

「つくりながら学ぶ! Python セキュリティプログラミング」

— Docker Desktop for Windows での環境構築

2021/06/25

手順

1. Docker Desktop for Windows のインストール ※1
2. Dockerfile を使い、書籍で使う Linux コンテナをビルド (書籍 12p と同じ)
3. コンテナの起動

※1 Docker Desktop for Windows のインストールについては公式 HP

<https://docs.docker.jp/docker-for-windows/install-windows-home.html>

に手順が載っているため、Linux コンテナのビルドと起動手順の解説のみ必要とされる場合は、手順 2. 以降を確認してください。

1. Docker Desktop for Windows のインストール

Docker Desktop は WSL2 をバックエンドとして使うため、WSL2 をインストールしてから Docker Desktop をインストールします。WSL2 のインストール手順は以下の URL に書いてあります。

<https://docs.docker.jp/docker-for-windows/wsl.html>

簡単にまとめると、

1. WSL2 の有効化(管理者権限で起動したコマンドプロンプトで 2 つコマンドを実行)
2. Linux カーネル更新プログラムパッケージをダウンロード & インストール

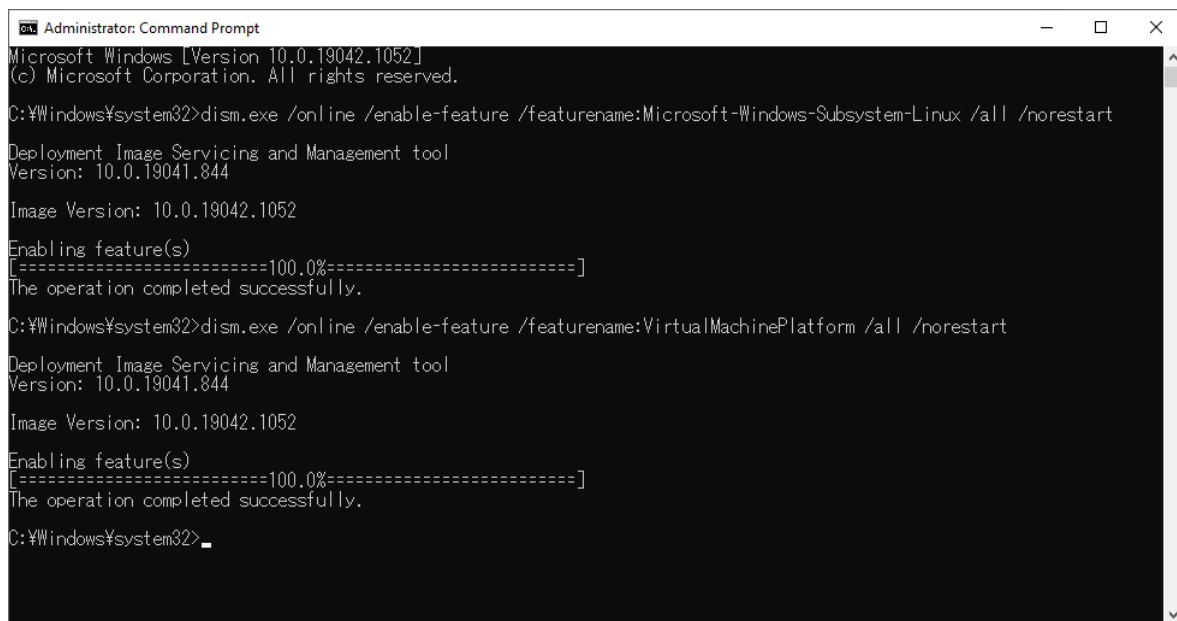
1.1 WSL2 の有効化

管理者権限でコマンドプロンプトを起動したら、以下の 2 つのコマンドを実行します。

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux  
/all /norestart
```

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

参考までに、上記 2 つのコマンドを実行したときのスクリーンショットは以下のようになります。



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19042.1052]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart

Deployment Image Servicing and Management tool
Version: 10.0.19041.844

Image Version: 10.0.19042.1052

Enabling feature(s)
[=====100.0%=====]
The operation completed successfully.

C:\Windows\system32>dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

Deployment Image Servicing and Management tool
Version: 10.0.19041.844

Image Version: 10.0.19042.1052

Enabling feature(s)
[=====100.0%=====]
The operation completed successfully.

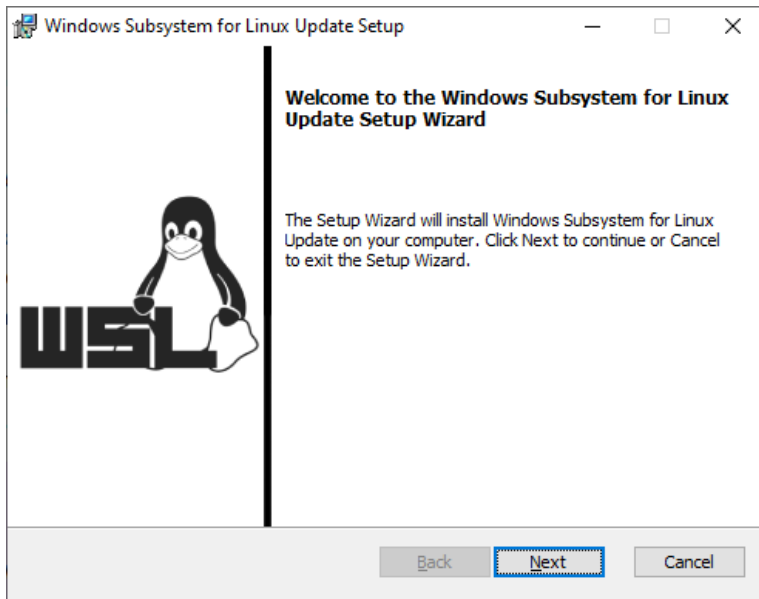
C:\Windows\system32>_
```

1.2 Linux カーネル更新プログラムパッケージをダウンロード & インストール

以下の URL にあるリンクをクリックしてインストーラーをダウンロードします。

<https://docs.microsoft.com/ja-jp/windows/wsl/install-win10#step-4---download-the-linux-kernel-update-package>

ダウンロードした exe ファイルを実行すればインストール完了です (参考までに、インストーラー起動時は以下のスクリーンショットに示すウィンドウが出ます)。



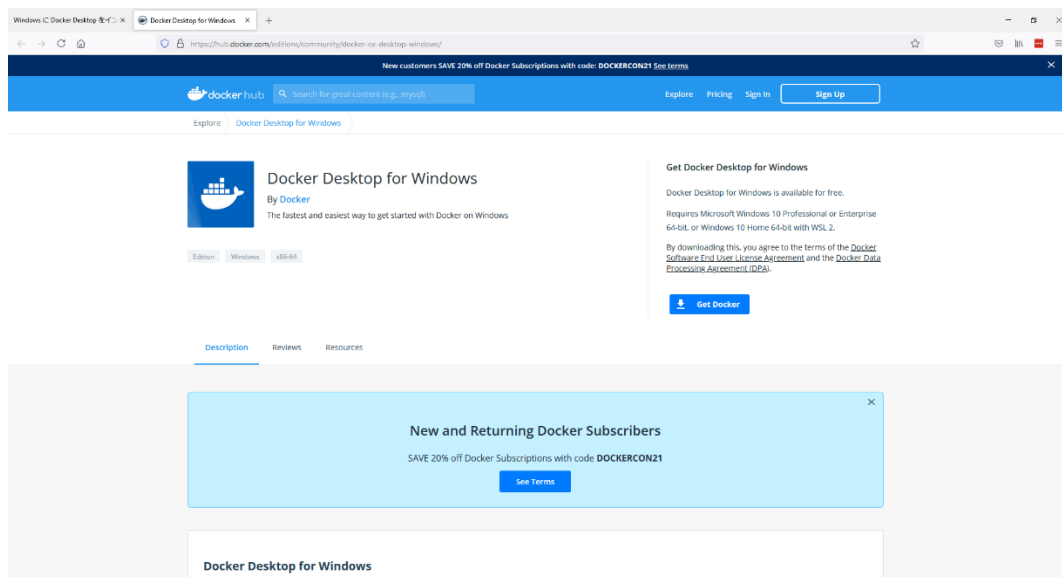
1.3 Docker Desktop のダウンロード & インストール

WSL2 がインストールできたら Docker Desktop をインストールします。インストール手順が記載されている以下の URL の公式 HP にアクセスすると次のような画面になります。

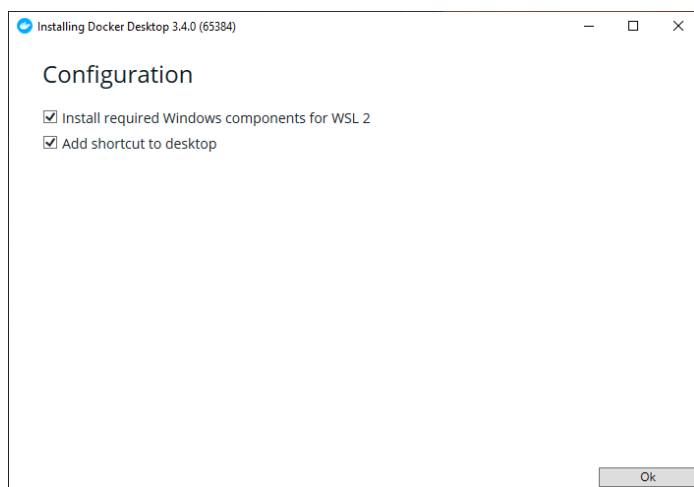
<https://docs.docker.jp/docker-for-windows/install-windows-home.html>



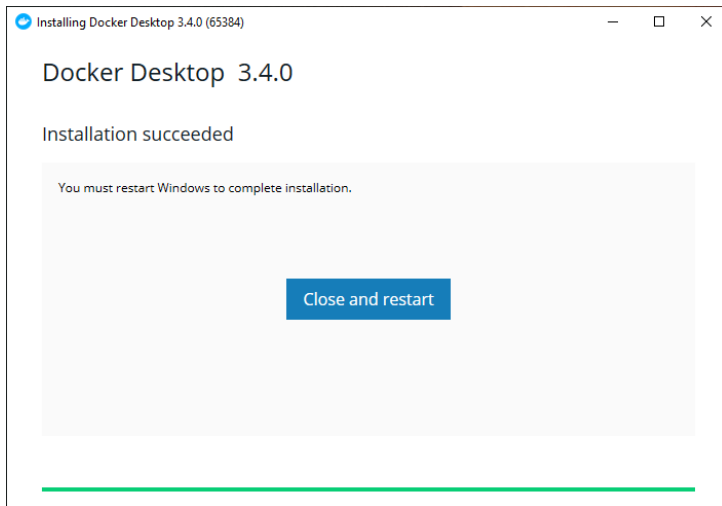
上のページ中にある「Docker Hub からダウンロード」をクリックすると、以下のダウンロードページに飛びます。



上の画面の「Get Docker」ボタンをクリックすればダウンロードが開始されます。ダウンロードした exe ファイルを起動すると以下のような画面になるので両方チェックがあることを確認して「OK」を押します。



インストールが終わると次のような画面になります。



これで Docker Desktop のインストールは完了ですが、念のためコマンドプロンプトで「docker」と打ってエラーが出ないことを確認すると良いです。インストールされていると下のような画面になります。(正しくインストールされていない点は未検証ですが、「コマンドが見つかりません」などと表示されると思います。)

```
Command Prompt
Microsoft Windows [Version 10.0.19042.1052]
(c) Microsoft Corporation. All rights reserved.

C:\Users\¥¥¥¥¥¥¥¥>docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default
                        "C:\Users\¥¥¥¥¥¥¥¥.docker")
  -c, --context string  Name of the context to use to connect to the
                        daemon (overrides DOCKER_HOST env var and
                        default context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level
                        ("debug"|"info"|"warn"|"error"|"fatal")
                        (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string    Trust certs signed only by this CA (default
                        "C:\Users\¥¥¥¥¥¥¥¥.docker\ca.pem")
  --tlscert string      Path to TLS certificate file (default
                        "C:\Users\¥¥¥¥¥¥¥¥.docker\cert.pem")
  --tlskey string       Path to TLS key file (default
                        "C:\Users\¥¥¥¥¥¥¥¥.docker\key.pem")
  --tlsverify           Use TLS and verify the remote
  -v, --version         Print version information and quit
```

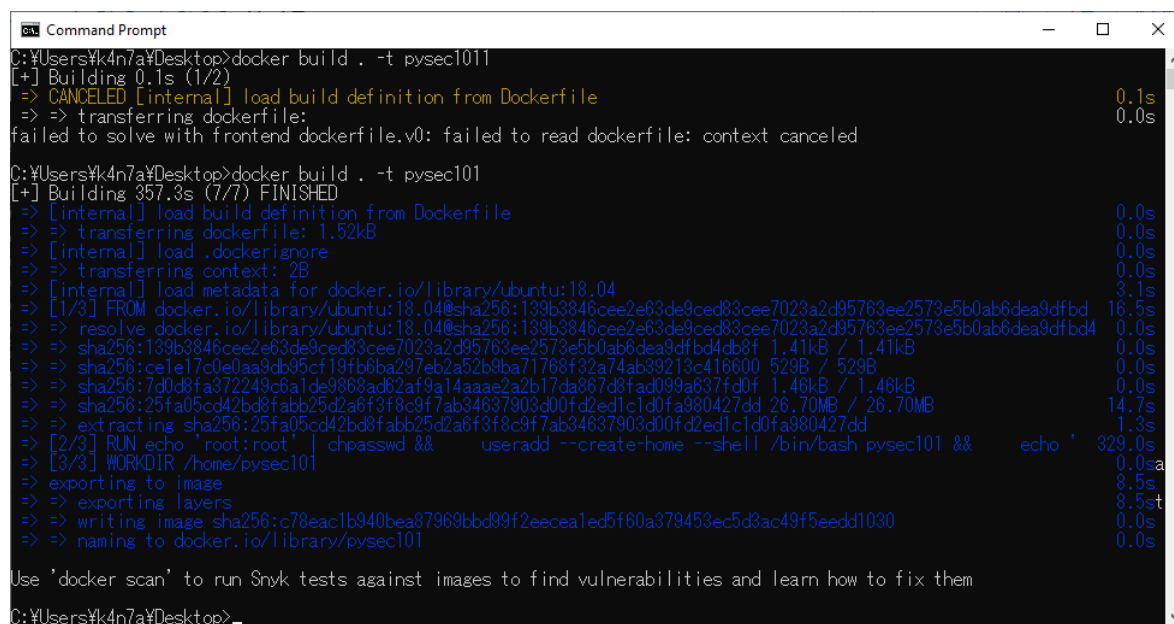
2. Dockerfile を使い、書籍で使う Linux コンテナをビルド (書籍 12p と同じ)

Docker Desktop がインストールできたら、書籍で使う Linux コンテナをビルドする作業に移りますが、こちらの手順は書籍 12p に記載している手順と同じです。

まずサポートサイトや gitlab から Dockerfile および run_linux をダウンロードします。ダウンロードした Dockerfile を適当なフォルダ(ここでは Desktop)に配置し、コマンドプロンプトに以下のコマンドを入力します。

```
docker build . -t pysec101
```

ビルドが完了したときのコマンドプロンプトの画面は以下の通りです。



```
Command Prompt
C:\Users\k4n7a\Desktop>docker build . -t pysec101
[+] Building 0.1s (1/2)
=> CANCELED [internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 1.52kB 0.0s
failed to solve with frontend dockerfile.v0: failed to read dockerfile: context canceled

C:\Users\k4n7a\Desktop>docker build . -t pysec101
[+] Building 357.3s (7/7) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 1.52kB 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/ubuntu:18.04 3.1s
=> [1/3] FROM docker.io/library/ubuntu:18.04@sha256:139b3846cee2e63de9ced83cee7023a2d95763ee2573e5b0ab6dea9dfbd4 16.5s
=> => resolve docker.io/library/ubuntu:18.04@sha256:139b3846cee2e63de9ced83cee7023a2d95763ee2573e5b0ab6dea9dfbd4 0.0s
=> => sha256:139b3846cee2e63de9ced83cee7023a2d95763ee2573e5b0ab6dea9dfbd4db8f 1.41kB / 1.41kB 0.0s
=> => sha256:ce1e17c0e0aa9db95cf19fb6ba297eb2a52b9ba71768f32a74ab39213c416600 529B / 529B 0.0s
=> => sha256:7d0d8fa372249c6a1de9868ad62af9a14aaae2a2b17da867c8fad099a637fd0f 1.46kB / 1.46kB 0.0s
=> => sha256:25fa05cd42bd8fab25d2a6f3f8c9f7ab34637903d00fd2ed1c1d0fa980427dd 26.70MB / 26.70MB 14.7s
=> => extracting sha256:25fa05cd42bd8fab25d2a6f3f8c9f7ab34637903d00fd2ed1c1d0fa980427dd 1.3s
=> [2/3] RUN echo 'root:root' | chpasswd && useradd --create-home --shell /bin/bash pysec101 && echo ' 329.0s
=> [3/3] WORKDIR /home/pysec101 0.0s
=> exporting to image 8.5s
=> => exporting layers 8.5s
=> => writing image sha256:c78eac1b940bea87969bbd99f2eecealed5f60a379453ec5d3ac49f5eedd1030 0.0s
=> => naming to docker.io/library/pysec101 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
C:\Users\k4n7a\Desktop>
```

3. コンテナの起動

3.1 通常の起動方法

書籍中では `run_linux` を使ってコンテナを起動していますが、`Docker Desktop` をインストールした場合は少し異なります。コマンドプロンプトを起動して以下のコマンドを実行することでコンテナが起動します。

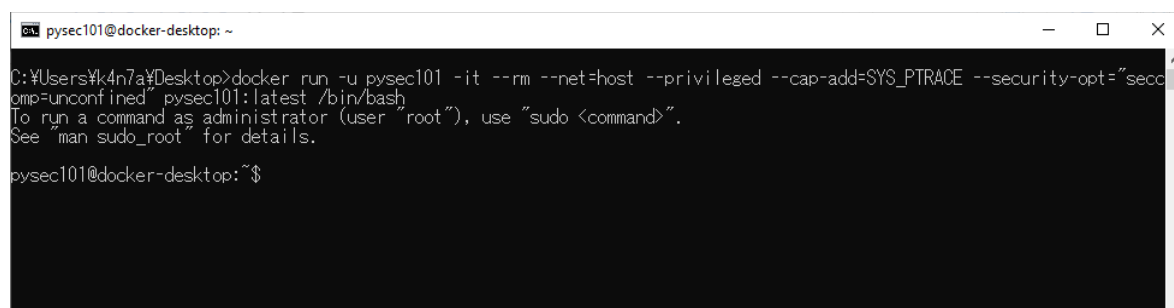
```
docker run -u pysec101 -it --rm --net=host --privileged --cap-add=SYS_PTRACE
--security-opt="seccomp=unconfined" pysec101:latest /bin/bash
```

また、`Windows` ホストと `Linux` コンテナ間でファイルを共有したい場合は、`-v` オプションを付与します。例えば以下のコマンドは、`Windows` ホストの `C:¥Users¥myname¥Desktop` フォルダを `Linux` コンテナ中の `/home/pysec101` ディレクトリに共有する例です。

`C:¥Users¥myname¥Desktop` の部分を変えることで、好きなフォルダのファイルを共有できます。

```
docker run -v C:¥Users¥myname¥Desktop:/home/pysec101 -u pysec101 -it --rm
--net=host --privileged --cap-add=SYS_PTRACE
--security-opt="seccomp=unconfined" pysec101:latest /bin/bash
```

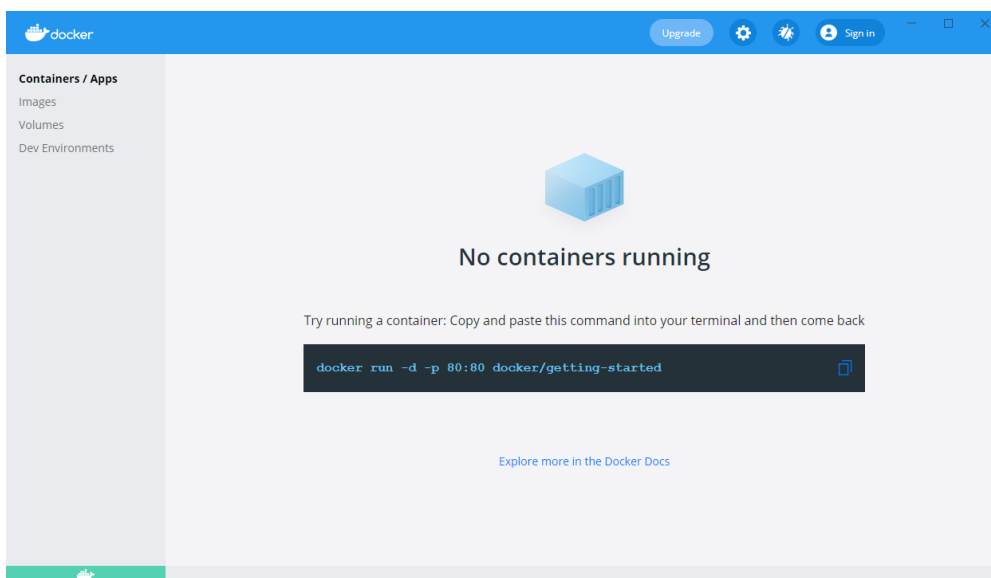
参考までに `Linux` コンテナの起動が完了した画面は以下のようになります。



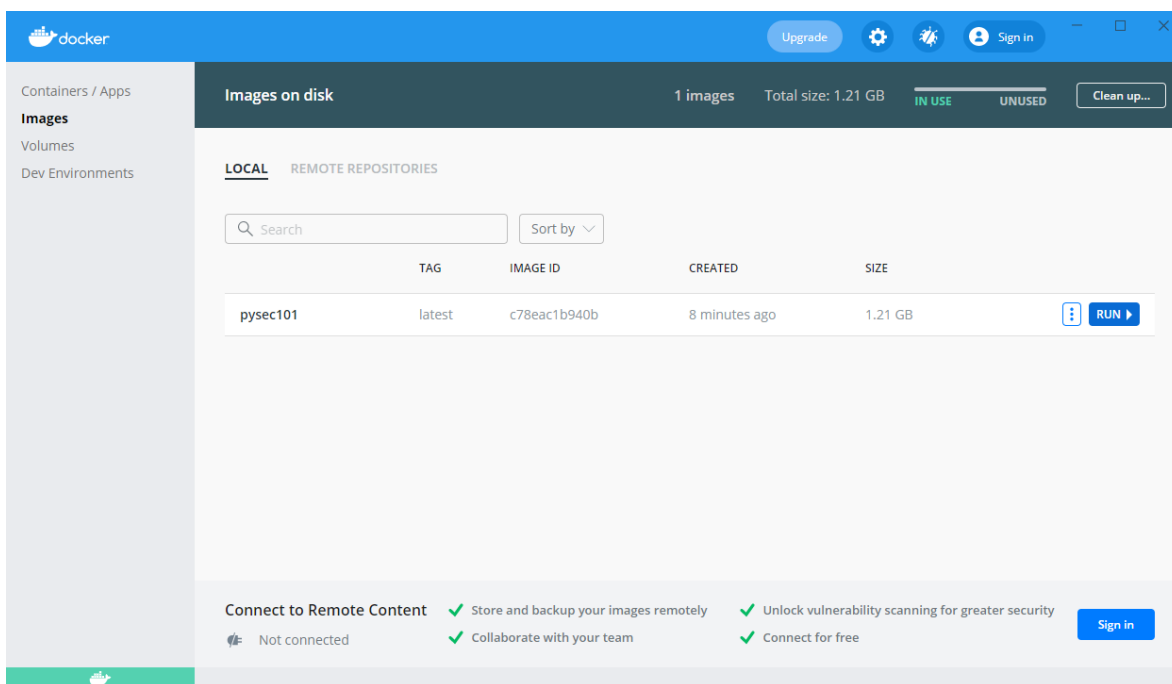
```
pysec101@docker-desktop: ~
C:¥Users¥k4n7a¥Desktop>docker run -u pysec101 -it --rm --net=host --privileged --cap-add=SYS_PTRACE --security-opt="seccomp=unconfined" pysec101:latest /bin/bash
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
pysec101@docker-desktop:~$
```

3.2 別の起動方法

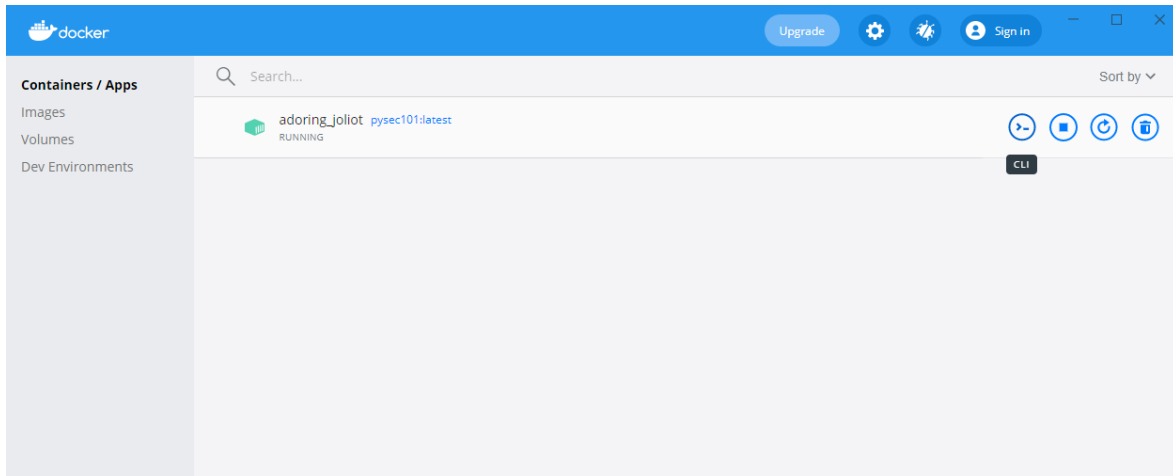
もう一つ別の起動方法として、**Docker Desktop** の GUI から立ち上げる方法もあります。インストールした **Docker Desktop** を起動すると以下のような画面になるので、左の「images」をクリックします。



クリックすると以下のような画面になる(ビルドした **pysec101** コンテナが確認できる)ので、「RUN」を押す。



「RUN」を押すと、「Containers/Apps」タブに起動中のコンテナが現れるので、右の「CLI」ボタンを押せばコマンドプロンプトが立ち上がり、Linux コンテナが起動できる。



この方法は GUI でできるので簡単ですが、一部のコードが動かないため、3.1 の手順を推奨します (3.2 の手順で起動したコンテナ中で 8 章のサンプルコードが動作しません)。

以上