

Ubuntu で spconv をコンパイルする.

リポジトリクローン

spconv の [git リポジトリ](#) を任意のディレクトリにクローンする.

build 用パッケージをインストール

次に、spconv をコンパイルするために以下の手順で追加パッケージをインストールする。

1. [spconv の git リポジトリ](#) のインストール手順に従い、`sudo apt install libboost-all-dev` を実行する.
2. `sudo apt install cmake` で cmake をインストールする.

CUDA インストール

インストール済み CUDA のアンインストール

[NVIDIA ドライバ・CUDA・CUDA NN をインストールして深層学習環境を整える](#)を参考に, CUDA のセットアップを行う.

1. まず以下のコマンドを使用して事前にインストールされている `nvidia` と `cuda` のパッケージをアンインストールする。

```
$ sudo apt purge -y nvidia-*  
$ sudo apt purge -y cuda-*  
$ sudo apt autoremove --purge -y  
$ reboot
```

ここで、`reboot` しないと、以下の `NVIDIA` のドライバリポジトリが追加できなくなる。

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:	NOTE: '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。

1. 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.
2. 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.
3. 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.

```
$ 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`

nouveau の無効化について

nouveau の無効化について、[NVIDIA ドライバ・CUDA・CUDNN をインストールして深層学習環境を整える](#) ではその処理を行っているが、[WSL 2 で GPU を使う \(WSL 2 の Ubuntu で, CUDA や PyTorch や TensorFlow 2.2 GPU 版を動かす. Docker は使わない, Windows 10 Insider Program, WSL 2 上 の Ubuntu を使用\)](#) では、以下のように説明されている.

手動で nouveau をブラックリストに追加する必要はありません
パッケージのインストール時に、`/lib/modprobe.d/nvidia-graphics-drivers.conf` が自動的に作成されます。

そこで、`/lib/modprobe.d/nvidia-graphics-drivers.conf` が作成されているか確認する.

```
$ cd /lib/modprobe.d/  
$ ls  
  
aliases.conf                fbdev-blacklist.conf  
blacklist_linux-hwe-5.8_5.8.0-43-generic.conf  nvidia-graphics-drivers.conf  
blacklist_linux-hwe-5.8_5.8.0-50-generic.conf  systemd.conf
```

その結果、`cuda-11-1` をインストールすると作成されることが確認された.

CUDA PATH 設定

これまでの手順で、CUDA のインストールが完了した。しかし、インストールしただけでは CUDA が環境変数に登録されていないため、`/home/USER` ディレクトリなどで、`nvcc -V` といった CUDA 関連のコマンドを使用することができない。そこで、以下を `~/.bashrc` ファイルの末尾に追加する。

```
# >>> 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。

再起動後、`~/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。

次に、nvidia-driver が正常にインストールされたかを以下のコマンドを実行することで確認する。

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

```
cat: /proc/driver/nvidia/version: そのようなファイルやディレクトリはありません
```

上記の結果から nvidia-driver は正常にインストールされなかったようである。理由として考えられるのは、最初の `purge nvidia-*` をしたときに `cd /lib/modprobe.d/` ファイルに記述が残ったままになったことが考えられる。そこで、以下の手順により再度 cuda 関係のパッケージをアンインストールした後に `cd /lib/modprobe.d/` の内容を確認する。

```
$ sudo apt purge -y nvidia-*  
$ sudo apt purge -y cuda-*  
$ sudo apt autoremove --purge -y
```

```
$ cd /lib/modprobe.d/
```

```
aliases.conf  
blacklist_linux-hwe-5.8_5.8.0-43-generic.conf  
blacklist_linux-hwe-5.8_5.8.0-50-generic.conf  
fbdev-blacklist.conf  
systemd.conf
```

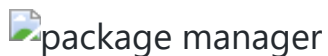
その結果、しっかりと `nvidia-graphics-drivers.conf` が削除されていた。この結果から、次の2つのことが考えられる。

1. nvidia-driver のインストール先が異なる
2. driver がインストールされていなかった

nvidia-graphics-drivers インストール

Ubuntu Linux インストール Nvidia ドライバー(最新のプロプライエタリドライバ) に、PC に搭載されている GPU の確認方法からドライバのインストール方法までが詳しく説明されていた。そこで、この記事参考にインストールを行う。

1. update manager を起動し、ソフトウェアを更新する。



2. ソフトウェアとアップデートから追加ドライバーを選択し、NVIDIA driver metapackage を nvidia-driver-460 から使用します を選択し、変更を適用

これで、ドライバのインストールは完了。

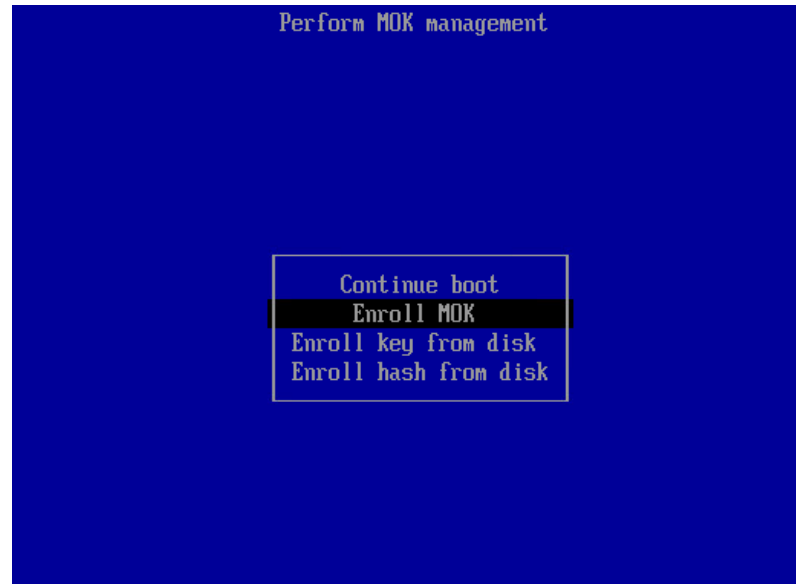
```
~$ cd /lib/modprobe.d/
$ ls

aliases.conf                fbdev-blacklist.conf
blacklist_linux-hwe-5.8_5.8.0-43-generic.conf  nvidia-graphics-drivers.conf
blacklist_linux-hwe-5.8_5.8.0-50-generic.conf  systemd.conf
```

しっかりと、nvidia-graphics-drivers.conf が書き足された。

Nvidia driver インストール成功

その後、再起動すると画像のような `Enroll MOK` というものが表示されていることに気づいた.



そこで、調べてみたところ[Ubuntu に NVIDIA のドライバをインストール（Enroll MOK に注意）](#) という記事を見つけた. この記事によると

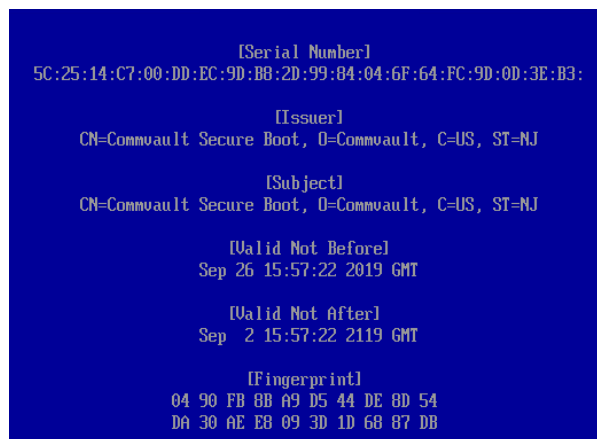
1. パスワードを入力する
 2. `sudo reboot` する
 3. OS の起動の前の選択肢で `Enroll MOK` を選ぶ ← 重要
 4. 先ほど入れたパスワードを入力
- 3.で何もせず OS を起動してしまうとドライバはインストールされない。

ということで、先程の記事を参考に操作をおこなった。ここでは、Enroll MOK 選択後の動作について説明する。

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。



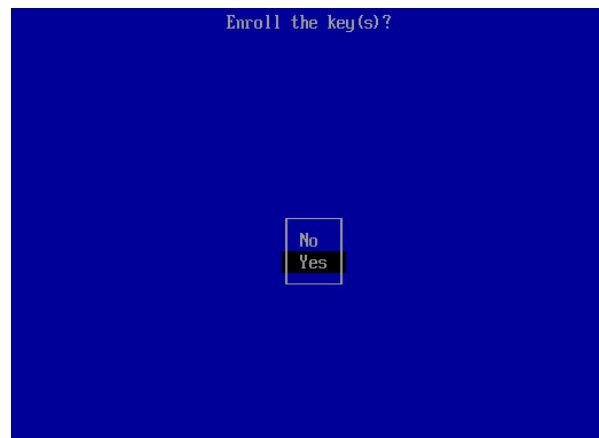
3. key の詳細が表示されるので、いずれかのキーを押して MOK 登録 画面に移動します。



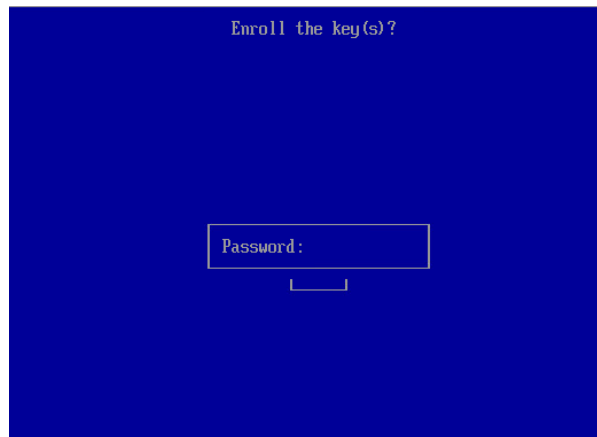
4. continue を選択



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

