

3.1. systemd を PID 1 に 続き

Ubuntu 20.04 LTS では、まず以下のコマンドを実行し、パッケージリポジトリを追加する。

```
wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
```

ここで、次のようなエラーが発生した。

```
failed: Temporary failure in name resolution.  
wget: unable to resolve host address 'packages.microsoft.com'
```

WSL2 で dns の名前解決ができなくなって ネット接続できなくなった場合の対処方法によると、このエラーは『WSL2 の DNS は WSL2 起動時に自動的に 仮想ネットワークの 172.25.208.1 等のアドレスになっている事があって、この仮想ネットワークの DNS が不調になっている場合に発生する。』らしく、『WSL2 の中でも Google の DNS サーバー 8.8.8.8 を利用するように設定する事で仮想ネットワーク上の DNS を利用しないようにする。』ことができるらしい。

3.1.1 DNS サーバを 8.8.8.8 にする.

その方法を以下に記す.

1. Ubuntu 20.04LTS を終了する.
2. コマンドプロンプトから WSL をシャットダウンさせる。

```
$ wsl --shutdown
```

3. 次に wsl にログインして 以下のコマンドを実し, vi を立ち上げる(vi とは Linux に標準搭載されているテキストエディタ『[vi コマンドについて詳しくまとめました](#)』).

```
$ sudo vi /etc/wsl.conf
```

4. vi に以下の手順で文字を入力する.
 - i を押す.
 - カーソル部分に以下のコマンドを貼り付け

```
[network]
generateResolvConf = false
```

- Esc ボタンを押して入力モードを終了
- :wq コマンドを入力して、ファイルに変更を保存して vi を閉じる.

5. 最後に `/etc/resolv.conf` ファイルを削除して DNS の設定を登録する。そのために以下のコマンドを実行する。

```
$ sudo rm /etc/resolv.conf
$ sudo sh -c "echo 'nameserver 8.8.8.8' > /etc/resolv.conf"
```

これで DNS 設定が変更されたはず。この処理は、再起動をすることなく有効となっているらしいので、改めて以下のコマンドを実行してみる。

```
$ wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
$ sudo dpkg -i packages-microsoft-prod.deb
```

その結果、

```
$ wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
--2021-04-13 09:52:26-- https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb
Resolving packages.microsoft.com (packages.microsoft.com)... 23.99.120.248
Connecting to packages.microsoft.com (packages.microsoft.com)|23.99.120.248|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3124 (3.1K) [application/octet-stream]
Saving to: 'packages-microsoft-prod.deb'

packages-microsoft- 100%[=====>] 3.05K --.-KB/s in 0s

2021-04-13 09:52:27 (185 MB/s) - 'packages-microsoft-prod.deb' saved [3124/3124]
```

と表示され、無事成功したみたいだった。

3.1.2. SDK をインストール

これまでの作業で, microsoft のパッケージリポジトリの署名キーが信頼できるキーのリストに追加されたので, 次は `.NET SDK` をインストールしていく.

以下のコマンドを実行する.

```
$ sudo apt-get update
$ sudo apt-get install -y apt-transport-https
$ sudo apt-get update
$ sudo apt-get install -y dotnet-sdk-5.0
```

その結果,

```
Welcome to .NET!
```

と表示されたので, これで `dotnet-runtime-5.0` のインストールは完了.

3.1.3 残りのパッケージをインストール

`dotnet-runtime-5.0` 以外のパッケージについては、ほとんどが `Ubuntu` のパッケージ管理システム上に存在することだったので、まとめてインストールする。

```
sudo apt -y install \  
daemonize \  
dbus \  
dotnet-runtime-5.0 \  
gawk \  
libc6 \  
libstdc++6 \  
policykit-1 \  
systemd
```

さらに、<https://arkane-systems.github.io/wsl-transdebian/>の手順に従って、`wsl-translinux` リポジトリを追加する。

それによると、まず次のコマンドを実行していく。

```
$ sudo -s apt install apt-transport-https  
$ sudo -s wget -O /etc/apt/trusted.gpg.d/wsl-transdebian.gpg \  
https://arkane-systems.github.io/wsl-transdebian/apt/wsl-transdebian.gpg  
$ sudo -s chmod a+r /etc/apt/trusted.gpg.d/wsl-transdebian.gpg  
$ sudo -s cat << EOF > /etc/apt/sources.list.d/wsl-transdebian.list  
cat << EOF > /etc/apt/sources.list.d/wsl-transdebian.list  
deb https://arkane-systems.github.io/wsl-transdebian/apt/ <distro> main  
deb-src https://arkane-systems.github.io/wsl-transdebian/apt/ <distro> main  
EOF  
  
$ sudo -s apt update
```

上記のコマンドを実行して、エラーが発生しなければ、ようやく次のコマンドを使用して `genie` をインストールできる。

```
$ sudo apt install -y systemd-genie
```

しかし、ここで以下のようなエラーが発生した。

```
E: Unable to locate package systemd-genie
```

インストールしようとしているパッケージの提供先の指示によると、『packagecloud.io リポジトリは非推奨になりました。まだ使用している場合は、上記のリポジトリを使用するようにシステムを更新してください。』とのことだが、[apt : E : パッケージ systemd-genie 見つかりません](#) の掲示板によると、同様のエラーが発生した場合、非推奨のリポジトリをしようするか、[パスクローニングエディション](#)を使用するかの2つの方法が記載されていた。

そこで、まず非推奨のパッケージからインストールできないかを試してみる。その方法は簡単で、以下の2つのコマンドを実行する。

```
$ curl -s https://packagecloud.io/install/repositories/arkane-systems/wsl-translinux/script.deb.sh | sudo bash  
$ sudo apt install -y systemd-genie
```

しかし、`packagecloud.io` 上のリポジトリは削除されており、インストールすることができなかった。

原因判明！！

インストールしようとしているパッケージの提供先にリンクが貼られている, <https://arkane-systems.github.io/wsl-transdebian/>を読んでいるとき,

`distro` をディストリビューションコードネームに置き換えます。

を実行していないことに気が付いた！！

そこで, 先ほどの

```
$ sudo -s cat << EOF > /etc/apt/sources.list.d/wsl-transdebian.list
cat << EOF > /etc/apt/sources.list.d/wsl-transdebian.list
deb https://arkane-systems.github.io/wsl-transdebian/apt/ <distro> main
deb-src https://arkane-systems.github.io/wsl-transdebian/apt/ <distro> main
EOF
```

で作成していたファイルを, `vi` を用いて次のように変更する. ここで, 使用している `Ubuntu` のディストリビューションコードネームを確認するには以下のコマンドを実行する.

```
$ lsb_release -c
```

その結果,

```
$ Codename: focal
```

と表示された. この情報をもとに先ほどのファイルを変更していく.

次に、先ほど作成したファイルを `vi` で開く.

```
$ sudo vi /etc/apt/sources.list.d/wsl-transdebian.list
```

`vi` 上で

```
$ deb https://arkane-systems.github.io/wsl-transdebian/apt/ focal main  
$ deb-src https://arkane-systems.github.io/wsl-transdebian/apt/ focal main
```

を記述する.

最後に,

```
sudo apt install -y systemd-genie
```

を実行する.

その結果、無事に `systemd-genie` をインストールすることができた.

3.1.4. Systemd genie 起動

まず、『[systemctl が動かない問題をきちんと解決する](#)』で書かれているように、以下のコマンドで `genie` を実行する。

```
genie -s
```

実行した結果、

```
Waiting for systemd....!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Timed out waiting for systemd to enter running state.
This may indicate a systemd configuration error.
Attempting to continue.
```

とタイムアウトが発生した。これは、[インストールしようとしているパッケージの提供先に『WARNING: TIMING OUT』](#)として対策方法が紹介されていた。

以下, その部分の翻訳.

ブロックされているようであれば、タイムアウト(デフォルトでは 180 秒)まで待って、`systemctl status` を実行してください。この問題の一般的な原因は、1 つ以上の `systemd` ユニツトが正しく起動していないことです。その場合、表示される `systemd` の状態は「degrade」、または「running」以外の別の状態になります。この場合、`systemd` の状態が「degrade」または「running」以外の状態で表示されます。この場合、それらのユニットを修正または無効にして `systemd` が正しく起動できるようにすると、`genie` も正しく起動できるようになります。

そこで, `systemctl status` を実行してみる.

```
$ systemctl status
● DESKTOP-Q123T6P-wsl
   State: degraded
   Jobs: 0 queued
  Failed: 2 units
   Since: Tue 2021-04-13 12:25:25 JST; 6min ago
  CGroup: /
          └─user.slice
              └─user-1000.slice
                  └─user@1000.service
                      └─init.scope
                          └─1051 /lib/systemd/systemd --user
                              └─1052 (sd-pam)
                  └─session-c1.scope
                      └─1049 /bin/bash -l
                          └─1063 (sd-pam)
                              └─1081 systemctl status
                                  └─1082 pager
          └─init.scope
              └─1 systemd
```

※ このコマンドを実行して表示されるエディタは, `:q` を押すことで閉じられる.

確かに、`State: degraded` , `Failed: 2 units` と出力されている。そこで、この状態を解決する方法を探す。そこで見つけた `systemctl status` で "degraded" と言われた場合の対処方法 のサイトを参考に、以下のコマンドを実行する。

```
$ systemctl --failed
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
● systemd-machined.service         loaded failed failed Virtual Machine and Container Registration Service
● systemd-remount-fs.service       loaded failed failed Remount Root and Kernel File Systems
● multipathd.socket                loaded failed failed multipathd control socket

LOAD    = Reflects whether the unit definition was properly loaded.
ACTIVE  = The high-level unit activation state, i.e. generalization of SUB.
SUB     = The low-level unit activation state, values depend on unit type.

3 loaded units listed.
```

`systemctl status` shows : "State: degraded"によると、この状態を正常に戻すには、`journalctl -xe` で出力されるログを見る必要があるようである。しかし、実際にログ表示を見ても直せる気がしなかった。そこで、『`systemctl` ステータスは、「State : degraded」を表示します』に記載されていたように、以下のコマンドを使用することで、強引にエラーを取り除く。

```
sudo systemctl reset-failed
```

再度, `systemctl --failed` を実行すると `0 loaded units listed.` と表示され, 実際にエラーが取り除かれていた. そこで, 再度 `genie -s` を実行すると, 何もエラーが発生しなかった.

では, 以下のコマンドを実行して実際に `systemd` が PID 1 で動作しているかを確認する.

```
$ ps aux
```

その結果, しっかりと一番上に `systemd` の文字があることが確認された！！！！！！

```
$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  58.9   0.0 175064 12936 ?        Ss   12:25   164:58 systemd
```

やった！！！！

3.2. Docker のインストール

これで `systemd` を用いたコマンドが使えるようになったはずなので、再度『[NVIDIA Docker って今どうなってるの？ \(20.09 版\)](#)』に戻り、`Docker` のインストールを行う。

まず、以下のコマンドを実行し、Docker をインストールする。

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

次に、前回躓いた `systemctl` コマンドを使用する。

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

前回の実行結果が、`Failed` であったのに対し、今回は、

```
Executing: /lib/systemd/systemd-sysv-install enable docker
```

と、正常に動作しているようである。続いて、以下のコマンドを使用して `sudo` なしに `docker` コマンドを実行可能にする。

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

このコマンドによって、ユーザが docker グループに加入していることを `id` コマンドを用いて確認する『[ユーザーのユーザーIDやグループIDを調べるには](#)』。

```
$ id miki
```

その結果,

```
uid=1000(miki)
gid=1000(miki)
groups=1000(miki),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),
30(dip),44(video),46(plugdev),117(netdev),998(docker)
```

`groups` の最後に (docker) が追加されたことが確認できた。

4. NVIDIA ContainerToolkitのセットアップ

まず、リポジトリと GPG キーを設定する。

```
$ 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
```

続いて、 `nvidia-docker2` をインストールする。

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

最後に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する。

```
$ sudo systemctl restart docker
```

5. CUDA コンテナの起動確認

以下のコマンドを使用することで、基本的な CUDA コンテナを実行し、動作セットアップをテストすることができる。

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

実行した結果、

```
docker: Error response from daemon:
OCI runtime create failed: container_linux.go:367:
starting container process caused: process_linux.go:495:
container init caused: Running hook #0::
error running hook: exit status 1,
stdout: , stderr: nvidia-container-cli:
initialization error: driver error:
failed to process request: unknown.
```

エラーを直接検索ボックスに打ち込んで検索したところ、『[OCIランタイムの作成に失敗しました : container_linux.go : 346 : コンテナプロセスを開始すると「process_linux.go : 449 : 」が発生しました](#)』に

docker system prune後、dockerサービス再起動

と書かれていた。そこで、まず

```
$ docker system prune
```

を実行してみる。その結果、

```
Got permission denied while trying to connect to the Docker daemon socket at unix:
///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/prune:
dial unix /var/run/docker.sock: connect: permission denied
```

というエラーが発生した。

これについて調べたところ、『[docker pull時に dial unix /var/run/docker.sock: connect: permission denied が出る場合の対処法](#)』ということで現在のユーザが `docker` グループに参加していないことが問題であると書かれていた。しかし、`groups` には `(docker)` が存在するので追加はできているはず....

もう一度、`genie` の起動から確認していく。

1. `ps aux` を先に実行していようと、していまいと `wsl --shutdown` を実行すると `sudo systemctl reset-failed` を実行する必要がある。