

# 高度サイバーセキュリティPBL II

## 2023 事前課題

株式会社ティアフォー / 大阪大学

高野 祐輝

# 事前課題

- Linux環境をセットアップし、課題用のソースコードをコンパイルせよ
- 本スライドではDockerを用いた方法を説明するが、スキルのある人は他の方法でも良い
- CPUアーキテクチャはx86-64のみとする
- Rustの講義スライドやRust by exampleを読んで、Rust言語の事前学習  
<https://doc.rust-jp.rs/rust-by-example-ja/>

# 事前レポート

- Linux環境を用意し、課題用ソースコードのコンパイルができたことを示すレポートを提出せよ  
(ChatGPTの**利用可**)
- The Rust Programming Language (<https://doc.rust-jp.rs/book-ja/>)の5章までを読んで、Rustの基本を解説した簡単なドキュメントを作成せよ。  
A4で2ページ以上。サンプルコードを自分でコンパイルしてみることに。  
(調査にはChatGPTの**利用可**。文章生成にはChatGPTの**利用禁止**)
- 締め切り：2023年6月5日（月） 23:50 (JST)

# ChatGPTの利用について

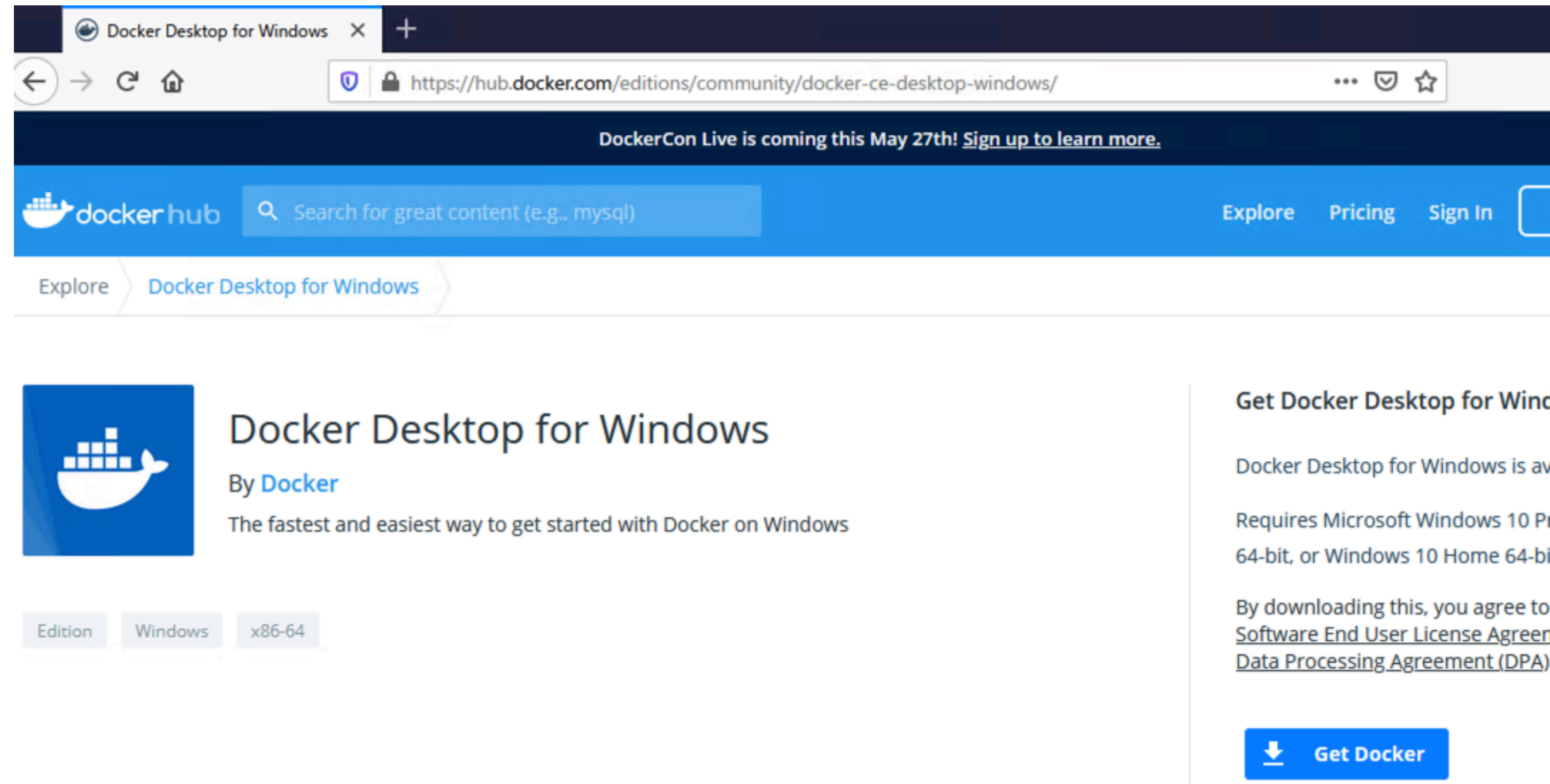
- ChatGPTは優れたツールなので、調査ではどんどん使ってください
- しかし、先端技術に関する作文や実装は、まだ人間が行う必要があります
- 大学の講義は、先端技術に関する作文能力や実装能力を身につけるのが目的です。文章生成や実装をそのままChatGPTにやらせるのはやめましょう

# Dockerのインストール

# Dockerのインストール Windows

以下のサイトからDockerを取得しインストールしてください

<https://hub.docker.com/editions/community/docker-ce-desktop-windows/>



The screenshot shows a web browser window with the Docker Hub page for Docker Desktop for Windows. The browser's address bar shows the URL <https://hub.docker.com/editions/community/docker-ce-desktop-windows/>. The page features a blue header with the Docker Hub logo, a search bar, and links for Explore, Pricing, and Sign In. Below the header, there's a navigation bar with 'Explore' and 'Docker Desktop for Windows'. The main content area displays the Docker Desktop for Windows logo, the title 'Docker Desktop for Windows', and the text 'By Docker' and 'The fastest and easiest way to get started with Docker on Windows'. There are also tabs for 'Edition', 'Windows', and 'x86-64'. On the right side, there's a section titled 'Get Docker Desktop for Windows' with a description, requirements, and a 'Get Docker' button.

Docker Desktop for Windows

By Docker

The fastest and easiest way to get started with Docker on Windows

Get Docker Desktop for Windows

Docker Desktop for Windows is available for Windows 10 Pro, Windows 10 Enterprise, Windows 10 Education, Windows 10 IoT Enterprise, Windows 10 Home 64-bit, or Windows 10 Home 64-bit.

By downloading this, you agree to [Software End User License Agreement](#) and [Data Processing Agreement \(DPA\)](#)

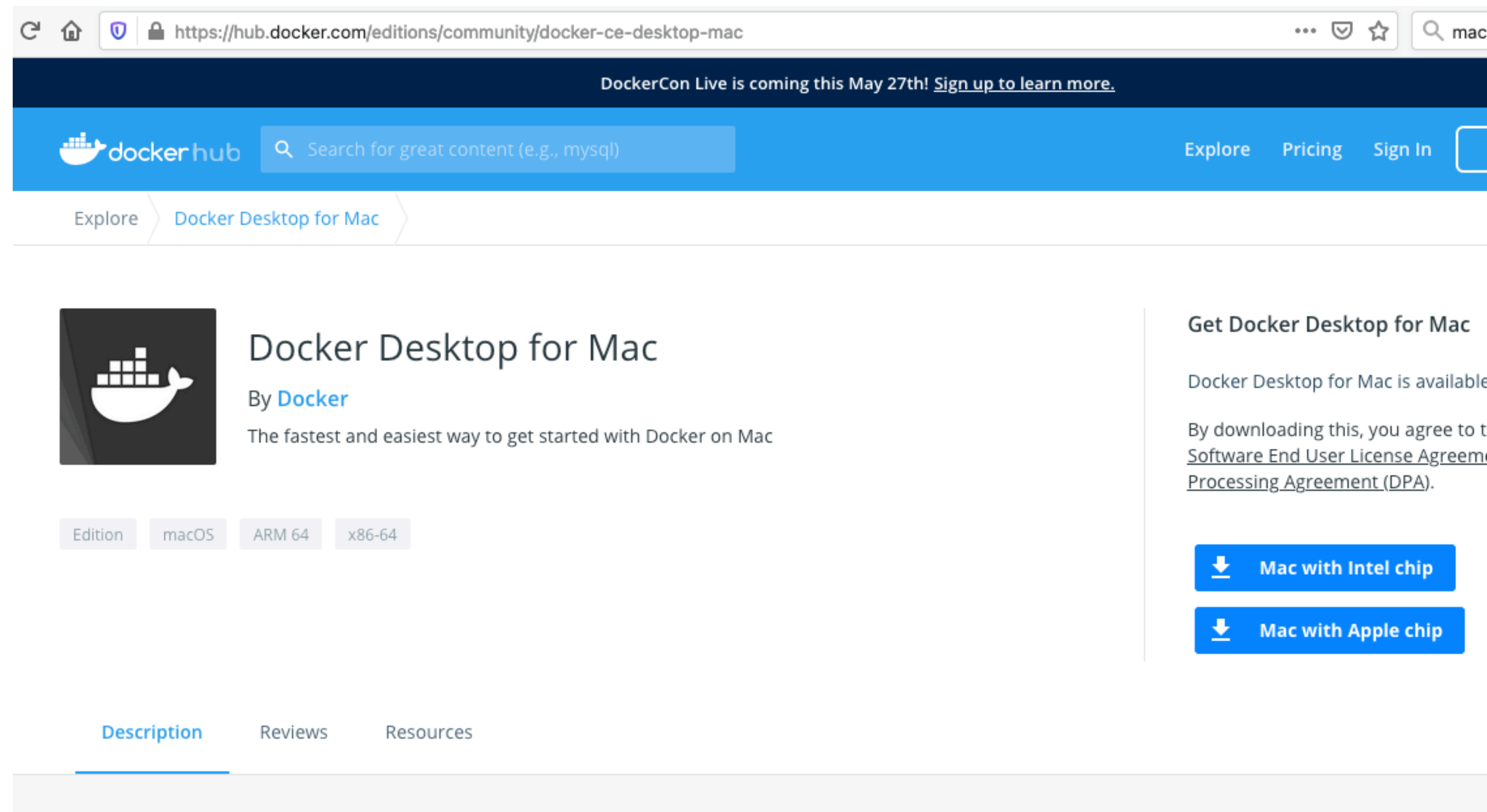
Get Docker

# Dockerのインストール

## Mac

Macの場合はこちら

<https://hub.docker.com/editions/community/docker-ce-desktop-mac>



The screenshot shows the Docker Hub page for 'Docker Desktop for Mac'. The browser address bar displays the URL <https://hub.docker.com/editions/community/docker-ce-desktop-mac>. The page features a dark blue header with the Docker Hub logo, a search bar, and navigation links for 'Explore', 'Pricing', and 'Sign In'. Below the header, a breadcrumb trail shows 'Explore' > 'Docker Desktop for Mac'. The main content area displays the Docker Desktop for Mac logo, the title 'Docker Desktop for Mac', and the text 'By Docker' and 'The fastest and easiest way to get started with Docker on Mac'. There are tabs for 'Edition', 'macOS', 'ARM 64', and 'x86-64'. On the right side, there is a section titled 'Get Docker Desktop for Mac' with the text 'Docker Desktop for Mac is available' and 'By downloading this, you agree to the [Software End User License Agreement](#) and [Data Processing Agreement \(DPA\)](#)'. Below this, there are two blue buttons: 'Mac with Intel chip' and 'Mac with Apple chip'. At the bottom, there are tabs for 'Description', 'Reviews', and 'Resources'.

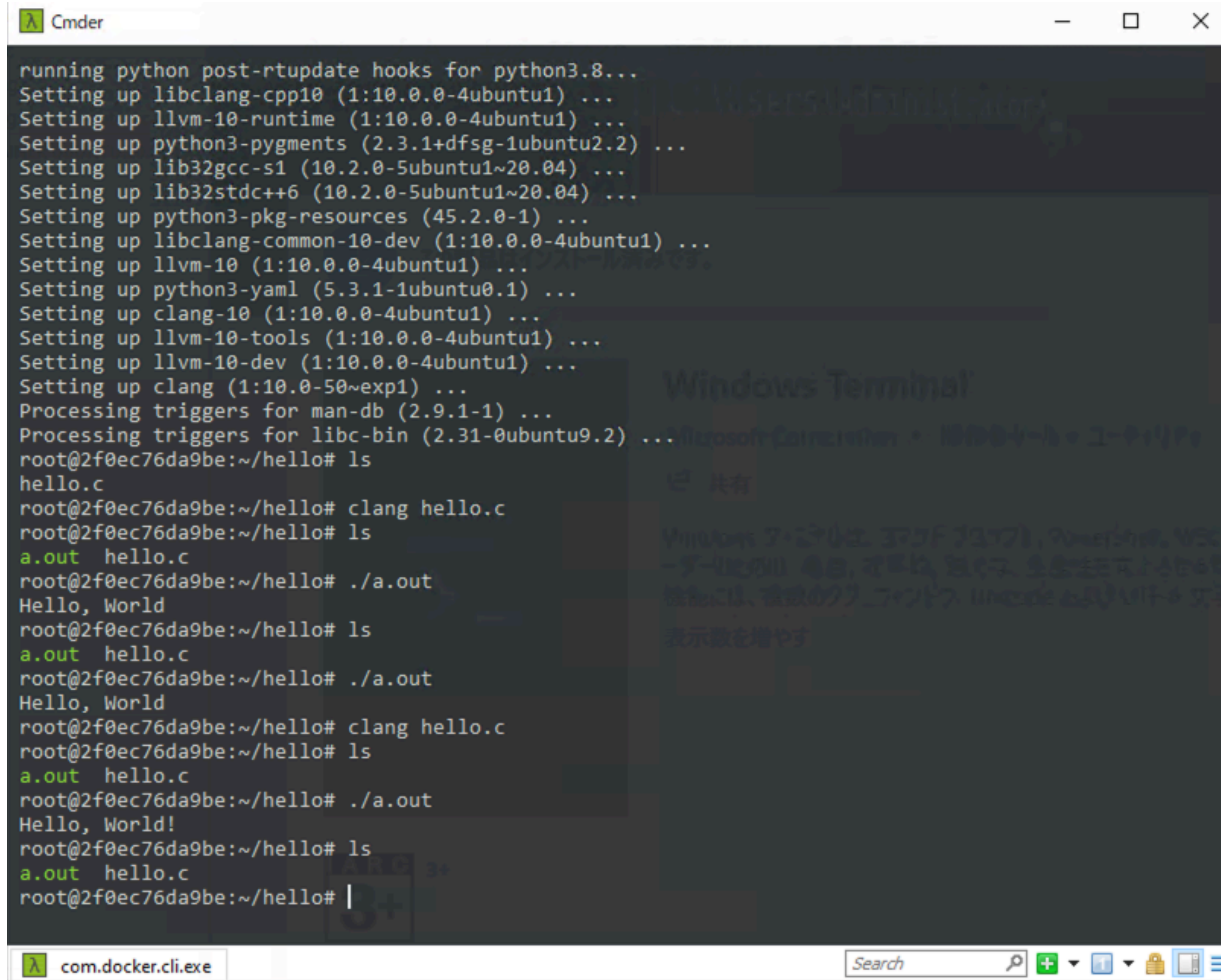
# ターミナルソフトウェアの インストール (Windowsのみ)



# ターミナルソフトウェアのインストール

- Windows標準のcmdは非常に使いにくいため、別のターミナルソフトウェアをインストールしてください。個人的なおすすめは以下の2つ。
- cmdr
  - 以下のサイトより取得
  - <https://cmdr.net/>
- Windows Terminal
  - Windows Storeでterminalと検索

# cmder



```
running python post-rtupdate hooks for python3.8...
Setting up libclang-cpp10 (1:10.0.0-4ubuntu1) ...
Setting up llvm-10-runtime (1:10.0.0-4ubuntu1) ...
Setting up python3-pygments (2.3.1+dfsg-1ubuntu2.2) ...
Setting up lib32gcc-s1 (10.2.0-5ubuntu1~20.04) ...
Setting up lib32stdc++6 (10.2.0-5ubuntu1~20.04) ...
Setting up python3-pkg-resources (45.2.0-1) ...
Setting up libclang-common-10-dev (1:10.0.0-4ubuntu1) ...
Setting up llvm-10 (1:10.0.0-4ubuntu1) ...
Setting up python3-yaml (5.3.1-1ubuntu0.1) ...
Setting up clang-10 (1:10.0.0-4ubuntu1) ...
Setting up llvm-10-tools (1:10.0.0-4ubuntu1) ...
Setting up llvm-10-dev (1:10.0.0-4ubuntu1) ...
Setting up clang (1:10.0-50~exp1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
root@2f0ec76da9be:~/hello# ls
hello.c
root@2f0ec76da9be:~/hello# clang hello.c
root@2f0ec76da9be:~/hello# ls
a.out hello.c
root@2f0ec76da9be:~/hello# ./a.out
Hello, World
root@2f0ec76da9be:~/hello# ls
a.out hello.c
root@2f0ec76da9be:~/hello# ./a.out
Hello, World
root@2f0ec76da9be:~/hello# clang hello.c
root@2f0ec76da9be:~/hello# ls
a.out hello.c
root@2f0ec76da9be:~/hello# ./a.out
Hello, World!
root@2f0ec76da9be:~/hello# ls
a.out hello.c
root@2f0ec76da9be:~/hello# |
```

The screenshot shows a Cmder terminal window with a dark background. The terminal output displays the installation of various LLVM and Clang components, followed by the compilation and execution of a 'Hello, World' program. The window title is 'Cmder' and the taskbar at the bottom shows 'com.docker.cli.exe' and a search bar.

# Windows Terminal

```
Windows PowerShell

ps      List containers
pull    Pull an image or a repository from a registry
push    Push an image or a repository to a registry
rename  Rename a container
restart Restart one or more containers
rm       Remove one or more containers
rmi     Remove one or more images
run     Run a command in a new container
save    Save one or more images to a tar archive (streamed to STDOUT by default)
search  Search the Docker Hub for images
start   Start one or more stopped containers
stats   Display a live stream of container(s) resource usage statistics
stop    Stop one or more running containers
tag      Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top     Display the running processes of a container
unpause Unpause all processes within one or more containers
update  Update configuration of one or more containers
version Show the Docker version information
wait    Block until one or more containers stop, then print their exit codes

Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at https://docs.docker.com/go/guides/
PS C:\Users\ytakano> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
2f0ec76da9be   ubuntu:20.04   "/bin/sh"               2 weeks ago   Up 47 minutes           naughty_rosalind
PS C:\Users\ytakano> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
2f0ec76da9be   ubuntu:20.04   "/bin/sh"               2 weeks ago   Up 47 minutes           naughty_rosalind
PS C:\Users\ytakano>
```

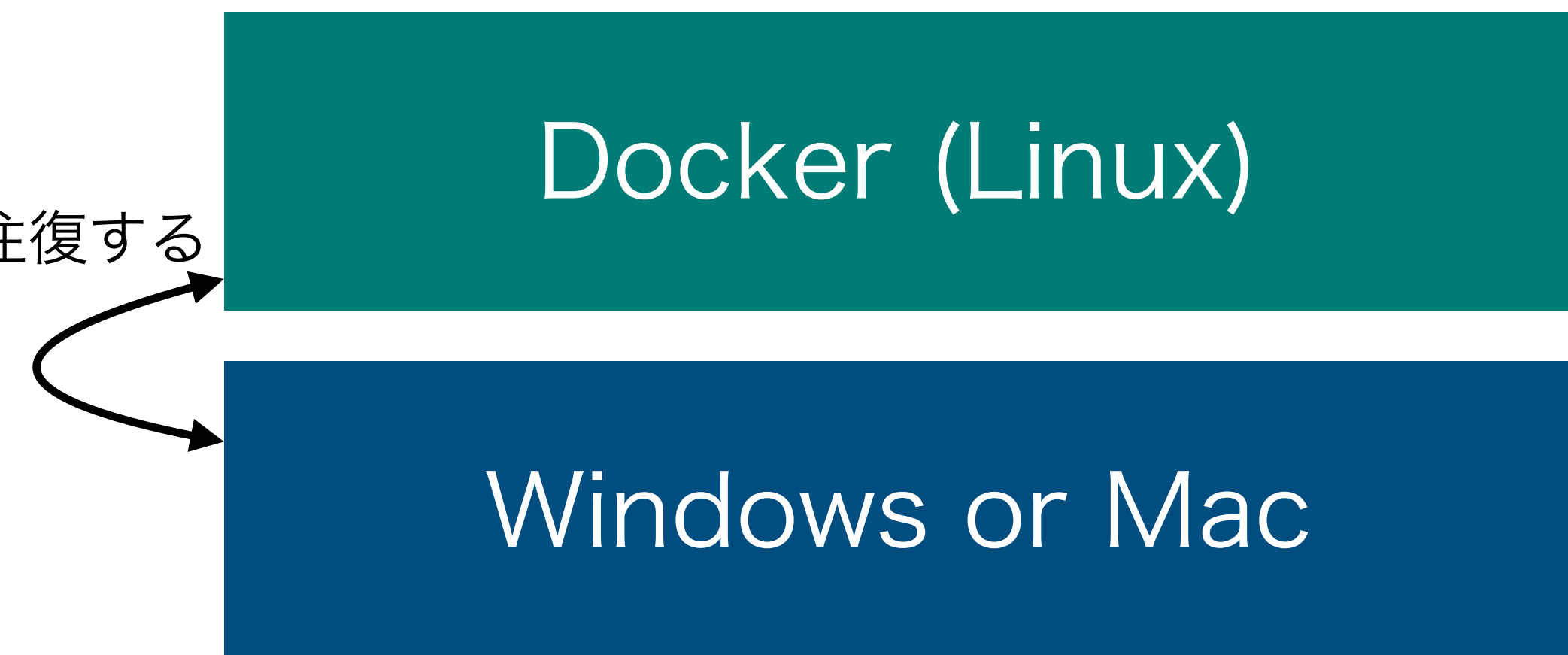
# Ubuntuコンテナの インストール



# どこに自分がいるかを考えながら作業しよう

- Windows or Macのホストか、Dockerのゲストか、今どちらで作業しているかをよく注意してください。
- ホスト側でLinuxコマンドは使えず、Linux側でDockerの操作はできません

ここを往復する



# Dockerイメージの作成と起動

Ubuntu 22.04のイメージを、以下のオプション付きで起動

```
$ docker run -itd --cap-add=SYS_PTRACE --security-opt="seccomp=unconfined"  
ubuntu:22.04 /bin/sh
```

docker psコマンドで、作成したコンテナのIDを取得

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
c2f037dc32c1	ubuntu:22.04	"/bin/sh"	4 minutes ago	Up 4 minutes

goofy\_wright

作成したコンテナ内で、/bin/bashを実行

```
$ docker exec -it c2 /bin/bash
```

# コンテナのパッケージ更新

コンテナ内でパッケージのリストを更新

```
root@c2f037dc32c1:/# apt update
```

パッケージをアップグレード

```
root@c2f037dc32c1:/# apt upgrade
```

Dockerイメージは縮小版となっているため、通常版に復元する

```
root@c2f037dc32c1:/# unminimize
```

...

```
Do you want to continue? [Y/n] Y
```

# ソフトウェアのインストール

## 講義で利用するソフトウェアをインストール

```
root@c2f037dc32c1:/# apt install build-essential git man curl libncurses5-dev  
libtinfo5
```

...

```
Do you want to continue? [Y/n] Y
```

## Rustのインストール

```
root@c2f037dc32c1:/# curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

インストールするバージョンを聞かれるがデフォルトでOK（そのままリターン）



# コンテナから抜けて、もう一度入る

一旦コンテナから抜ける

```
root@c2f037dc32c1:/# exit
```

再度、コンテナ内で/bin/bashを実行

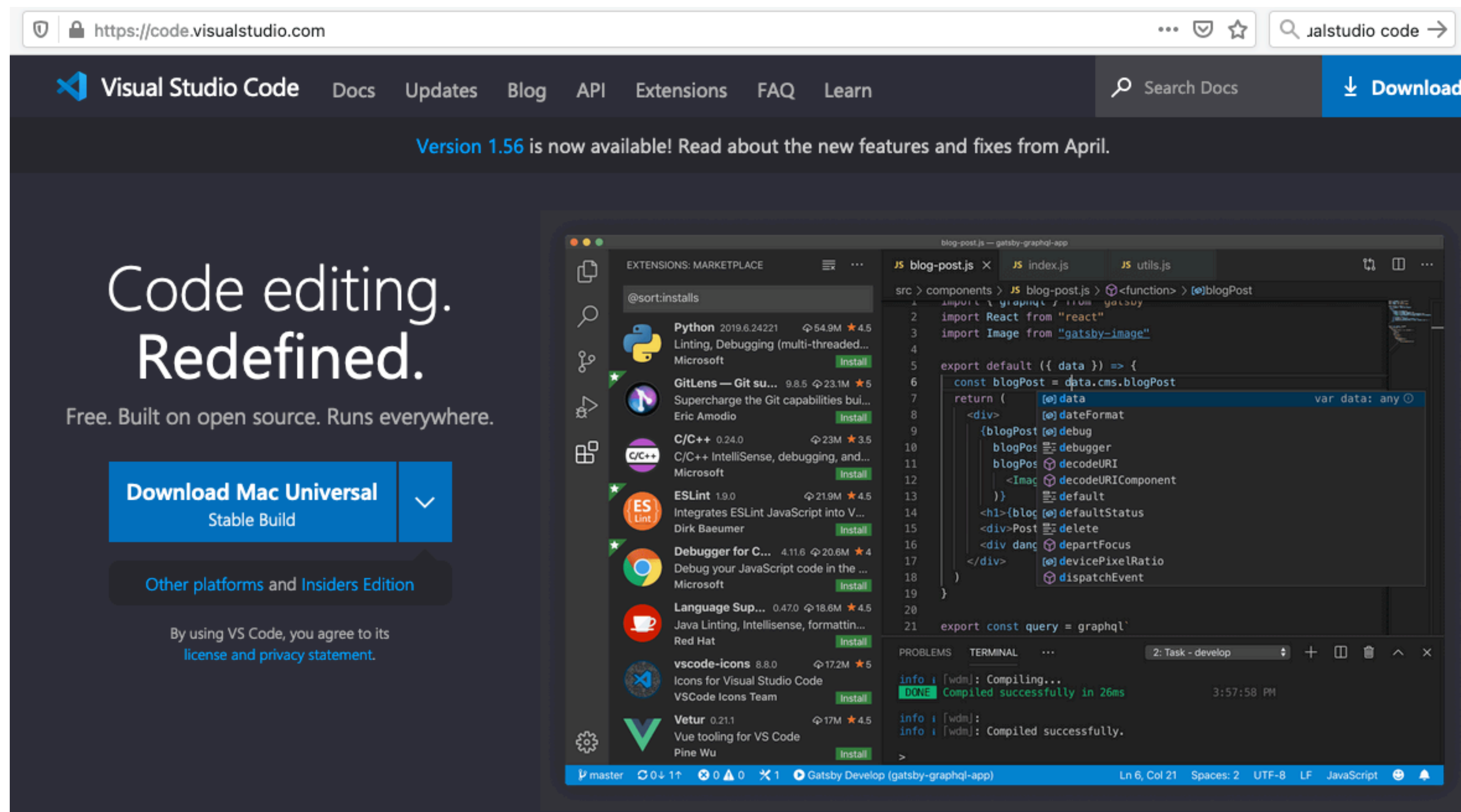
```
$ docker exec -it c2 /bin/bash
```

# VisualStudio Codeの インストール

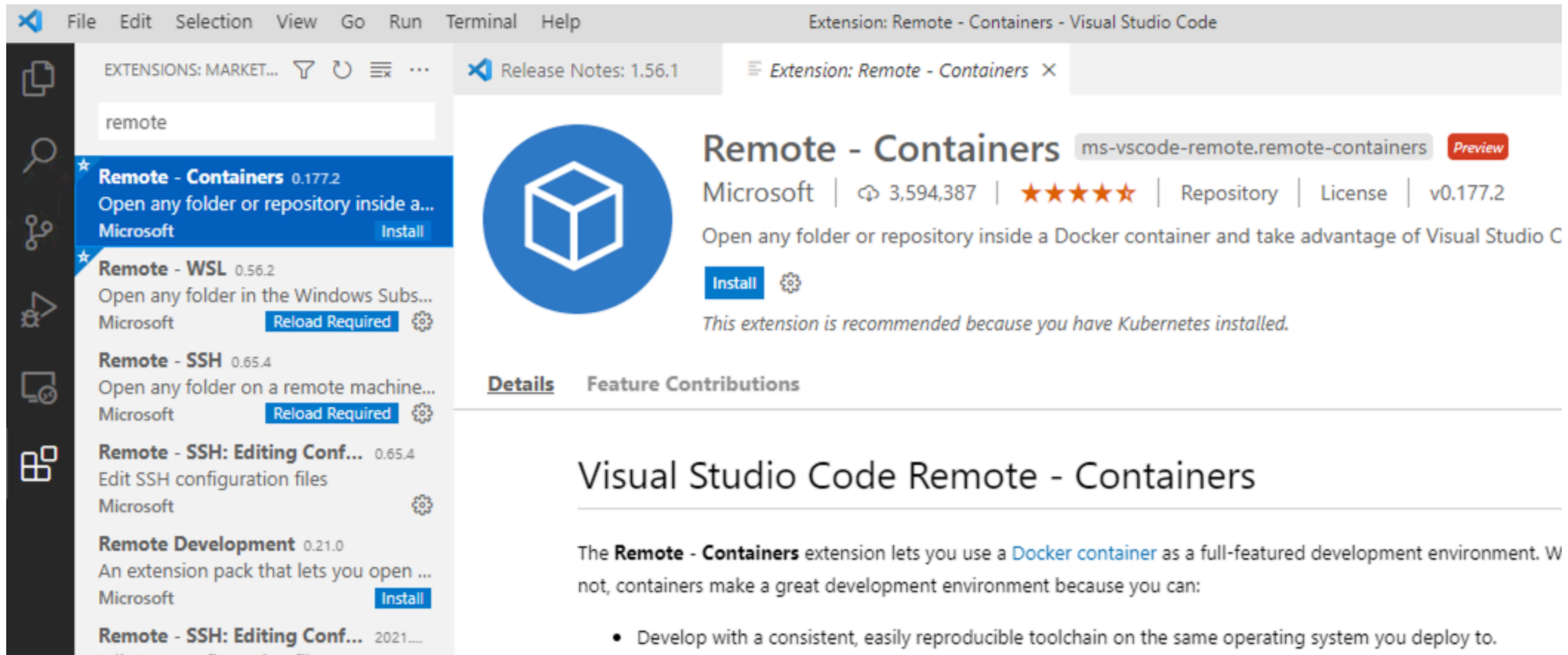
# VisualStudio Codeのダウンロード

以下のサイトより、VisualStudio Codeをダウンロードし、インストールしてください。

<https://code.visualstudio.com/>



# Visual Studio Codeの Remote Containers拡張をインストール



The screenshot shows the Visual Studio Code interface with the 'Remote - Containers' extension page open. The left sidebar displays a list of extensions, with 'Remote - Containers' selected. The main panel shows the extension's details, including its icon, name, publisher (Microsoft), version (0.177.2), and a list of features. The extension is marked as 'Preview' and has a high rating of 5 stars. The description states that it allows users to open any folder or repository inside a Docker container and take advantage of Visual Studio Code's features. The 'Install' button is visible, along with a note that the extension is recommended because the user has Kubernetes installed. The bottom section of the page is titled 'Visual Studio Code Remote - Containers' and provides a brief overview of the extension's capabilities.

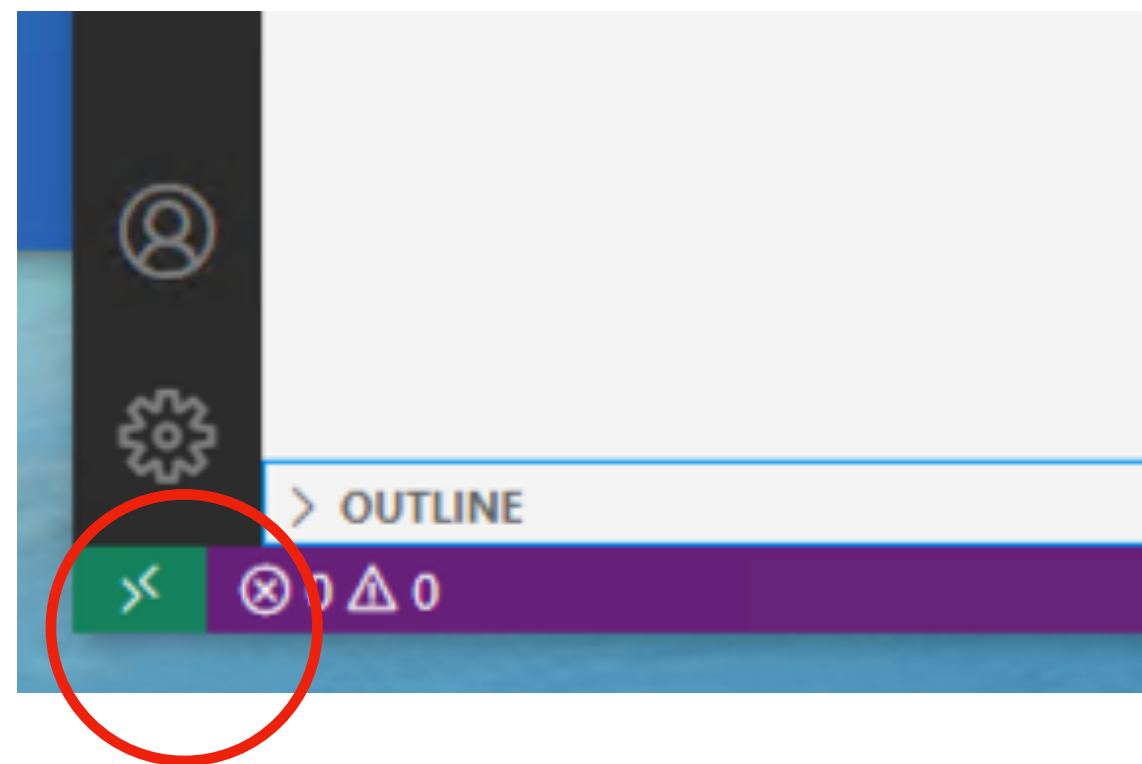
Visual Studio Code Remote - Containers

The **Remote - Containers** extension lets you use a [Docker container](#) as a full-featured development environment. Without containers, containers make a great development environment because you can:

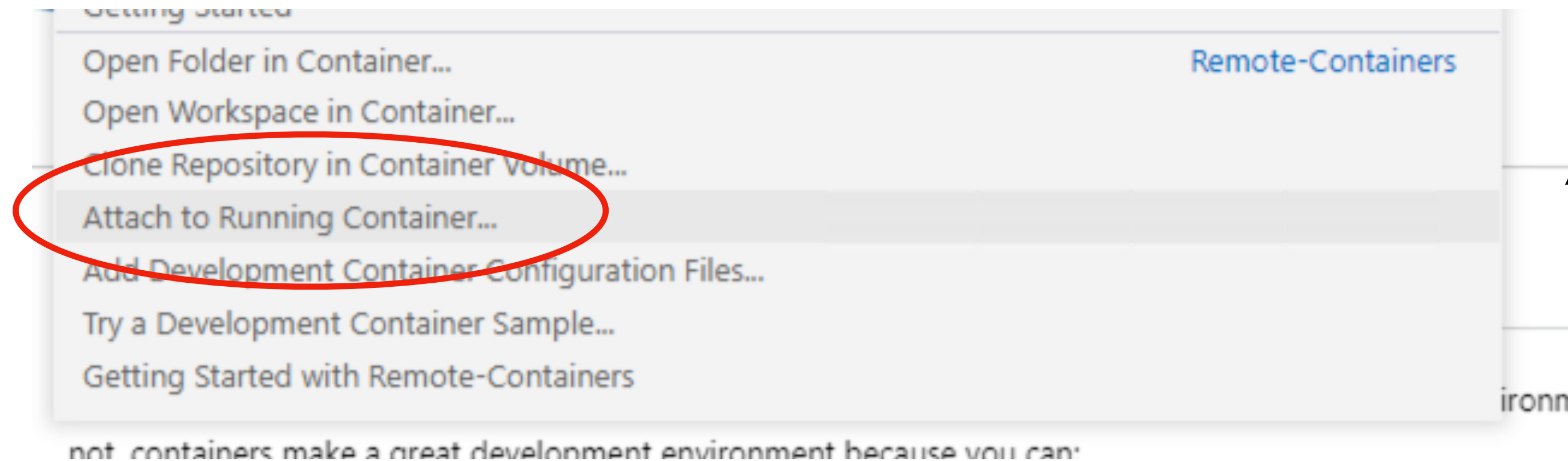
- Develop with a consistent, easily reproducible toolchain on the same operating system you deploy to.



# コンテナにアタッチ

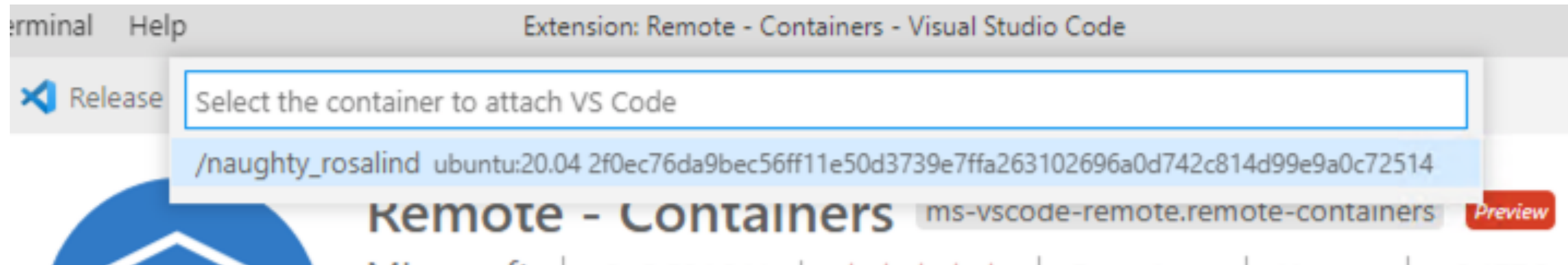


左下の緑をクリックし



Attach to Running Container...  
を選択

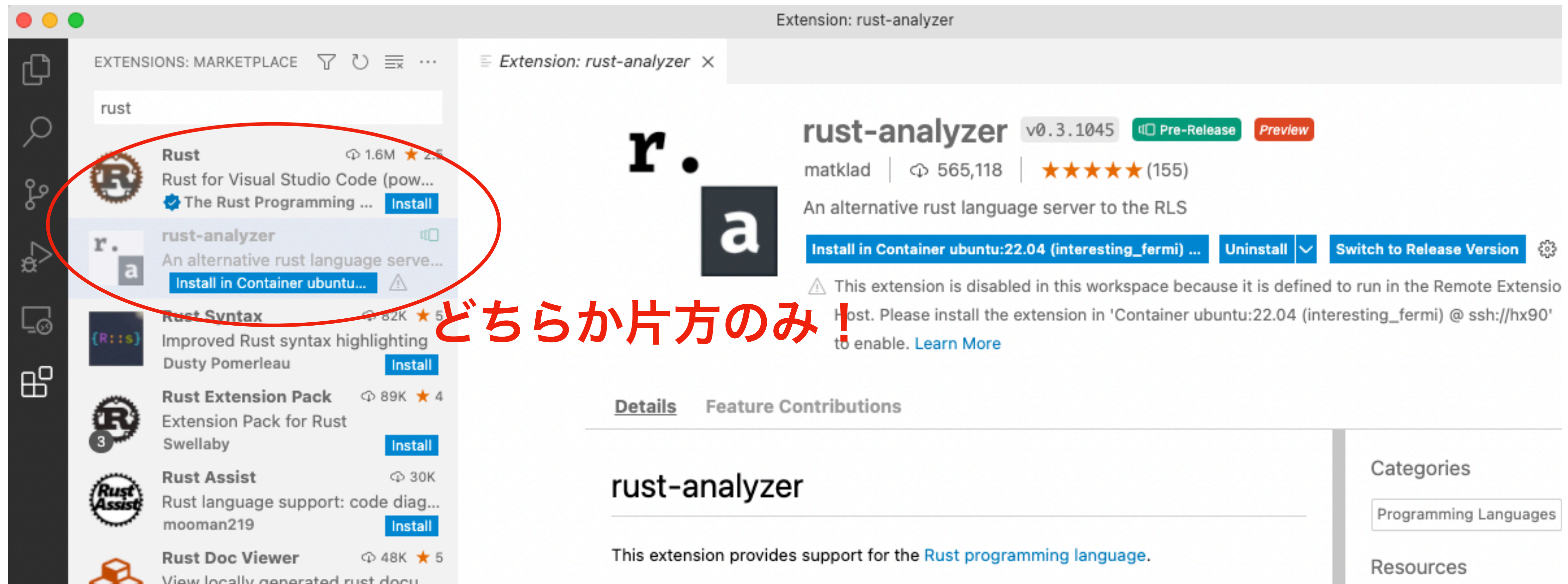
# コンテナにアタッチ



先に作成したコンテナを選択



# rust-analyzerかRust機能拡張のインストール



The screenshot shows the Visual Studio Code Extensions Marketplace. On the left, a list of extensions is displayed, with 'rust-analyzer' and 'Rust' circled in red. A red text overlay 'どちらか片方のみ!' (Only one of them!) is placed over the list. On the right, the details for 'rust-analyzer' are shown. It is version v0.3.1045, a Pre-Release, and has 565,118 downloads and 155 stars. It is described as 'An alternative rust language server to the RLS'. A warning message states: 'This extension is disabled in this workspace because it is defined to run in the Remote Extension Host. Please install the extension in 'Container ubuntu:22.04 (interesting\_fermi)' @ ssh://hx90' to enable. Learn More'. The 'Details' tab is selected, showing the extension's name and description.

rust-analyzer v0.3.1045 Pre-Release Preview

matklad | 565,118 | ★★★★★ (155)

An alternative rust language server to the RLS

Install in Container ubuntu:22.04 (interesting\_fermi) ... Uninstall Switch to Release Version

⚠ This extension is disabled in this workspace because it is defined to run in the Remote Extension Host. Please install the extension in 'Container ubuntu:22.04 (interesting\_fermi)' @ ssh://hx90' to enable. [Learn More](#)

Details Feature Contributions

rust-analyzer

This extension provides support for the [Rust programming language](#).

Categories

Programming Languages

Resources

rust-analyzerがおすすめだが、PCの搭載メモリが少ない場合はRustを利用すること

# 課題の取得とコンパイル



# zdbg

zdbgのソースコードをダウンロードしコンパイル

```
$ cd  
$ git clone https://github.com/ytakano-lecture/zdbg.git  
$ cd zdbg  
$ cargo build
```

# tiny\_regex\_rs

tiny\_regex\_rsのソースコードをダウンロードしコンパイル

```
$ cd
```

```
$ git clone https://github.com/ytakano-lecture/tiny_regex_rs.git
```

```
$ cd tiny_regex_rs
```

```
$ cargo build
```

# VSCodeによるファイル編集

- VSCodeでコンテナに接続し、先にcloneしたフォルダを開くと、各種ファイルを編集可能

