

参考

sudo dpkg -i docker-ce_17.09.0~ce-0~debian_amd64.deb

<https://www.glamenv-septzen.net/view/1422>

<https://i.loveruby.net/ja/rhg/cd/win.html>

概要

--	WSL2	WSL	VirtualBox	Cygwin	Msys2
動作条件	1903以降、ビルド18362以上	Windows10なら動くと思う	あまり気にしなくていい	あまり気にしなくていい	あまり気にしなくていい
Linuxカーネル	あり	なし	あり(LinuxOSを使えば)	なし	なし
Dockerの実行	可能	動くけど不具合が多い	可能	不可	不可
ローカルサービスへのポートフォワードの必要性	不要 .wslconfig に localhostForwarding=True	不要	不要ホストネットワークのNICを追加して別IPを降る	不要	不要
バックアップ(スナップショット)	export、importが可能	export、importが可能	スナップショットもバックアップも可能	ディレクトリ丸ごとコピー	ディレクトリ丸ごとコピー
Windows実行体のコンパイル	MinGW	MinGW	MinGW	gcc、MinGW	MinGW
おすすめ	動作条件を満たすならこれが一番いい	Xdockerがまともにも動作しないのでおすすめしない	WSL2より起動は遅いけど何かと使いやすい	sshクライアントとかちょっとコンパイルするだけならあり	sshクライアントとかちょっとコンパイルするだけならあり

Windows上でUnix環境を利用するには大きく2つの方法がある。

- ・ Windows上でLinuxを動作させる
- ・ Windows上でUnix互換環境(UnixのコマンドをWindows用に移植したもの)を使う

個人的にはWSL2が使い勝手が良くて、動作の安定性も高いと思う。

Unix 仮想環境

WSL2

WSL2の設定

WSL

WSL単独でのdockerの動作に問題が多々ある。hello-world程度の簡単なコンテナであれば動作可能。

設定手順

1. WSL 起動
 1. 管理者権限でコマンドプロンプトを実行 (wsl は管理者権限を使わないと docker サービスが起動しない)
 2. wsl -d Ubuntu-20.04 -u user
2. apt アップデート
 1. sudo apt update
3. docker-ce の設定 (最新版だと正常に動作しないので古いものをダウンロードする)
 1. curl -O https://download.docker.com/linux/debian/dists/stretch/pool/stable/amd64/docker-ce_17.09.0~ce-0~debian_amd64.deb
 2. sudo dpkg -i docker-ce_17.09.0~ce-0~debian_amd64.deb
4. docker グループに追加設定
 1. sudo gpasswd -a user docker
 2. ログインしなおす (id で docker グループが追加されていれば OK)
5. サービス起動と hello-world 実行
 1. sudo service docker start
 2. docker run --rm hello-world

Virtualbox

WSL2 が使えないときはこちらの方法がおすすめ。

OS のインストールが面倒なときは <https://www.osboxes.org/> からイメージをダウンロードする。

Ubuntu Server

1. キーボードの設定
 1. sudo dpkg-reconfigure keyboard-configuration
2. SSH を使う場合
 1. ホストオンリーアダプタを追加して、ホスト OS とゲスト OS を通信できるようにする必要がある
 1. ゲスト OS の設定画面 -> ネットワーク -> アダプタ 2 -> 割当 をホストオンリーアダプタにする
 2. 起動後にアダプタ 2 に DHCP を設定する
 1. sudo vim /etc/netplan/00-installer-config.yaml

```
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: true
```

1. 上記のように追加したデバイスの dhcp を有効にする

1. docker の設定
 1. curl -fsSL <https://download.docker.com/linux/ubuntu/gpg> | sudo apt-key add -
 2. sudo add-apt-repository "deb [arch=amd64] <https://download.docker.com/linux/ubuntu> \$(lsb_release -cs) stable"
 3. sudo apt update
 4. sudo apt install docker-ce docker-compose
 5. sudo gpasswd -a ユーザ名 docker
 6. 追加したグループを反映させるためにログオフして、ログインしなおし
2. hello-world 実行
 1. docker run --rm hello-world

Unix 互換環境

Windows 上に Unix 互換環境を構築するには、Msys2 や Cygwin を使う必要がある。

Unix と Windows ではパスの区切文字 (\ と /) やデバイスの指定方法 (c:\Users と /home) が異なるため、Unix 互換ツールを利用するには、その違いを意識せず利用するための仕組み (POSIX エミュレーション) が必要となる。

ツール	説明
Cygwin	Unix 互換環境として優秀。パッケージも多く、X 関連も動作させることができる。
Msys2	Cygwin から派生し、必要なツールを必要最低限使えるようにして、Mingw の利用を目的として作られた。

MinGW

Windows の GCC コンパイラ。POSIX エミュレーションはなく、普通の Windows アプリケーションであり、パスの指定も通常のパス (c:\data\src\test.c) が使える。

Cygwin や Msys2 と一緒に使われることが多いが、Cygwin や Msys2 にも gcc が存在するので区別して利用する必要がある。

例えば、Cygwin の gcc でコンパイルした exe の実行には、Cygwin1.dll が必要。

同様に Msys2 の gcc でコンパイルした exe の実行には、msys-2.0.dll が必要。

MinGW の gcc でコンパイルした exe の実行は、そのまま実行できるし、内部で利用するパスも Windows のパス指定が利用できる。

使い分け

個人的には、

ツール	用途
Cygwin	Windows 上にがっつり Unix 互換環境を構築する。または Msys2 で動かなかったものを動かしたい。(WSL や仮想マシンでもいい気がするけど・・・)
Msys2	X とかは必要なく、CUI の Unix 互換環境を構築したい。MinGW で Windows 上に gcc の環境が欲しい。

環境設定

Cygwin

Cygwin の設定

Cygwin はアーカイブを展開後すぐに使える Cygwin 環境。ただし、2018 年以降更新されていない。

gnupack の設定

Msys2

MSYS2 Portable の環境を作成する