[CUDA Installation Guide for Linux (nvidia.com)](https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html" \l "pre-installation-actions)

# 2. Pre-installation Actions

lspci | grep -i nvidia

**2.1. Verify You Have a CUDA-Capable GPU**



**2.2. Verify You Have a Supported Version of Linux**

To determine which distribution and release number you’re running

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

Graphical user interface, text, application

Description automatically generated

The x86\_64 line indicates you are running on a 64-bit system. The remainder gives information about your distribution.

## 2.3. Verify the System Has gcc Installed

gcc --version

Text

Description automatically generated

An error message displays, so we need to install the development tools from our Linux distribution or obtain a version of gcc and its accompanying toolchain from the Web.

After installing using the code suggested, we get this output.

Text

Description automatically generated

## 2.4. Verify the System has the Correct Kernel Headers and Development Packages Installed

The version of the kernel our system is running can be found by running the following command.

uname -r



This is the version of the kernel headers and development packages that must be installed prior to installing the CUDA Drivers. This command will be used multiple times.

The kernel headers and development packages for the currently running kernel can be installed with:

sudo apt-get install linux-headers-$(uname -r)

## 2.7. Download the NVIDIA CUDA Toolkit

<https://developer.nvidia.com/cuda-downloads>

Diagram

Description automatically generated

**Download Installer for Linux Ubuntu 18.04 x86\_64**

wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86\_64/cuda-ubuntu1804.pin

sudo mv cuda-ubuntu1804.pin /etc/apt/preferences.d/cuda-repository-pin-600

wget https://developer.download.nvidia.com/compute/cuda/12.0.0/local\_installers/cuda-repo-ubuntu1804-12-0-local\_12.0.0-525.60.13-1\_amd64.deb

sudo dpkg -i cuda-repo-ubuntu1804-12-0-local\_12.0.0-525.60.13-1\_amd64.deb

sudo cp /var/cuda-repo-ubuntu1804-12-0-local/cuda-\*-keyring.gpg /usr/share/keyrings/

sudo apt-get update

sudo apt-get -y install cuda

Text

Description automatically generated



After about 20 minutes this came up:

Timeline

Description automatically generated

# 3. Package Manager Installation

[CUDA Quick Start Guide (nvidia.com)](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html#ubuntu)

Debian Installer

Install the repository meta-data, update the GPG key, update the apt-get cache, and install CUDA:

sudo dpkg --install cuda-repo-<distro>-<version>.<architecture>.deb

sudo apt-key del 7fa2af80

wget https://developer.download.nvidia.com/compute/cuda/repos/$distro/$arch/cuda-keyring\_1.0-1\_all.deb

sudo dpkg -i cuda-keyring\_1.0-1\_all.deb

sudo add-apt-repository contrib

sudo apt-get update

sudo apt-get -y install cuda

This translates into:

sudo dpkg --install cuda-repo-ubuntu1804-12-0-local\_12.0.0-525.60.13-1\_amd64.deb

sudo apt-key del 7fa2af80

wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86\_64/cuda-keyring\_1.0-1\_all.deb

sudo dpkg -i cuda-keyring\_1.0-1\_all.deb

sudo add-apt-repository contrib

sudo apt-get update

sudo apt-get -y install cuda

cuda-repo-ubuntu1804-12-0-local\_12.0.0-525.60.13-1\_amd64.deb

<distro> ubuntu1804

<version> local\_12.0.0

<arch> x86\_64

“sudo add-apt-repository contrib” gave the error:



A reboot followed by “sudo apt-get update” fixed it.

# Post-installation Actions

Add PATH to Environment

echo $PATH



export PATH=/usr/local/cuda-12.0/bin${PATH:+:${PATH}}

echo $PATH



Modify System-Wide Environment Variables for Logged-In Users

sudo nano /etc/profile

Add code to end of file.

export PATH=/usr/local/cuda-12.0/bin${PATH:+:${PATH}}

save and quit by pressing CTRL + X, followed by Y, and then ENTER.

Text

Description automatically generated

Reboot and test new PATH variable.

### Verify the Installation

When the driver is loaded, the driver version can be found using this code.

cat /proc/driver/nvidia/version



Verify the CUDA Toolkit version

*nvcc -V*

Text

Description automatically generated

#### Running the Binaries

Download the CUDA samples from <https://github.com/nvidia/cuda-samples>

Under the 1\_Utilities directory, compile the deviceQuery program, and run it.

cd cuda-samples-master/cuda-samples-master/Samples/1\_Utilities/deviceQuery

sudo make

./deviceQuery

Graphical user interface, text

Description automatically generated

We now have CUDA installed!

Also test the bandwidthTest program for good measure.

cd cuda-samples-master/cuda-samples-master/Samples/1\_Utilities/bandwidthTest

sudo make

./bandwidthTest

Some CUDA samples use third-party libraries which may not be installed by default on your system. These samples attempt to detect any required libraries when building.

Graphical user interface, text, application

Description automatically generated

From the CUDA installation documentation:

If a library is not detected, it waives itself and warns you which library is missing. To build and run these samples, you must install the missing libraries. These dependencies may be installed if the RPM or Deb cuda-samples-12-0 package is used. In cases where these dependencies are not installed, follow the instructions below.