WSL + docker + GPU (3)

1 Docker のインストール

Docker 公式の便利スクリプトで、最新の安定板をすぐにインストールで着る。インストール後の Docker サービスの開始と、自動起動設定をしておく。

1.1 Uninstall old versions

```
sudo apt remove docker docker-engine docker.io containerd runc
```

```
kohei@DESKTOP-Q123T6P:~$ sudo apt remove docker docker-engine docker.io containerd runc
[sudo] password for kohei:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package 'docker-engine' is not installed, so not removed
Package 'containerd' is not installed, so not removed
Package 'runc' is not installed, so not removed
Package 'docker' is not installed, so not removed
Package 'docker.io' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

1.2 SET UP THE REPOSITORY

1. 以下のコマンドで、必要なリポジトリを追加する。

```
sudo apt update

sudo apt install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg-agent \
    software-properties-common
```

```
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]
Hit:4 https://download.docker.com/linux/ubuntu focal InRelease
Hit:5 http://jp.archive.ubuntu.com/ubuntu focal InRelease
Get:6 http://jp.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:7 http://jp.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Hit:9 https://nvidia.github.io/libnvidia-container/experimental/ubuntu18.04/amd64  InRelease
Fetched 317 kB in 1s (225 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
kohei@DESKTOP-Q123T6P:~$ sudo apt install \
pt-trans>     apt-transport-https \
>     ca-certificates \
>     curl \
>     gnupg-agent \
>     software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20190110ubuntu1.1).
curl is already the newest version (7.68.0-1ubuntu2.2).
software-properties-common is already the newest version (0.98.9.2).
software-properties-common set to manually installed.
apt-transport-https is already the newest version (2.0.2ubuntu0.1).
The following NEW packages will be installed:
  gnupg-agent
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 5236 B of archives.
After this operation, 46.1 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://jp.archive.ubuntu.com/ubuntu focal/universe amd64 gnupg-agent all 2.2.19-3ubuntu2 [5236 B]
Fetched 5236 B in 0s (58.6 kB/s)
Selecting previously unselected package gnupg-agent.
(Reading database ... 93636 files and directories currently installed.)
Preparing to unpack .../gnupg-agent_2.2.19-3ubuntu2_all.deb ...
Unpacking gnupg-agent (2.2.19-3ubuntu2) ...
Setting up gnupg-agent (2.2.19-3ubuntu2) ...
```

2. Docker の official GPG key を追加する。

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
kohei@DESKTOP-Q123T6P:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
```

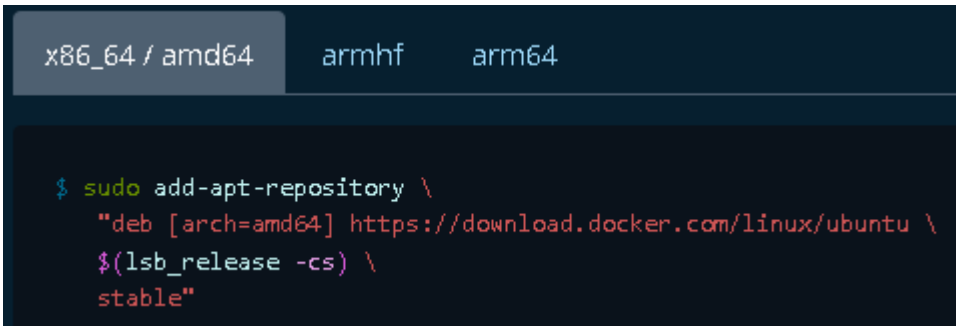
fingerprint の最後の 8 文字を検索して、手元の key が `9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88` であることを確認する。

```
sudo apt-key fingerprint 0EBFCD88
```

```
kohei@DESKTOP-Q123T6P:~$ sudo apt-key fingerprint 0EBFCD88
pub   rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid           [ unknown] Docker Release (CE deb) <docker@docker.com>
sub   rsa4096 2017-02-22 [S]
```

3. 次のコマンドを使用して、安定したリポジトリを追加する。nightly または test リポジトリを追加するには、以下のコマンドで stable という単語の後に nightly または test (または両方) という単語を追加する。

今回は、`x86_64 / amd64` を使用する。




```
sudo add-apt-repository \
   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
   $(lsb_release -cs) \
   stable"
```

```
kohei@DESKTOP-Q123T6P:~$ sudo add-apt-repository \
>    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
>    $(lsb_release -cs) \
>    stable"
[sudo] password for kohei:
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]
Hit:4 https://download.docker.com/linux/ubuntu focal InRelease
Hit:5 http://jp.archive.ubuntu.com/ubuntu focal InRelease
Get:6 http://jp.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:7 http://jp.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Hit:8 https://nvidia.github.io/libnvidia-container/experimental/ubuntu18.04/amd64  InRelease
Get:9 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Fetched 317 kB in 1s (215 kB/s)
Reading package lists... Done
```

1.3 INSTALL DOCKER ENGINE

1. apt パッケージインデックスを更新し、Docker Engine と containerd の最新バージョンをインストールするか、次の手順に進んで特定のバージョンをインストールする。

```
sudo apt update

sudo apt install docker-ce docker-ce-cli containerd.io
```

```
kohei@DESKTOP-Q123T6P:~$ sudo apt install docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
containerd.io is already the newest version (1.2.13-2).
containerd.io set to manually installed.
docker-ce-cli is already the newest version (5:19.03.12~3-0~ubuntu-focal).
docker-ce-cli set to manually installed.
docker-ce is already the newest version (5:19.03.12~3-0~ubuntu-focal).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

2. Docker Engine の特定のバージョンをインストールするには、利用可能なバージョンをリポジトリにリストしてから、選択してインストールします。

- リポジトリで利用可能なバージョンを一覧表示します。

```
apt-cache madison docker-ce
```

```
kohei@DESKTOP-Q123T6P:~$ apt-cache madison docker-ce
 docker-ce | 5:19.03.12~3-0~ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
 docker-ce | 5:19.03.11~3-0~ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
 docker-ce | 5:19.03.10~3-0~ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
 docker-ce | 5:19.03.9~3-0~ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
```

- 2 番目の列のバージョン文字列 `5:19.03.11~3-0~ubuntu-focal` を使用して、特定のバージョンをインストールする場合は、以下のコマンドの `<VERSION_STRING>` 部分をそのバージョン文字列に置き換えて使用する。

```
sudo apt-get install docker-ce=<VERSION_STRING> docker-ce-cli=<VERSION_STRING>
containerd.io
```

3. hello-world イメージを実行して、Docker エンジンが正しくインストールされていることを確認します。

```
sudo service docker stop

sudo service docker start

sudo docker run hello-world
```

```
kohei@DESKTOP-Q123T6P:~$ sudo service docker start
 * Starting Docker: docker                                                    [ OK ]
kohei@DESKTOP-Q123T6P:~$ docker run hello-world

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/

kohei@DESKTOP-Q123T6P:~$
```

4. docker サービスの開始と自動起動設定

docker サービスの開始と、自動起動設定をする。

```
curl https://get.docker.com | sh

sudo systemctl start docker && sudo systemctl enable docker
```

```
kohei@DESKTOP-Q123T6P:~$ curl https://get.docker.com | sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 13857  100 13857    0     0  44413      0 --:--:-- --:--:-- --:--:-- 44700
# Executing docker install script, commit: 26ff363bcf3b3f5a00498ac43694bf1c7d9ce16c
warning: the "docker" command appears to already exist on this system.

If you already have Docker installed, this script can cause trouble, which is
why we're displaying this warning and provide the opportunity to cancel the
installation.

If you installed the current Docker package using this script and are using it
again to update Docker, you can safely ignore this message.

You may press Ctrl+C now to abort this script.
+ sleep 20

WSL DETECTED: We recommend using Docker Desktop for Windows.
Please get Docker Desktop from https://www.docker.com/products/docker-desktop


You may press Ctrl+C now to abort this script.
+ sleep 20
+ sudo -E sh -c apt-get update -qq >/dev/null
[sudo] password for kohei:
Sorry, try again.
[sudo] password for kohei:
+ sudo -E sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/null
+ sudo -E sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | apt-key add -qq - >/dev/null
warning: apt-key output should not be parsed (stdout is not a terminal)
+ sudo -E sh -c echo "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable" > /etc/apt/sources.list.d/docker.list
+ sudo -E sh -c apt-get update -qq >/dev/null
+ [ -n  ]
+ sudo -E sh -c apt-get install -y -qq --no-install-recommends docker-ce >/dev/null
+ sudo -E sh -c docker version
Client: Docker Engine - Community
 Version:           19.03.12
 API version:       1.40
 Go version:        go1.13.10
 Git commit:        48a66213fe
 Built:             Mon Jun 22 15:45:44 2020
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.12
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.13.10
  Git commit:       48a66213fe
  Built:            Mon Jun 22 15:44:15 2020
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.2.13
  GitCommit:        7ad184331fa3e55e52b890ea95e65ba581ae3429
 runc:
  Version:          1.0.0-rc10
  GitCommit:        dc9208a3303feef5b3839f4323d9beb36df0a9dd
 docker-init:
  Version:          0.18.0
  GitCommit:        fec3683
If you would like to use Docker as a non-root user, you should now consider
adding your user to the "docker" group with something like:

  sudo usermod -aG docker kohei

Remember that you will have to log out and back in for this to take effect!

WARNING: Adding a user to the "docker" group will grant the ability to run
         containers which can be used to obtain root privileges on the
         docker host.
         Refer to https://docs.docker.com/engine/security/security/#docker-daemon-attack-surface
         for more information.
```

sudo なしに dokcer コマンドを実行可能にするため、次のように対象ユーザを dokcer グループに追加する。

実際に使用する場合は、`$USER` 部分を現在使用している `$USER` 名に変更する。

```
sudo usermod -aG docker $USER
```

1.4 NVIDIA Container Toolkit のインストール

> 2020 年 9 月 の更新により、nvidia-docker2 パッケージのインストールがとてもシンプルになった。

ここからは、nvidia の公式ドキュメントのインストールガイド に従い、NVIDIA Container Toolkit のインストールを行う。

1. 安定したリポジトリと GPG キーを設定します。

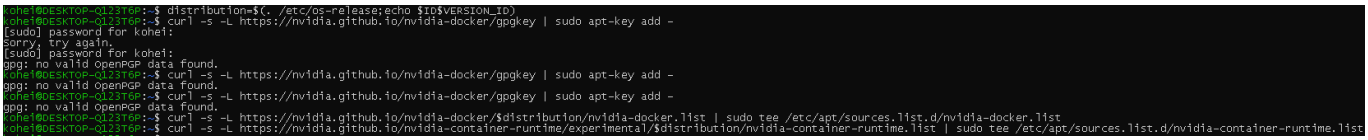
```
distribution=$(. /etc/os-release;echo $ID$VERSION_ID)
```

```
curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add -

curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list
| sudo tee /etc/apt/sources.list.d/nvidia-docker.list
```

> WSL の CUDA や A100 の新しい MIG 機能などの実験的機能にアクセスするには、実験的ブランチをリポジトリリストに追加できます。

```
curl -s -L https://nvidia.github.io/nvidia-container-
runtime/experimental/$distribution/nvidia-container-runtime.list | sudo tee
/etc/apt/sources.list.d/nvidia-container-runtime.list
```



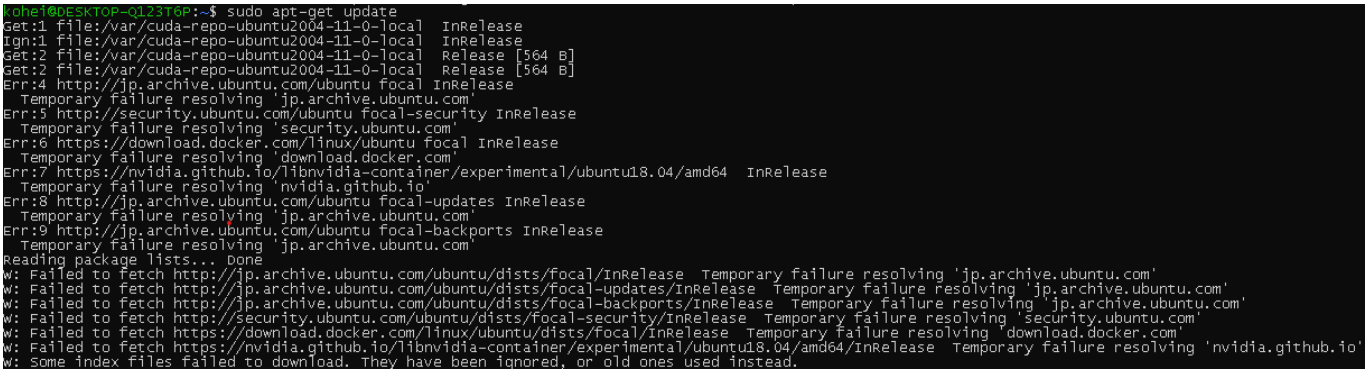
2 つ目の GPG は、追加することができなかった。

2. パッケージリストを更新した後、nvidia-docker2 パッケージ（および依存関係）をインストールします。

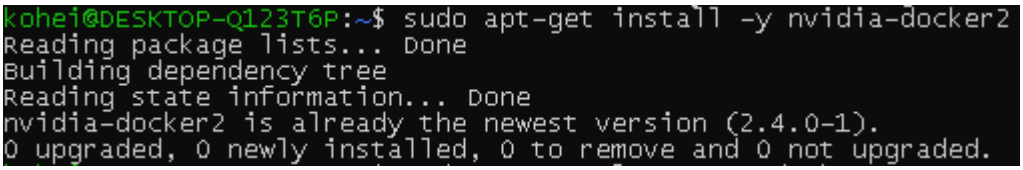
```
sudo apt-get update

sudo apt-get install -y nvidia-docker2
```

しかし、やはりリポジトリの追加が上手く行っていないのか、エラーが出た。



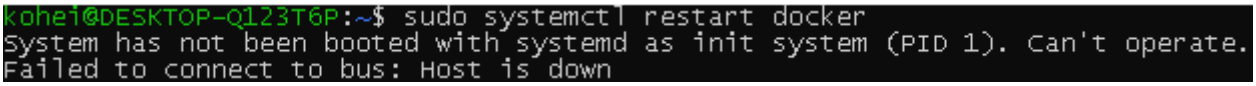
一応、`nvidia-docker2` のインストールも試みたが、こちらは一度インストールしていたため、次のような文章が表示された。



3. デフォルトのランタイムを設定した後、Docker デーモンを再起動してインストールを完了します。

```
sudo systemctl restart docker
```

これまたエラー.....



4. この時点で、ベースの CUDA コンテナーを実行することで、機能しているセットアップをテストできます。

```
sudo docker run --rm --gpus all nvidia/cuda:11.0-base nvidia-smi
```

これでインストール終了のはずだが....

`Docker stop` , `Docker start` を使用しても動作させることができなかった。

今日はここまで....