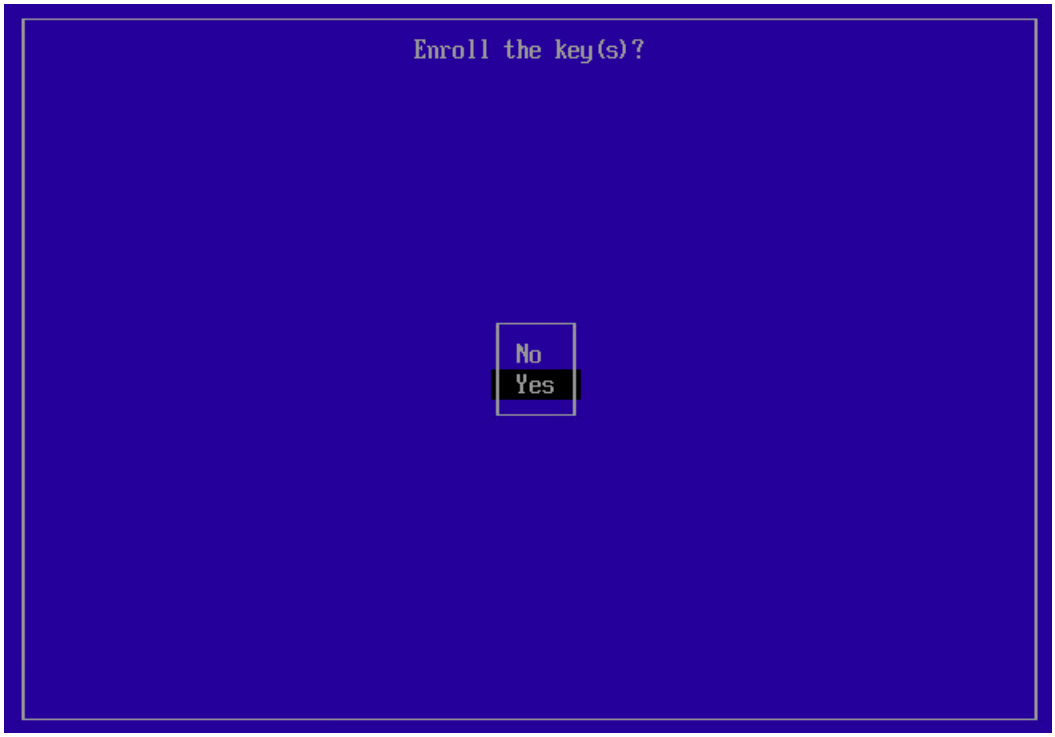

Figure 20: follow the instruction

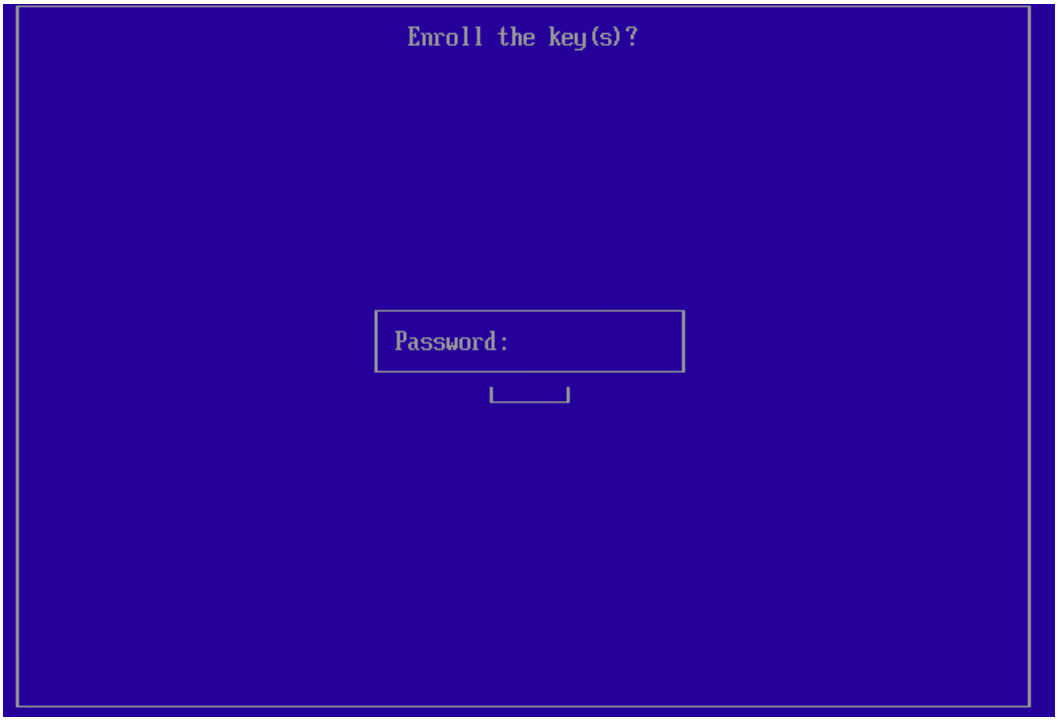

Figure 21: follow the instruction

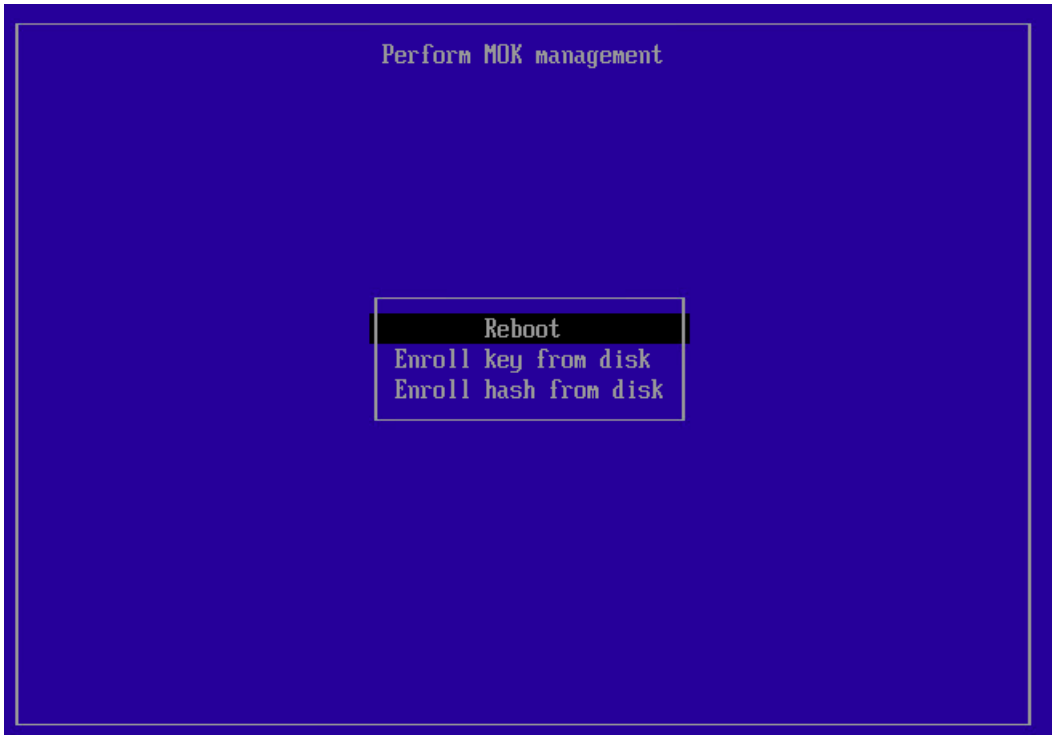

Figure 22: follow the instruction

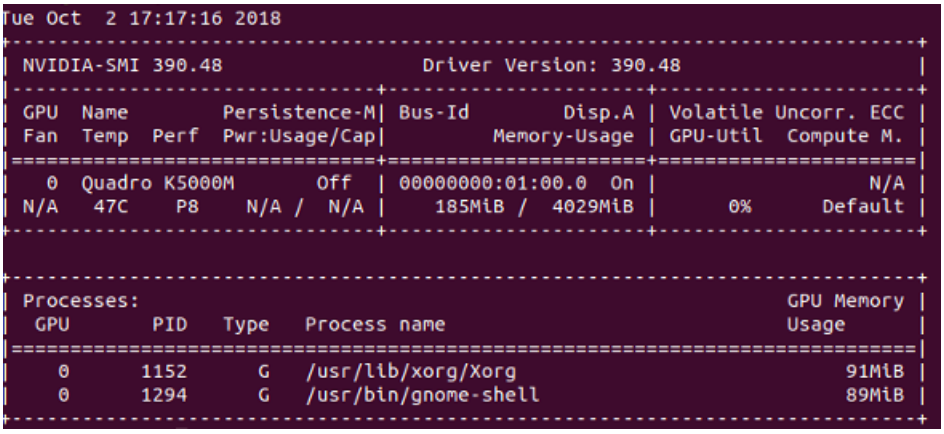

Figure 23: type in nvidia-smi and check whether the GPU card has successfully detected.

5.2 cuda install

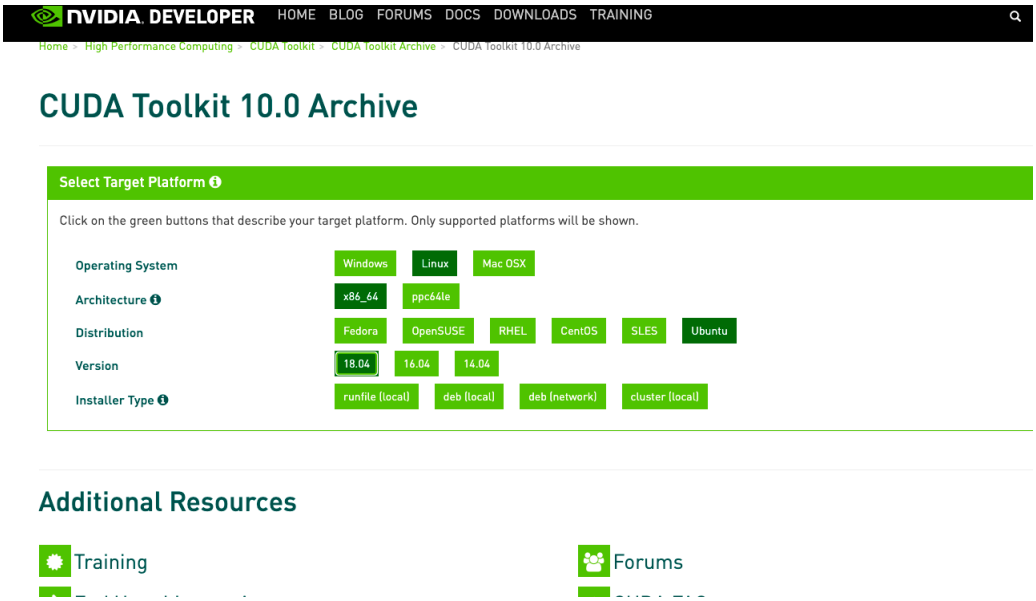

Figure 24: go to https://developer.nvidia.com/cuda-10.0-download-archive choose Linux - x86 64 - Ubuntu - 18.04 - runfile and click download

```
$ sudo sh cuda_10.0.130_410.48_linux.run
```

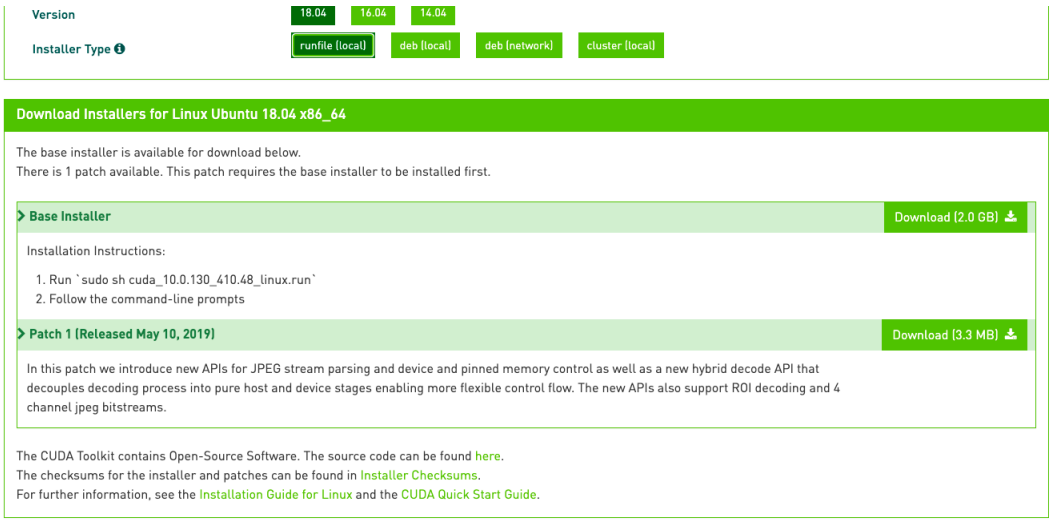

Figure 25: sudo sh the file PLEASE choose the second options as No(it will help you to install driver, but that Nvidia driver may not support your computer). After installation type in nvidia-smi and check whether the GPU card has successfully detected.

5.3 CUDA 10 cuDNN 7.6

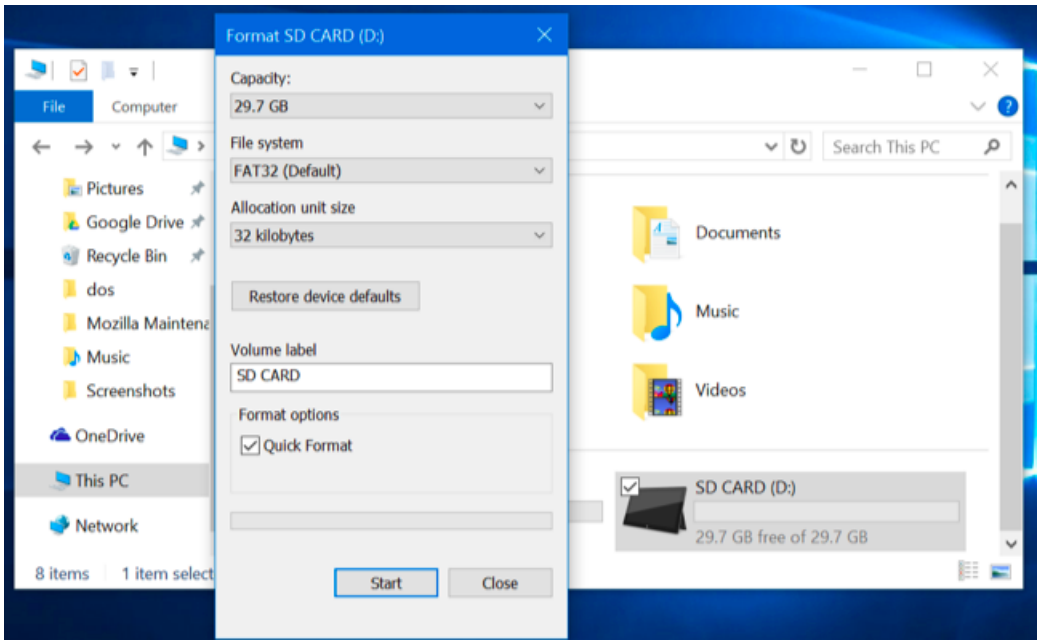

Figure 26: go to https://developer.nvidia.com/rdp/form/cudnn-download-survey It will need to join the member. choose linex and you will need to find archived file which is suitabled for cuda 10.

```
$ sudo dpkg -i libcudnn7_7.6.4.38+cuda10.0_amd64.deb
```

5.4 opencv2 installation with cuda 10.0 in Ubuntu 18.04

```
$ sudo apt update
$ sudo apt upgrade

$ sudo apt install build-essential cmake pkg-config unzip yasm git
   checkinstall
$ sudo apt install libjpeg-dev libpng-dev libtiff-dev
$ sudo apt install libavcodec-dev libavformat-dev libswscale-dev
   libavresample-dev
$ sudo apt install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev
$ sudo apt install libxvidcore-dev x264 libx264-dev libfaac-dev
   libmp3lame-dev libtheora-dev
$ sudo apt install libfaac-dev libmp3lame-dev libvorbis-dev
$ sudo apt install libopencore-amrnb-dev libopencore-amrwb-dev
$ sudo apt-get install libdc1394-22 libdc1394-22-dev libxine2-dev
   libv4l-dev v4l-utils
$ cd /usr/include/linux
$ sudo ln -s -f ../libv4l1-videodev.h videodev.h
$ cd ~
$ sudo apt-get install libgtk-3-dev
$ sudo apt-get install python3-dev python3-pip
$ sudo -H pip3 install -U pip numpy
$ sudo apt install python3-testresources
$ sudo apt-get install libtbb-dev
$ sudo apt-get install libatlas-base-dev gfortran
$ sudo apt-get install libprotobuf-dev protobuf-compiler
$ sudo apt-get install libgoogle-glog-dev libgflags-dev
$ sudo apt-get install libgphoto2-dev libeigen3-dev libhdf5-dev doxygen


$ cd ~
$ wget -O opencv.zip https://github.com/opencv/opencv/archive/4.1.0.zip
$ wget -O opencv_contrib.zip
   https://github.com/opencv/opencv_contrib/archive/4.1.0.zip
$ unzip opencv.zip
$ unzip opencv_contrib.zip


# create virtual environment
$ sudo pip install virtualenv virtualenvwrapper
$ sudo rm -rf ~/.cache/pip
```

```
$ echo "Edit ~/.bashrc"
$ export WORKON_HOME=$HOME/.virtualenvs
$ export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
$ source /usr/local/bin/virtualenvwrapper.sh
$ mkvirtualenv cv -p python3
$ pip install numpy



make file
$ cd opencv-4.1.0
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE
-D CMAKE_INSTALL_PREFIX=/usr/local
-D INSTALL_PYTHON_EXAMPLES=ON
-D INSTALL_C_EXAMPLES=OFF
-D WITH_TBB=ON
-D WITH_CUDA=ON
-D BUILD_opencv_cudacodec=OFF
-D ENABLE_FAST_MATH=1
-D CUDA_FAST_MATH=1
-D WITH_CUBLAS=1
-D WITH_V4L=ON
-D WITH_QT=OFF
-D WITH_OPENGL=ON
-D WITH_GSTREAMER=ON
-D OPENCV_GENERATE_PKGCONFIG=ON
-D OPENCV_PC_FILE_NAME=opencv.pc
-D OPENCV_ENABLE_NONFREE=ON
-D
   OPENCV_PYTHON3_INSTALL_PATH=~/.virtualenvs/cv/lib/python3.6/site-packages
-D
   OPENCV_EXTRA_MODULES_PATH=~/downloads/opencv/opencv_contrib-4.1.0/modules
-D PYTHON_EXECUTABLE=~/.virtualenvs/cv/bin/python
-D BUILD_EXAMPLES=ON ..

$ nproc
$ make -j8
$ sudo make install

$ sudo /bin/bash -c 'echo "/usr/local/lib" >>
   /etc/ld.so.conf.d/opencv.conf'
$ sudo ldconfig


# change the executable environment to global environment
```

```
$ sudo cp -r ~/.virtualenvs/cv/lib/python3.6/site-packages/cv2
   /usr/local/lib/python3.6/dist-packages

$ echo "Modify config-3.6.py to point to the target directory"
$ sudo nano /usr/local/lib/python3.6/dist-packages/cv2/config-3.6.py

‘‘‘
    PYTHON_EXTENSIONS_PATHS = [
    os.path.join('/usr/local/lib/python3.6/dist-packages/cv2',
       'python-3.6')
    ] + PYTHON_EXTENSIONS_PATHS
‘‘‘
```

5.5 libpq-dev

Required package in nano

```
sudo apt install libpq-dev
```

5.6 postgres

sudo apt install postgresql postgresql-contrib

```
$ sudo apt install postgresql postgresql-contrib
$ systemctl status postgresql.service
$ sudo apt install libpq-dev
```