

橋梁診断支援 AI システム  
インストールマニュアル  
(WSL2 + Docker 版)  
プロキシ設定あり

令和 7 年 6 月

国立研究開発法人 土木研究所

## 目 次

|                                                         |      |
|---------------------------------------------------------|------|
| 1. 概要 .....                                             | 1-1  |
| 2. システム概要 .....                                         | 2-2  |
| 3. 動作環境 .....                                           | 3-3  |
| 3.1. Windows の設定変更 .....                                | 3-3  |
| 4. インストール手順 .....                                       | 4-5  |
| 4.1. Windows Subsystem for Linux 2 (WSL2) のインストール ..... | 4-5  |
| 4.2. WSL2 への Docker のインストール .....                       | 4-12 |
| 4.3. 橋梁診断支援 AI システムのインストール .....                        | 4-15 |
| 4.4. 動作確認 .....                                         | 4-22 |
| 5. 起動・停止方法 .....                                        | 5-23 |
| 5.1. 橋梁診断支援 AI システムの起動 .....                            | 5-23 |
| 5.2. 橋梁診断支援 AI システムの停止 .....                            | 5-25 |
| 6. コンテナとホストシステム .....                                   | 6-27 |
| 6.1. コンテナとホストシステムの関係 .....                              | 6-27 |
| 6.2. ポートの競合 .....                                       | 6-28 |
| 6.2.1. ポート競合の確認 .....                                   | 6-29 |
| 6.2.2. 競合プログラムを停止する場合 .....                             | 6-30 |
| 6.2.3. マッピングするポートを変更する場合 .....                          | 6-30 |

## 目 次

|                                       |      |
|---------------------------------------|------|
| 図 2-1 システムコンテナ化イメージ .....             | 2-2  |
| 図 3-1 コントロールパネルの起動 .....              | 3-3  |
| 図 3-2 プログラムと機能 .....                  | 3-4  |
| 図 3-3 Windows の機能の有効化または無効化 .....     | 3-4  |
| 図 4-1 更新プログラムパッケージのダウンロード .....       | 4-5  |
| 図 4-2 更新プログラムパッケージからの変更確認 .....       | 4-6  |
| 図 4-3 更新プログラムパッケージのインストール .....       | 4-6  |
| 図 4-4 コマンドプロンプトの管理者権限での起動 .....       | 4-7  |
| 図 4-5 WSL のデフォルトバージョンの設定 .....        | 4-7  |
| 図 4-6 WSL 上への Ubuntu のインストール .....    | 4-8  |
| 図 4-7 WSL 上の Ubuntu のコマンドプロンプト .....  | 4-8  |
| 図 4-8 管理者権限での Ubuntu の起動 .....        | 4-8  |
| 図 4-9 プロキシ設定の確認 .....                 | 4-9  |
| 図 4-10 Ubuntu のプロキシ設定 .....           | 4-10 |
| 図 4-11 apt のプロキシ設定 .....              | 4-11 |
| 図 4-12 curl のプロキシ設定 .....             | 4-11 |
| 図 4-13 Docker サービスの起動確認 .....         | 4-13 |
| 図 4-14 Ubuntu 上のファイルへのアクセス .....      | 4-15 |
| 図 4-15 橋梁診断支援 AI システムの配置 .....        | 4-16 |
| 図 4-16 docker のプロキシ設定 .....           | 4-16 |
| 図 4-17 docker compose の設定 .....       | 4-17 |
| 図 4-18 docker compose のプロキシ設定 .....   | 4-18 |
| 図 4-19 橋梁診断支援 AI システムのインストール .....    | 4-19 |
| 図 4-20 橋梁診断支援 AI システムのインストール完了 .....  | 4-19 |
| 図 4-21 橋梁診断支援 AI システムの停止 .....        | 4-20 |
| 図 4-22 橋梁診断支援 AI システムの再起動 .....       | 4-20 |
| 図 4-23 橋梁診断支援 AI システムの起動確認 .....      | 4-20 |
| 図 4-24 橋梁診断支援 AI システムのトップ画面 .....     | 4-22 |
| 図 5-1 Ubuntu の管理者権限での起動 .....         | 5-23 |
| 図 5-2 WSL2 上の Ubuntu のコマンドプロンプト ..... | 5-23 |
| 図 5-3 「appcont」フォルダへの移動 .....         | 5-24 |
| 図 5-4 橋梁診断支援 AI システムの起動 .....         | 5-24 |
| 図 5-5 橋梁診断支援 AI システムの起動確認 .....       | 5-24 |
| 図 5-6 橋梁診断支援 AI システムのログ情報 .....       | 5-25 |

|                                          |      |
|------------------------------------------|------|
| 図 5-7 橋梁診断支援 AI システムの「appcont」フォルダ ..... | 5-25 |
| 図 5-8 橋梁診断支援 AI システムの停止 .....            | 5-26 |
| 図 6-1 橋梁診断支援 AI システムとホストシステムの関係.....     | 6-28 |
| 図 6-2 ポート競合時の netstat コマンド実行時の一例.....    | 6-29 |
| 図 6-3 ポート競合プログラムの確認.....                 | 6-29 |

## 1. 概要

国立研究開発法人土木研究所(以下、当研究所)では、開発中の橋梁診断支援 AI システムについて、システムのインストールに係る負担軽減のため、システムのコンテナ化を行いました。

本図書は、プロキシ設定がされた端末において、コンテナ化した橋梁診断支援 AI システムを Windows Subsystem for Linux2 および Docker を使用してインストールする場合のインストールマニュアルです。プロキシ設定がされていない端末をご使用の場合は、通常の「橋梁診断支援 AI システムインストールマニュアル(WSL2 + Docker 版)」をご参照ください。

## 2. システム概要

当研究所で開発中の橋梁診断支援 AI システムは、Web サーバ(Apache)上の CGI(PHP)を用いた Web アプリケーションとして動作し、バックエンドとしてデータベース(DB:PostgreSQL)を使用しています。そのため、PC に Apache や PHP、PostgreSQL 等を個別にインストールを行い、各々が正しく動作するように設定を行う必要があり、システムが使用できる状態にするまでに大きな負担がありました。

そこで、Docker コンテナエンジンを使用し、橋梁診断支援 AI システムをコンテナ化することで、Docker コンテナエンジンが使用可能な環境であれば容易にインストールできるようなシステムとしました。

図 2-1 に示すように、Apache や PHP、PostgreSQL 等が各種設定内容も含めてコンテナ化されているため、コンテナエンジン上にコンテナとして一括でインストールすることができ、Docker コンテナエンジン環境が準備されている PC 上であれば同じ設定内容でシステムを再現することが可能です。

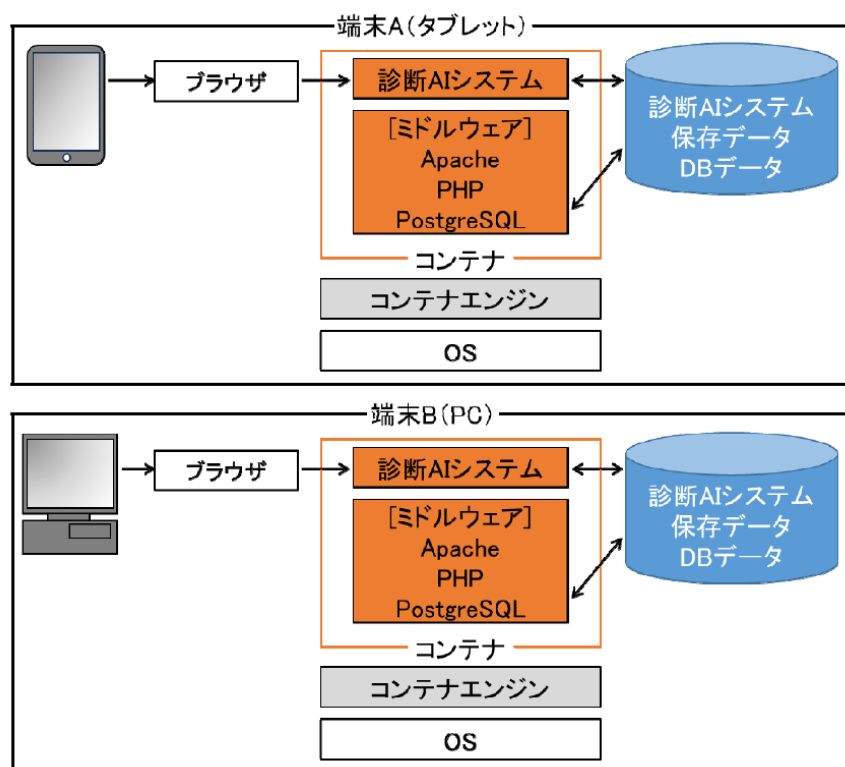


図 2-1 システムコンテナ化イメージ

### 3. 動作環境

橋梁診断支援 AI システムの動作環境を以下に示します。

表 3-1 動作環境

| 項目      | 名称                 | バージョン等                                                                   |
|---------|--------------------|--------------------------------------------------------------------------|
| OS      | Windows10/11 64bit | Pro, Enterprise, Education<br>(備考)<br>Hyper-V、Windows コンテナ、WSL 機能の有効化が必要 |
| 開発言語    | PHP                | 7.4.33                                                                   |
| データベース  | Postgre SQL        | 11                                                                       |
| Web サーバ | Apache             | 2.4 以上                                                                   |
| 使用ブラウザ  | Google Chrome      | Ver.83 以降                                                                |

#### 3.1. Windows の設定変更

Windows 上で Docker コンテナエンジンを使用するには、Hyper-V 等の機能を有効化する必要があります。Docker Desktop for Windows をインストールする場合、これらの機能はインストール時に自動的に有効化されますが、必要に応じて設定の変更を行ってください。

以下ではこれらの機能を有効化する方法について示します。なお、ここでは Windows11 を使用する場合の設定変更について示します。Windows10 の場合も同様の設定変更を行うことで使用することが可能です。

- ① Windowsのスタートメニューから「コントロールパネル」を選択します。

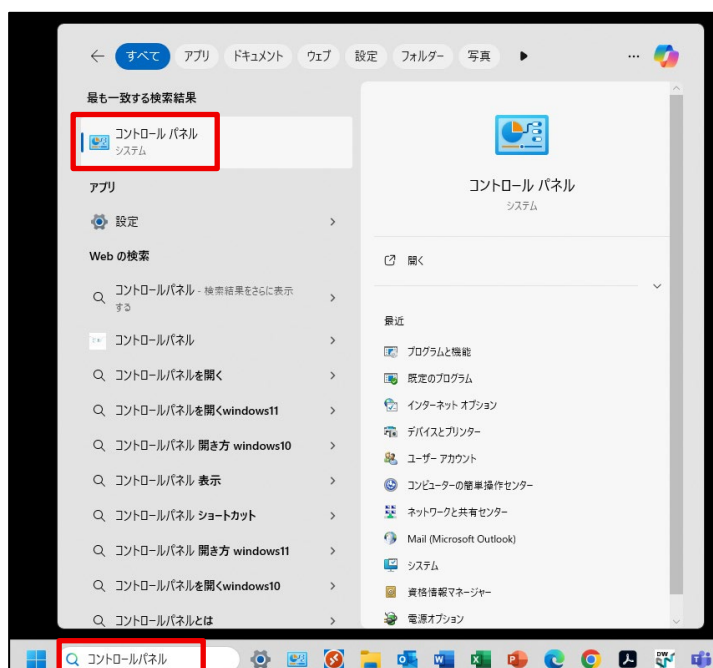


図 3-1 コントロールパネルの起動

- ② 「プログラムと機能」画面から「Windows の機能の有効化または無効化」を選択します。

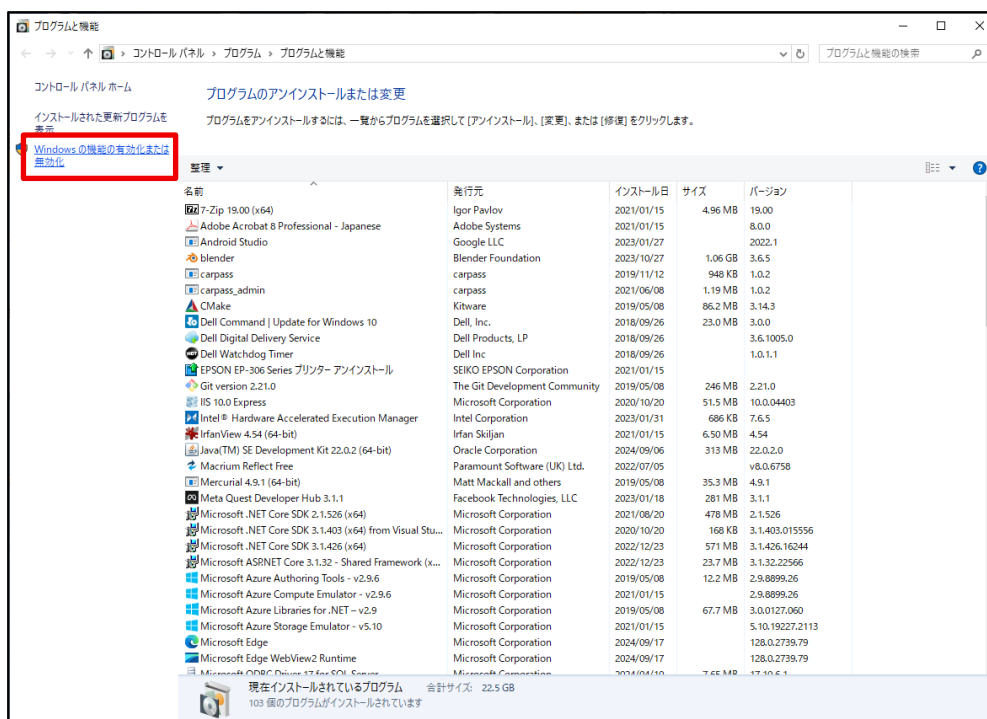


図 3-2 プログラムと機能

- ③ 「Windows の機能の有効化または無効化」画面において、「Hyper-V」と「Linux 用 Windows サブシステム」と「Windows ハイパーバイザープラットフォーム」と「コンテナー」にチェックをつけ「OK」を選択します。

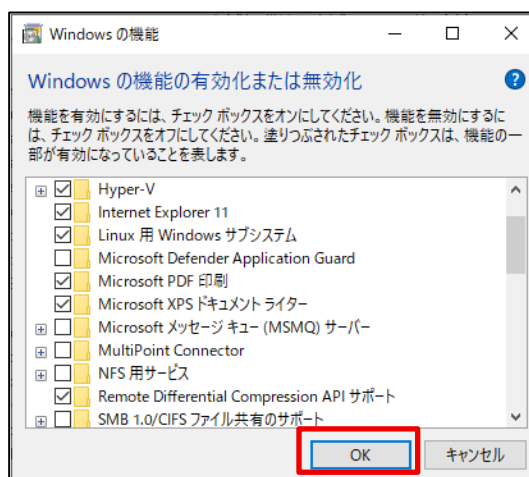


図 3-3 Windows の機能の有効化または無効化

- ④ 再起動後に設定が有効化されます。

## 4. インストール手順

Windows Subsystem for Linux 2(WSL2)および WSL2 上にインストールした Docker を使用する  
場合の橋梁診断支援 AI システムのインストール手順を以下に示します。

### 4.1. Windows Subsystem for Linux 2(WSL2)のインストール

Windows Subsystem for Linux には Windows Subsystem for Linux(WSL)と Windows  
Subsystem for Linux 2(WSL2)が存在しますが、Docker を使用するには WSL2 をインストールする  
必要があります。以下では WSL2のインストール手順について示します。

- ① 以下の URL から「x64 マシン用 WSL2Linux カーネル更新プログラムパッケージ」をダウンロードし  
ます。(図中赤枠からダウンロード可能)

URL: <https://aka.ms/wsl2kernel>

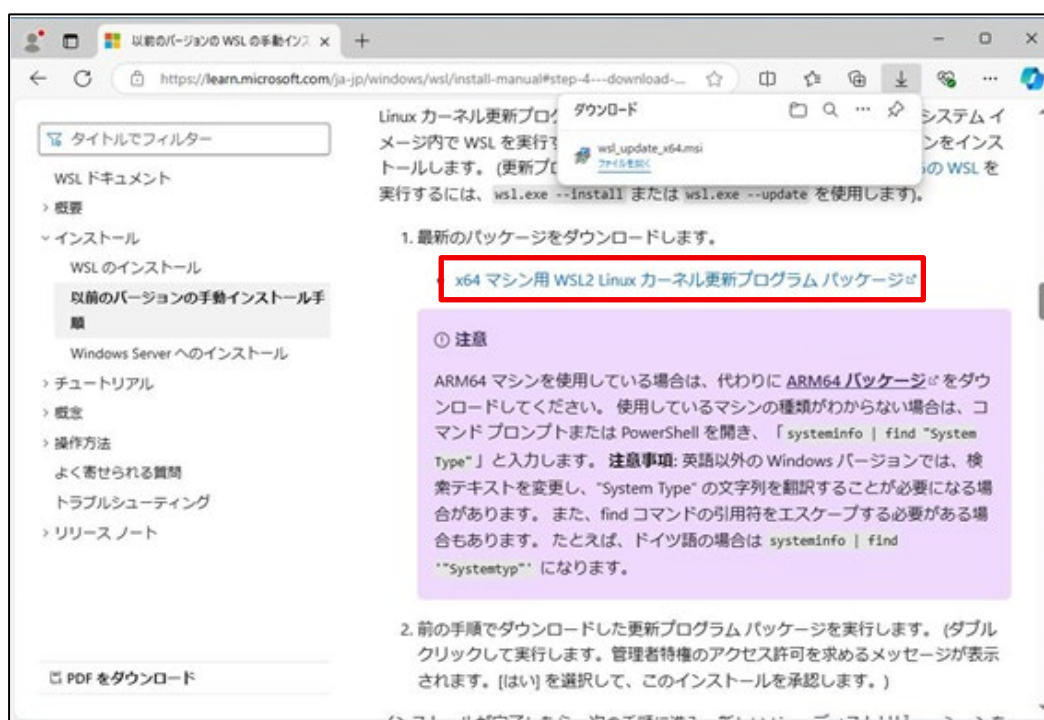


図 4-1 更新プログラムパッケージのダウンロード

- ② ダウンロードされた「wsl\_update\_x64.mis」を実行します。

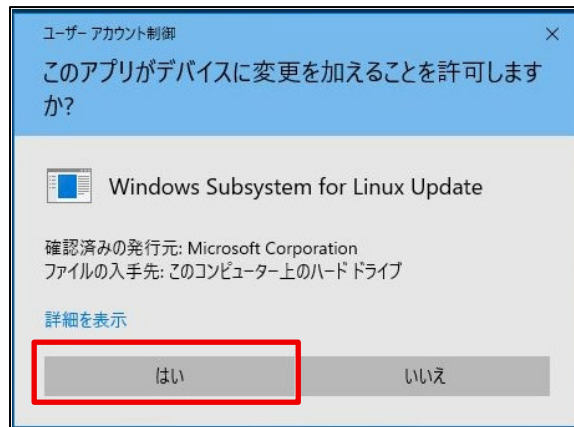


図 4-2 更新プログラムパッケージからの変更確認

変更を加えることの許可が要求されるので、「はい」を選択します。

「x64 マシン用 WSL2Linux カーネル更新プログラムパッケージ」のインストールが開始され、環境すると以下の画面が表示されるので、「Finish」を選択します。

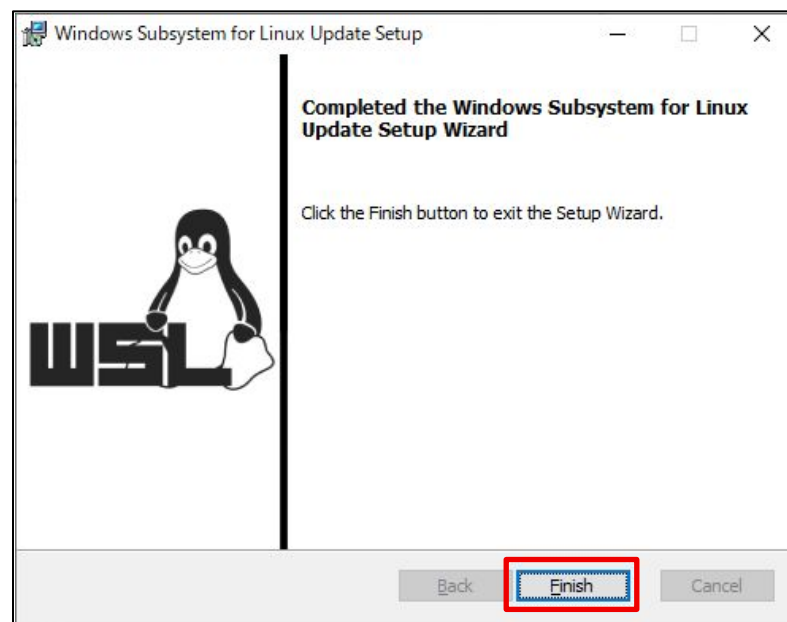


図 4-3 更新プログラムパッケージのインストール

- ③ 「コマンドプロンプト」を管理者権限で実行します。

検索フォームに「cmd」と入力すると「コマンドプロンプト」が候補として検索されるので、「管理者として実行」を選択します。

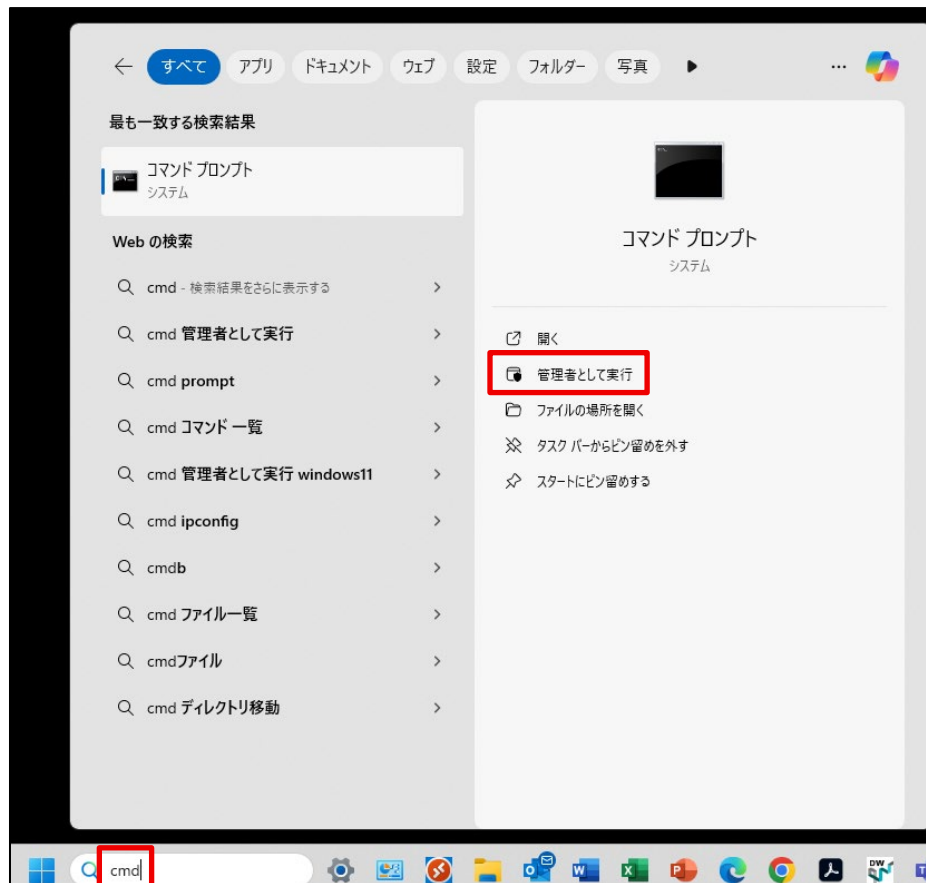


図 4-4 コマンドプロンプトの管理者権限での起動

- ④ 管理者権限で開いた「コマンドプロンプト」で以下のコマンドを実行し、WSL2 をデフォルトのバージョンに設定します。

コマンド: `wsl --set-default-version 2`

```

管理者: コマンドプロンプト
Microsoft Windows [Version 10.0.19045.5073]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>
C:\WINDOWS\system32>
C:\WINDOWS\system32>
C:\WINDOWS\system32>wsl --set-default-version 2
WSL 2 との主な違いについては、https://aka.ms/wsl2
を参照してください。
この操作を正しく終了しました。

C:\WINDOWS\system32>
  
```

図 4-5 WSL のデフォルトバージョンの設定

- ⑤ 「コマンドプロンプト」で以下のコマンドを実行し、WSL2 を更新します。

コマンド: `wsl --update`

- ⑥ 「コマンドプロンプト」で以下のコマンドを実行し、WSL2に「Ubuntu」をインストールします。  
最後の「Ubuntu」は先頭が大文字のUになります。

コマンド: `wsl --install -d Ubuntu`

- ⑦ コマンドを実行すると「Ubuntu」のインストールが開始し、最後に「Ubuntu」上でのユーザとパスワードの設定が求められますので、設定を行います。

```
C:\WINDOWS\system32>
C:\WINDOWS\system32>wsl --install -d Ubuntu
インストール中: Ubuntu
Ubuntu がインストールされました。
Ubuntu を起動しています...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username:
```

図 4-6 WSL 上への Ubuntu のインストール

- ⑧ ユーザとパスワードの設定が完了するとWSL2上で「Ubuntu」が起動します。



図 4-7 WSL 上の Ubuntu のコマンドプロンプト

以降、「Ubuntu」を起動するには、スタートメニューから「Ubuntu」を選択し、管理者権限で実行します。

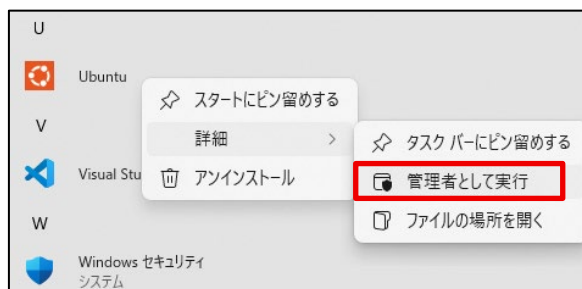


図 4-8 管理者権限での Ubuntu の起動

⑨ 端末のプロキシ設定を確認します。

Windows の場合は、「設定」-「ネットワークとインターネット」-「プロキシ」の順に開くことで、プロキシサーバの IP アドレス(または URL)とポート番号を確認できます。

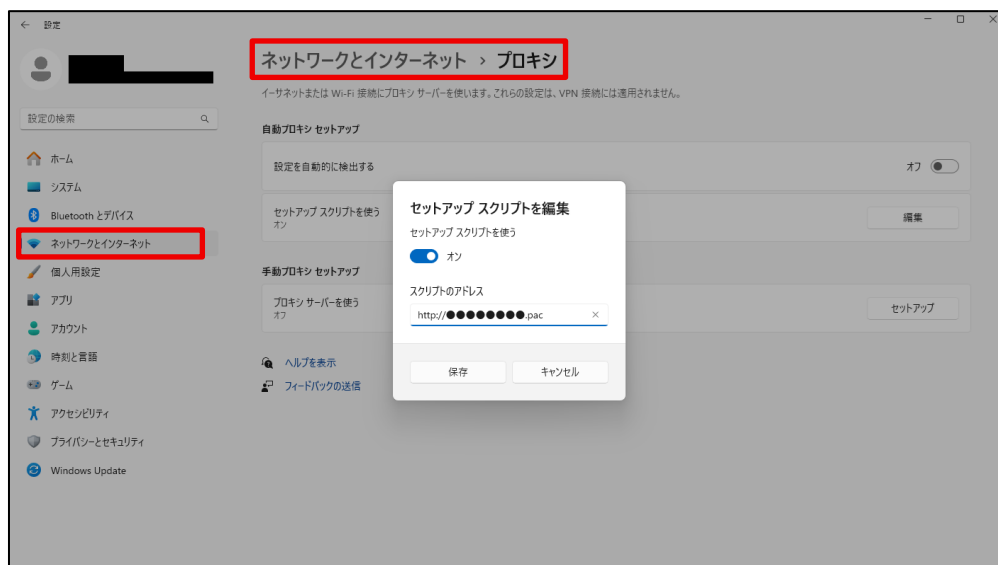


図 4-9 プロキシ設定の確認

また、自動構成スクリプト(<http://...pac>)などにより、サーバの IP アドレス(または URL)とポート番号が分からない場合は、システム管理者にご確認ください。

なお、本マニュアルの以降の節において、『プロキシ設定』と記載している場合、表 4-1 の最右列に記載する文字列を指しています。

表 4-1 プロキシ設定の記載の例

| プロキシサーバの指定方法  | プロキシサーバのアドレス                                                                 | ポート番号 | 『プロキシ設定』                                                                                            |
|---------------|------------------------------------------------------------------------------|-------|-----------------------------------------------------------------------------------------------------|
| IP アドレス       | 192.0.2.255                                                                  | 1080  | <a href="http://192.0.2.255:1080">http://192.0.2.255:1080</a>                                       |
| URL           | <a href="http://exampleproxy">http://exampleproxy</a>                        | 3128  | <a href="http://exampleproxy:3128">http://exampleproxy:3128</a>                                     |
| URL<br>(認証あり) | <a href="http://exampleproxy">http://exampleproxy</a><br>(username、password) | 8080  | <a href="http://username:password@exampleproxy:8080">http://username:password@exampleproxy:8080</a> |

⑩ Ubuntu を管理者として開き、以下のコマンドを実行して Ubuntu の設定ファイルを開きます。

コマンド: `sudo nano .bashrc`

- ⑪ ファイルの末尾に以下の文字列を追加し、Ctrl+X キーで上書き保存します。

```
export http_proxy="http://『プロキシ設定』"  
export https_proxy="http://『プロキシ設定』"  
export HTTP_PROXY="http://『プロキシ設定』"  
export HTTPS_PROXY="http://『プロキシ設定』"
```

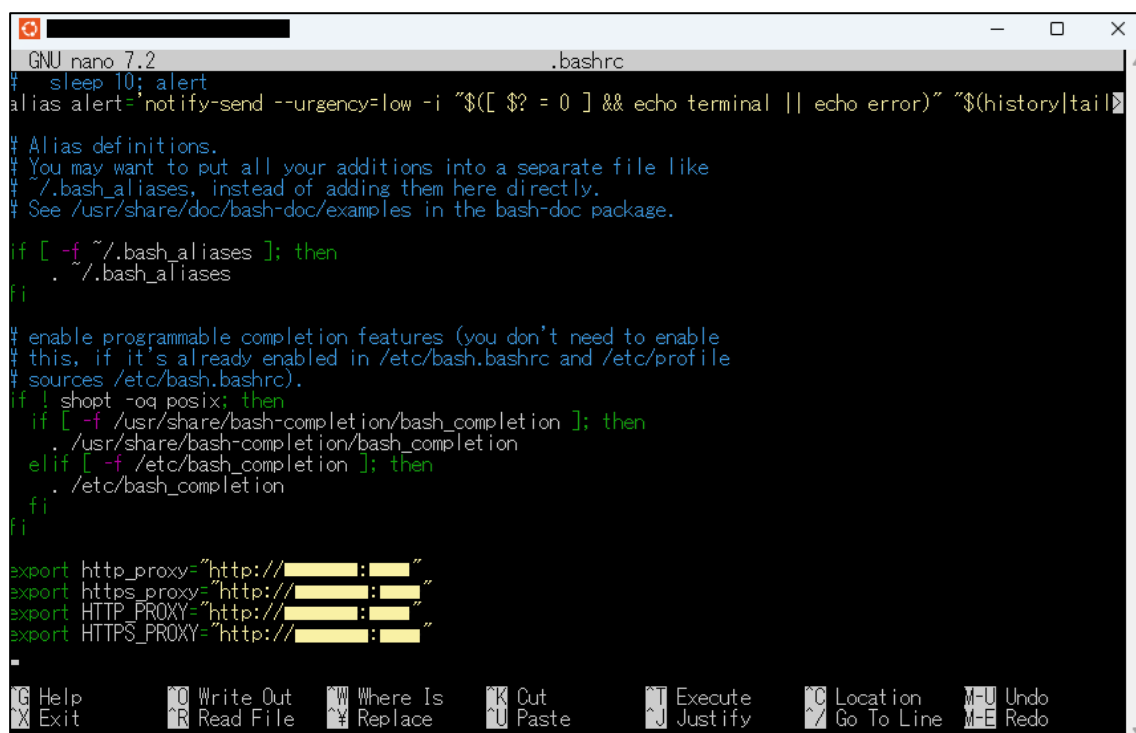


図 4-10 Ubuntu のプロキシ設定

- ⑫ 以下のコマンドで一度 Ubuntu を終了してから、再度管理者として起動します。

コマンド: exit

- ⑬ 以下のコマンドを実行して apt の設定ファイルを開きます。

コマンド: sudo nano /etc/apt/apt.conf

- ⑭ ファイルに以下の文字列を追加し、Ctrl+X キーで上書き保存します。

```
Acquire::http::Proxy "http://『プロキシ設定』";  
Acquire::https::Proxy "http://『プロキシ設定』";
```

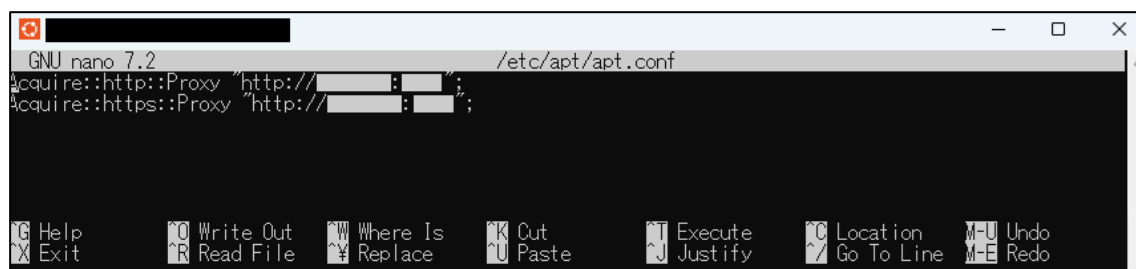


図 4-11 apt のプロキシ設定

- ⑮ 以下のコマンドを実行して apt をアップデートします。

```
コマンド: sudo apt update  
          sudo apt upgrade
```

- ⑯ 以下のコマンドを実行して curl の設定ファイルを開きます。

```
コマンド: sudo nano ~/.curlrc
```

- ⑰ ファイルに以下の文字列を追加し、Ctrl+X キーで上書き保存します。

```
proxy=http://『プロキシ設定』
```



図 4-12 curl のプロキシ設定

## 4.2. WSL2 への Docker のインストール

WSL2 上にインストールした「Ubuntu」への Docker インストール手順を示します。

- ① 「Ubuntu」上で以下のコマンドを実行し、パッケージの更新を行います。  
「sudo」で実行するため、実行時にパスワードの入力が求められるので、「Ubuntu」インストール時に設定したパスワードを入力します。「sudo」では一度パスワードを入力すると一定時間は再度実行してもパスワードの入力を求められません。以降「sudo」で実行時にパスワードの入力を求められた場合には、同様に「Ubuntu」インストール時に設定したパスワードを入力して下さい。

コマンド: `sudo apt-get update`

- ② 「Ubuntu」上で以下のコマンドを実行し、Docker のインストールに必要なパッケージのインストールを行います。インストール中にインストールを続行するか確認があるので、「Y」を選択し、インストールを続行します。

コマンド: `sudo apt-get install ca-certificates curl gnupg lsb-release`

- ③ 「Ubuntu」上で以下のコマンドを実行し、GPG キーを格納するフォルダを作成します。

コマンド: `sudo mkdir -p /etc/apt/keyrings`

- ④ 「Ubuntu」上で以下のコマンドを実行し、GPG キーを追加します。

コマンド: `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg`

- ⑤ 「Ubuntu」上で以下のコマンドを実行し、GPG キーのセットアップを行います。

コマンド: `echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null`

- ⑥ 「Ubuntu」上で以下のコマンドを実行し、GPG キーのセットアップ後に再度パッケージの更新を行います。

コマンド: `sudo apt-get update`

- ⑦ 「Ubuntu」上で以下のコマンドを実行し、Docker をインストールします。  
インストール中にインストールを続行するか確認があるので、「Y」を選択しインストールを続行します。

コマンド: `sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin`

- ⑧ 「Ubuntu」上で以下のコマンドを実行し、Docker-compose をインストールします。インストール中にインストールを続行するか確認があるので、「Y」を選択し、インストールを続行します。

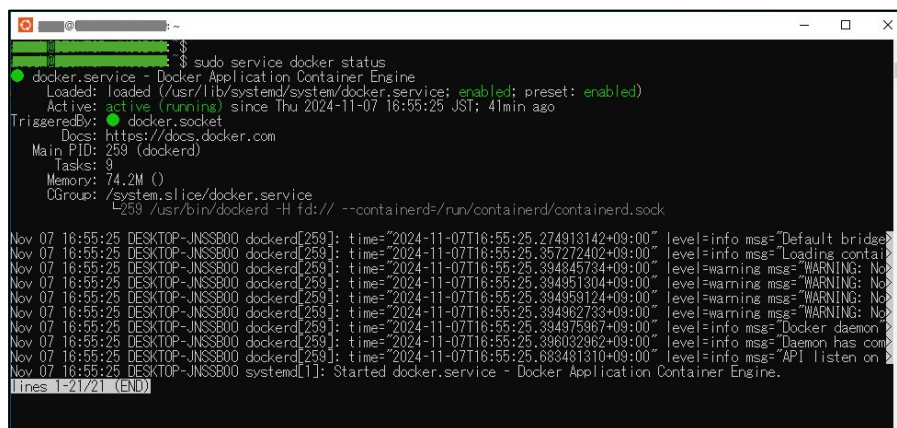
コマンド: `sudo apt install docker-compose`

- ⑨ 「Ubuntu」上で以下のコマンドを実行し、Docker のサービスを起動します。

コマンド: `sudo service docker start`

- ⑩ 「Ubuntu」上で以下のコマンドを実行し、Docker サービスの状態を確認します。

コマンド: `sudo service docker status`

A terminal window showing the command `sudo service docker status` and its output. The output indicates that the Docker service is loaded and active. It also shows the Docker daemon logs, which include warnings about the default bridge network and information about the daemon's status.

```
$ sudo service docker status
docker.service - Docker Application Container Engine
Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
Active: active (running) since Thu 2024-11-07 16:55:25 JST; 41min ago
TriggeredBy: ● docker.socket
Docs: https://docs.docker.com
Main PID: 259 (dockerd)
Tasks: 9
Memory: 74.2M ()
CGroup: /system.slice/docker.service
        └─259 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 07 16:55:25 DESKTOP-JNSSEB00 dockerd[259]: time="2024-11-07T16:55:25.274913142+09:00" level=info msg="Default bridge
Nov 07 16:55:25 DESKTOP-JNSSEB00 dockerd[259]: time="2024-11-07T16:55:25.357272402+09:00" level=info msg="Loading contain
Nov 07 16:55:25 DESKTOP-JNSSEB00 dockerd[259]: time="2024-11-07T16:55:25.394845734+09:00" level=warning msg="WARNING: No
Nov 07 16:55:25 DESKTOP-JNSSEB00 dockerd[259]: time="2024-11-07T16:55:25.394951304+09:00" level=warning msg="WARNING: No
Nov 07 16:55:25 DESKTOP-JNSSEB00 dockerd[259]: time="2024-11-07T16:55:25.394959124+09:00" level=warning msg="WARNING: No
Nov 07 16:55:25 DESKTOP-JNSSEB00 dockerd[259]: time="2024-11-07T16:55:25.394962733+09:00" level=warning msg="WARNING: No
Nov 07 16:55:25 DESKTOP-JNSSEB00 dockerd[259]: time="2024-11-07T16:55:25.394975967+09:00" level=info msg="Docker daemon
Nov 07 16:55:25 DESKTOP-JNSSEB00 dockerd[259]: time="2024-11-07T16:55:25.396032962+09:00" level=info msg="Daemon has com
Nov 07 16:55:25 DESKTOP-JNSSEB00 dockerd[259]: time="2024-11-07T16:55:25.883481310+09:00" level=info msg="API listen on 2
Nov 07 16:55:25 DESKTOP-JNSSEB00 systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-21/21 (END)
```

図 4-13 Docker サービスの起動確認

Active の行で「active(running)」となっていれば、Docker サービスの起動が完了です。(Q キーで閉じることができます。)

以上で、WSL2 上にインストールした「Ubuntu」への Docker インストールは完了です。

コンテナの作成を行うため、引き続き「Ubuntu」上で以下の橋梁診断支援 AI システムのインストール作業を行います。

### 4.3. 橋梁診断支援 AI システムのインストール

橋梁診断支援 AI システムのインストール手順について以下に示します。

- ① 「Ubuntu」を管理者権限で起動し、「Ubuntu」上で以下のコマンドを実行して、インストール時に作成したユーザのホームフォルダに移動します。

コマンド: `cd`

- ② 「Ubuntu」上で以下のコマンドを実行し、ホームフォルダにコンテナインストール用のフォルダを作成します。

コマンド: `mkdir docker`

- ③ Windows のエクスプローラーを開き、アドレスバーに以下のパスを入力します。

パス: `¥¥wsl$`



図 4-14 Ubuntu 上のファイルへのアクセス

これにより Windows 上から WSL2上の「Ubuntu」配下のファイルやフォルダにエクスプローラーからアクセスできるようになります。

- ④ エクスプローラーから「Ubuntu」上の以下のパスにアクセスします。

パス: `¥¥wsl.localhost¥Ubuntu¥home¥『ユーザ』¥docker`

ここで、『ユーザ』の部分は、4.1.⑦において「Ubuntu」上で作成したユーザ名になります。

これにより、事前に「Ubuntu」上で作成した docker フォルダに Windows 上からアクセスすることができます。

- ⑤ 「docker\_橋梁診断支援 AI システム」フォルダを、「Ubuntu」上で作成した docker フォルダにコピーします。

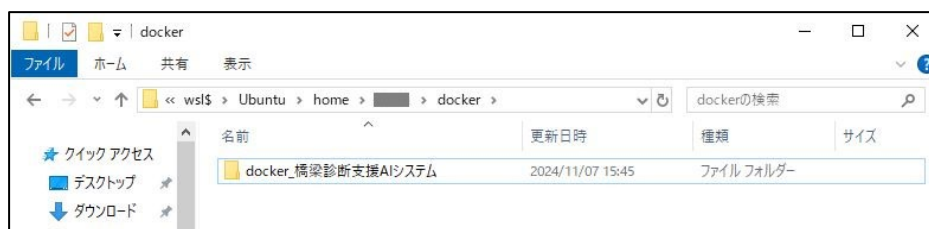


図 4-15 橋梁診断支援 AI システムの配置

- ⑥ 以下のコマンドを実行して docker のプロキシ設定ファイルを作成します。

```
コマンド: sudo mkdir /etc/systemd/system/docker.service.d
          sudo nano /etc/systemd/system/docker.service.d/http-proxy.conf
```

- ⑦ ファイルに以下の文字列を追加し、Ctrl+X キーで上書き保存します。

```
[Service]
Environment="HTTP_PROXY=http://『プロキシ設定』"
Environment="HTTPS_PROXY=http://『プロキシ設定』"
```

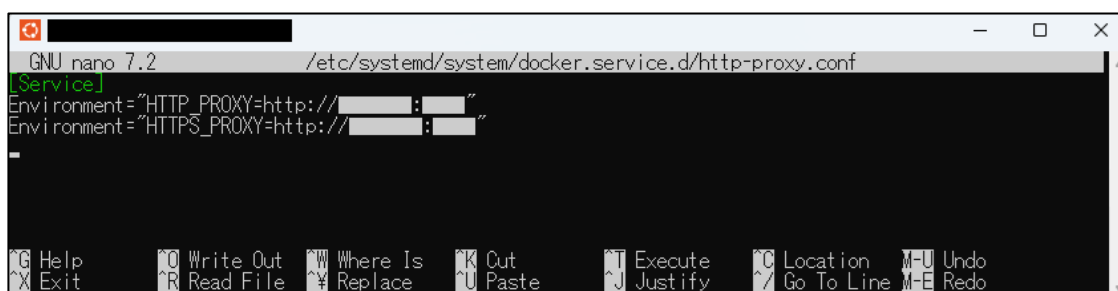


図 4-16 docker のプロキシ設定

- ⑧ 以下のコマンドを実行して docker compose の設定ファイルを開きます。

```
コマンド: sudo nano ~/docker/docker_橋梁診断支援 AI システム/appcont/docker-compose.yml
```

- ⑨ yml ファイルの「dockerfile: Dockerfile」の下に、以下の黒色文字で記載した文字列を追加します。このとき、「dockerfile:」と「args:」の書き出し位置、及び「build:」と「environment:」の書き出し位置を、必ず合わせてください。追加したら Ctrl+X キーで上書き保存します。

```
build:
  context: ../WebApp
  dockerfile: Dockerfile
# ビルド中にプロキシを使うための設定
args:
  - http_proxy=$http_proxy
  - https_proxy=$https_proxy
  - HTTP_PROXY=$HTTP_PROXY
  - HTTPS_PROXY=$HTTPS_PROXY
# 立ち上がったコンテナでプロキシを使うための設定
environment:
  - http_proxy=$http_proxy
  - https_proxy=$https_proxy
  - HTTP_PROXY=$HTTP_PROXY
  - HTTPS_PROXY=$HTTPS_PROXY
```

```
GNU nano 7.2 /home/.../docker/docker 橋梁診断支援AIシステム/appcont/docker-compose.yml
#version: "3.0"

services:
  # PHP + Apache
  webapp:
    container_name: webapp
    # ../WebApp/Dockerfile でイメージをビルド
    build:
      context: ../WebApp
      dockerfile: Dockerfile
      # ビルド中にプロキシを使うための設定
      args:
        - http_proxy=$http_proxy
        - https_proxy=$https_proxy
        - HTTP_PROXY=$HTTP_PROXY
        - HTTPS_PROXY=$HTTPS_PROXY
      # 立ち上がったコンテナでプロキシを使うための設定
      environment:
        - http_proxy=$http_proxy
        - https_proxy=$https_proxy
        - HTTP_PROXY=$HTTP_PROXY
        - HTTPS_PROXY=$HTTPS_PROXY

    # ローカルの '../html' ディレクトリをコンテナの '/var/www/html' ディレクトリにマウント (コンテンツ配置用)
    volumes:
      - ../html:/var/www/html
    # ホストの80番ポートをコンテナの80番ポートに割り当て
    ports:
      - 80:80
```

図 4-17 docker compose の設定

- ⑩ 以下のコマンドを実行して docker compose のプロキシ設定ファイルを作成します。

コマンド: `sudo nano ~/docker/docker_橋梁診断支援 AI システム/appcont/.env`

- ⑪ ファイルに以下の文字列を追加し、Ctrl+X キーで上書き保存します。

```
http_proxy=http://『プロキシ設定』
https_proxy=http://『プロキシ設定』
HTTP_PROXY=http://『プロキシ設定』
HTTPS_PROXY=http://『プロキシ設定』
```



図 4-18 docker compose のプロキシ設定

- ⑫ 以下のコマンドを実行して docker を再起動します。

コマンド: `systemctl daemon-reload`  
`systemctl restart docker`

- ⑬ 「Ubuntu」上で以下のコマンドを実行し、Windows 上からコピーしたフォルダに移動します。これによりコピーした「docker\_橋梁診断支援 AI システム」フォルダ配下の「appcont」フォルダに移動します。

コマンド: `cd docker/docker_橋梁診断支援 AI システム/appcont`

「Ubuntu」上で以下のコマンドを実行し、コンテナを作成します。コンテナの作成にはしばらく時間が掛かります。なお、「sudo」で実行するため、実行時にパスワードの入力が求められるので、「Ubuntu」インストール時に設定したパスワードを入力します。「sudo」では一度パスワードを入力すると一定時間は再度実行してもパスワードの入力を求められません。以降「sudo」で実行時にパスワードの入力を求められた場合には、同様に「Ubuntu」インストール時に設定したパスワードを入力して下さい。

コマンド: `sudo docker compose up -d`

```
WARN[0000] /home/.../docker/docker_橋梁診断支援AIシステム/appcont: the attribute `version` is obso
etc, it will be ignored, please remove it to avoid potential confusion
[+] Running 1/14
[+] Running 2/14
[+] Running 2/14e Downloading [=====] 4.422MB/... 7.3s
[+] Running 2/14e Downloading [=====] 12.07MB/... 7.4s
[+] Running 2/14e Downloading [=====] 229.4kB/2... 1.8s
[+] Running 2/14e Extracting [====] 2.523MB/2... 7.5s
[+] Running 2/14e Extracting [====] 2.523MB/2... 1.9s
[+] Running 2/14e Extracting [====] 2.523MB/2... 7.6s
[+] Running 2/14e Extracting [====] 2.523MB/2... 2.0s
[+] Running 2/14e Extracting [====] 2.523MB/2... 7.7s
[+] Running 2/14e Extracting [====] 2.523MB/2... 2.1s
[+] Running 2/14e Extracting [====] 2.523MB/2... 7.8s
[+] Running 2/14e Extracting [====] 2.523MB/2... 2.2s
[+] Running 2/14e Extracting [====] 2.523MB/2... 7.9s
```

図 4-19 橋梁診断支援 AI システムのインストール

- ⑭ コンテナの作成が完了すると、自動的に橋梁診断支援 AI システムのコンテナが起動します。

```
@ ~:~/docker/docker_橋梁診断支援AIシステム/appcont
=> extracting sha256:8b5d88f73ad62e88c0c22819eed-bc7eda30bdf0c1e171cabcaf81cb8b1c40cb 0.0s
=> extracting sha256:d2c43c5efbc861f83ee6565c7102ca660d8f35e158324fbb042de5017e43afe8 1.0s
=> extracting sha256:ab590b48ea476386dd7b07c34de9eff7cf2103c4668ade985fe31e59f15deee8 0.0s
=> extracting sha256:80682ae2d067c8358112c56490a2a97f69ef395fd8f7662a31498644c9a813ef 0.0s
=> extracting sha256:05e465aaa98a358add4acecdade8f39843089069f31fa0201533d3a09a98c9a 0.0s
[webapp] RUN apt-get update && apt-get install -y libpq-dev libonig-dev libpng-dev libzip-dev && dock 32.8s
[webapp] exporting to image 0.2s
=> exporting layers 0.1s
=> writing image sha256:302d8b3f3ecd8ee0521c7f8e9e82f437a6f973a5e123bf5b6bbb16aab68d4a9b 0.0s
=> naming to docker.io/library/appcont-webapp 0.0s
[webapp] resolving provenance for metadata file 0.0s
[+] Running 1/1
[+] Running 1/3cont default Created 0.2s
[+] Running 1/3cont default Created 0.2s
[+] Running 1/3cont default Created 0.2s
[+] Running 1/3cont default Created 0.2s
[+] Running 1/3cont default Created 0.2s
[+] Running 1/3cont default Created 0.2s
[+] Running 3/3cont default Created 0.2s
[+] Network appcont_default Created 0.2s
[+] Container appdb Starting 0.6s
[+] Container appdb Started 0.7s
[+] Container webapp Started 0.7s
```

図 4-20 橋梁診断支援 AI システムのインストール完了

- ⑮ 「Ubuntu」上で以下のコマンドを実行し、起動したコンテナを一度停止します。
- コンテナ作成後、一度停止せずに橋梁診断支援 AI システムにアクセスすると、コンテナの起動順の関係で Web アプリケーションから DB に上手くアクセスできず、システム内でエラーが発生します。

コマンド: `sudo docker compose stop`



以降は、「Ubuntu」上で「sudo docker compose up -d」と「sudo docker compose stop」を実行することで、橋梁診断支援 AI システムの起動および停止を行う事ができます。

橋梁診断支援 AI システムは、「webapp」コンテナと、「appdb」コンテナから構成されます。ここで、「webapp」コンテナは PHP を含む Apache Webサーバコンテナであり、「appdb」コンテナは PostgreSQL DBコンテナです。

なお、「docker\_橋梁診断支援 AI システム」フォルダ内の「html」フォルダが「webapp」コンテナと結びつき、橋梁診断支援 AI システムのコンテンツフォルダになります。

また、「docker\_橋梁診断支援 AI システム」フォルダ配下の「appcont」フォルダ配下の「dbdata」フォルダが「appdb」コンテナと結びつき、PostgreSQL のデータフォルダとなります。

上述のように、どちらのフォルダも Windows 上からエクスプローラーを介してアクセス可能です。

以上で、橋梁診断支援 AI システムのインストールおよび起動は完了です。ブラウザからアクセスすることで橋梁診断支援 AI システムを使用することができます。

#### 4.4. 動作確認

橋梁診断支援 AI システムの動作確認を行うには、ブラウザから以下の URL にアクセスします。橋梁診断支援 AI システムのトップ画面が表示されればインストールおよび起動は成功です。

URL: <http://localhost>

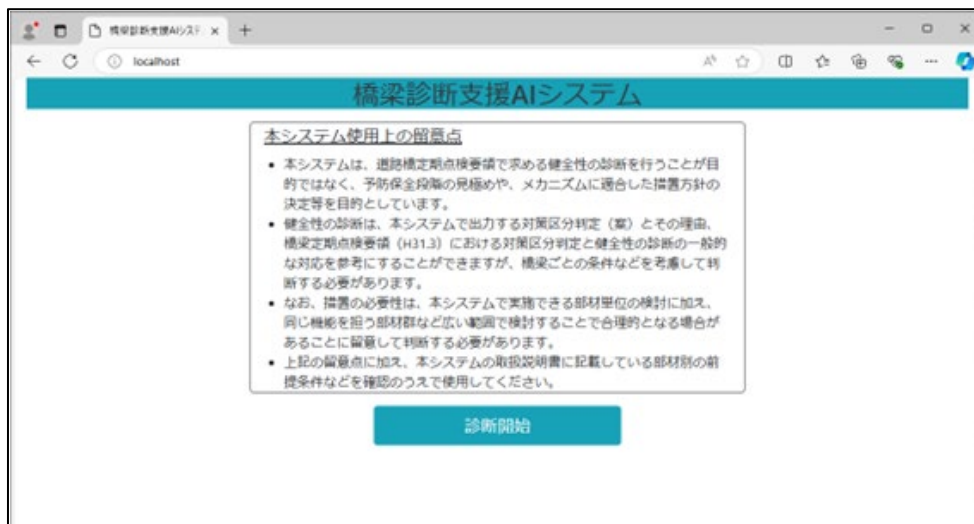


図 4-24 橋梁診断支援 AI システムのトップ画面

## 5. 起動・停止方法

WSL2 および WSL2 上にインストールした Docker を使用する場合の橋梁診断支援 AI システムの起動・停止方法を以下に示します。

### 5.1. 橋梁診断支援 AI システムの起動

WSL2 および WSL2 上にインストールした Docker を使用する場合、橋梁診断支援 AI システムの起動および停止は「Ubuntu」上でのコマンドラインでの操作となります。

WSL2 上の「Ubuntu」を起動するには、Windows のスタートメニューから「Ubuntu」を選択し、管理者権限で実行します。

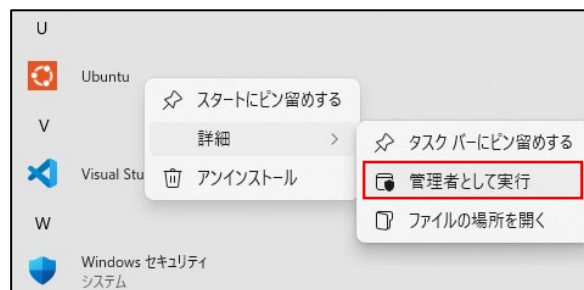


図 5-1 Ubuntu の管理者権限での起動

これにより WSL2 上の「Ubuntu」が起動し、図 5-1 に示すような「Ubuntu」のプロンプトが表示されます。



図 5-2 WSL2 上の Ubuntu のコマンドプロンプト

橋梁診断支援 AI システムの起動および停止はこのプロンプトからコマンド操作で実行します。

橋梁診断支援 AI システムを起動するには、まず、「Ubuntu」上で以下のコマンドを実行し、「appcont」フォルダへ移動します。

コマンド: `cd ~/docker/docker_橋梁診断支援 AI システム/appcont`

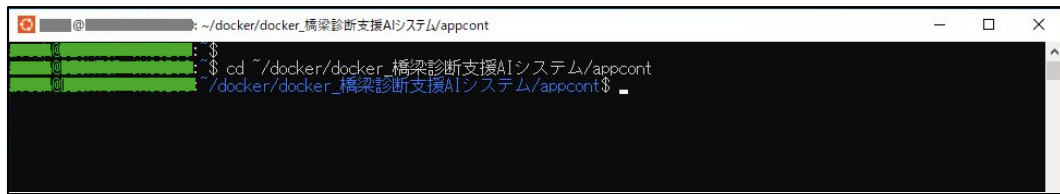


図 5-3 「appcont」フォルダへの移動

「appcont」フォルダに移動後、「Ubuntu」上で以下のコマンドを実行し、橋梁診断支援 AI システムを起動します。

コマンド: `sudo docker compose up -d`

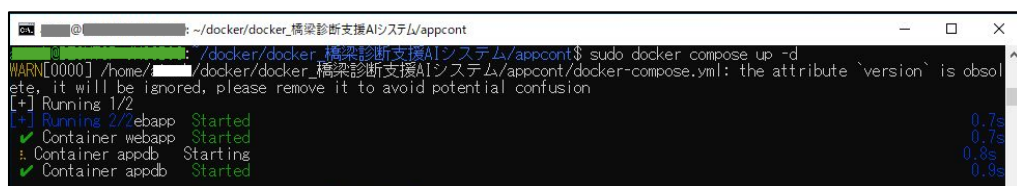


図 5-4 橋梁診断支援 AI システムの起動

橋梁診断支援 AI システムを起動後、「Ubuntu」上で以下のコマンドを実行し、橋梁診断支援 AI システムの起動状況を確認します。

コマンド: `sudo docker compose ps --format "{{.Service}} {{.State}}"`



図 5-5 橋梁診断支援 AI システムの起動確認

コマンド実行後、「webapp」コンテナと「appdb」コンテナがどちらも「running」になっていれば橋梁診断支援 AI システムの起動が成功しています。

ブラウザから橋梁診断支援 AI システムにアクセス可能な状態となっています。

なお、「exit」コマンドで、「Ubuntu」から抜け出す、もしくは「Ubuntu」のコマンドプロンプトを終了しても、橋梁診断支援 AI システムの各コンテナは動作を継続しますが、Windows を再起動した場合、各コンテナが停止するため、再度コンテナを起動する必要があります。

また、「Ubuntu」上で以下のコマンドを実行することで、橋梁診断支援 AI システムの詳細な状況をログから確認することができます。

コマンド: `sudo docker compose logs -f -t`

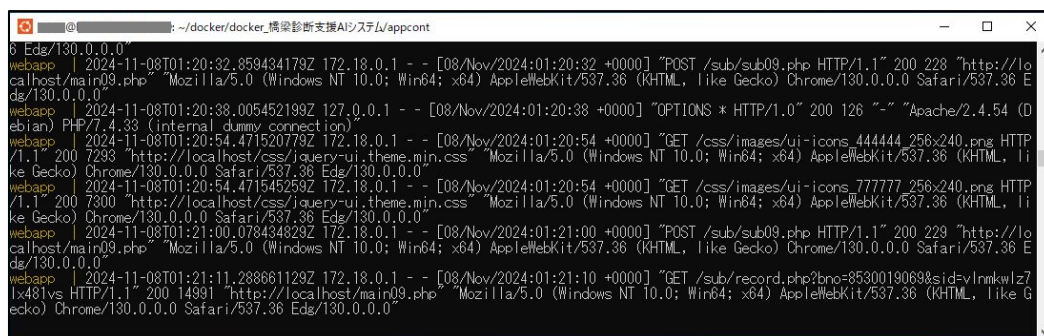


図 5-6 橋梁診断支援 AI システムのログ情報

## 5.2. 橋梁診断支援 AI システムの停止

橋梁診断支援 AI システムを起動するには、まず、「Ubuntu」上で以下のコマンドを実行し、「appcont」フォルダへ移動します。

コマンド: `cd ~/docker/docker_橋梁診断支援 AI システム/appcont`



図 5-7 橋梁診断支援 AI システムの「appcont」フォルダ

「appcont」フォルダへの移動が完了したら、「Ubuntu」上で以下のコマンドを実行し、橋梁診断支援 AI システムを停止します。

コマンド: `sudo docker compose stop`

```
~/docker/docker_橋梁診断支援AIシステム/appcont$ sudo docker compose stop
WARN[0000] /home/~/docker/docker_橋梁診断支援AIシステム/appcont/docker-compose.yml: the attribute `version` is obso
ete, it will be ignored, please remove it to avoid potential confusion
[+] Stopping 1/0
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 1/2pdb Stopped 0.0s
[+] Stopping 2/2pdb Stopped 0.0s
[+] Container appdb Stopped 0.0s
[+] Container webapp Stopping 1.5s
[+] Container webapp Stopped 1.5s
~/docker/docker_橋梁診断支援AIシステム/appcont$
~/docker/docker_橋梁診断支援AIシステム/appcont$
~/docker/docker_橋梁診断支援AIシステム/appcont$
```

図 5-8 橋梁診断支援 AI システムの停止

## 6. コンテナとホストシステム

WSL2 上に構築した Docker 環境にインストールした橋梁診断支援 AI システムとホストシステム (Windows) との関係について以下に示します。

### 6.1. コンテナとホストシステムの関係

上述のように、橋梁診断支援 AI システムをインストールすると appcont コンテナがインストールされます。また、appcont コンテナは以下の 2 つのコンテナから構成されています。

(webapp コンテナ)

- ・Webサーバ機能を担当するコンテナ
- ・PHPを使用可能なApacheが動作
- ・Apacheのドキュメントルートがホストシステム上のフォルダにマウント
- ・ホストシステムの 80 番ポートがコンテナの 80 番ポートとマッピング
- ・ホストシステム上のブラウザから「http://localhost」にアクセスすることで接続可能

(appdbコンテナ)

- ・DB機能を担当するコンテナ
- ・PostgreSQLが動作
- ・コンテナ構築時にホストシステム上のバックアップデータからDBを復元
- ・コンテナ上のDBのデータ格納フォルダをホストシステム上のフォルダにマウント
- ・ホストシステムの 5432 番ポートをコンテナの 5432 番ポートとマッピング
- ・ホストシステムの 5432 番ポートからDBに接続可能

図 6-1 に Docker 環境上のコンテナとホストシステムの関係を示します。

図 6-1 に示すように、webapp コンテナの 80 番ポートがホストシステムの 80 番ポートとマッピングされています。また、appdb コンテナの 5432 番ポートがホストシステムの 5432 番ポートにマッピングされています。

WSL2 を使用している場合、より正確には、webapp コンテナの 80 番ポートは WSL2 (Ubuntu 等) の 80 番ポートと、appdb コンテナの 5432 番ポートは WSL2 (Ubuntu 等) の 5432 番ポートとマッピングされますが、WSL2 のポートはホストシステム (Windows) の同じ番号のポートにマッピングされているため、ホストシステムのポートにマッピングされる事と同じ意味となります。

そのため、ホストシステムの 80 番ポートから webapp コンテナの 80 番ポートに、ホストシステムの 5432 番ポートから appdb コンテナの 5432 番ポートに接続することが可能となっています。これにより、ホストシステム (localhost) に Web ブラウザからアクセスすることで、webapp コンテナ上の橋梁診断支援

AIシステムにアクセス可能であり、また、ホストの 5432 番ポートを経由して appdb コンテナ上の PostgreSQL に接続可能となります。

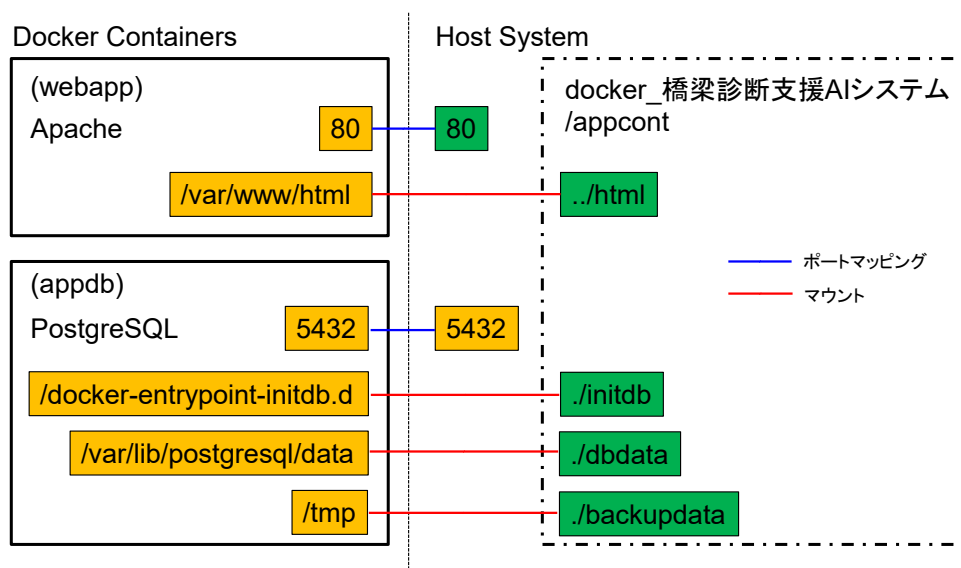


図 6-1 橋梁診断支援 AI システムとホストシステムの関係

また、ホストシステム上の「docker\_橋梁診断支援AIシステム」フォルダの「html」フォルダが、webapp コンテナ上の Apache の Document Root である「/var/www/html」フォルダにマウントされています。これによりホストシステム上のファイルを編集することで、webapp コンテナのWebアプリケーションの内容を変更することが可能となります。

同様に、ホストシステム上の「./initdb」フォルダが appdb コンテナ上の「/docker-entrypoint-initdb.d」フォルダに、ホストシステム上の「./dbdata」フォルダが appdb コンテナ上の「/var/lib/postgresql/data」フォルダに、ホストシステム上の「./backupdata」フォルダが appdb コンテナ上の「/tmp」フォルダに、それぞれマウントされています。これにより、ホストシステム上にDBのデータが保持され、コンテナ終了時にデータが削除されることはありません。

## 6.2. ポートの競合

上述のように、橋梁診断支援 AI システムではホストシステム(Windows)のポートと各コンテナのポートをマッピングすることで、ホストシステム上から各コンテナへのアクセスが可能となります。

一方で、コンテナのポートとマッピングするポートがホストシステムで既に使用済みの場合、コンテナ起動時にポートマッピングが失敗し、コンテナの起動エラーとなってしまいます。

橋梁診断支援 AI システムでは、webapp コンテナと appdb コンテナはそれぞれ、Apache と PostgreSQL の標準ポートをマッピング先のポートとして使用しています。そのため、例えば、ホストシステム上でApache等のWebサーバを動作させている場合や、ホスト上でPostgreSQLを動作させている場合には、ポートが使用されているためポートマッピングが失敗することがあります。

この場合、ホスト上のポートが競合しているプログラムを停止するか、コンテナとマッピングするポートの設定を変更する必要があります。以下、ポートが競合した場合の回避方法について示します。

### 6.2.1. ポート競合の確認

ポートの競合が発生するかを確認するにはホストシステム(Windows)のコマンドプロンプトで以下のコマンドを実行してください。

コマンド: `netstat -ano | findstr ":ポート番号"`

ここで、ポート番号には競合を確認したいポートの番号を設定します。webapp との競合を確認する場合には 80、appdb との競合を確認する場合には 5432 を設定します。

コマンドを実行した結果、以下のようにリストが表示される場合、使用ポートの競合が発生しています。以下は 5432 番ポートの競合を確認した場合の出力例です。

|     |              |            |             |      |
|-----|--------------|------------|-------------|------|
| TCP | 0.0.0.0:5432 | 0.0.0.0:0  | LISTENING   | 6656 |
| TCP | :::5432      | :::0       | LISTENING   | 6656 |
| TCP | :::1:5432    | :::1:49723 | ESTABLISHED | 6656 |
| TCP | :::1:49723   | :::1:5432  | ESTABLISHED | 7148 |

図 6-2 ポート競合時の netstat コマンド実行時の一例

リストが表示された場合には、PID に相当するリストの最後の列の数字(リストの一番上行の例であれば 6656)を使用して以下のコマンドを実行してください。

コマンド: `tasklist /FI "PID eq 最後の列の数字"`

コマンドを実行すると、以下のように指定した PID に該当するプログラムの情報が表示されます。以下の例では、PID 6656 に該当するプログラムは postgres.exe となります。

| イメージ名        | PID  | セッション名   | セッション# | メモリ使用量   |
|--------------|------|----------|--------|----------|
| =====        |      |          |        |          |
| postgres.exe | 6656 | Services | 0      | 22,244 K |

図 6-3 ポート競合プログラムの確認

このように、ホストシステム上のコマンドプロンプトでコマンドを実行することで、ホストシステム上で橋梁診断支援 AI システムのコンテナとポートが競合するプログラムが動作しているかを確認することができます。

### 6.2.2. 競合プログラムを停止する場合

ポートの競合を回避する最も簡単な方法は、ホストシステム上で動作しているポートが競合するプログラムを停止することです。

6.2.1 に示した方法で、競合するプログラムを確認後、該当するプログラムをホストシステム上で停止します。6.2.1 に示した例では、postgres.exe が 5432 番ポートを使用しているため、appdb コンテナとポートが競合するホスト上の PostgreSQL を停止します。

競合するプログラムの停止後、「docker\_橋梁診断支援AIシステム」フォルダの「appcont」フォルダで再度「docker compose up -d」コマンドを実行します。これによりポートの競合を回避して、橋梁診断支援AIシステムの各コンテナを起動することができます。

### 6.2.3. マッピングするポートを変更する場合

ホストシステム上の競合するプログラムを停止することが難しい場合、橋梁診断支援AIシステムの各コンテナとマッピングするホストシステムのポートを変更する必要があります。

マッピングするポートを変更するには、「docker\_橋梁診断支援AIシステム」フォルダの「appcont」フォルダにある「docker-compose.yml」ファイルを編集する必要があります。なお、「docker-compose.yml」ファイルを編集する場合、事前に、「appcont」フォルダで「docker compose stop」コマンドを実行し、各コンテナを停止してから以下の作業を行ってください。

(webapp コンテナの設定を変更する場合)

「docker-compose.yml」ファイルの 16 行目にある以下の行の左側の値を編集します。

- "80:80"

ここで左側の値がホストシステム側のポート番号、右側の値がコンテナ側のポート番号となります。例えば以下のように編集した場合、ホストシステム上の 8080 番ポートとコンテナの 80 番ポートとがマッピングされることになります。

- "8080:80"

この場合、ホストシステム上のブラウザから webapp コンテナの Web サーバにアクセスするには「http://localhost:8080/」のようにポート番号まで指定する必要があります。

設定変更後、「appcont」フォルダで「docker compose up -d」コマンドを実行することで、ポートの競合を回避して、コンテナを起動することができます。

(appdb コンテナの設定を変更する場合)

「docker-compose.yml」ファイルの 36 行目にある以下の行の左側の値を編集します。

- "5432:5432"

ここで左側の値がホストシステム側のポート番号、右側の値がコンテナ側のポート番号となります。例えば以下のように編集した場合、ホストシステム上の 5431 番ポートとコンテナの 5432 番ポートとがマッピングされることになります。

- "5431:5432"

この場合、ホストシステムから appdb コンテナの PostgreSQL にアクセスするには接続するポート番号として 5431 を指定する必要があります。

設定変更後、「appcont」フォルダで「docker compose up -d」コマンドを実行することで、ポートの競合を回避して、コンテナを起動することができます。

以上のように、「docker-compose.yml」ファイルを編集し、コンテナとマッピングするホストシステムのポートを変更することで、ホスト上の既存のプログラムを動作させた状態で、ポートの競合を回避して橋梁診断支援AIシステムを起動することができます。

なお、上述のようにマッピングするポートを変更した場合、ブラウザから橋梁診断支援AIシステムにアクセスする際の URL が以下のように変更されます。

URL: <http://localhost:ポート番号/>

ここで、ポート番号は変更したポート番号の値(上の例であれば 8080)となります。

また、DB に接続する際に指定するポート番号も変更したポート番号の値(上の例であれば 5431)を指定する必要があります。