

WSL + docker + GPU (2)

1.1. ベンチマークが動作するようになるまで

まず, docker が動作しているかを確認するため, 以下のコマンドを実行

```
docker run hello-world
```

その結果, docker は正常に動作していることを確認した.

```
kohei@DESKTOP-Q123T6P:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:4cf9c47f86df71d48364001ede3a4fcd85ae80ce02ebad74156906caff5378bc
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

1.2. nvidia-docker2.0

一度 nvidia-docker2.0 について調べてみることにした.

参考

- [nvidia-docker コンテナ Deep Learning](#)
- [nvidia docker って今どうなってるの? \(19.11 版\)](#)
- [nvidia docker って今どうなってるの? \(20.09 版\)](#)

その結果, このプロジェクト名は既に NVIDIA Docker ではないことを知った.

1.3. NVIDIA Docker の歴史

1.3.1. 誕生からバージョン 1.0.1 まで

第一世代の NVIDIA Docker は、標準の `docker` コマンドとは別に `nvidia-docker` というコマンドが提供されていた。

ホスト側の CUDA 関連ファイルをコンテナで利用かのようにするために、`nvidia-docker-plugin` という Docker ボリュームプラグラインのデーモンが動いており、`nvidia-docker` コマンドはこのデーモンと通信してコンテナで GPU を使うための環境を整えていた。

この頃はまだパッケージのリポジトリを公開しておらず、`.deb` や `.rpm` ファイルを直接配布していた。

1.3.2. バージョン 2.0.2 から 2.1.1 まで

このバージョンが、いわゆる「NVIDIA Docker 2」.

nvidia-docker2 パッケージでインストール可能に.

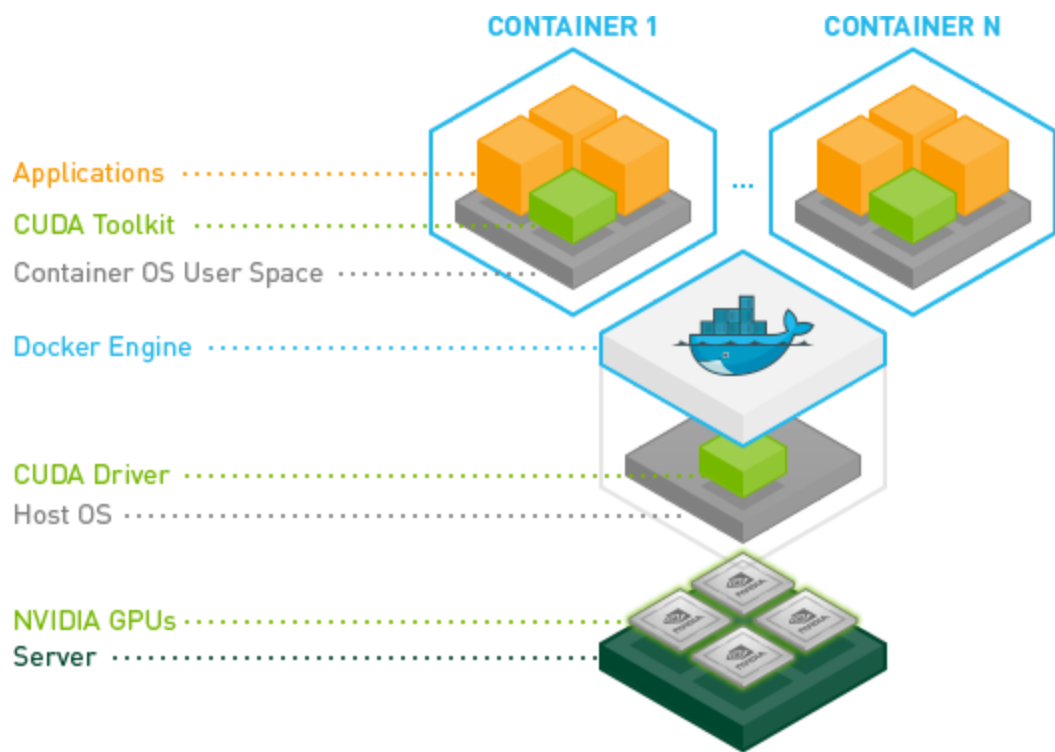
第 1 世代とはアーキテクチャが変わり、コンテナ実行時は `docker run --runtime=nvidia` として NVIDIA によるカスタマイズ版の runc を指定する形になった. `nvidia-docker` コマンドは `--runtime=nvidia` オプションなどを補って `docker` コマンドを実行するラッパースクリプトとして残った.

プロジェクトの名称は、当初 "*Docker Engine Utility for NVIDIA GPUs*" で、後に "*NVIDIA Container Runtime for Docker*" に変わった.

1.3.3. バージョン 2.2.1 以降

Docker 19.03 が GPU をサポートし `--gpus` オプションが追加されたことに対応したのがこの世代. 名称は "*NVIDIA Container Toolkit*" になった. 引き続き, `nvidia-docker2` パッケージでインストールできるが, パッケージ構成が以前と少し異なる.

2020 年 9 月現在, この [NVIDIA Container Toolkit](#) が最新の「NVIDIA Docker」です.



1.4. 環境構築方法

次の 3 つのステップで環境構築を行う.

1. NVIDIA ドライバのインストール
2. Docker のインストール
3. NVIDIA Container Toolkit のインストール

1.4.1. NVIDIA ドライバのインストール

オススメの方法は, NVIDIA 提供の cuda-drivers パッケージを使用すること.

The recommended way is to use your package manager and install the cuda-drivers package (or equivalent).

When no packages are available, you should use an official "runfile".

適当約: 「オススメの方法はご利用の環境に応じた応じたパッケージマネージャで cuda-drivers パッケージ (あるいはその相当品) をインストールすることです. パッケージが利用できない場合に, 公式の "runfile" を使ってください. 」

cuda-drivers パッケージをインストールする場合, まず [CUDA Toolkit の web サイト](#) にアクセスする.

Home > High Performance Computing > CUDA Toolkit

CUDA Toolkit

Develop, Optimize and Deploy GPU-Accelerated Apps

The NVIDIA® CUDA® Toolkit provides a development environment for creating high performance GPU-accelerated applications. With the CUDA Toolkit, you can develop, optimize, and deploy your applications on GPU-accelerated embedded systems, desktop workstations, enterprise data centers, cloud-based platforms and HPC supercomputers. The toolkit includes GPU-accelerated libraries, debugging and optimization tools, a C/C++ compiler, and a runtime library to deploy your application.

Using built-in capabilities for distributing computations across multi-GPU configurations, scientists and researchers can develop applications that scale from single GPU workstations to cloud installations with thousands of GPUs.

[Download Now](#)

CUDA 11 Features

そこで、自分が欲しいディストリビューションに合わせてダウンロードするファイルを選択する。

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

Operating System

[Windows](#) [Linux](#)

Architecture

[x86_64](#) [ppc64le](#) [sbsa](#)

Distribution

[OpenSUSE](#) [RHEL](#) [CentOS](#) [SLES](#) [Ubuntu](#)

Version

[20.04](#) [18.04](#) [16.04](#)

Installer Type

[runtime \(local\)](#) [deb \(local\)](#) [deb \(network\)](#)

Download Installer for Linux Ubuntu 20.04 x86_64

The base installer is available for download below.

> Base Installer

Installation Instructions:

```
$ 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/11.0.3/local_installers/cuda-repo-ubuntu2004-11-0-local_11.0.3-450.51.06-1_amd64.deb
$ sudo dpkg -i cuda-repo-ubuntu2004-11-0-local_11.0.3-450.51.06-1_amd64.deb
$ sudo apt-key add /var/cuda-repo-ubuntu2004-11-0-local/7fa2af80.pub
$ sudo apt-get update
$ sudo apt-get -y install cuda
```

The CUDA Toolkit contains Open-Source Software. The source code can be found [here](#).
The checksums for the installer and patches can be found in [Installer Checksums](#).
For further information, see the [Installation Guide for Linux](#) and the [CUDA Quick Start Guide](#).

ここで、最後の行を

```
sudo apt-get -y install cuda-drivers
```

とすると、最新のドライバだけがきれいにインストールされる。以下、CUDA Toolkit に記されていた通りにインストールする。

```
$ 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/11.0.3/ \
local_installers/cuda-repo-ubuntu2004-11-0-local_11.0.3-450.51.06-1_amd64.deb
$ sudo dpkg -i cuda-repo-ubuntu2004-11-0-local_11.0.3-450.51.06-1_amd64.deb
$ sudo apt-key add /var/cuda-repo-ubuntu2004-11-0-local/7fa2af80.pub
$ sudo apt update
$ sudo apt -y install cuda-drivers
```



```
kohei@DESKTOP-Q123T6P:~$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin
--2020-09-11 15:47:48-- https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin
Resolving developer.download.nvidia.com (developer.download.nvidia.com)... 152.199.39.144
Connecting to developer.download.nvidia.com (developer.download.nvidia.com)|152.199.39.144|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 190 [application/octet-stream]
Saving to: 'cuda-ubuntu2004.pin'

cuda-ubuntu2004.pin      100%[=====>]          190  --.-KB/s    in 0s
2020-09-11 15:47:48 (11.0 MB/s) - 'cuda-ubuntu2004.pin' saved [190/190]
```

```
kohei@DESKTOP-Q123T6P:~$ sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-repository-pin-600
kohei@DESKTOP-Q123T6P:~$ wget https://developer.download.nvidia.com/compute/cuda/11.0.3/local_installers/cuda-repo-ubuntu2004-11-0-local_11.0.3-450.51.06-1_amd64.deb
--2020-09-11 15:49:32-- https://developer.download.nvidia.com/compute/cuda/11.0.3/local_installers/cuda-repo-ubuntu2004-11-0-local_11.0.3-450.51.06-1_amd64.deb
Resolving developer.download.nvidia.com (developer.download.nvidia.com)... 152.199.39.144
Connecting to developer.download.nvidia.com (developer.download.nvidia.com)|152.199.39.144|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2306981744 (2.1G) [application/x-deb]
Saving to: 'cuda-repo-ubuntu2004-11-0-local_11.0.3-450.51.06-1_amd64.deb'

cuda-repo-ubuntu2004-11-0-local_11.0.3-450.51.06-1_amd64.de 100%[=====>]          2.1G  7.07MB/s    in 5m 19s
2020-09-11 15:54:52 (6.89 MB/s) - 'cuda-repo-ubuntu2004-11-0-local_11.0.3-450.51.06-1_amd64.deb' saved [2306981744/2306981744]
```

```
kohei@DESKTOP-Q123T6P:~$ sudo dpkg -i cuda-repo-ubuntu2004-11-0-local_11.0.3-450.51.06-1_amd64.deb
Selecting previously unselected package cuda-repo-ubuntu2004-11-0-local.
(Reading database ... 32175 files and directories currently installed.)
Preparing to unpack cuda-repo-ubuntu2004-11-0-local_11.0.3-450.51.06-1_amd64.deb ...
Unpacking cuda-repo-ubuntu2004-11-0-local (11.0.3-450.51.06-1) ...
Setting up cuda-repo-ubuntu2004-11-0-local (11.0.3-450.51.06-1) ...

The public CUDA GPG key does not appear to be installed.
To install the key, run this command:
sudo apt-key add /var/cuda-repo-ubuntu2004-11-0-local/7fa2af80.pub
```

```
kohei@DESKTOP-Q123T6P:~$ sudo apt-key add /var/cuda-repo-ubuntu2004-11-0-local/7fa2af80.pub
OK
kohei@DESKTOP-Q123T6P:~$ sudo apt update
Get:1 file:/var/cuda-repo-ubuntu2004-11-0-local InRelease
Ign:1 file:/var/cuda-repo-ubuntu2004-11-0-local InRelease
Get:2 file:/var/cuda-repo-ubuntu2004-11-0-local Release [564 B]
Get:2 file:/var/cuda-repo-ubuntu2004-11-0-local Release [564 B]
Get:3 file:/var/cuda-repo-ubuntu2004-11-0-local Release.gpg [836 B]
Get:3 file:/var/cuda-repo-ubuntu2004-11-0-local Release.gpg [836 B]
Get:4 file:/var/cuda-repo-ubuntu2004-11-0-local Packages [24.4 kB]
Hit:5 https://download.docker.com/linux/ubuntu focal InRelease
Hit:6 https://nvidia.github.io/libnvidia-container/experimental/ubuntu18.04/amd64 InRelease
Hit:7 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:8 http://jp.archive.ubuntu.com/ubuntu focal InRelease
Get:9 http://jp.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:10 http://jp.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Fetched 210 kB in 1s (179 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
```

今日はここまで.