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の影	没定変更	3-3
4. インストール手順		4-5
4.1. Windows Sul	bsystem for Linux 2(WSL2)のインストール	4-5
4.2. WSL2 への Do	ockerのインストール	4-12
4.3. 橋梁診断支援 A	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 A	AI システムの起動	5-23
5.2. 橋梁診断支援 A	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	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

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	I
図 4-10 Ubuntu のプロキシ設定4-1	0
図 4-11 aptのプロキシ設定4-1	1
図 4-12 curl のプロキシ設定4-1	1
図 4-13 Docker サービスの起動確認4-1	3
図 4-14 Ubuntu 上のファイルへのアクセス4-1	5
図 4-15 橋梁診断支援 AI システムの配置4-1	6
図 4-16 docker のプロキシ設定4-1	6
図 4-17 docker composeの設定4-1	7
図 4-18 docker compose のプロキシ設定4-1	8
図 4-19 橋梁診断支援 AI システムのインストール4-1	9
図 4-20 橋梁診断支援 AI システムのインストール完了4-1	9
図 4-21 橋梁診断支援 AI システムの停止	0
図 4-22 橋梁診断支援 AI システムの再起動4-2	0
図 4-23 橋梁診断支援 AI システムの起動確認4-2	0
図 4-24 橋梁診断支援 AI システムのトップ画面4-2	2
図 5-1 Ubuntuの管理者権限での起動5-2	3
図 5-2 WSL2上の Ubuntu のコマンドプロンプト5-2	3
図 5-3 「appcont」フォルダへの移動5-2	4
図 5-4 橋梁診断支援 AI システムの起動5-2	4
図 5-5 橋梁診断支援 AI システムの起動確認5-2	4
図 5-6 橋梁診断支援 AI システムのログ情報5-2	5

図目次

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。

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 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。

☑ プログラムと機能						-		×
← → · ↑ □ · ביארם-ル	パネル > プログラム > プログラムと機能				v ö 7⊡	コグラムと機能の検索		Q
コントロール バネル ホーム	プログラムのアンインストールまたは変更							
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ま	尺して 「アンインストール」、「変更」、または 「	修復1 をクリックし	ます。				
表示								
 Windows の機能の有効化または 無効化 	at 72						D	•
	至注 ▼						855	•
	名前	発行元	インストール日	サイズ	バージョン			^
	2 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			
	III carpass	carpass	2019/11/12	948 KB	1.0.2			
	📧 carpass_admin	carpass	2021/06/08	1.19 MB	1.0.2			
	🛕 CMake	Kitware	2019/05/08	86.2 MB	3.14.3			
	둲 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			
	EIIS 10.0 Express	Microsoft Corporation	2020/10/20	51.5 MB	10.0.04403			
	Manager Intel® Hardware Accelerated Execution Manager	Intel Corporation	2023/01/31	686 KB	7.6.5			
	FifanView 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			
	Wicrosoft .NET Core SDK 2.1.526 (x64)	Microsoft Corporation	2021/08/20	478 MB	2.1.526			
	Wicrosoft .NET Core SDK 3.1.403 (x64) from Visual Stu	Microsoft Corporation	2020/10/20	168 KB	3.1.403.015556			
	Wicrosoft .NET Core SDK 3.1.426 (x64)	Microsoft Corporation	2022/12/23	571 MB	3.1.426.16244			
	Wicrosoft 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	7.001.00	128.0.2739.79			~
	現在インストールされているプログラム 合計 103 個のプログラムがインストールされています	サイズ: 22.5 GB						

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。

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。

👷 Windows Subsystem for Lin	nux Update Setup	50 — 53		×
	Completed the Windows Update Setup Wizard	Subsystem	for Lin	ux
	Click the Finish button to exit th	e Setup Wizard	ł.	
2	Back	Finish	Cano	el

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。

 خ عبر المركز الم	ウェブ	設定 フォルダー 写真 🕨 🛛 🗸 🗸	
最も一致する検索結果			
ביגיג לעדב אקקע			
Web の検索		ביעד לעדר ביעד ביעד ביעד ביעד ביעד ביעד	
Q cmd - 検索結果をさらに表示する	>		
Q cmd 管理者として実行	>	 ☑ 關 	
Q cmd prompt	>	同 管理者として実行	
Q、cmd コマンド 一 覧	>	つ ファイルの場所を開く	
Q cmd 管理者として実行 windows11	>		
Q cmd ipconfig	>		
Q cmd b	>		
Q cmd ファイル一 覧	>		
ር cmd ファ1ル	>		
Q cmd ディレクトリ移動	>		
C cmd	2 🚺	📜 📲 🚾 🔹 😰 💟 🐺	Ę

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

雪狸者: コマンドブロンプト	
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>wslset-default-version 2 E:¥WINDOWS¥system32>wslset-default-version 2 に、WINDOWS¥system32>wslset-default-version 2 この操作を正しく終了しました。	
C:VWINDOWS¥system32>	



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、「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。



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。

← 設定				- 0 ×	ſ
	ネットワークとイン	ターネット > プロキシ			
	イーサネットまたは Wi-Fi 接続にブ	ロキシ サーバーを使います。これらの設定は、VPN 接続には3	適用されません。		
設定の検索 Q	自動プロキシ セットアップ				
▲ ホーム	設定を自動的に検出する			77 💽	
 システム Bluetooth とデバイス 	セットアップ スクリプトを使う オン	セットアップ スクリプトを編集		編集	
 ネットワークとインターネット / 個人用設定 	手動プロキシ セットアップ				
 アプリ アカウント 	プロキシ サーバーを使う オフ	スクリフトのアドレス http://●●●●●●●●.pac ×		セットアップ	
 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	▲ ヘルプを表示 ₽ フィードバックの送信	保存 キャンセル			
🏋 アクセシビリティ					
🔍 プライバシーとセキュリティ					
Windows Update					

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。

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	http://192.0.2.255:1080
URL	http://exampleproxy	3128	http://exampleproxy:3128
URL (認証あり)	http://exampleproxy (username、password)	8080	http://username:password@ex ampleproxy:8080

表 4-1 プロキシ設定の記載の例

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

① 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

② 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

③ 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』";

GNU papo 7-2			/etc/ant/ant	conf		-		×
Acquire::http:: Acquire::http::	Proxy "http:// :Proxy "http:/	2	/etc/apt/apt.	.cont	276			
GHelp XExit	U Write Out <mark>^R</mark> Read File	₩ Where Is [^] ¥ Replace	^K Uut [^] U Paste	J Execute J Justify	U Location [™] Go To Line	M-U Und M-E Red	0 0	

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
- ⑥ 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	2
-------	------	------	-----	---------	---

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。

proxy=http://『プロキシ設定』

-		×
(EU) Unda)	
M-E Redo		
		— □ 11-11 Undo 11-E Redo

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) signed-by=/etc/apt/ke yrings/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 dockercompose-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

	1000		×
in the service docker status			^
docker.service - Docker Application Container Engine			
Loaded: Toaded (/usr/Tib/systemd/system/docker.service; enabled; preset: enabled) Active: active (running) since Thu 2024-11-07 16:55:25 JST; 41min ago			
TriggeredBy: Odocker.socket			
Main PID: 259 (dockerd)			
Memory: 74.2M ()			
CGroup: /system.slice/docker.service 1959. /usr/bip/dockerd -H.fd://containerd=/rup/containerd/containerd.sock			
Mar. 07 10-55-25 DEGYTOP, MCCDOO Jalara 70503, 1:4-770024-11-07710-55-25 274012142400-00" laurel-:-fa are"T		. Lucia	
Nov 07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time=2024-11-07T16:55:25.357272402+09:00 // level=info msg="L	oadin	s conta	
Nov U/ 16:55:25 DESKTUP-JNSSBUU dockerd[259]: time= 2024-11-0/116:55:25.394845/34+09:00 Tevel=warning msg Nov 07 16:55:25 DESKTUP-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.394951304+09:00" Tevel=warning msg	= wart = "Wart	VING: N VING: N	
Nov 07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.394959124+09:00" level=warning msg Nov.07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.394969733+09:00" level=warning msg Nov.07 16:55:25 394969733+09:00" level=warning msg Nov.07 16:55:25 394969733+09:00" level=warning msg Nov.07 16:55:25 394969733+09:00" level=warning msg	= "WARM = "WARM	VING: N	
Nov 07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.394975967+09:00" level=info msg="D	ocker	daemor	
Nov 07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time= 2024-11-07116:55:25.683481310+09:00″ level=info msg="A Nov 07 16:55:25 DESKTOP-JNSSB00 dockerd[259]: time="2024-11-07T16:55:25.683481310+09:00″ level=info msg="A	aemon PI lis	nas co sten or	n >
Nov 07 16:55:25 DESKTOP-JNSSB00 systemd[1]: Started docker.service - Docker Application Container Engine.			

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

「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 からアクセスすることが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。

📙 🛃 📙 🖛 docker					
ファイル ホーム 共有	表示				~ (
$\leftarrow \rightarrow \cdot \uparrow \square \ll wsl$	> Ubuntu > home >	ت ~	dockerの検索		Q
^	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 ★		2024/11/07 15:45	ファイル フォルダー	-	

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』"



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-co mpose.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。



- 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	A
#version, s, u	
services: # PHP + Apacha 書き出し位置を合わせる	
Sontainer name: webapp # ./WebApp/Dockerfile でイメージをビルド build: App	
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(コンテンツ配置所 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: * concor	Ħ)
- 	
a nerp i d'intere det in intere is in contra i execute i d'Education in d'ondo îX Exit îR Read File î¥ Replace îU Paste îJ Justify î∕ Go To Line M-E Redo	-

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

① 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』

C								-	×
	GNU nano 7.2	/home/	/	/docker/docker	_橋梁診断支援AIシ:	ステム/appcont/.	env		
ht	tp_proxy=http:	//:							
ht	tps_proxy=http	:/ <u>/ </u>							
ΗĻ	IP_PROXY=http:	// <u>,</u> :							
ΗI	TPS_PRUXY=http	://							
-									
îG	Help î	0 Write Out	🎬 Where Is	<mark>îK</mark> Cut	îT Execute	<u>°C</u> Location	<mark>M−U</mark> Undo		
ŶΧ	Exit î	R Read File	î¥ Replace	^U Paste	📵 Justify	^∕ Go To Line	M-E Redo		_
									V

図 4-18 docker compose のプロキシ設定

- ② 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
- ③「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

	sudo docker compose up -d locker-compose.yml: the attribute `ver	rsion`is obsol
ete, it will be ignored, please remove it to avoid potential confusio [+] Running 1/14 [+] Running 2/14] Pulling	'n	7.8s
::appdb[[::::]]Pulling [+]Runnings2/14e Downloading:[======>>] 4.422MB/	7.4s 1.8s
L: appdb L:### 」 Pulling [+] Running 2/14e Downloading [==================> - appdb [==== -] Pulling] 12.07MB/	7.6s 1.9s 7.6s
[+] Running 2/14e Extracting [> appd [===]] 229.4kB/2	2.0s
[+] Running 2/14e Extracting [=====> appdb [###] Pulling] 2.523MB/2	2.1s 7.8s
[+] Running 2/14e Extracting [=====> ; appdb [====] Pulling] 2.523MB/2	2.2s 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。

Cav.	@: ~/docker/docker_博强診断支援AIシステ/Jappcont	<u>un</u>	
22222222222	 extracting sha256:6660897/3acb62e88c0c226f9eadebc7eda305df0c1e171ca5caf81cb8bic40cb extracting sha256:243c5efbc881f83ee686c710/2a6804675e185824fbb042de5017e43afe8 extracting sha256:2504265ae47638647607c34de9eff7c12103c4660459e185824fbb042de5017e43afe8 extracting sha256:050465aaa93a558a112c56490a2a97f69ef395fd87f6052a1898644c9a813ef extracting sha256:050465aaa939a558add4acecdade8f398f31fea0cb8844c9a813ef6426830498963a extracting sha256:050465aaa939a558add4acecdade8f398f31fea0cb8844c9a813ef6426830498963a extracting sha256:050465aaa939a558add4acecdade8f398f31fea0cb8844c9a813ef642683049898c9a extracting sha256:050465aaa93a558add4acecdade8f398f31fea0cb98451fea0c9158343049898c9a extracting sha256:050465aaa89a358add4acecdade8f398f31fea0cb98451fea0c9158343049898c9a extracting sha256:050465aaa89a358add4acecdade8f28f38f38608986981fea020158343049898c9a extracting sha256:050465aaa89a358add4acecdade8f21678e982f437a6f973a5e129bf5b6bbb16aab68d4a9b extracting sha256:02d8b3f3acd8ee0521c7f8e982f437a6f973a5e129bf5b6bbb16aab68d4a9b extracting sha256:02d8b3f3acd8ee0521c7f8e982f437a6f973a5e129bf5b6bbb16aab68d4a9b extracting sha256:02d8b3f3acd8ee0521c7f8e982f437a6f973a5e129bf5b6bbb16aab68d4a9b extracting sha256:02d8b3f3acd8ee0521c7f8e982f437a6f973a5e129bf5b6bbb16aab68d4a9b extracting sha256:02d8b3f3acd8ee0521c7f8e982f437a6f973a5e129bf5b6bbb16aab68d4a9b 	dock	0.0s 1.0s 0.0s 0.0s 32.8s 0.2s 0.1s 0.0s 0.0s 0.0s
[] _] _] _] _] _] _] _] _] _]	Running 1/3cont_default Created Running 1/3cont_default Created Running 1/3cont_default Created Running 1/3cont_default Created Running 1/3cont_default Created Running 1/3cont_default Created Running 3/3cont_default Created Container appdb Starting Container appdb Started		0.2s 0.2s 0.2s 0.2s 0.2s 0.2s 0.2s 0.2s
V	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

⑤「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

國 @	<u>vini</u> /	
###################################	ersion`	is obsol
(+) Stopping 1/2 (+) Stopping 1/2pdb Stopped (+) Stopping 1/2pdb Stopped (+) Stopping 1/2pdb Stopped (+) Stopping 1/2pdb Stopped		0.0s 0.0s 0.0s 0.0s
[+] Stopping 1/2pdb Stopped [+] Stopping 1/2pdb Stopped [+] Stopping 1/2pdb Stopped [+] Stopping 1/2pdb Stopped		0.0s 0.0s 0.0s 0.0s 0.0s
+ Stopping 1/2pdb Stopped + Stopping 1/2pdb Stopped + Stopping 1/2pdb Stopped + Stopping 1/2pdb Stopped + Stopping 1/2pdb Stopped		0.0s 0.0s 0.0s 0.0s
t+j Stopping 1/2pab Stopped ↓ Stopping 2/2pdb Stopped ↓ Container appdb Stopped ↓ Container webapp Stopping ↓ Container webapp Stopped		0.0s 0.0s 1.5s 1.5s

図 4-21 橋梁診断支援 AI システムの停止

⑯ 「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

国	<u> </u>		×
	`version`	is c	bso l ^
 ✓ Container webapp Started ✓ Container webapp Started Container appdb Starting ✓ Container appdb Started 			0.7s 0.7s .8s 0.9s

図 4-22 橋梁診断支援 AI システムの再起動

⑦ 「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}}"



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。

2 D D REPERTANZE X	+						-	0	×
← C () localhost	,	A ⁶	$\dot{\Omega}$	Φ	¢±	۲	-	***	٥
	橋梁診断支援AIシステム								
	本システム使用上の留意点								
	 本システムは、通路構定和点便要領で求める健全性の診断を行うことが 的ではなく、予防保全段準の局極がや、メカニズムに適合した措置方針 決定等を目的としています。 健全性の診断は、本システムで出力する対策区分判定と健全性の診断の一般 な対応を整考にすることができますが、機楽ごとの条件などを考慮して 断する必要があります。 なお、措置の必要性は、本システムで実施できる部材里位の検討に加え 同じ機能を担う部材群など広い範囲で検討することで合理的となる場合 あることに留意して判断する必要があります。 上記の留意点に加え、本システムの取扱説書に記載している部材別の 提条件などを確認のうえで使用してください。 	相のは、酸料は、が、潮							
	10966 BR348								

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}}"

	docker/docker.構梁診断支援AIシステム/appcont	_0		×
appdb running webapp running	'docker/docker_橋梁診断支援A!システム/appcont\$ 'docker/docker_橋梁診断支援A!システム/appcont\$ sudo docker compose psformat "[[.Service]] [[.St	ate}}	"	^

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

💽 🔤 @ Contraction Contraction (福保診断支援AIシステム/appcont	_2		×
6 Edg/130 0 0 0 0			~
weben 1 2024-11-08T01+20+32 8594341797 172 18 0 1 [08/May/2024+01+20+32 +0000] "POST /cub/cub/g abo HTTP/1 1" 200 2	28 "h	++n•77	16 ⁽¹⁾
	fori	E07 06	Ë
24/190 0.0 °C (NTML), TRE decko/ Chrome/190.0.0 MT10.0, MT104, X04/ Appremediativity 337.30 (NTML), TRE decko/ Chrome/190.0.0.0 au	lai 17	557.50	L .
		1 51	(D
Webabb [2024-11-08101:20:38.0034321882.127.0.0.1 - LOS/NOV/2024:01:20:38 +0000] UPITUNS * HIP/1.0 200 126 - Apa	cne/ z	.4.04	(U
ebian) PHP/7.4.33 (internal_dummy_connection)			
webapp 2024-[1-08101:20:54.4715207792 172.18.0.1 L087Nov/2024:01:20:54 +0000] GET /css/images/uj-icons_44444_256	x24U.	png HI	IP
/1.1~200 7293 ~http://localhost/css/jquery-ui.theme.min.css~~Mozilla/5.0 (\indows NT 10.0; \indox; x64) Apple\ebit/537.	36 (K	HTML,	11
ke Gecko) Chrome/130.0.0.0 Safari/537.36 Edg/130.0.0.0″			
webapp 2024-11-08T01:20:54.4715452597 172.18.0.1 [08/Nov/2024:01:20:54 +0000] "GET /css/images/ui-icons 77777 256	×240.	ong HT	TP
/1 1″ 200 7300 "http://localbost/css/iguery-ui theme min css" "Mozilla/5 0 (Windows NT 10 0: Win64: x64) AppleWebKit/537	36 (K	HTM .	11
ke Gedo) Chrome/130 0 0 0 Safari/537 36 Edg/130 0 0 0"			
2024-11-08101.0101010101010101010101010101010101	20 "h	++6.11	16
Websph (2024 H 0001,21,00,0446024,172,10,0) (2014 12,10,0) (2014,012,10,0) (2017,200,0) (2017,200,0) (2017,0) (£0 £:/	E07 00	1e
Calificational sprip wozitrazio (mindows Ni 10.0, mino4, x04) Apprementitzizzi, so (Nitmil, tike decko) chromeziso.0.0.0 sa	rar iz	007.00	E
webapp [_2024-1]-08101:21:11.2880611292_172.18.0.1 [U8/Nov/2024:01:21:10 +0000] GET /sub/record.php?bno=83300190698	sid=v	Inmkwl	zi
1x48]vs HIIP/1.1 200 14991 "http://localhost/main09.php" "Mozilla/5.0 (Windows NI 10.0; Win64; x64) AppleWebKit/537.36 (KHIML	, like	e G
ecko) Chrome/130.0.0.0 Satari/537.36 Edg/130.0.0.0"			
			22
			~

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

□□ @L · ·/docker/docker. 捕梁診断支援AIシステム/appcont -			×
amene (1997) (Adocker/docker/橋梁診断支援AIシステム/appcont\$ sudo docker compose stop WARN[0000] /home/unad/docker/docker_橋梁診断支援AIシステム/appcont/docker-compose.yml: the attribute `versig ete, it will be ignored, please remove it to avoid potential confusion	n`i	is ob	sol ^
<pre>t+ Stopping 1/2 + Stopping 1/2pdb Stopped + Stopping 1/2pdb Stopped - Stopping 2/2pdb Stopped - Stopping 2/2pdb Stopped - Container webapp Stopping * Container webapp Stopping</pre>			.0s .0s .0s .0s .0s .0s .0s .0s .0s .0s
「「「「「「「」」」」)、「」」(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-entrypointinitdb.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。

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。

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A I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。