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)

令和6年12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	4-7
4.	1. Windows Subsystem for Linux 2(WSL2)のインストール	4-7
4.	2. WSL2 への Docker のインストール	4-11
4.	3. 橋梁診断支援 AI システムのインストール	4-14
4.	4. 動作確認	4-20
5.	起動·停止方法	5-21
5.	1. 橋梁診断支援 AI システムの起動	5-21
5.	2. 橋梁診断支援 AI システムの停止	5-23
6.	コンテナとホストシステム	6-25
6.	1. コンテナとホストシステムの関係	6-25
6.	2. ポートの競合	6-26
(6.2.1. ポート競合の確認	. 6-27
(6.2.2. 競合プログラムを停止する場合	. 6-28
(6.2.3. マッピングするポートを変更する場合	. 6-28

図 2-1 システムコンテナ化イメージ2-2	2
図 3-1 Windowsの設定画面の起動3-3	3
図 3-2 Windowsの設定3-4	1
図 3-3 アプリと機能	1
図 3-4 プログラムと機能3-5	5
図 3-5 Windowsの機能の有効化または無効化3-6	5
図 4-1 更新プログラムパッケージのダウンロード4-7	7
図 4-2 更新プログラムパッケージからの変更確認4-8	3
図 4-3 更新プログラムパッケージのインストール4-8	3
図 4-4 コマンドプロンプトの管理者権限での起動4-9)
図 4-5 WSL のデフォルトバージョンの設定4-9)
図 4-6 WSL 上への Ubuntu のインストール	0
図 4-7 WSL 上の Ubuntu のコマンドプロンプト4-1	0
図 4-8 WSL 上の Ubuntu の起動4-1	l 1
図 4-9 Docker サービスの起動確認4-1	13
図 4-10 WSL 上の Ubuntu の起動4-1	14
図 4-11 WSL 上の Ubuntu のプロンプト4-1	14
図 4-12 コマンドプロンプトの起動4-1	15
図 4-13 コマンドプロンプト上で起動した Ubuntu4-1	15
図 4-14 Ubuntu 上のファイルへのアクセス4-1	16
図 4-15 橋梁診断支援 AI システムの配置4-1	17
図 4-16 橋梁診断支援 AI システムのインストール4-1	17
図 4-17 橋梁診断支援 AI システムのインストール完了4-1	18
図 4-18 橋梁診断支援 AI システムの停止4-1	18
図 4-19 橋梁診断支援 AI システムの再起動4-1	19
図 4-20 橋梁診断支援 AI システムの起動確認4-1	19
図 4-21 橋梁診断支援 AI システムのトップ画面4-2	20
図 5-1 Ubuntuの管理者権限での起動5-2	21
図 5-2 WSL2上の Ubuntu のコマンドプロンプト5-2	21
図 5-3 「appcont」フォルダへの移動5-2	22
図 5-4 橋梁診断支援 AI システムの起動 5-2	22
図 5-5 橋梁診断支援 AI システムの起動確認	22
図 5-6 橋梁診断支援 AI システムのログ情報5-2	23
図 5-7 橋梁診断支援 AI システムの「appcont」フォルダ	23

図目次

図 5-8	橋梁診断支援 AI システムの停止	5-24
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	6-26
図 6-2	ポート競合時の netstat コマンド実行時の一例	6-27
図 6-3	ポート競合プログラムの確認	6-28

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、コンテナ化した橋梁診断支援 AI システムを Windows Subsystem for Linux 2 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。

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。

項目	名称	バージョン等
OS	Windows10/11 64bit	Pro, Enterprise, Education (備考)
		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

表 3-1 動作環境

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Windows10を 使用する場合の設定変更について示します。Windows11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「設定」を選択します。



図 3-1 Windows の設定画面の起動

② 設定画面から「アプリ」を選択します。



図 3-2 Windowsの設定

③ 「アプリと機能」画面から「プログラムと機能」を選択します。

4 pz			- D X
@ #~L	アプリと機能		
REGHT	アプリを入手する場所の選択		間連接定 システム コンボーネント
דע	Microsoft Store からのみアプリモインストールする。 課することに役立ちます。	と、お使いのデバイスを保	プログラムと戦略
目言 アプリと権能	構現を描述ない シー		A 1788-
15 HI0779	アプリと機能		24-F/19203-0
ロモ オフラインマップ	7798014972		
DD Web ライト用のアプリ	総面や並べ替えを行ったり、ドライブでフィルターをた ンインストールまたは移動する場合は、一覧で目的	N5たりできます。アプリをア のアプリを選びます。	
CP 2776#1	このリストを映象 ア		
Q 29-1797	王べ見え: 右府 ~ 2(1)ター: すべてのドライ	ブ~	
	150 個のアプリが見つかりました		
	3D ビューアー Microsoft Corporation	65.7 KB 2024/08/26	
	7-Zip 19.00 (x64)	4.96 MB 2021/01/15	
	Adobe Acrobat II Professional - Japa	2021/01/15	
	Android Studio	2023/01/27	
	tiender	1.07 GB 2023/10/27	
	Canon Office Printer Utility	15.2 MB 2022/12/26	

図 3-3 アプリと機能

④ 「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。

1 ノロシラムと1歳用2						-		×
	ルパネル > プログラム > プログラムと機能				ٽ ~	プログラムと機能の検索		Ą
コントロール パネル ホーム	プログラムのアンインストールまたけ変更							
	10/140/1/1/AT Walcia							
インストールされた更新プログラムを	プログラムをアンインストールするには、一覧からプログラムを選が	ペして [アンインストール]、[変更]、または [ウ	修復] をクリックし	ます。				
500 V								
Windows の機能の有効化または 毎効化	i							
<u>mm///10</u>	整理 ▼						9== •	r
		発行元	インストール日	サイズ	バージョン			
	Zz 7-Zip 19.00 (x64)	Igor Pavlov	2021/01/15	4.96 MB	19.00			
	Adobe Acrobat 8 Professional - Japanese	Adobe Systems	2021/01/15		8.0.0			
	Android Studio	Google LLC	2023/01/27		2022.1			
	🔊 blender	Blender Foundation	2023/10/27	1.06 GB	3.6.5			
	■ carpass	carpass	2019/11/12	948 KB	1.0.2			
	carpass_admin	carpass	2021/06/08	1.19 MB	1.0.2			
	A CMake	Kitware	2019/05/08	86.2 MB	3.14.3			
	o Dell Command Update for Windows 10	Dell, Inc.	2018/09/26	23.0 MB	3.0.0			
	Dell Digital Delivery Service	Dell Products, LP	2018/09/26		3.6.1005.0			
	Dell Watchdog Timer	Dell Inc	2018/09/26		1.0.1.1			
	💼 EPSON EP-306 Series プリンター アンインストール	SEIKO EPSON Corporation	2021/01/15					
	Oit version 2.21.0	The Git Development Community	2019/05/08	246 MB	2.21.0			
	IIS 10.0 Express	Microsoft Corporation	2020/10/20	51.5 MB	10.0.04403			
	🔀 Intel® Hardware Accelerated Execution Manager	Intel Corporation	2023/01/31	686 KB	7.6.5			
	🍀 IrfanView 4.54 (64-bit)	Irfan Skiljan	2021/01/15	6.50 MB	4.54			
	🕌 Java(TM) SE Development Kit 22.0.2 (64-bit)	Oracle Corporation	2024/09/06	313 MB	22.0.2.0			
	Macrium Reflect Free	Paramount Software (UK) Ltd.	2022/07/05		v8.0.6758			
	Mercurial 4.9.1 (64-bit)	Matt Mackall and others	2019/05/08	35.3 MB	4.9.1			
	Meta Quest Developer Hub 3.1.1	Facebook Technologies, LLC	2023/01/18	281 MB	3.1.1			
	Hicrosoft .NET Core SDK 2.1.526 (x64)	Microsoft Corporation	2021/08/20	478 MB	2.1.526			
	Microsoft .NET Core SDK 3.1.403 (x64) from Visual Stu	Microsoft Corporation	2020/10/20	168 KB	3.1.403.015556			
	Hicrosoft .NET Core SDK 3.1.426 (x64)	Microsoft Corporation	2022/12/23	571 MB	3.1.426.16244			
	Hicrosoft ASP.NET Core 3.1.32 - Shared Framework (x	Microsoft Corporation	2022/12/23	23.7 MB	3.1.32.22566			
	📒 Microsoft Azure Authoring Tools - v2.9.6	Microsoft Corporation	2019/05/08	12.2 MB	2.9.8899.26			
	Microsoft Azure Compute Emulator - v2.9.6	Microsoft Corporation	2021/01/15		2.9.8899.26			
	Microsoft Azure Libraries for .NET – v2.9	Microsoft Corporation	2019/05/08	67.7 MB	3.0.0127.060			
	Microsoft Azure Storage Emulator - v5.10	Microsoft Corporation	2021/01/15		5.10.19227.2113			
	C Microsoft Edge	Microsoft Corporation	2024/09/17		128.0.2739.79			
	Microsoft Edge WebView2 Runtime	Microsoft Corporation	2024/09/17		128.0.2739.79			
	Histore of ODBC Driver 17 for SOL Server	Microsoft Composition	2024/04/10	7 CE MD	17 10 6 1			

図 3-4 プログラムと機能

⑤ 「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 3-5 Windows の機能の有効化または無効化

⑥ 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。

 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



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

「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。



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 WSL2 から「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 WSL上の Ubuntu の起動

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の確認があるので、「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) signedby=/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の確認があるので、「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、Docke-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の確認があるので、「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

	1		×
and Containing the service docker status			^
docker.service - Docker Application Container Ensine			
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			
IriggeredBy: • docker.socket			
uocs: https://docs.docker.com Main PID: 259 (dockerd)			
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-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.274913142+09:00" evel=info msg="D Nov 07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.387484734749:00" evel=warning msg Nov 07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.387484734749:00" evel=warning msg Nov 07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.387484734749:00" evel=warning msg Nov 07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.384895733495100" evel=warning msg Nov 07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.38495737349:00" evel=warning msg Nov 07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.384957367*09:00" evel=warning msg Nov 07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.384957367*09:00" evel=warning msg Nov 07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.384957367*09:00" evel=warning msg	efault oadina ="WARM ="WARM ="WARM ="WARM ocker aemon	brid cont IING: IING: IING: daemo has c	se ai No No No No No No No No
Nov 07 16:00:20 DESKTUP-JNSOBUU dockerd[209]: time=2024-11-07116:00:20,683481310+09:00 Tevel=infomss= A Nov 07 16:56:25 DESKTUP-JNSOBU0 conternal[1]: Started docker convises - Decker Application Container Empire	ri lis	sten o	n 🖄
loss 1-2/21 (END)			
			~

図 4-9 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。

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。

なお、「Ubuntu」を終了する場合には、以下のコマンドを実行します。これにより「Ubuntu」 を終了することができます。

コマンド: exit

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。

まず、WSL2 上にインストールした「Ubuntu」を起動します。「Ubuntu」を起動する方法は以下に示す 2 通りの方法が存在します。どちらの方法で起動した場合でも②以降の作業を行う事が可能です。
 1700日は スクートメニューの「Ubuntu」からお動する方法になります。

1通り目は、スタートメニューの「Ubuntu」から起動する方法になります。 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 4-10 WSL 上の Ubuntu の起動



この場合、下図のように左上のアイコンがオレンジ色のマークのプロンプトが開きます。

図 4-11 WSL 上の Ubuntu のプロンプト

もう1通りの方法は、コマンドプロンプトから起動する方法になります。 スタートメニューの Windows システムツールからコマンドプロンプトを選択し、管理者 権限で実行します。



図 4-12 コマンドプロンプトの起動

コマンドプロンプトが開いた後、以下のコマンドを実行します。これにより「Ubuntu」が 起動します。

コマンド: wsl

この場合、下図のようにコマンドプロンプト上で「Ubuntu」が起動します。



図 4-13 コマンドプロンプト上で起動した Ubuntu

② 「Ubuntu」が起動したら、「Ubuntu」上で以下のコマンドを実行し、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

パス: <u>¥¥wsl\$</u>

 ③「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

④ 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。



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することができるようになります。

⑤ 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: <u>¥¥wsl.localhost¥Ubuntu¥home¥</u>ユーザ¥docker

ここで、ユーザの部分は「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。

 ⑥ 「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。

📙 🛃 📒 🖛 docker			<u></u>		×
ファイル ホーム 共有	表示				~ (?
$\leftarrow \rightarrow \checkmark \uparrow \square \ll wsl$	> Ubuntu > home > 🗾 > docker >	5 V	dockerの検索		P
^	名前	更新日時	種類	サイズ	
★ クイック アクセス ■ デスクトップ ★	docker_橋梁診断支援AIシステム	2024/11/07 15:45	ファイル フォルダー		
➡ ダウンロード 🖈					

図 4-15 橋梁診断支援 AI システムの配置

⑦「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

	do docker compose up -d er-compose.vml: the attribute `vers	sion` is obsol
ete, it will be ignored, please remove it to avoid potential confusion		
[+] Running 1/14		
[+] Running 2/14] Pulling		<u>7.3s</u>
Lappdb Ling J Pulling	7	7.4s
L+J Running 2/14e Downloading L======>	J 4.422MB/	1.8s
Lappdb, Lapp, J. Pulling		(.þs
L+J Running 2/14e Downloading L====================================	J 12.07MB/	1.98
Lappdb Lass J Fulling	7 000 41 D/0	(.6S
L+J Kunning Z/14e Extracting L>	J ZZ9.4KB/Z	- <u>4</u> .48
Li appob Li J Fulling	- 	6 (S
LTJ Running 2/14e Extracting L2	J Z.3Z3MD/Z	4. IS
Fill Dumping 2/14p Extracting Economy	1 0 E00MD/0	2.08
r oppdb Fran Pulling	J 2.020MD/2	5.68
[+] Rupping 3/14e Extracting [second	1 / 258MB/2	2.35
	1 4.000MD/2	8 Ne

図 4-16 橋梁診断支援 AI システムのインストール

⑧ 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ジステ/」/appcont	<u></u>		×
 >> extracting sha256:66d98f73acb62e86c0c226f9eedcbc7eda305df0c1e171ca5caf81cb8b1c40cb >> extracting sha256:6d2c43c5efbc801f83ee6565c7102ca60d6f35e158324fbb042e601r423afe8 >> extracting sha256:ab50bb4ead768386dfb0c7c34de9eff7cf2103c4668de98fe31e59f15dee88 >> extracting sha256:b5465aa993a558add4acecdade8f39843089069f31fea0201533d309a98cda > extracting sha256:b5465aa993a558add4acecdade8f39843089069f31fea0201533d309a98cda > extracting sha256:b5465aa993a558add4acecdade8f39843089069f31fea0201533d309a98cda > extracting sha256:b6465aa993a558add4acecdade8f39843089069f31fea0201533d309a98cda > extracting sha256:b6465aa993a558add4acecdade8f39843089069f31fea0201533d309a98cda > [webapp 2/2] RUN aptraget update & aptraget install -y libpq-dev libonig-dev libpng-dev libzip-dev && > webapp1 exporting to image > exporting layers > writing image sha256:302d8b3f3ecd8ee0521c7f8e9e82f437a6f973a5e129bf5b6bbb16aab68d4a9b > [webapp1 resolving provenance for metadata file 	dock	0.0s 1.0s 0.0s 0.0s 32.8s 0.2s 0.1s 0.0s 0.0s 0.0s	<
the number // Acont_default Created Terrated Terrateed Terrateed Terrateed Terrateed Terrateed Terrat			0000000000
🗸 Container webapp Started			s
International Content of the Conte			~

図 4-17 橋梁診断支援 AI システムのインストール完了

⑨「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

Come @ Come Content Con	1227	
WARN[0000] /home/www.docker/docker 橋梁診断支援A[システム/appcont\$ sudo docker compose stop WARN[0000] /home/www.docker/docker_橋梁診断支援A[システム/appcont/docker-compose.yml: the attribute `v ete, it will be ignored, please remove it to avoid potential confusion	version`	is obsol
 Stopping 1/2 Stopped Stopped Stopping 1/2 Stopped Stopping 1/2 Stopped Stopped Stopped Stopping 1/2 Stopped Stopping 1/2 Stopped Stopped Stopped Stopping 1/2 Stopped Stopped Stopped Stopped Stopped 		0.0s 0.0s 0.0s 0.0s 0.0s 0.0s 0.0s 0.0s
* Container Hoopp School 		

図 4-18 橋梁診断支援 AI システムの停止

⑩ 「Ubuntu」上で以下のコマンドを実行し、再度コンテナを起動します。

コマンド: sudo docker compose up -d



図 4-19 橋梁診断支援 AI システムの再起動

① 「Ubuntu」上で以下のコマンドを実行し、コンテナの起動状態を確認します。

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



図 4-20 橋梁診断支援 AI システムの起動確認

コマンドを実行し「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。

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 4-21 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

Occupied Control	_0		×
6 Edg/130.0.0.0″ webapp 2024-11-08T01:20:32.859434179Z 172.18.0.1 [08/Nov/2024:01:20:32 +0000] ″POST /sub/sub09.php HTTP/1.1″ 200 calhost/maing09.php ″Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.38 (KHTML, like Gecko) Chrome/130.0.0.0 S	228 "H Safari,	nttp:// /537.36	lo î
dg/330.0.0.0" webapp 2024-11-08T01:20:38.005452199Z 127.0.0.1 - [08/Nov/2024:01:20:38 +0000] "OPTIONS * HTTP/1.0" 200 126 "-" "Ac ebian) PHP/7.4.33 (internal_dummy_connection)" of the control of the condition of the condition of the condition	ache/2	2.4.54	(D
webagp [2024-1]-08101:20:34.4/15201/32 172.18.0.1108/Nov/2024101:20:34 +00000] GET /css/images/ui-icons_444444_25 /1.1 200 [228] "http://localhost/css/iguery-ui.them.min.css" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/587 ke Gecko) Chrome/130.0.0.0 Safari/587.38 Edg/130.0.0.0" [00.4] (0004.01.00/f0000] [0007 (001.0)]	6×240. .36 (H	.png HI (HTML,	IP I i
Webago [2024-]]-usiu:2014.47/942592 172.18.0.1 - [08/Nov/2024:01/20194 +00000] GET /CESS/Images/ui-lcons_/////29 V1.1 200 [300 [http://localhost/cess/iguery-ui.them.min.css [Mozilla/5.0] (Windows NT 10.0; Win64; x64) AppleWebKit/587 ke Gecko) Chrome/130.0.0.0 Safari/587.38 Edg/130.0.0.0"	6×240. .36 (ł	HTML,	li
Webapp / 2024-11-05101;21:100_074334032 172.18.0.1 - 106/N072024:0121:00 +00000 _PDS1 /SubySub03.ppp H117/1.1 200 calhost/main09.php ^ Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 S dg/130.1.0204 11.00701;01:11 2006011207 172.10.0 1 _ F09 Alway(2004.01:01:01.00000] ^CTT (-k/subs-sub-sub-sub-	229 r kafari∕ №_:	/537.36	Ĕ
Medapp [2024-11-05101.2].11.2500611292 112.15.01 - "Dio/Nov/2024.01.2].10 40000 GET/SUD/FEORG.pre:DNO-050001905 1x481vs HTP/1.1 2001 14991 "http://ocalhost/main(9.php "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 ecko) Chrome/130.0.0.0 Safari/537.36 Edg/130.0.0.0	&sta- (KHTML	_, like	G
			- U

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

an @1@1@1@1@1@1@1@1@1@1@1@1@1@1@1@1@1@1@1	<u> </u>		×
an monoperative Adocker/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 ` ete, it will be ignored, please remove it to avoid potential confusion	version`	is ob	sol ^
<pre>[*] Stopping 1/2 + Stopping 1/2pdb Stopped * Stopping 1/2pdb Stopped</pre>			90000000000000000000000000000000000000
✔ Container webapp Stopped ■ Container webapp Stopp			.05

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「/dockerentrypoint-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。

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、使用ポートの競合は発生いたします。なお、以下の例は 5432 番ポートの競合を確認した場合の例となります。

ТСР	0.0.0.0:5432	0.0.0.0:0	LISTENING	6656
ТСР	[::]:5432	[::]:0	LISTENING	6656
ТСР	[::1]:5432	[::1]:49723	ESTABLISHED	6656
ТСР	[::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。

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: <u>http://localhost</u>: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。