

# NVIDIA driver のインストール

## PyTorch で使用できる CUDA と cuDNN のバージョン確認

次に、[PyTorch-github](#) でサポートされている CUDA と cuDNN のバージョンを確認する.

CUDAサポートを使用してコンパイルする場合は、インストールします

- NVIDIA CUDA9.2 以降
- NVIDIA cuDNNの V7以上
- コンパイラ CUDAと互換性のある 注： については、 参照でき サポートさ cuDNNサポートマトリックス れている  
さまざまなCUDA、CUDAドライバー、およびNVIDIAハードウェアを備えたcuDNNバージョンを ます。

2021 年 4 月 21 日時点では、上記のような表示だった.

さらに, [PyTorch](#) からインストールしたい PyTorch がサポートしている CUDA のバージョンを確認する.

## START LOCALLY

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, 1.9 builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g., numpy)**, depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also [install previous versions of PyTorch](#). Note that LibTorch is only available for C++.

PyTorch Build	Stable (1.8.1)		Preview (Nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
Compute Platform	CUDA 10.2	CUDA 11.1	ROCm 4.0 (beta)	CPU
Run this Command:	<b>NOTE:</b> 'conda-forge' channel is required for cudatoolkit 11.1 <code>conda install pytorch torchvision torchaudio cudatoolkit=11.1 -c pytorch -c conda-forge</code>			

2021 年 4 月 21 日時点では、上記のような表示だった.

# PyTorch がサポートしている最新バージョンの CUDA をインストール

これまでの情報をもとに、PyTorch がサポートしている最新バージョンの CUDA をインストールする。

- まず、[CUDA-Toolkit-Archive](#) からインストールしたい CUDA のバージョンを選択する (Installer Type として deb (local) を選択).
  - もし、インストールすべき CUDA のバージョンがわからない場合は、先に CUDA 付き PyTorch をインストール後、`conda list` コマンドを使用し、インストールされた `culatoolkit` のバージョンをもとに選ぶと良い。
- 表示されたコマンドに従い、パッケージリポジトリを登録。
  - 今回は、バージョン 11.1 を選択した。
- 表示されたコマンドの最後の部分を以下のように変更して、パッケージをインストールする。

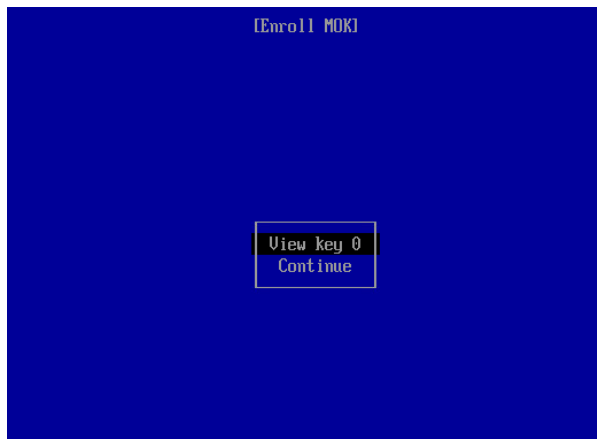
```
$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin
$ sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-repository-pin-600
$ wget https://developer.download.nvidia.com/compute/cuda/11.1.1/local_installers/cuda-repo-ubuntu2004-11-1-local_11.1.1-455.32.00-1_amd64.deb
$ sudo dpkg -i cuda-repo-ubuntu2004-11-1-local_11.1.1-455.32.00-1_amd64.deb
$ sudo apt-key add /var/cuda-repo-ubuntu2004-11-1-local/7fa2af80.pub
$ sudo apt update
$ sudo apt install -y cuda-drivers
```

もし、このときにパスワード入力を求められた場合は以下の手順でドライバインストールを完了する。求められなかった場合は `reboot` する。

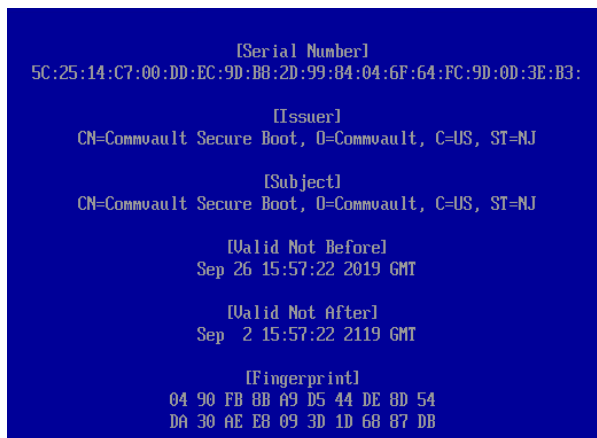
# インストール時にパスワード入力を求められた場合

ドライバインストール時にパスワードを求められた場合は、以下の手順でインストールを完了する『[Enrolling the Commvault Keys with the UEFI MOK \(Machine Owned Key\) List](#)』。

1. OS の起動の前の選択肢で Enroll MOK を選ぶ
2. View key 0 を選択する.



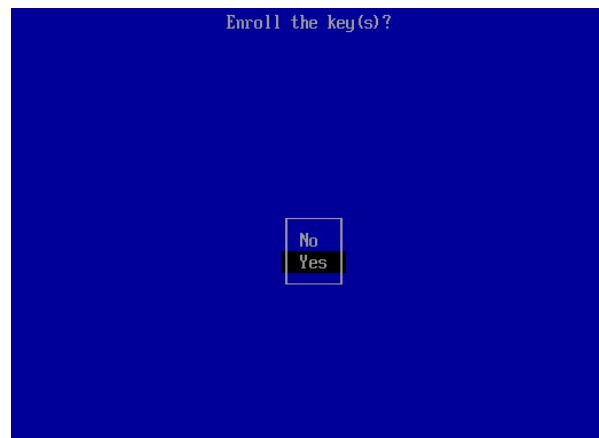
3. key の詳細が表示されるので、いずれかのキーを押して MOK 登録 画面に移動します。



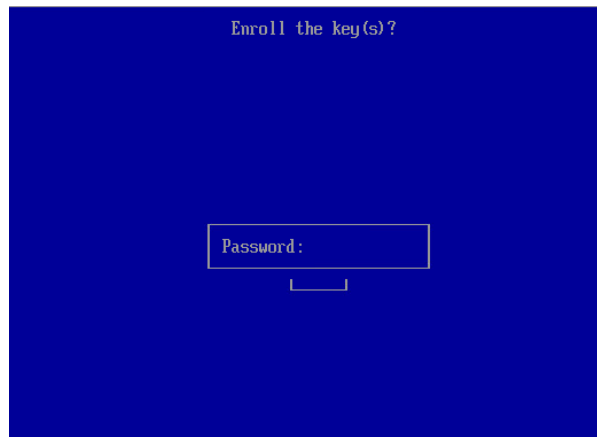
#### 4. continue を選択



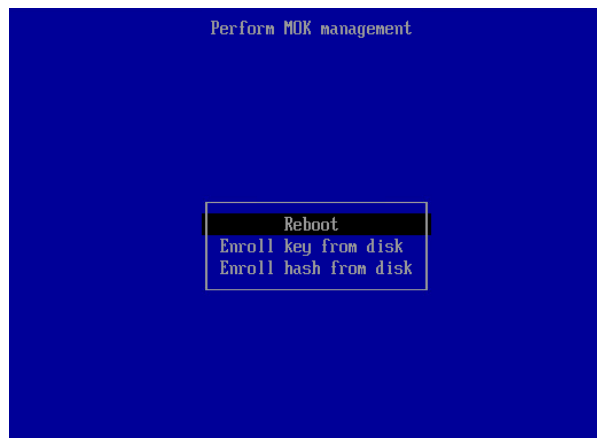
#### 5. Yes を選択



## 6. ドライバインストール時に設定したパスワードを入力



## 7. 再々度 MOK 登録画面が表示されるので、 **Reboot** を選択



## Nvidia driver インストールの確認

Nvidia-driver がインストールできたかの確認は、以下の 2 つのコマンドで行える。

```
$ nvidia-smi
```

```
Thu Apr 22 15:24:38 2021
```

+-----+-----+-----+-----+-----+-----+									
NVIDIA-SMI		455.32.00		Driver Version: 455.32.00			CUDA Version: 11.1		
+-----+-----+-----+-----+-----+-----+									
GPU	Name	Persistence-M			Bus-Id	Disp.A	Volatile	Uncorr. ECC	
Fan	Temp	Perf	Pwr:Usage/Cap		Memory-Usage		GPU-Util	Compute M.	MIG M.
=====									
0	GeForce GTX 166...	On	00000000:01:00.0			Off	N/A		
N/A	53C	P8	5W / N/A		194MiB / 5944MiB		9%	Default	N/A
+-----+-----+-----+-----+-----+-----+									

Processes:						
GPU	GI ID	CI ID	PID	Type	Process name	GPU Memory Usage
0	N/A	N/A	1147	G	/usr/lib/xorg/Xorg	45MiB
0	N/A	N/A	1746	G	/usr/lib/xorg/Xorg	54MiB
0	N/A	N/A	1929	G	/usr/bin/gnome-shell	83MiB

```
$ cat /proc/driver/nvidia/version
```

```
NVRM version: NVIDIA UNIX x86_64 Kernel Module 455.32.00 Wed Oct 14 22:46:18 UTC 2020
GCC version: gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)
```

# Nvidia Driver に合う CUDA のインストール

## CUDA のインストール

次に、先程インストールした `Nvidia Driver` に合う CUDA をインストールする.

```
$ sudo apt install cuda-NN-M
```

例えば、CUDA 10.1 をインストールするのであれば、`sudo apt install cuda-10-1`

## パス設定

以下を `~/.bashrc` ファイルの末尾に追加することで、CUDA を環境変数に登録する.

```
# >>> cuda path >>>
export PATH="/usr/local/cuda-11.1/bin:$PATH"
export LD_LIBRARY_PATH="/usr/local/cuda-11.1/lib64:$LD_LIBRARY_PATH"
export LIBRARY_PATH="/usr/local/cuda-11.1/include:$LIBRARY_PATH"
# <<< cuda path <<<
```

これらを追加後、`source ~/.bashrc` でパスを更新する.



## インストールの確認

~/home/ ディレクトリを端末で開き、以下のコマンドが実行できるか確認. 成功すると、以下のような結果が返ってくる.

```
$ nvcc -V

nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2020 NVIDIA Corporation
Built on Mon_Oct_12_20:09:46_PDT_2020
Cuda compilation tools, release 11.1, V11.1.105
Build cuda_11.1.TC455_06.29190527_0
```

ここで、release バージョンがインストールしたものと同じであるかも確認すること.

さらに、以下のコマンドでサンプルプログラムを実行できる.

```
$ cd /usr/local/cuda-11/samples
$ sudo make
$ cd ./bin/x86_64/linux/release
$ sudo ./deviceQuery
```

先程のサンプルプログラムの実行に成功すると、以下のような結果が表示される。

```
./deviceQuery Starting...

  CUDA Device Query (Runtime API) version (CUDA static linking)

Detected 1 CUDA Capable device(s)

Device 0: "GeForce GTX 1660 Ti"
  CUDA Driver Version / Runtime Version      11.1 / 11.1
  CUDA Capability Major/Minor version number: 7.5
  Total amount of global memory:              5945 MBytes (6233391104 bytes)
  (24) Multiprocessors, ( 64) CUDA Cores/MP: 1536 CUDA Cores
  GPU Max Clock rate:                        1590 MHz (1.59 GHz)
  Memory Clock rate:                         6001 Mhz
  Memory Bus Width:                           192-bit
  L2 Cache Size:                             1572864 bytes
  Maximum Texture Dimension Size (x,y,z)      1D=(131072), 2D=(131072, 65536), 3D=(16384, 16384, 16384)
  Maximum Layered 1D Texture Size, (num) layers 1D=(32768), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers 2D=(32768, 32768), 2048 layers
  Total amount of constant memory:             65536 bytes
  Total amount of shared memory per block:     49152 bytes
  Total shared memory per multiprocessor:      65536 bytes
  Total number of registers available per block: 65536
  Warp size:                                  32
  Maximum number of threads per multiprocessor: 1024
  Maximum number of threads per block:         1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                      2147483647 bytes
  Texture alignment:                          512 bytes
  Concurrent copy and kernel execution:        Yes with 3 copy engine(s)
  Run time limit on kernels:                   Yes
  Integrated GPU sharing Host Memory:           No
  Support host page-locked memory mapping:      Yes
  Alignment requirement for Surfaces:           Yes
  Device has ECC support:                       Disabled
  Device supports Unified Addressing (UVA):      Yes
  Device supports Managed Memory:               Yes
  Device supports Compute Preemption:           Yes
  Supports Cooperative Kernel Launch:          Yes
  Supports MultiDevice Co-op Kernel Launch:     Yes
  Device PCI Domain ID / Bus ID / location ID:  0 / 1 / 0
  Compute Mode:
    < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 11.1, CUDA Runtime Version = 11.1, NumDevs = 1
Result = PASS
```

# cuDNN をインストール

まず、[NVIDIA Developer Program Membership Required](#) で ユーザ登録を行う。

次に、以下の 3 つのファイルをダウンロードする。

- cuDNN Runtime Library for < OS >
- cuDNN Developer Library for < OS >
- cuDNN Code Samples and Users Guide for < OS >

## Library for Windows and Linux, Ubuntu(x86\_64, armsbsa, PPC architecture)

cuDNN Library for Linux (aarch64sbsa)

cuDNN Library for Linux (x86\_64)

cuDNN Library for Linux (PPC)

cuDNN Library for Windows (x86)

cuDNN Runtime Library for Ubuntu20.04 x86\_64 (Deb)

cuDNN Developer Library for Ubuntu20.04 x86\_64 (Deb)

cuDNN Code Samples and User Guide for Ubuntu20.04 x86\_64 (Deb)

cuDNN Runtime Library for Ubuntu20.04 aarch64sbsa (Deb)

cuDNN Developer Library for Ubuntu20.04 aarch64sbsa (Deb)

cuDNN Code Samples and User Guide for Ubuntu20.04 aarch64sbsa (Deb)

そして、`cd Downloads/` でダウンロードディレクトリへ移動し、以下のコマンドを実行して cuDNN をインストールする.

```
$ sudo apt install ./libcudnn*
```

## cuDNN の確認

サンプルプログラムをコンパイルし、実行する.

```
$ cuda-install-samples-11.1.sh ~  
$ cd ~/NVIDIA_CUDA-11.1_Samples/  
$ make -k  
$ cd 2_Graphics/volumeRender  
$ ./volumeRender
```

ここで生成されるサンプルプログラムは、古くなっているためコンパイルエラーが発生した.