

1. **先裝好Ubuntu如下網站所示**

**先製作Live USB隨身碟的軟體Unetbootin，如下網站所示**

<https://blog.xuite.net/yh96301/blog/57645340-Ubuntu+18.04%E8%A3%BD%E4%BD%9CLive+USB%E9%9A%A8%E8%BA%AB%E7%A2%9F%E7%9A%84%E8%BB%9F%E9%AB%94Unetbootin>

**以USB安裝Ubuntu的步驟**

<https://blog.xuite.net/yh96301/blog/242333268>

**裝好後請記得 reboot 謝謝**

**Ubuntu 18.04 安裝NVIDIA驅動程式**

<http://chiustin.blogspot.com/2019/01/ubuntu-1804-nvidia.html>

**安裝前的套件**  
$ sudo apt-get install gcc  
$ sudo apt-get install make  
$ sudo service gdm3 stop  
$ reboot

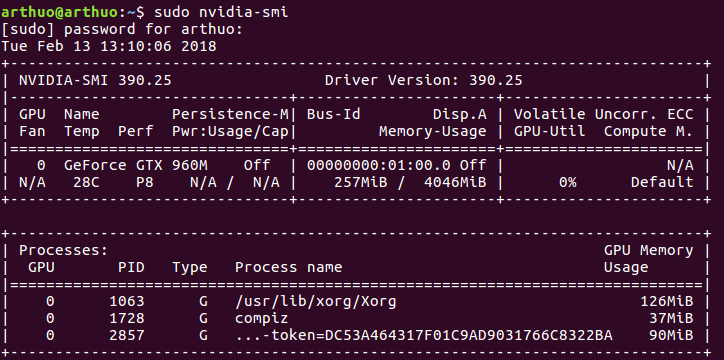
**安裝驅動**  
$ sudo add-apt-repository ppa:graphics-drivers/ppa

$ sudo apt-get update  
$ ubuntu-drivers devices  
$ sudo apt-get install nvidia-driver-415

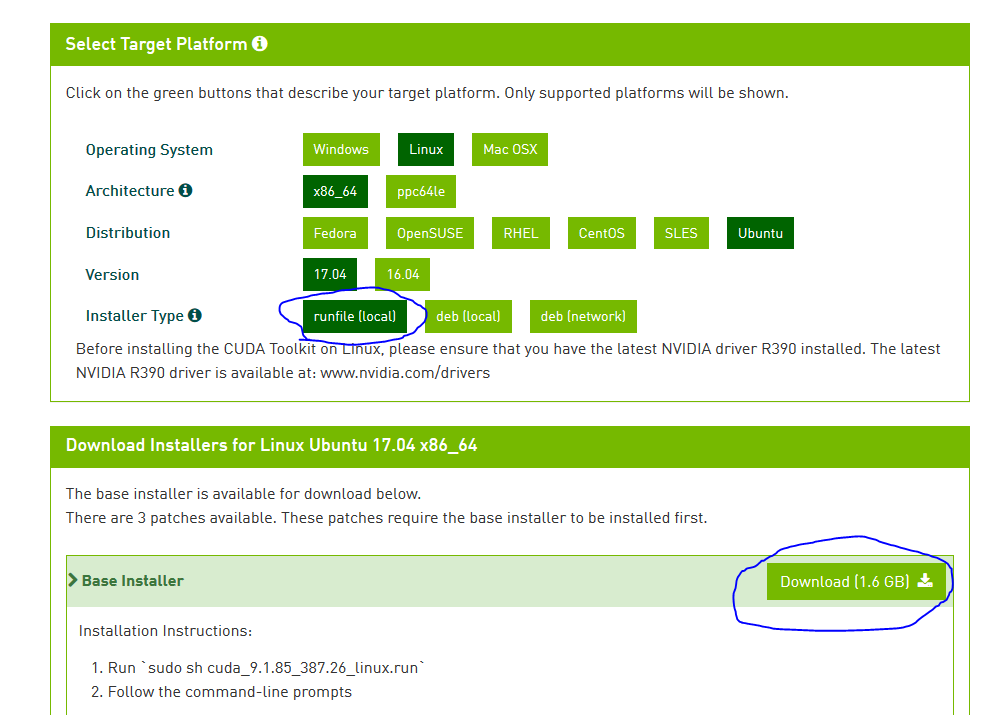
$ sudo service gdm3 start  
$ reboot

**進入桌面，執行下面的命令，查看驅動的安裝狀態**

$ sudo nvidia-smi

**若有安裝成功應該出現以下畫面**

1. **CUDA installation**
   1. **先上Cuda官網安裝toolkit(請記得選擇是當的OS系統，如下，但請看你真實的系統)**

<https://developer.nvidia.com/cuda-downloads?target_os=Linux&target_arch=x86_64&target_distro=Ubuntu&target_version=1704&target_type=runfilelocal>

* 1. **接下來，如下網站進行裏頭的三行指令，cuda\_9.1這邊要自己更動**

<https://mark-down-now.blogspot.com/2018/05/pytorch-gpu-ubuntu-1804.html?fbclid=IwAR1clPJPvKOX0po0NkaL0rs96dH9Vv9gKQhmmSKGg-YpI0w9WzRsxMebgl4>

只取這三行執行

cd ~/Downloads

chmod 755 cuda\_9.1.85\_387.26\_linux.run

sudo ./cuda\_9.1.85\_387.26\_linux.run –override

1. **Docker installation**

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04?fbclid=IwAR3D4T707CRGAX-iIL0OEUoZD8b4u3V-czKF8wPrfFkTCkJCCgN2EfEL4pY>

## Step 1 — Installing Docker

First, update your existing list of packages:

* sudo apt update

Next, install a few prerequisite packages which let apt use packages over HTTPS:

* sudo apt install apt-transport-https ca-certificates curl software-properties-common

Then add the GPG key for the official Docker repository to your system:

* curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

Add the Docker repository to APT sources:

* sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"

Next, update the package database with the Docker packages from the newly added repo:

* sudo apt update

Make sure you are about to install from the Docker repo instead of the default Ubuntu repo:

* apt-cache policy docker-ce

Finally, install Docker:

* sudo apt install docker-ce

Docker should now be installed, the daemon started, and the process enabled to start on boot. Check that it's running:

* sudo systemctl status docker

The output should be similar to the following, showing that the service is active and running:

Output

● docker.service - Docker Application Container Engine

Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)

Active: active (running) since Thu 2018-07-05 15:08:39 UTC; 2min 55s ago

Docs: https://docs.docker.com

Main PID: 10096 (dockerd)

Tasks: 16

CGroup: /system.slice/docker.service

├─10096 /usr/bin/dockerd -H fd://

└─10113 docker-containerd --config /var/run/docker/containerd/containerd.toml

## Step 2 — Executing the Docker Command Without Sudo (Optional)

By default, the docker command can only be run the **root** user or by a user in the **docker** group, which is automatically created during Docker's installation process. If you attempt to run the docker command without prefixing it with sudo or without being in the **docker** group, you'll get an output like this:

Output

docker: Cannot connect to the Docker daemon. Is the docker daemon running on this host?.

See 'docker run --help'.

If you want to avoid typing sudo whenever you run the docker command, add your username to the docker group:

* sudo usermod -aG docker ${USER}

To apply the new group membership, log out of the server and back in, or type the following:

* su - ${USER}

You will be prompted to enter your user's password to continue.

Confirm that your user is now added to the **docker** group by typing:

* id -nG

Output

sammy sudo docker

## Step 4 — Working with Docker Images

You can search for images available on Docker Hub by using the docker command with the search subcommand. Search for the ncuipvr/keras-with-ssh image, type:

* docker search ncuipvr/keras-with-ssh