

## Installation of Nvidia Drivers + CUDA for x2 Systems

Genalyn Cabilao - 2019-03-06 - in Infrastructure

# Installation of Nvidia Drivers + CUDA for x2 Systems

*This article describes the steps on installing the official Nvidia drivers and CUDA toolkit for Ubuntu 16.04 and 18.04 for x2 systems in Packet.*

### Systems Requirements:

- CUDA-capable GPU
- Supported version of Linux with gcc compiler and toolchain
- NVIDIA CUDA Toolkit

### Pre-installation Actions:

1. Verify the system has a CUDA-capable GPU.

```
lspci | grep -i nvidia
```

To install lspci, run:

```
apt-get install pciutils
```

If you do not see any settings, update the PCI hardware database that Linux maintains by

entering update-pciids (generally found in /sbin or /usr/sbin) at the command line and rerun the previous lspci command.

2. Verify the system is running a supported version of Linux.

```
uname -m && cat /etc/*release
```

```
Output: x86_64DISTRIB_ID=UbuntuDISTRIB_RELEASE=16.04DISTRIB_CODENAME=
xenialDISTRIB_DESCRIPTION="Ubuntu 16.04.5
LTS"NAME="Ubuntu"VERSION="16.04.5 LTS (Xenia
Xerus)"ID=ubuntuID_LIKE=debianPRETTY_NAME="Ubuntu 16.04.5
LTS"VERSION_ID="16.04"HOME_URL="http://www.ubuntu.com/"SUPPORT_URL="h
ttp://help.ubuntu.com/"BUG_REPORT_URL="http://bugs.launchpad.net/ubun
tu/"VERSION_CODENAME=xenialUBUNTU_CODENAME=xenial
```

3. Verify the system has the correct kernel headers and development packages installed. Below is the command to check the current kernel version the system is running.

```
uname -r
```

The development packages and kernel headers are needed for the CUDA driver. It should match the running version of the kernel at the time of the driver installation, also whenever the driver is rebuilt. (Reference: [CUDA Installation Guide Linux - 2.4 Verify System has Correct Kernel Headers and Development Packages Installed](#))

**Note:** If system update is performed and changes to the version of the linux kernel being used, make sure to rerun the commands below to ensure you have the kernel headers installed. Otherwise, the CUDA Driver will fail to work with the new kernel. (Reference: [CUDA Installation Guide Linux - 2.4 Verify System has Correct Kernel Headers and Development Packages Installed](#))

To install the development packages and kernel headers for the currently running kernel:

```
apt-get install build-essentialapt-get install linux-headers-$(uname
-r)
```

4. Verify the system has gcc installed.

```
gcc --version
```

## Installation of Nvidia driver and CUDA from the official repo:

1. Perform the pre-installation actions.
2. Install official repository. Browse the repos depending on your operating system.

<https://developer.download.nvidia.com/compute/cuda/repos/>

**Note:** At the time of writing, the latest version is 10.0.130-1.

For **Ubuntu 16.04** (32 and 64-bit):

wget

[https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86\\_64/cuda-repo-ubuntu1604\\_<version>\\_amd64.deb](https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/cuda-repo-ubuntu1604_<version>_amd64.deb)

Example:

wget

[https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86\\_64/cuda-repo-ubuntu1604\\_10.0.130-1\\_amd64.deb](https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/cuda-repo-ubuntu1604_10.0.130-1_amd64.deb)

For **Ubuntu 18.04** (32 and 64-bit):

wget

[https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86\\_64/cuda-repo-ubuntu1804\\_<version>\\_amd64.deb](https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-repo-ubuntu1804_<version>_amd64.deb)

Example:

wget

[https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86\\_64/cuda-repo-ubuntu1804\\_10.0.130-1\\_amd64.deb](https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-repo-ubuntu1804_10.0.130-1_amd64.deb)

Then install.

```
dpkg -i cuda-repo-ubuntu<ubuntu_version>_<cuda_version>_amd64.deb
```

**Fetch the gpg keys for the repo.**

```
apt-key adv --fetch-keys
```

```
https://developer.download.nvidia.com/compute/cuda/repos/<distro>/<architecture>/7fa2af80.pub
```

#### **For Ubuntu 16.04**

```
apt-key adv --fetch-keys
```

```
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86\_64/7fa2af80.pub
```

#### **For Ubuntu 18.04:**

```
apt-key adv --fetch-keys
```

```
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86\_64/7fa2af80.pub
```

To check, navigate to the /etc/source.list.d and check if there is repo list for CUDA.

```
ls -l /etc/apt/sources.list.d
```

```
cat /etc/apt/sources.list.d/cuda.list
```

#### **Output:**

```
deb
```

```
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64 /
```

Update apt cache

```
apt-get update
```

3. Install the nvidia driver and the CUDA toolkit.

```
apt-get install --no-install-recommends cuda
```

4. Reboot machine.

5. Once rebooted, double check if the nvidia driver has been loaded automatically.

```
lsmod | grep -i nvidia
```

**Output:**

```
nvidia_uvm          790528  0nvidia_drm          45056
0nvidia_modeset    1040384  1 nvidia_drmnvidia
16592896  2 nvidia_modeset,nvidia_uvmipmi_msghandler    49152  3
ipmi_ssif,nvidia,ipmi_sidrm_kms_helper    155648  1 nvidia_drmdrm
364544  3 drm_kms_helper,nvidia_drm
```

6. Check nvidia gpu and driver version.

```
~# nvidia-debugdump --list
```

Found 1 NVIDIA devices

```
Device ID:          0
Device name:        Tesla P4
GPU internal ID:    0421818006236
```

```
~# cat /proc/driver/nvidia/version
```

NVRM version: NVIDIA UNIX x86\_64 Kernel Module 410.79 Thu Nov 15 10:41:04 CST 2018

GCC version: gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.11)

```
~# nvidia-smi +-----+
-----+| NVIDIA-SMI 410.79      Driver Version:
410.79      CUDA Version: 10.0      ||-----+
+-----+-----+| GPU Name
Persistence-M| Bus-Id      Disp.A | Volatile Uncorr. ECC || Fan
Temp Perf Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute
M.
||=====+=====+=====
=====||  0 Tesla P4          Off | 00000000:D8:00.0 Off |
          0 || N/A  37C   P0   23W / 75W |      0MiB /
7611MiB |      0%      Default |+-----+
-----+-----+
```

```

+-----+
-----+| Processes:
      GPU Memory || GPU      PID   Type   Process name
              Usage
||=====
=====|| No running processes found
      |+-----+
-----+

```

7. Check CUDA version.

```
nvcc -V
```

**Output:**

```

nvcc: NVIDIA (R) Cuda compiler driverCopyright (c) 2005-2018 NVIDIA
CorporationBuilt on Sat_Aug_25_21:08:01_CDT_2018Cuda compilation
tools, release 10.0, V10.0.130

```

**Note:** For Ubuntu 18.04 image for x2 system, the *nouveau* driver is integrated and loaded on boot automatically. Because of this it interferes with the loading of the official Nvidia driver (installed from above instruction).

To unload *nouveau*, do this:

```
~# rmmod nouveau
```

However, *nouveau* will not prevent it to load on-boot.

Reference:

[CUDA Installation Guide - Linux](#)