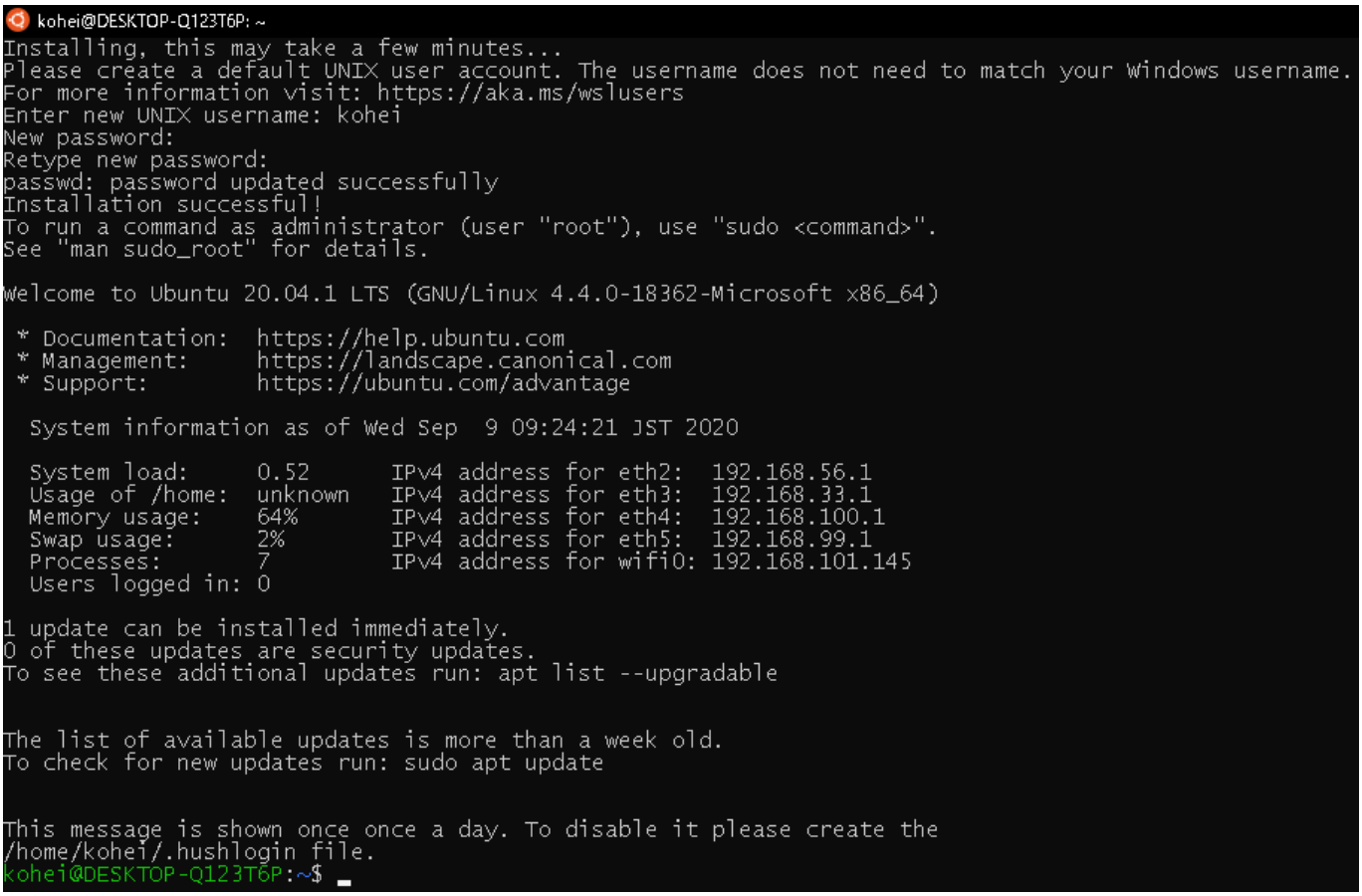
WSL + docker + GPU

1.1 Ubuntu 20.04.1 LTS をインストール

Microsoft Store より、Ubuntu をインストール



1.2 Ubuntu 初期設定

ユーザ名とパスワードを設定

パッケージアップデート

```
sudo apt update
sudo apt upgrade
```

WSL kernel update

参考：WSL2 で docker-compose を使えるようにするまで

2.1 WSL kernenl の確認

WLA2 の正しいカーネルで動作しているかを確認

```
uname -r
```

```
kohei@DESKTOP-Q123T6P:~$ uname -r
4.4.0-18362-Microsoft
```

4.19.121-microsoft-WSL2-standardと表示されれば成功らしいが.....

明らかに文字数が少ないので、少し調べてみた。

コマンドプロンプトにて、現在の Ubuntu が動作している WSL のバージョンを確認

```
wsl -l -v
```

```
PS C:\Users\rurus> wsl -l -v
  NAME      STATE           VERSION
* Ubuntu    Running         1
```

WSL のバージョンがやはり、1 だった.....

2.2 kernel update

このバージョンを 2 に変更する

```
wsl --set-version Ubuntu 2
```

```
PS C:\Users\rurus> wsl --set-version Ubuntu 2
変換中です。この処理には数分かかることがあります...
WSL 2 との主な違いについては、https://aka.ms/wsl2 を参照してください
WSL 2 を実行するには、カーネル コンポーネントの更新が必要です。詳細については https://aka.ms/wsl2kerne
l を参照してください
```

カーネルの viersion が古かったのか、その更新を要求された。

また、Ubuntu を起動した状態だと、

```
kohei@DESKTOP-Q123T6P:~$ Linux 用 Windows サブシステム インスタンスが強制終了されました。
Press any key to continue...
```

このように、Ubuntu が強制終了された。

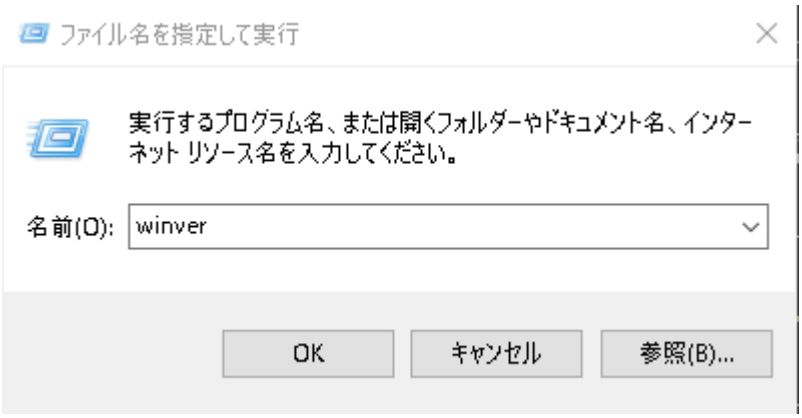
先ほどのサイトから WSL.exe のアップデートファイルをダウンロードすることができた。

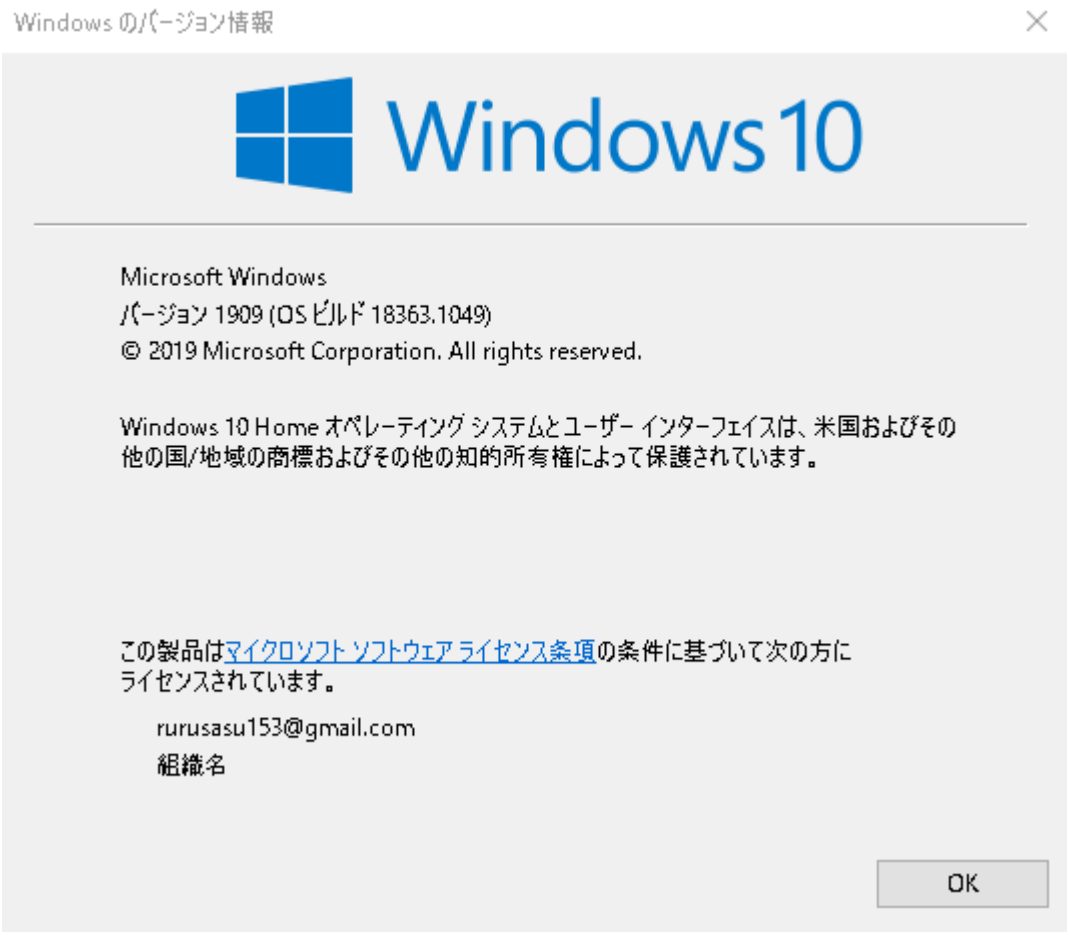
しかし、コマンドラインでアップデートを行いたかったので、今回はこちらは使用しなかった。

2.3.1 コマンドラインでの WSL2 のアップデート

参考サイト：WSL2 導入

windows のバージョンを確認





目的の「version 2004，build 19041 以上」になっていなければ WSL2 は使えないが、現在の windows の version は「version 1909 build 18363」なので、まずこちらのアップデートを行わなければならない。

2.3.2 Windows 10 のダウンロード

windows の更新プログラムでは「version 2004」にアップデートできなかったので、次のサイトから更新プログラムをダウンロードする。

https://www.microsoft.com/ja-jp/software-download/windows10

windows の versoin が 2004 になっていることを確認

再度、Ubuntu の version を確認する。

```
wsl -l -v
```

```
PS C:\Users\rurus> wsl -l -v
  NAME      STATE           VERSION
* Ubuntu    Stopped         1
```

windows を更新しただけでは、kernel version はアップデートされていなかったので、再度 versionup を試みる。

```
wsl --set-version Ubuntu 2
```

```
PS C:\Users\rurus> wsl --set-version Ubuntu 2
変換中です。この処理には数分かかることがあります...
WSL 2 との主な違いについては、https://aka.ms/wsl2 を参照してください
WSL 2 を実行するには、カーネル コンポーネントの更新が必要です。詳細については https://aka.ms/wsl2kerne
l を参照してください
```

しかし、これでも同じ結果になった。

2.3.3 WSL 2 に更新する

WSL 2 に更新するための条件を次に示す。

WSL 2 に更新する

WSL 2 に更新するには、次の条件を満たす必要があります。

- バージョン 1903 以降、**ビルド 18362** 以上に更新された x64 システム用の Windows 10 を実行している。
- バージョン 2004 以降、**ビルド 19041** に更新された ARM64 システム用の Windows 10 を実行している。
- Windows 10 バージョン 1903 または 1909 を使用している場合は、適切なバックポートがあることを確認する必要があります。手順については、こちらを参照してください。
- Windows のバージョンを確認するには **Windows ロゴ キー + R** キーを押します。次に「**winver**」と入力し、**[OK]** を選択します (または、Windows コマンド プロンプトで ver コマンドを入力します)。お使いのビルドが 18361 より前の場合は、最新の Windows バージョンに更新してください。Windows 更新アシスタントを入手する。

> WSL 2 をインストールする前に、"仮想マシンプラットフォーム" オプション機能を有効にする必要があります。
> 管理者として PowerShell を開き、以下を実行します。

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all
/norestart
```

```
PS C:¥WINDOWS¥system32> dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

展開イメージのサービスと管理ツール
バージョン: 10.0.19041.329

イメージのバージョン: 10.0.19041.508

機能を有効にしています
[==========================100.0%==========================]
操作は正常に完了しました。
```

2.3.4 WSL 2 を基底の version として設定する

> PowerShell で次のコマンドを実行して、新しい Linux ディストリビューションをインストールするときに WSL 2 を既定の verion として設定する。

```
wsl --set-default-version 2
```

ダメ....

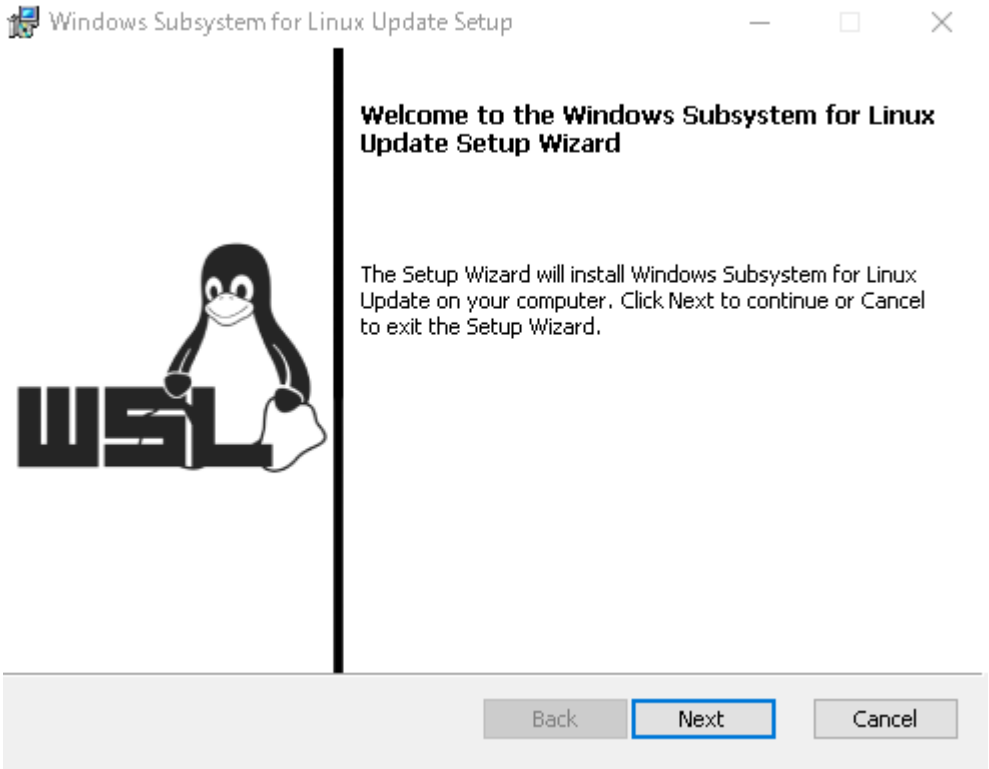
```
PS C:¥WINDOWS¥system32> wsl --set-default-version 2
WSL 2 を実行するには、カーネル コンポーネントの更新が必要です。詳細については https://aka.ms/wsl2kernel を参照してください
WSL 2 との主な違いについては、https://aka.ms/wsl2 を参照してください
```

2.4 wsl_update_x64.msi を使う

結局、wsl_update_x64.msi を使うことに。

https://aka.ms/wsl2kernelにアクセスし、`wsl_update_x64.msi`をダウンロードする。

.msi ファイルを実行すると、以下の画面が表示されるので、「Next」を選択。



すぐにアップデートが開始される。

2.4.1 再度 WSL 2 を基底の version として設定する

再度、以下のコマンドをコマンドプロンプトに打ち込み、WSL 2 が正常に交換されるか確認。

```
wsl --set-default-version 2
```

```
PS C:\Users\rurus> wsl --set-default-version 2
WSL 2 との主な違いについては、https://aka.ms/wsl2 を参照してください
```

今度はエラーが表示されず、正常に交換された模様。

しかし、デフォルトの kernel を変更しても、既に動作している WSL の Version は変わらないらしい。

```
PS C:\Users\rurus> wsl -l -v
  NAME      STATE           VERSION
* Ubuntu    Stopped         1
```

よって、下記のコマンドを再々度打ち込む

```
wsl --set-default-version 2
```

```
PS C:\Users\rurus> wsl --set-version Ubuntu 2
変換中です。この処理には数分かかることがあります...
WSL 2 との主な違いについては、https://aka.ms/wsl2 を参照してください
変換が完了しました。
```

ようやく WSL の version が 2 に変更された。

```
PS C:\Users\rurus> wsl -l -v
  NAME      STATE           VERSION
* Ubuntu    Stopped         2
```

Ubuntu 側でも、しっかりと WSL 2 で動作していることを確認

```
kohei@DESKTOP-Q123T6P:~$ uname -r
4.19.104-microsoft-standard
```

4.19.121-microsoft-WSL2-standardではなく、

4.19.104-microsoft-standardと表示されたので失敗.....

2.5 Windows Insider Program

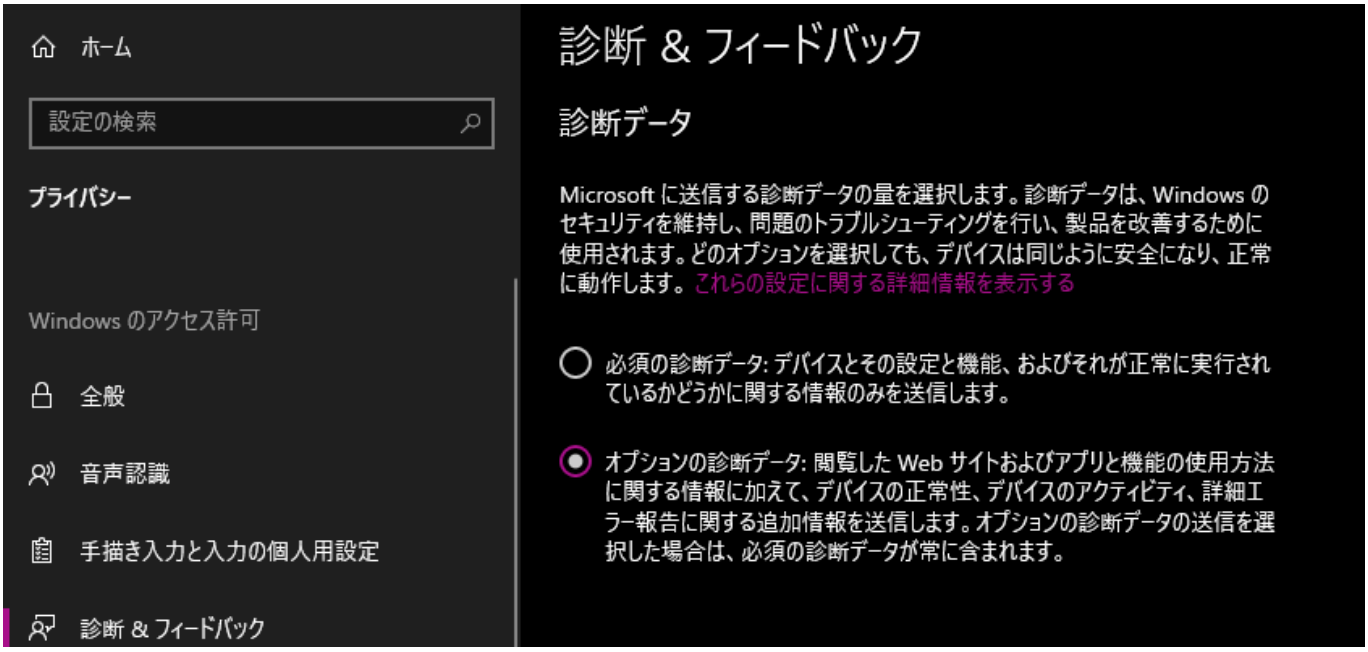
4.19.121 以上の version は、github上に存在したが、コンパイルを行う必要があったため、断念.....

そこで、windows の Insider Program に加入し、Linux krenel のアップデートを行うことにした。

診断とフィードバックの変更

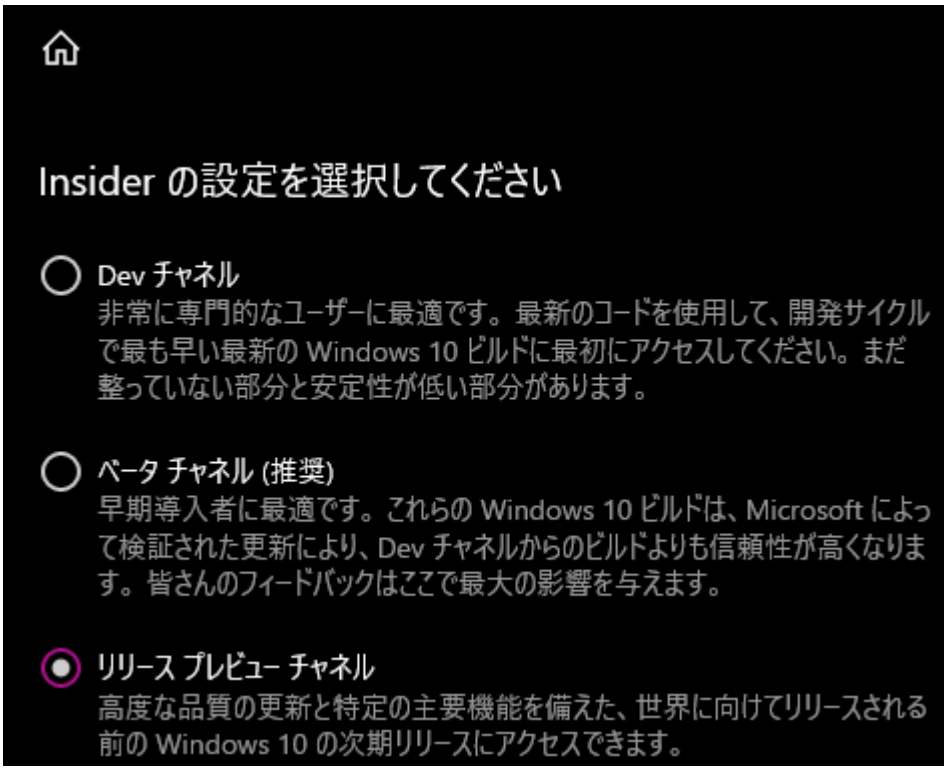
Windows Insider Program に加入するためには、診断 & フィードバックデータの取得を「完全」に設定する必要がある。

しかし、Windows のアップデートの影響なのか、「完全」の欄が消えていたため、オプションの診断データを選択。

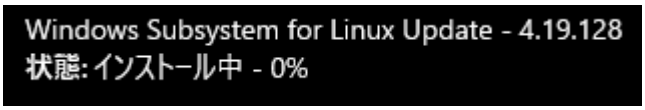


その結果、Windows Insider Program に加入することができた。

Insider Program には、3 つの選択肢があるが、今回はリリースプレビューを選択した。



その結果、無事 Linux kernel 4.19.128 をダウンロードすることができた。



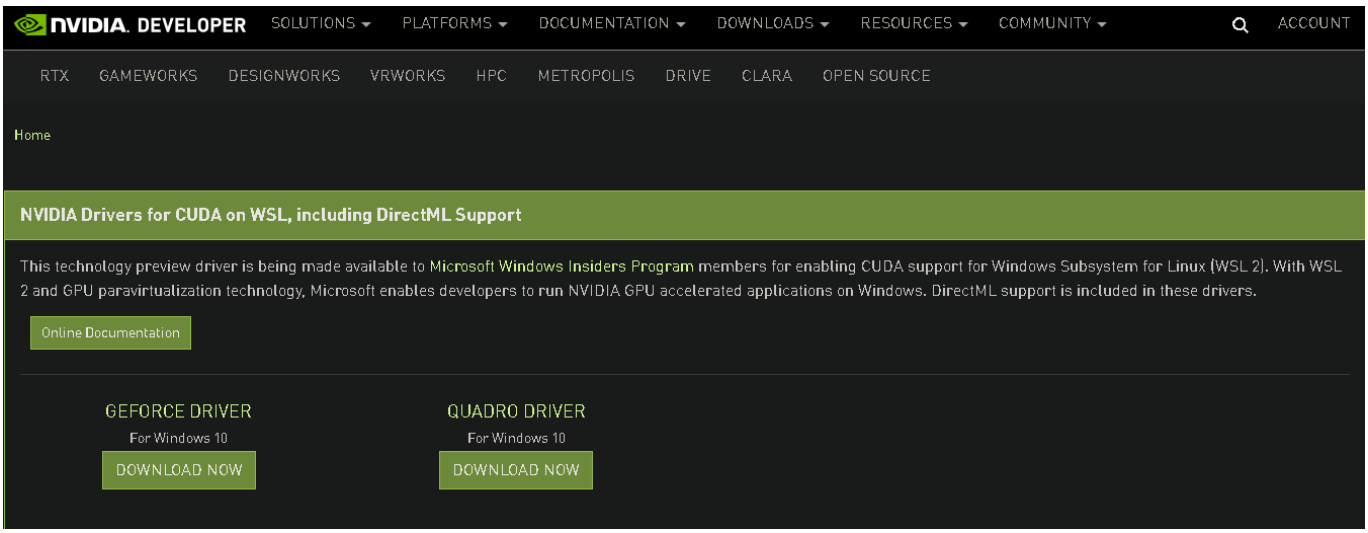
最後に、Ubuntu 上でも確認を行う。

```
kohei@DESKTOP-Q123T6P:~$ uname -r
4.19.128-microsoft-standard
```

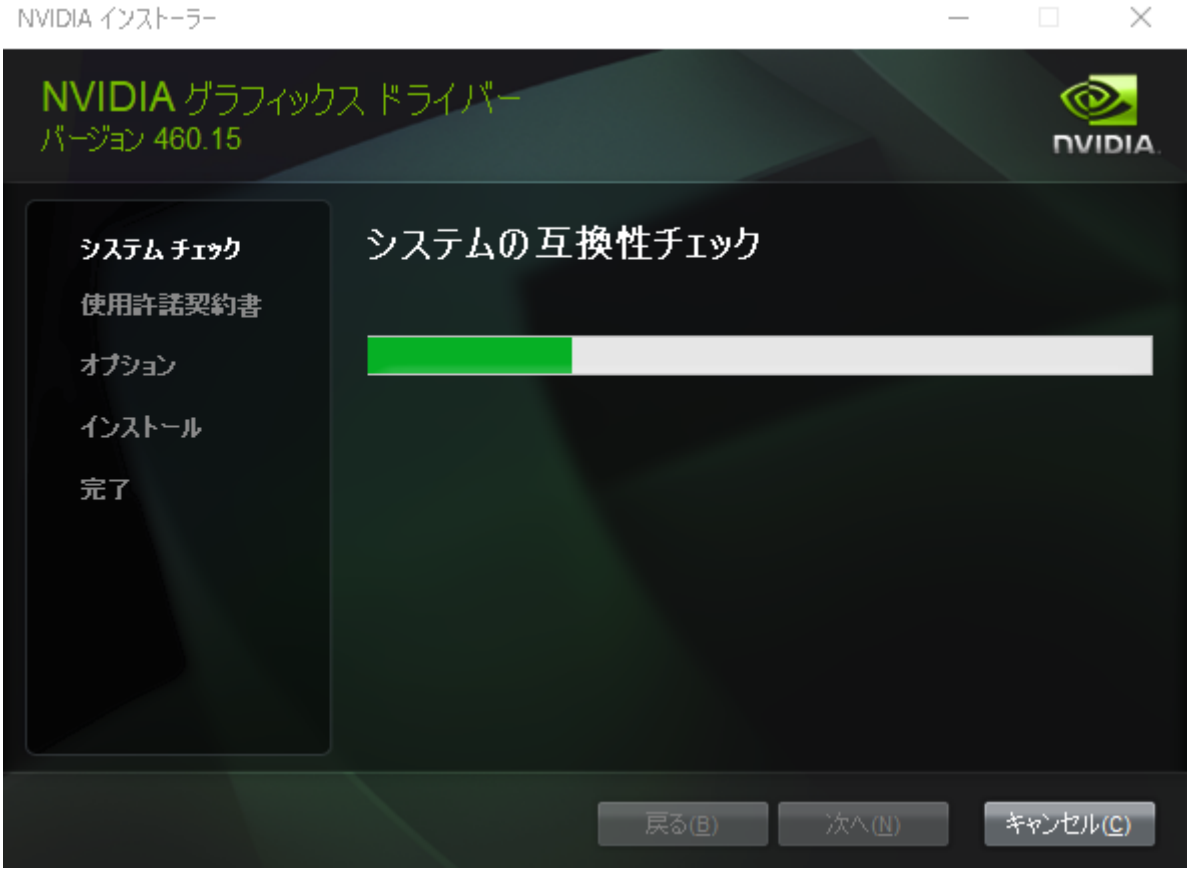
4 NVIDIA Drivers for CUDA on WSL のインストール

以下のリンクから、NVIDIA Driver をインストールする。

https://developer.nvidia.com/cuda/wsl/download



ダウンロードした exe ファイルを起動し、適当なフォルダにインストールする。



5 一端、Ubuntu の設定を再開

ダウンロードした Ubuntu にこちらの設定を行う

```
sudo sed -i 's/\/\/archive.ubuntu.com/\/\/jp.archive.ubuntu.com/g'
/etc/apt/sources.list
sudo sed -i 's/\/\/us.archive.ubuntu.com/\/\/jp.archive.ubuntu.com/g'
/etc/apt/sources.list
sudo sed -i 's/\/\/fr.archive.ubuntu.com/\/\/jp.archive.ubuntu.com/g'
/etc/apt/sources.list
```

```
sudo apt update
sudo apt -yV upgrade
```

6 Ubuntu 上に Docker をインストールする

ここまで設定が終了したら、続いて Ubuntu 上に Docker をインストールする。

コマンドライン上に次のコマンドを打ち込む。

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

```
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  96229      0 --:--:-- --:--:-- --:--:-- 96229
# Executing docker install script, commit: 26ff363bcf3b3f5a00498ac43694bf1c7d9ce16c

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
```

これが終わると、次に nvidia container toolkit をインストールする。

そのためにまず、リポジトリを追加する。

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

追加が完了すると、このように「OK」と表示される。

```
kohei@DESKTOP-Q123T6P:~$ distribution=$(. /etc/os-release;echo $ID$VERSION_ID)
curl -s -L https://nvidia.github.io/nvidia-docker/gpgkekohei@DESKTOP-Q123T6P:~$ curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add -
curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list | sudo tee /etc/apt/sources.list.d/nvidia-docker.list
curl -s -L https://nvidia.github.io/libnvidia-container/experimental/$distribution[sudo] password for kohei:
ia-container-experimental.list | sudo tee /etc/apt/sources.list.d/libnvidia-container-experimental.list
Sorry, try again.
[sudo] password for kohei:
kSorry, try again.
[sudo] password for kohei:
OK
```

※ 一度、コマンドラインに打ち込んでみたが、読み込まれていなかったのか、以下のコマンドが動作しないことがあった。

```
kohei@DESKTOP-Q123T6P:~$ sudo apt install nvidia-docker2
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package nvidia-docker2
```

その際は、再度上のコマンドをコピペすることで、動作した。

そして、Ubuntu リポジトリを更新してから、Nvidia ランタイムをインストールする。

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

インストールに成功すると、このように大量の履歴が表示される。

```
kohei@DESKTOP-Q123T6P:~$ sudo apt-get update
 apt-get install -y nvidia-docker2
Hit:1 http://jp.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://jp.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Hit:3 https://download.docker.com/linux/ubuntu focal InRelease
Get:4 http://jp.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:5 https://nvidia.github.io/libnvidia-container/experimental/ubuntu18.04/amd64  InRelease [1158 B]
Get:6 https://nvidia.github.io/libnvidia-container/stable/ubuntu18.04/amd64  InRelease [1139 B]
Get:7 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu18.04/amd64  InRelease [1136 B]
Get:8 https://nvidia.github.io/nvidia-docker/ubuntu18.04/amd64  InRelease [1129 B]
Get:9 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:10 https://nvidia.github.io/libnvidia-container/experimental/ubuntu18.04/amd64  Packages [3392 B]
Get:11 https://nvidia.github.io/libnvidia-container/stable/ubuntu18.04/amd64  Packages [8496 B]
Get:12 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu18.04/amd64  Packages [5828 B]
Get:13 https://nvidia.github.io/nvidia-docker/ubuntu18.04/amd64  Packages [4180 B]
Fetched 343 kB in 1s (255 kB/s)
Reading package lists... Done
kohei@DESKTOP-Q123T6P:~$ sudo apt-get install -y nvidia-docker2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libnvidia-container-tools libnvidia-container1 nvidia-container-runtime nvidia-container-toolkit
The following NEW packages will be installed:
  libnvidia-container-tools libnvidia-container1 nvidia-container-runtime nvidia-container-toolkit nvidia-docker2
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 1470 kB of archives.
After this operation, 4683 kB of additional disk space will be used.
Get:1 https://nvidia.github.io/libnvidia-container/experimental/ubuntu18.04/amd64  libnvidia-container1 1.3.0~rc.1-1
[66.8 kB]
Get:2 https://nvidia.github.io/libnvidia-container/experimental/ubuntu18.04/amd64  libnvidia-container-tools 1.3.0~rc
.1-1 [20.4 kB]
Get:3 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu18.04/amd64  nvidia-container-toolkit 1.2.1-1 [7
62 kB]
Get:4 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu18.04/amd64  nvidia-container-runtime 3.3.0-1 [6
15 kB]
Get:5 https://nvidia.github.io/nvidia-docker/ubuntu18.04/amd64  nvidia-docker2 2.4.0-1 [5792 B]
Fetched 1470 kB in 1s (1319 kB/s)
Selecting previously unselected package libnvidia-container1:amd64.
(Reading database ... 32145 files and directories currently installed.)
Preparing to unpack .../libnvidia-container1_1.3.0~rc.1-1_amd64.deb ...
Unpacking libnvidia-container1:amd64 (1.3.0~rc.1-1) ...
Selecting previously unselected package libnvidia-container-tools.
Preparing to unpack .../libnvidia-container-tools_1.3.0~rc.1-1_amd64.deb ...
Unpacking libnvidia-container-tools (1.3.0~rc.1-1) ...
Selecting previously unselected package nvidia-container-toolkit.
Preparing to unpack .../nvidia-container-toolkit_1.2.1-1_amd64.deb ...
Unpacking nvidia-container-toolkit (1.2.1-1) ...
Selecting previously unselected package nvidia-container-runtime.
Preparing to unpack .../nvidia-container-runtime_3.3.0-1_amd64.deb ...
Unpacking nvidia-container-runtime (3.3.0-1) ...
Selecting previously unselected package nvidia-docker2.
Preparing to unpack .../nvidia-docker2_2.4.0-1_all.deb ...
Unpacking nvidia-docker2 (2.4.0-1) ...
Setting up libnvidia-container1:amd64 (1.3.0~rc.1-1) ...
Setting up libnvidia-container-tools (1.3.0~rc.1-1) ...
Setting up nvidia-container-toolkit (1.2.1-1) ...
Setting up nvidia-container-runtime (3.3.0-1) ...
Setting up nvidia-docker2 (2.4.0-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
```

実際の実行は、apt-get で行ったが、apt でも動作すると思われる。

6.1 動作確認

先ほどインストールした nvidia-docker2 を動作させて GPU が正常に動作してるかを確認してく。

まず、タスクバー上の Ubuntu アイコンを **Shift キーを押しながらクリックすることで、新しいウインドウを立ち上げる。**

新しいウインドウにて、以下のコマンドが動作するか確認

```
sudo service docker stop
sudo service docker start
```

```
kohei@DESKTOP-Q123T6P:~$ sudo service docker stop
 * Stopping Docker: docker                                                [ OK ]
kohei@DESKTOP-Q123T6P:~$ sudo service docker start
 * Starting Docker: docker                                                [ OK ]
```

正常に動作すると、「OK」が表示される。

6.2 GPUを動かしてみる

nvidia-docker2 を動作させていく。

```
docker run --gpus all nvcr.io/nvidia/k8s/cuda-sample:nbody nbody -gpu -benchmark
```

しかし、このコマンドを新しいウインドウに打ち込むと、以下のようなエラーが出た。

どうやら、docker コマンドはデフォルトでは root 権限でないと動かないらしい(一般ユーザでは動かせない)。

6.2.1 行った対応

参考：「Got permission denied while trying to connect to the Docker daemon socket」への対応

一般ユーザを 「docker」 グループに追加する。

```
grep -i docker /etc/group
```

```
kohei@DESKTOP-Q123T6P:~$ grep -i docker /etc/group
docker:x:999:
```

```
sudo gpasswd -a username docker
```

ここで、`username` は自分がログイン時に使っている `name` に変更する必要がある。

```
kohei@DESKTOP-Q123T6P:~$ sudo gpasswd -a kohei docker
Adding user kohei to group docker
```

```
grep -i docker /etc/group
```

```
kohei@DESKTOP-Q123T6P:~$ grep -i docker /etc/group
docker:x:999:kohei
```

グループに入ったことを確認した後、一旦ログアウトを行い、再度ログインする。

docker start ・ stop には、root 権限が必要らしい。

```
kohei@DESKTOP-Q123T6P:~$ service docker stop
 * Docker must be run as root
kohei@DESKTOP-Q123T6P:~$ sudo service docker start
[sudo] password for kohei:
 * Starting Docker: docker
```

その後、ベンチマークのコンテナを run

```
kohei@DESKTOP-Q123T6P:~$ docker run --gpus all nvcr.io/nvidia/k8s/cuda-sample:nbody nbody -gpu -benchmark
Unable to find image 'nvcr.io/nvidia/k8s/cuda-sample:nbody' locally
nbody: Pulling from nvidia/k8s/cuda-sample
22dc81ace0ea: Pull complete
1a8b3c87dba3: Pull complete
91390a1c435a: Pull complete
07844b14977e: Pull complete
b78396653dae: Pull complete
95e837069dfa: Pull complete
fef4aadda783: Pull complete
343234bd5cf3: Pull complete
d1e57bfda6f0: Pull complete
c67b413dfc79: Pull complete
529d6d22ae9f: Pull complete
d3a7632db2b3: Pull complete
4a28a573fcc2: Pull complete
71a88f11fc6a: Pull complete
11019d591d86: Pull complete
10f906646436: Pull complete
9b617b771963: Pull complete
6515364916d7: Pull complete
Digest: sha256:aaca690913e7c35073df08519f437fa32d4df59a89ef1e012360fbec46524ec8
Status: Downloaded newer image for nvcr.io/nvidia/k8s/cuda-sample:nbody
docker: Error response from daemon: OCI runtime create failed: container_linux.go:349: starting container process cau
sed "process_linux.go:449: container init caused \"process_linux.go:432: running prestart hook 0 caused \\\"error run
ning hook: exit status 1, stdout: , stderr: nvidia-container-cli: initialization error: driver error: failed to proce
ss request\\\\n\\\"\"": unknown.
ERRO[0029] error waiting for container: context canceled
```

エラー.....

root 権限なしで動作はできるようになったが、別の問題が発生した模様。

今日はここまで....