

## 参考

<https://qiita.com/snaka/items/a8eee4cfc8f7d733e6ab>  
<https://www.atmarkit.co.jp/ait/articles/1904/05/news027.html>  
<https://www.atmarkit.co.jp/ait/articles/1806/28/news043.html>  
<https://qiita.com/yoichiwo7/items/e3e13b6fe2f32c4c6120>  
<https://blog.mohyo.net/2022/02/11591/>

## 初期設定

- ・ BIOS で VT-x を有効にする
- ・ Linux 用 Windows サブシステムと仮想マシンプラットフォームをインストール
  - ・ Windows の機能の有効化または無効化から以下をチェックして OK
    - ・ Linux 用 Windows サブシステム
    - ・ 仮想マシンプラットフォームをインストール
- ・ 場合によってはカーネルコンポーネントの更新が必要
  - ・ <https://docs.microsoft.com/ja-jp/windows/wsl/install-win10#step-4---download-the-linux-kernel-update-package>

## ディレクトリビューションを手動でインストール

通常は Microsoft Store で利用したいディストリビューションをインストールする。  
Microsoft Store を使わない場合は、以下の URL からダウンロードする

<https://docs.microsoft.com/ja-jp/windows/wsl/install-manual>

PowerShell

```
Add-AppxPackage Ubuntu-20.04.appx
```

とか

## ディストリビューションの設定

1. localhost で WSL に接続できるようにする
  1. c:\Users\< ユーザ名 >\.wslconfig を作成して以下の内容を記載する
    1. localhostForwarding=True
    2. [wsl2]
    3. memory=6GB
    4. WSL のメモリの上限を設定できる。物理メモリの 50% ~ 80% くらいがおすすめ
2. locale の設定
  1. sudo apt -y install language-pack-ja
  2. sudo update-locale LANG=ja\_JP.UTF8
3. timezone の設定
  1. sudo dpkg-reconfigure tzdata
4. ssh server の設定
  1. sudo apt install openssh-server
  2. sudo ssh-keygen -A
  3. sudo vim /etc/ssh/sshd\_config
    1. PasswordAuthentication yes に変更
5. docker-ce の設定
  1. Add Docker's official GPG key:
    1. sudo apt-get update

2. `sudo apt-get install ca-certificates curl gnupg`
3. `sudo install -m 0755 -d /etc/apt/keyrings`
4. `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg`
5. `sudo chmod a+r /etc/apt/keyrings/docker.gpg`
2. Add the repository to Apt sources:
  1. `echo \`
  2. `"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]`  
`https://download.docker.com/linux/ubuntu \`
  3. `$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \`
  4. `sudo tee /etc/apt/sources.list.d/docker.list > /dev/null`
  5. `sudo apt-get update`
3. `sudo apt install docker-ce docker-compose-plugin`
4. `sudo gpasswd -a ユーザ名 docker`
6. 起動用 bat
  1. `wsl -u root -- service ssh start`
  2. `wsl -u root -- service docker start`
7. `git-prompt.sh`
  1. `sudo mkdir /usr/share/git-core/contrib/completion/`
  2. `sudo curl -o /usr/share/git-core/contrib/completion/git-prompt.sh`  
`https://raw.githubusercontent.com/git/git/master/contrib/completion/git-prompt.sh`
8. XServer 等で画面転送する場合

## もし GUI を使う場合

### x11 のキーレイアウト

もしレイアウトがおかしい場合は以下で変更できる。

```
setxkbmap -layout jp
```

## wslconfig

Windows のホームディレクトリ直下に

```
.wslconfig
```

を作成すると wsl のメモリ制限などの設定をすることができる

### メモリを制限する

```
[wsl2]
memory=8GB
```

### WSL が確保したメモリを自動的に開放する

```
[experimental]
autoMemoryReclaim=dropcache
```

指定できる値は

- gradual : アイドル時に段階的に開放していく
- dropcache : 未使用分を一括で開放する
- disabled : 自動メモリ解放をしない

## 自動ディスクスペースクリーンアップ

```
[experimental]
sparseVhd=true
```

でディスクの未使用領域を開放する。

エクスプローラでファイルのプロパティで確認すると、ディスク上のサイズが小さくなる。

既存のディスクイメージには効かないので、既存のディスクには以下のコマンドすることで有効になる

```
wsl.exe --manage <ディストリビューション名> --set-sparse
```

## WSLg を docker で使う

docker の設定に以下を指定することで WSL の docker で wslg が使える

```
environment:
  - DISPLAY=$DISPLAY
  - WAYLAND_DISPLAY=$WAYLAND_DISPLAY
  - XDG_RUNTIME_DIR=$XDG_RUNTIME_DIR
  - PULSE_SERVER=$PULSE_SERVER
volumes:
  - /tmp/.X11-unix:/tmp/.X11-unix
  - /mnt/wslg:/mnt/wslg
```

## Windows 上のサービス

WSL のためのサービスは以下のもの

```
LxssManager (WSLService に変わった )
WSLService
```

## デフォルトユーザの変更

wsl を起動したときのデフォルトユーザを変更する方法

### 方法 1

wsl の仮想マシンに以下のファイルを作成する

```
/etc/wsl.conf
```

```
[user]
default=user-name
```

仮想マシンを落落とす

```
wsl -t 仮想マシン名
```

### 方法 2

仮想マシンが一つだけの場合

PowerShell で以下を実行する

```
ubuntu config --default-user xxxx
```

これを行うことで以下のレジストリが書き換わる

```
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Lxss\{distro-id}\DefaultUid
```

## VPN 環境で通信ができない場合

### SSL 証明書の問題の場合

#### SASE 環境で docker を使う場合の証明書問題

#### 一部のサイトに通信できない場合

ホストの VPN 経由のみ通信できる場合は、ホストに proxy を立てて WSL、docker コンテナからその proxy を経由させることで通信できる。

#### Windows で Proxy サーバを立てる

で proxy を立てて、WSL で以下を実行する。

```
export http_proxy=http://$(cat /etc/resolv.conf | grep nameserver | awk '{print $2}'):8081
export https_proxy=${http_proxy}
```

また各コンテナは docker-compose.yml, dockerfile で proxy の設定を行う

docker-compose.yml

```
build:
  context: .
  dockerfile: ./environment/php-apache/Dockerfile
  args:
    - http_proxy=${http_proxy}
    - https_proxy=${https_proxy}
```

dockerfile

```
ARG http_proxy
ARG https_proxy
ENV http_proxy ${http_proxy}
ENV https_proxy ${https_proxy}
```

## 以下自分用メモ

```
sudo apt update
sudo apt -y install language-pack-ja
sudo update-locale LANG=ja_JP.UTF8
sudo dpkg-reconfigure tzdata
sudo apt install openssh-server
sudo ssh-keygen -A
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release
-cs) stable"
sudo apt update
sudo apt install docker-ce docker-compose
sudo gpasswd -a user docker
sudo vim /etc/wsl.conf
sudo vim /etc/ssh/sshd_config
パスワード認証許可
sudo apt install tmux
sudo apt install tig git
sudo apt install meld
sudo apt install make
mkdir temp
```

```

cd temp
cp /mnt/c/data/temp/Download/Linux.zip ./
sudo apt install unzip
unzip Linux.zip
cp -rT Linux /
cd
ls
vim /.tmux.conf
cd bin
chmod +x *
sudo apt install peco
sudo apt install xclip
sudo apt install fonts-vlgothic
sudo apt install fonts-ricty-diminished
sudo apt install gnome-icon-theme
sudo apt install gnome-shell

```

## systemd

個人的には systemd は不要なので止めて良い

vim /etc/wsl.conf

```
systemd=true
```

はコメントアウト

## docker

docker に関してはオフィシャルの方法を使う

<https://docs.docker.com/engine/install/ubuntu/>

```

# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg

# Add the repository to Apt sources:
echo ¥
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu ¥
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | ¥
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

```