

Magic xpi 4.14 のインストール

マジックソフトウェア・ジャパン株式会社



Agenda

- Magic xpi 4.14の構成
- Linux上でのインストール/設定
- Windows上でのインストール/設定
- Magic xpi 4.14のインストール
- IMMのデプロイ
- 動作確認
- その他注意事項



Magic xpi 4.14の構成



Magic xpi 4.14の構成



Magic xpi 4.14ではシステムの構成に

- Linux(Ubuntu)
- Windows

の2つのOSを使用します。

【それぞれの役割】

- Windows

Magic xpi Server(MagicxpiSrver.exe)でプロジェクトを実行

- Linux

メッセージング層を構成する**IMM(In Memory Middleware)**を実行し、
リクエストの受付、プロジェクトの起動/停止、ログ(アクティビティログ)の管理を行う

IMM (In Memory Middleware)



IMM(In Memory Middleware)



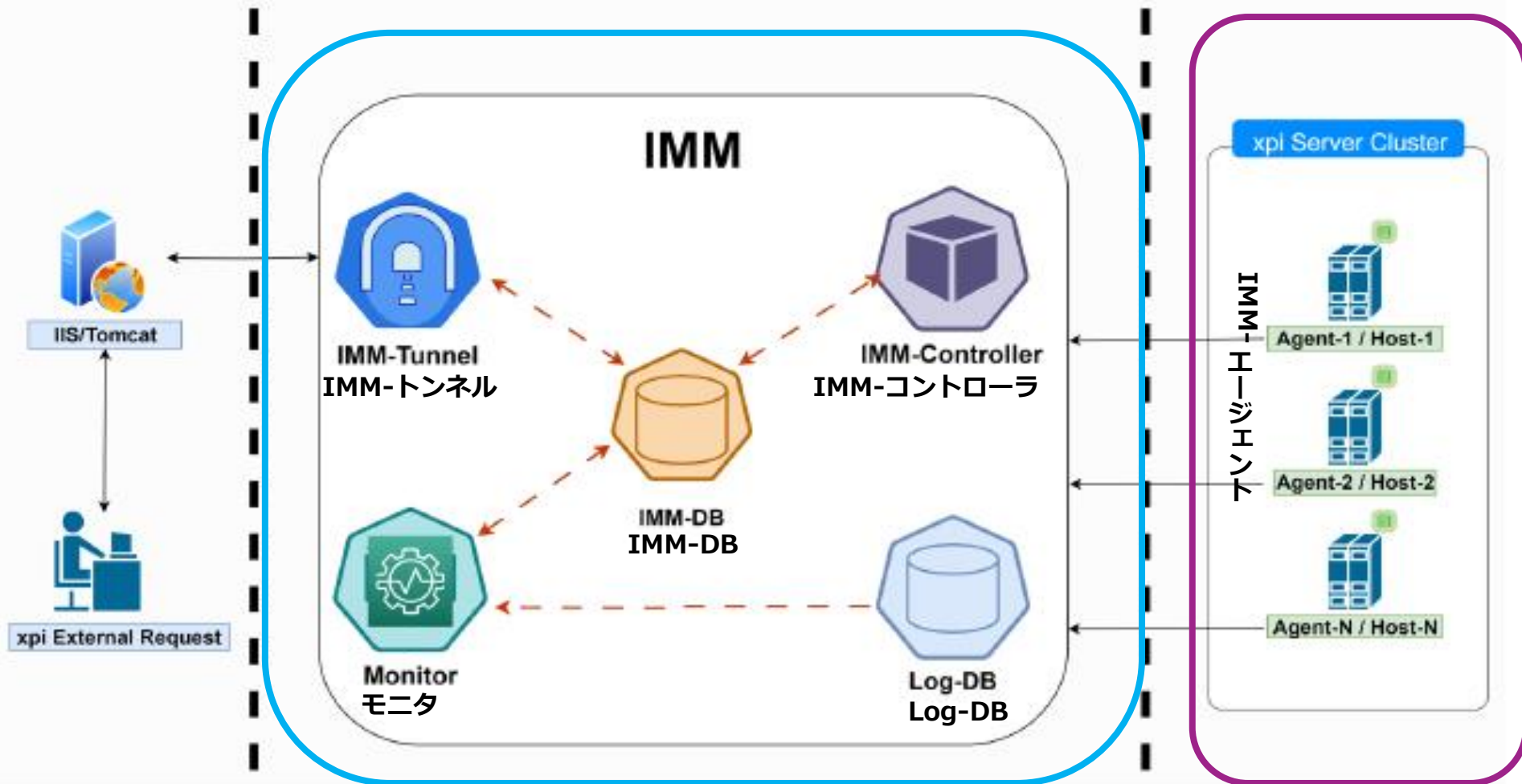
【IMMアーキテクチャ概要】

- メッセージング層として構成
 - 独自コンテナ(Docker/Kubernetes)として構成
 - **Docker/ Kubernetes : Linux上の最新コンテナ技術**
 - Linux上でIMMは稼働
- Magic xpi サーバ(プロジェクト)との通信
 - プロジェクトは今まで同様Windows上で稼働
 - IMMエージェント(Windows上のプロセス)を使用してIMMと通信
- IMMとMagic xpi サーバ
 - 1つのIMMに対し、複数のMagic xpiサーバが接続可能
 - 負荷分散、可用性

IMM(In Memory Middleware)



Windows上



Linux上のコンテナ
複数サービスがIMM内のノードとしてデプロイ

IMM(In Memory Middleware)



- IMM-エージェント
 - xpiサーバ(プロジェクト)が実行されるサーバ上で必ず稼働
 - プロジェクトの起動/停止、正常性のチェック
 - 複数のエージェントもIMMで一括管理
- IMM-コントローラ
 - IMM エージェントと通信し、xpiサーバの開始/停止の要求を受け付ける
- IMM-DB
 - IMM が機能するために必要なすべての運用データが保存されるセントラルリポジトリすべての xpi 関連のランタイム データ、プロジェクト メタデータ、およびリアルタイム実行データが含まれます
 - インメモリデータベースのRedisを使用
- LogDB
 - xpiサーバによって生成されるすべてのアクティビティ ログの保管コンテナ
 - 高速なMongoDBを使用

IMM(In Memory Middleware)



- IMM-トンネル

- IMM アーキテクチャ内に存在する Webサービス
- すべての外部リクエスト (HTTPリクエスト、TCPリクエストなど) を処理し、リクエストをIMM コントローラにリダイレクト。

注: HTTP Web サーバー トリガーなどに必要なWeb サーバを代替するものではありません

- モニタ

- IMM とその配下で実行されている xpi プロジェクトに関する情報と状態関連データを提供するインターフェイスです。
- 主な機能

ライセンスの使用状況、サーバーの負荷、フローの実行状況

注: 過去バージョンとは異なり、xpi モニタは IMM インフラストラクチャ内に常駐します(Windowsサービスとして提供されません)。

IMM(In Memory Middleware)



【開発環境構築】

- WSL2を使用
 - WSL : Windows Subsystem for Linux ⇒ Windows上でLinuxを動かす仕組み
 - PowerShell等のコマンドを実行する必要はありません
 - **Rancher Desktop(SUSE)**でらくらく環境構築
 - 自動的にDocker/Kubernetesをインストール/構成します
 - Linux上にIMMをデプロイしてももちろん良い
- 簡単デプロイ
 - Magic xpi4.14をインストール
 - インストールフォルダ内にデプロイ用のバッチファイルが用意 ⇒ 実行するだけ
 - 前述5つのサービスが自動的にデプロイ

IMM(In Memory Middleware)



【実行環境構築】

- Linux(Ubuntu)を準備
 - MicroK8Sインストール
 - Kubernetes 接続情報を出力し、Windowsへ
- 簡単デプロイ
 - 接続情報をLinuxからコピー
 - Magic xpi4.14をインストール
 - インストールフォルダ内にデプロイ用のバッチファイルが用意 ⇒ 実行するだけ
WindowsからLinux上のKubernetesに接続し、デプロイ
 - 前述5つのサービスがLinux上にデプロイ

IMM(In Memory Middleware)



【システム要件】

- 開発環境
 - CPU : 2.66Ghz以上 4コア以上
 - メモリ : 16G以上
 - 空きディスク容量 : 25G以上
 - OS : Windows 10 pro, Windows 11 pro WSL2が有効である必要あり
(クラウド/仮想環境ではWSL2が無効の可能性あり)
Ubuntu 22.04.x LTS

注意) 上記仕様は、xpi 開発環境とIMMを同一環境で実行する場合を想定。

IMM(In Memory Middleware)



【システム要件】

- xpi 実行環境とIMMを同一環境で実行する場合
 - CPU : 2.66Ghz以上 8コア以上
 - メモリ : 32G以上
 - 空きディスク容量 : 25G以上
 - OS : Windows Server 2022 WSL2が有効である必要あり (クラウド/仮想環境ではWSL2が無効の可能性あり)
Ubuntu 22.04.x LTS

IMM(In Memory Middleware)



【システム要件】

- xpi実行環境とIMMを別環境で実行する場合

【Windowsサーバ】

- CPU : 2.66Ghz以上 4コア以上
- メモリ : 32G以上
- 空きディスク容量 : 25G以上
- OS : Windows Server 2016、2019、2022

【Linuxサーバ】

- CPU : 2.66Ghz以上 4コア以上
- メモリ : 16G以上
- 空きディスク容量 : 10G以上
- OS : Ubuntu 22.04.x LTS

注意 1) AWSでは、EKSでもIMMは動作するが、料金体系が高額なので、Linuxでの動作が現実的。

注意 2) 本社開発部では、引き続き、WSL2以外にもWindowsOS上(Hyper-Vなど) でIMMを実行できる環境を検討/検証中。
正式サポートが確認された場合は、追って通知。

×

Linux上でのインストール/設定

×



Linux上でのインストール/設定



- Linux上でのインストール/設定手順
 - Linux(Ubuntu 22.04.x LTS)のインストール
Ubuntu 22.04.x LTSをインストールします
本資料ではOSのインストールについては割愛します
 - microk8sのインストール
 - Metallb(Loadbalancer)の設定
 - 確認とconfigによる接続情報の取得

Linux上でのインストール/設定



【microk8sのインストール】

- Linux上で以下のコマンドを実行し、microk8sをインストールします

> `sudo snap install microk8s --classic`

```
chin@Chin830G8: ~  
chin@Chin830G8:~$ sudo snap install microk8s --classic
```

- コマンド実行結果

```
chin@Chin830G8:~$ sudo snap install microk8s --classic  
microk8s (1.28/stable) v1.28.3 from Canonical ✓ installed  
chin@Chin830G8:~$
```

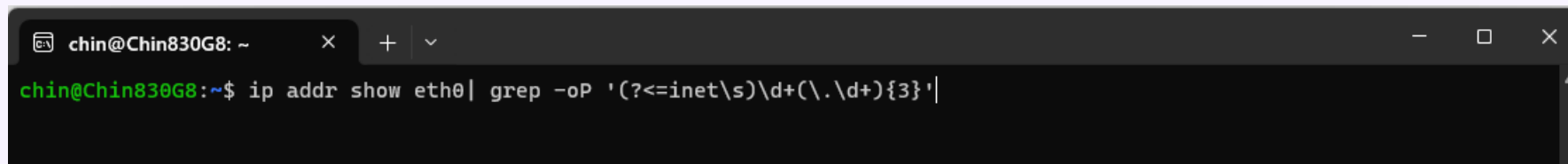
Linux上でのインストール/設定



【 Metallb(Loadbalancer)の設定】

- Linux上で以下のコマンドを実行し、IPアドレスを調べます

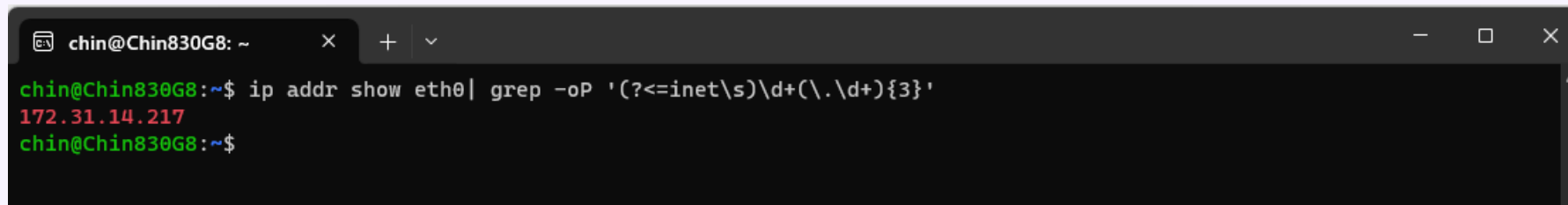
```
> ip addr show eth0 | grep -oP '(?<=inet\s)\d+(\.\d+){3}'
```



```
chin@Chin830G8: ~  
chin@Chin830G8:~$ ip addr show eth0 | grep -oP '(?<=inet\s)\d+(\.\d+){3}'
```

※ I/Fがeth0でない場合は、>ip a コマンド等を使用し、I/Fを調べます

- コマンド実行結果：このIPアドレスをメモしておきます



```
chin@Chin830G8: ~  
chin@Chin830G8:~$ ip addr show eth0 | grep -oP '(?<=inet\s)\d+(\.\d+){3}'  
172.31.14.217  
chin@Chin830G8:~$
```

Linux上でのインストール/設定



【 Metallb(Loadbalancer)の設定】

- Linux上で以下のコマンドを実行し、metallbを設定します

> sudo microk8s enable dns storage metallb

```
chin@Chin830G8: ~  
chin@Chin830G8:~$ sudo microk8s enable dns storage metallb|
```

Linux上でのインストール/設定



【 Metallb(Loadbalancer)の設定】

- Linux上で以下のコマンドを実行し、metallbを設定します

コマンドを実行すると、以下の画面でIPアドレスの入力を求められます。先にメモしたIPアドレスを指定します

```
chin@Chin830G8: ~  
Infer repository core for addon metallb  
WARNING: Do not enable or disable multiple addons in one command.  
This form of chained operations on addons will be DEPRECATED in the future.  
Please, enable one addon at a time: 'microk8s enable <addon>'  
Enabling DNS  
Using host configuration from /etc/resolv.conf  
Applying manifest  
serviceaccount/coredns unchanged  
configmap/coredns unchanged  
deployment.apps/coredns unchanged  
service/kube-dns unchanged  
clusterrole.rbac.authorization.k8s.io/coredns unchanged  
clusterrolebinding.rbac.authorization.k8s.io/coredns unchanged  
CoreDNS service deployed with IP address 10.152.183.10  
DNS is enabled  
DEPRECATION WARNING: 'storage' is deprecated and will soon be removed. Please use 'hostpath-storage' instead.  
  
Infer repository core for addon hostpath-storage  
Enabling default storage class.  
WARNING: Hostpath storage is not suitable for production environments.  
A hostpath volume can grow beyond the size limit set in the volume claim manifest.  
  
deployment.apps/hostpath-provisioner created  
storageclass.storage.k8s.io/microk8s-hostpath created  
serviceaccount/microk8s-hostpath created  
clusterrole.rbac.authorization.k8s.io/microk8s-hostpath created  
clusterrolebinding.rbac.authorization.k8s.io/microk8s-hostpath created  
Storage will be available soon.  
Enabling MetalLB  
Enter each IP address range delimited by comma (e.g. '10.64.140.43-10.64.140.49,192.168.0.105-192.168.0.111'): |
```

Linux上でのインストール/設定



【 Metallb(Loadbalancer)の設定】

- 「IPアドレス-IPアドレス」と入力します。この例では、「172.31.14.217-172.31.14.217」と入力します

```
chin@Chin830G8: ~  
WARNING: Do not enable or disable multiple addons in one command.  
This form of chained operations on addons will be DEPRECATED in the future.  
Please, enable one addon at a time: 'microk8s enable <addon>'  
Enabling DNS  
Using host configuration from /etc/resolv.conf  
Applying manifest  
serviceaccount/coredns unchanged  
configmap/coredns unchanged  
deployment.apps/coredns unchanged  
service/kube-dns unchanged  
clusterrole.rbac.authorization.k8s.io/coredns unchanged  
clusterrolebinding.rbac.authorization.k8s.io/coredns unchanged  
CoreDNS service deployed with IP address 10.152.183.10  
DNS is enabled  
DEPRECATION WARNING: 'storage' is deprecated and will soon be removed. Please use 'hostpath-storage' instead.  
  
Infer repository core for addon hostpath-storage  
Enabling default storage class.  
WARNING: Hostpath storage is not suitable for production environments.  
A hostpath volume can grow beyond the size limit set in the volume claim manifest.  
  
deployment.apps/hostpath-provisioner created  
storageclass.storage.k8s.io/microk8s-hostpath created  
serviceaccount/microk8s-hostpath created  
clusterrole.rbac.authorization.k8s.io/microk8s-hostpath created  
clusterrolebinding.rbac.authorization.k8s.io/microk8s-hostpath created  
Storage will be available soon.  
Enabling MetalLB  
Enter each IP address range delimited by comma (e.g. '10.64.140.43-10.64.140.49,192.168.0.105-192.168.0.111'): 172.31.14  
.217-172.31.14.217
```

Linux上でのインストール/設定



【 Metallb(Loadbalancer)の設定】

➤ コマンド実行結果

```
chin@Chin830G8: ~  
Applying Metallb manifest  
customresourcedefinition.apiextensions.k8s.io/addresspools.metallb.io created  
customresourcedefinition.apiextensions.k8s.io/bfdprofiles.metallb.io created  
customresourcedefinition.apiextensions.k8s.io/bgpadvertisements.metallb.io created  
customresourcedefinition.apiextensions.k8s.io/bgppeers.metallb.io created  
customresourcedefinition.apiextensions.k8s.io/communities.metallb.io created  
customresourcedefinition.apiextensions.k8s.io/ipaddresspools.metallb.io created  
customresourcedefinition.apiextensions.k8s.io/l2advertisements.metallb.io created  
namespace/metallb-system created  
serviceaccount/controller created  
serviceaccount/speaker created  
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created  
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created  
role.rbac.authorization.k8s.io/controller created  
role.rbac.authorization.k8s.io/pod-lister created  
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller created  
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker created  
rolebinding.rbac.authorization.k8s.io/controller created  
secret/webhook-server-cert created  
service/webhook-service created  
rolebinding.rbac.authorization.k8s.io/pod-lister created  
daemonset.apps/speaker created  
deployment.apps/controller created  
validatingwebhookconfiguration.admissionregistration.k8s.io/validating-webhook-configuration created  
Waiting for Metallb controller to be ready.  
deployment.apps/controller condition met  
ipaddresspool.metallb.io/default-addresspool created  
l2advertisement.metallb.io/default-advertise-all-pools created  
Metallb is enabled  
chin@Chin830G8: ~$
```

Linux上でのインストール/設定



【確認とconfigによる接続情報の取得】

- Linux上で以下のコマンドを実行し、statusを確認します

> sudo microk8s status

```
chin@Chin830G8: ~  
chin@Chin830G8:~$ sudo microk8s status
```

- コマンド実行結果

```
chin@Chin830G8: ~  
chin@Chin830G8:~$ sudo microk8s status  
microk8s is running  
high-availability: no  
datastore master nodes: 127.0.0.1:19001  
datastore standby nodes: none  
addons:
```

Linux上でのインストール/設定



【確認とconfigによる接続情報の取得】

- Linux上で以下のコマンドを実行し、config (接続情報)を取得します

> sudo microk8s config



```
chin@Chin830G8: ~  
chin@Chin830G8:~$ sudo microk8s config
```




Windows上でのインストール/設定



Windows上でのインストール/設定



- Windows上でのインストール/設定手順
 - Package管理システム chocolatey のインストール
 - minikube のインストール
 - helmのインストール
 - configファイルの作成

Windows上でのインストール/設定



【 Package管理システム chocolatey のインストール】

➤ PowerShellを管理者として実行し、以下のコマンドを実行します

```
> Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object  
System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps  
1'))
```

Windows上でのインストール/設定



【Package管理システム chocolatey のインストール】

➤ コマンド実行結果

```
Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProt
ocol
for 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
forcing web requests to allow TLS v1.2 (Required for requests to Chocolatey.org)
getting latest version of the Chocolatey package for download.
not using proxy.
getting Chocolatey from https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2.
downloading https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2 to C:\Users\ADMINI~1\AppData\Local\Temp\2\chocolatey\chocoInstall\chocolatey.zip
not using proxy.
extracting C:\Users\ADMINI~1\AppData\Local\Temp\2\chocolatey\chocoInstall\chocolatey.zip to C:\Users\ADMINI~1\AppData\Local\Temp\2\chocolatey\chocoInstall
installing Chocolatey on the local machine
creating ChocolateyInstall as an environment variable (targeting 'Machine')
Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
before you can use choco.
restricting write permissions to Administrators
we are setting up the Chocolatey package repository.
the packages themselves go to 'C:\ProgramData\chocolatey\lib'
(i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
a shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.

creating Chocolatey folders if they do not already exist.

chocolatey.nupkg file not installed in lib.
Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
警告: Not setting tab completion: Profile file does not exist at 'C:\Users\Administrator\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
chocolatey (choco.exe) is now ready.
you can call choco from anywhere, command line or powershell by typing choco.
run choco /? for a list of functions.
you may need to shut down and restart powershell and/or consoles
first prior to using choco.
ensuring Chocolatey commands are on the path
ensuring chocolatey.nupkg is in the lib folder
PS C:\Users\Administrator>
```

Windows上でのインストール/設定



【minikube のインストール】

➤ 引き続き、PowerShell(管理者)画面から、以下のコマンドを実行します

```
> choco install minikube
```

Windows上でのインストール/設定



【minikube のインストール】

➤ コマンド実行結果

```
管理: Windows PowerShell
PS C:\Users\Administrator> choco install minikube
Chocolatey v2.2.2
Installing the following packages:
minikube
By installing, you accept licenses for the packages.
Progress: Downloading kubernetes-cli 1.28.2... 100%
kubernetes-cli v1.28.2 [Approved]
kubernetes-cli package files install completed. Performing other installation steps.
The package kubernetes-cli wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): A
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar.gz to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
ShimGen has successfully created a shim for kubectl-convert.exe
ShimGen has successfully created a shim for kubectl.exe
The install of kubernetes-cli was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\kubernetes-cli\tools'
Progress: Downloading Minikube 1.32.0... 100%
Minikube v1.32.0 [Approved]
Minikube package files install completed. Performing other installation steps.
ShimGen has successfully created a shim for minikube.exe
The install of Minikube was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\Minikube'
Chocolatey installed 2/2 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Users\Administrator>
```

「A」を入力します

Windows上でのインストール/設定



【helmのインストール】

➤ 引き続き、PowerShell(管理者)画面から、以下のコマンドを実行します

```
> choco install -y kubernetes-helm --version=3.12.3
```


Windows上でのインストール/設定



【helmのインストール】

➤ コマンド実行結果

```
管理: Windows PowerShell
PS C:\Users\Administrator> choco install -y kubernetes-helm --version=3.12.3
chocolatey v2.2.2
Installing the following packages:
kubernetes-helm
By installing, you accept licenses for the packages.
Progress: Downloading kubernetes-helm 3.12.3... 100%

kubernetes-helm v3.12.3 [Approved]
kubernetes-helm package files install completed. Performing other installation steps.
Downloading kubernetes-helm 64 bit
  from 'https://get.helm.sh/helm-v3.12.3-windows-amd64.zip'
Progress: 100% - Completed download of C:\Users\Administrator\AppData\Local\Temp\chocolatey\kubernetes-helm\3.12.3\helm-v3.12.3-windows-amd64.zip (15.43 MB).
Download of helm-v3.12.3-windows-amd64.zip (15.43 MB) completed.
Hashes match.
Extracting C:\Users\Administrator\AppData\Local\Temp\chocolatey\kubernetes-helm\3.12.3\helm-v3.12.3-windows-amd64.zip to C:\ProgramData\chocolatey\lib\kubernetes-helm\tools...
C:\ProgramData\chocolatey\lib\kubernetes-helm\tools
ShimGen has successfully created a shim for helm.exe
The install of kubernetes-helm was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\kubernetes-helm\tools'

chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Users\Administrator>
```

minikubeと同様に
Do you want to run the script ?
と質問された場合は「A」を入力します

Windows上でのインストール/設定



【configファイルの作成】

- ログインアカウント フォルダ配下 (Administratorでログインしているなら、C:¥Users¥Administrator)に以下を作成します。
 - (1) .kube フォルダ
 - (2) .kube フォルダ内に config ファイル
config ファイルをエディタで開き、P.25で保存した内容をconfig ファイルに保存します

これでWindows側のインストール/設定作業は終了です。



Magic xpi 4.14のインストール



Magic xpi 4.14のインストール



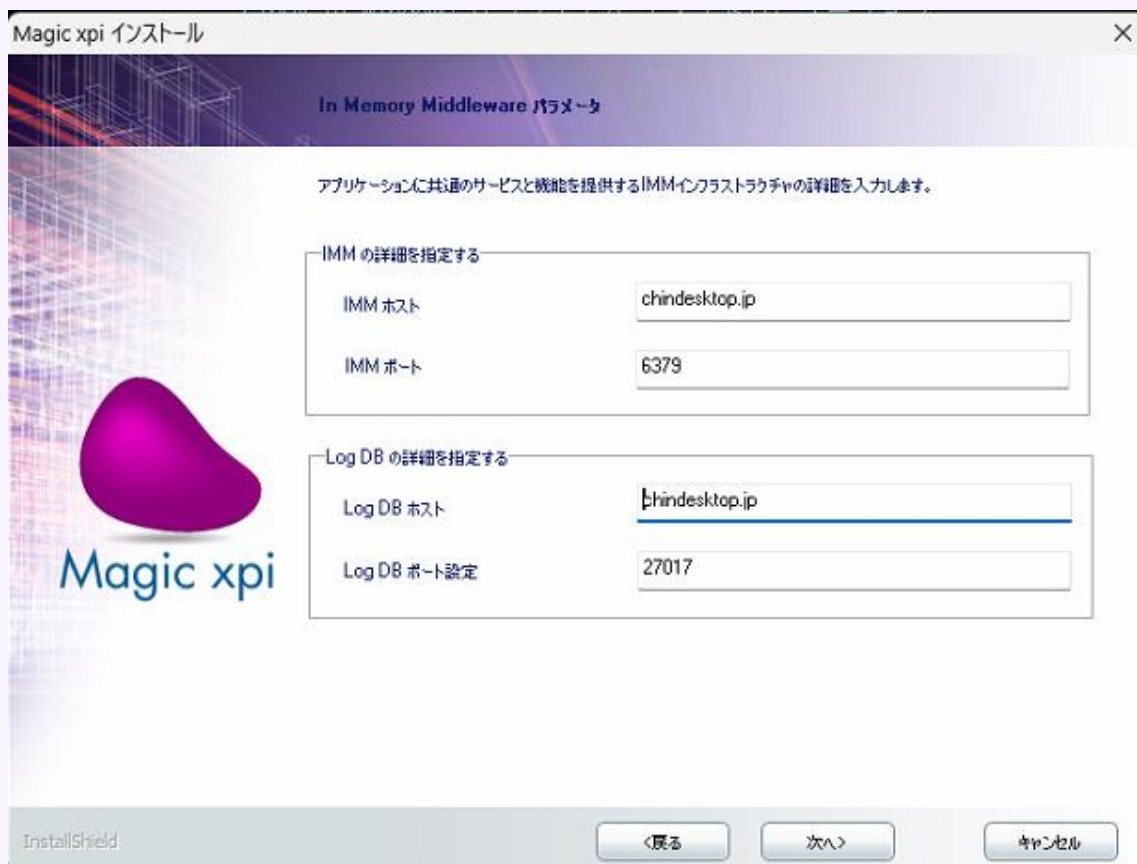
- Magic xpi 4.14のインストール
 - setup.exe の実行

Windows上でのインストール/設定



【setup.exeの実行】

- setup.exeを実行し、インストールを開始します



In Memory Middlewareパラメータ画面：

【IMMホストおよびLog DBホスト】
Linuxに付けるホスト名を入力します
任意の値を入力可も、英小文字である必要があります

【IMMホストおよびLog DBホスト】
デフォルト値のままでOKです



IMMのデプロイ



IMMのデプロイ



- IMMをデプロイします
 - deploy-imm.batの実行
 - hostsファイルの修正

Windows上でのインストール/設定



【deploy-imm.batの実行】

- <インストール先>¥InMemoryMiddleware¥deploy フォルダ内のdeploy-imm.bat を実行します
- コマンド実行結果

```
C:\Windows\System32\cmd.exe
INFO: Deploying magic-xpi-imm-chart
NOTE: Waiting for deployment to complete...
Pulled: devmcsworkspaceacr.azurecr.io/magic-xpi-helm/xpi-imm-chart:4.14.01
Digest: sha256:8c4ab34692a185673a0172303742457ae05a3347e8335e960fc4f8745553dbaf
Error: INSTALLATION FAILED: client rate limiter Wait returned an error: rate: Wait(n=1) would exceed context deadline
=== Done ===

INFO: xpi-imm successfully deployed.
===== IMM Deployment complete =====

===== Please make note of the deployed Endpoints =====
IMM-DB at: chindesktop.jp:6379
LOG-DB at: chindesktop.jp:27017
IMM Controller URL: chindesktop.jp/controller
IMM-Tunnel URL: http://chindesktop.jp/immunnel
Magic xpi monitor URL: http://chindesktop.jp/magicmonitor
=====Action needed=====
IMP Add the following to you System host file for name resolution.
for e.g. in windows "C:\WINDOWS\system32\drivers\etc\hosts"
-----
172.20.159.210    chindesktop.jp
-----

NOTE: The end points and the host entry are also saved in "EndPoints.txt"
You may review the results, or press "x" to exit:
```

エラーがない場合は「x」を入力し、バッチファイルを終了します。

Windows上でのインストール/設定



【hostsファイルの修正】

- deploy-imm.bat 実行結果画面に表示されている内容：本例では 172.20.159.210 ホスト名を C:¥Windows¥system32¥drivers¥etc¥hosts ファイルに追加します

```
C:\Windows\System32\cmd.exe
INFO: Deploying magic-xpi-imm-chart
NOTE: Waiting for deployment to complete...
Pulled: devmcsworkspaceacr.azurecr.io/magic-xpi-helm/xpi-imm-chart:4.14.01
Digest: sha256:8c4ab34692a185673a0172303742457ae05a3347e8335e960fc4f8745553dbaf
Error: INSTALLATION FAILED: client rate limiter Wait returned an error: rate: Wait(n=1) would exceed context deadline
=== Done ===

INFO: xpi-imm successfully deployed.
===== IMM Deployment complete =====

===== Please make note of the deployed Endpoints =====
IMM-DB at: chindesktop.jp:6379
LOG-DB at: chindesktop.jp:27017
IMM Controller URL: chindesktop.jp/controller
IMM-Tunnel URL: http://chindesktop.jp/immunnel
Magic xpi monitor URL: http://chindesktop.jp/magicmonitor
=====Action needed =====
IMP Add the following to you System host file for name resolution.
for e.g. in windows "C:\WINDOWS\system32\drivers\etc\hosts"
-----
172.20.159.210    chindesktop.jp
-----

NOTE: The end points and the host entry are also saved in "EndPoints.txt"
You may review the results, or press "x" to exit:
```



動作確認



動作確認



- 動作確認
 - WindowsサービスよりIMMエージェントを起動
 - Magicモニタで確認
 - kubectlコマンドでの確認

動作確認



【 WindowsサービスよりIMMエージェントを起動】

- Windowsサービスより、
Magic xpi 4.14 IMM Agent
を起動します

Magic xpi 4.14 Soap Service	Magic xpi 4.14 Soap Server	実行中	自動	.\Magicxpi414Soap
Magic xpi 4.14 Debugger	Magic xpi 4.14 Debug Server	実行中	自動	.\Magicxpi414Debug
Magic xpi 4.14 IMM Agent		実行中	自動	.\Magicxpi414IMM
Magic xpi 4.14 Soap Service	Magic xpi Soap Server	実行中	自動	.\Magicxpi414Soap

動作確認



【 Magicモニタで確認】

➤ デスクトップ上のショートカットよりMagicモニタを起動します

この3つの○が緑なら
IMMが正しく動作している
ことを示しています



ダッシュボード

Home > ダッシュボード

プロジェクトのロードと保留中のメッセージ

ライセンスの使用状況と利用可能なワーカーの経時変化

重大度	説明	発生日時	スターテス
HIGH	Agent is alive.	17-12-2023 10:33:00 PM	Alive

トリガーアクティビティ

動作確認



【 kubectl コマンドでの確認】

➤ コマンドプロンプトより以下のコマンドを実行

> kubectl get pods -A

➤ NAMESPACEが
magic-xpi-imm-ns
の行が6行表示されたらOK

```
コマンド プロンプト
C:\Users\chin>kubectl get pods -A
NAMESPACE          NAME                                                    READY   STATUS    RESTARTS
AGE
kube-system        helm-delete-traefik-crd-fnzs8                         0/1     Completed 0
75d
kube-system        helm-delete-traefik-nm8rl                             0/1     Completed 0
75d
kube-system        svclb-xpi-ingress-controller-ingress-nginx-controller-ba66pfr2r 4/4     Running   56 (11m ago)
20d
kube-system        local-path-provisioner-84db5d44d9-wvcsw              1/1     Running   91 (11m ago)
75d
kube-system        coredns-6799fbc5-dnf9w                                1/1     Running   92 (151m ago)
75d
magic-xpi-imm-ns   logdb-74c7d886d4-7d6b9                               1/1     Running   14 (11m ago)
20d
magic-xpi-imm-ns   imm-db-0                                               1/1     Running   14 (11m ago)
20d
default           xpi-ingress-controller-ingress-nginx-controller-598ffd65fcrhj4k 1/1     Running   22 (11m ago)
20d
magic-xpi-imm-ns   imm-tunnel-deployment-7bc966b767-l9zxj               1/1     Running   14 (11m ago)
20d
magic-xpi-imm-ns   imm-controller-7ffc9f6897-qj5kl                      1/1     Running   14 (11m ago)
20d
kube-system        metrics-server-67c658944b-f4ght                      1/1     Running   96 (154m ago)
75d
magic-xpi-imm-ns   xpi-monitor-d5bdff6b6-49bnr                          1/1     Running   14 (11m ago)
20d
magic-xpi-imm-ns   xpi-imm-server-deployment-5b4b7fcc47-gkfdw           1/1     Running   14 (11m ago)
20d
C:\Users\chin>
```



その他注意事項



その他注意事項



- 注意事項
 - コンピュータ名に「-」などいわゆる禁則文字は使用しないでください
 - IMMホスト名には英小文字を使用します
 - Networkの状態によってはデプロイがエラーとなる場合があります
その際は再度デプロイを行ってください

その他注意事項



- 注意事項

- Proxy環境下ではKubernetes エンジン : microk8sを使用する際にProxyの設定を行う必要があります

参考サイト : <https://microk8s.io/docs/install-proxy>

(1) /etc/environmentファイル

http_proxy="<プロトコル>://プロキシサーバ:ポート" 例 : "http://ProxyServer.co.jp:8080"

https_proxy="<プロトコル>://プロキシサーバ:ポート"

HTTP_PROXY="<プロトコル>://プロキシサーバ:ポート"

HTTPS_PROXY="<プロトコル>://プロキシサーバ:ポート"

NO_PROXY=10.0.0.0/8,192.168.0.0/16,127.0.0.1,172.16.0.0/16,127.0.0.1,.svc,.svc.cluster.loc

no_proxy=10.0.0.0/8,192.168.0.0/16,127.0.0.1,172.16.0.0/16,127.0.0.1,.svc,.svc.cluster.loc

その他注意事項



- 注意事項

(2) /var/snap/microk8s/current/args/containerd-envファイル

http_proxy= "<プロトコル>://プロキシサーバ:ポート" 例 : "http://ProxyServer.co.jp:8080"

https_proxy="<プロトコル>://プロキシサーバ:ポート"

HTTP_PROXY="<プロトコル>://プロキシサーバ:ポート"

HTTPS_PROXY="<プロトコル>://プロキシサーバ:ポート"

NO_PROXY=10.0.0.0/8,192.168.0.0/16,127.0.0.1,172.16.0.0/16,127.0.0.1,.svc,.svc.cluster.loc

no_proxy=10.0.0.0/8,192.168.0.0/16,127.0.0.1,172.16.0.0/16,127.0.0.1,.svc,.svc.cluster.loc

その他注意事項



- 注意事項

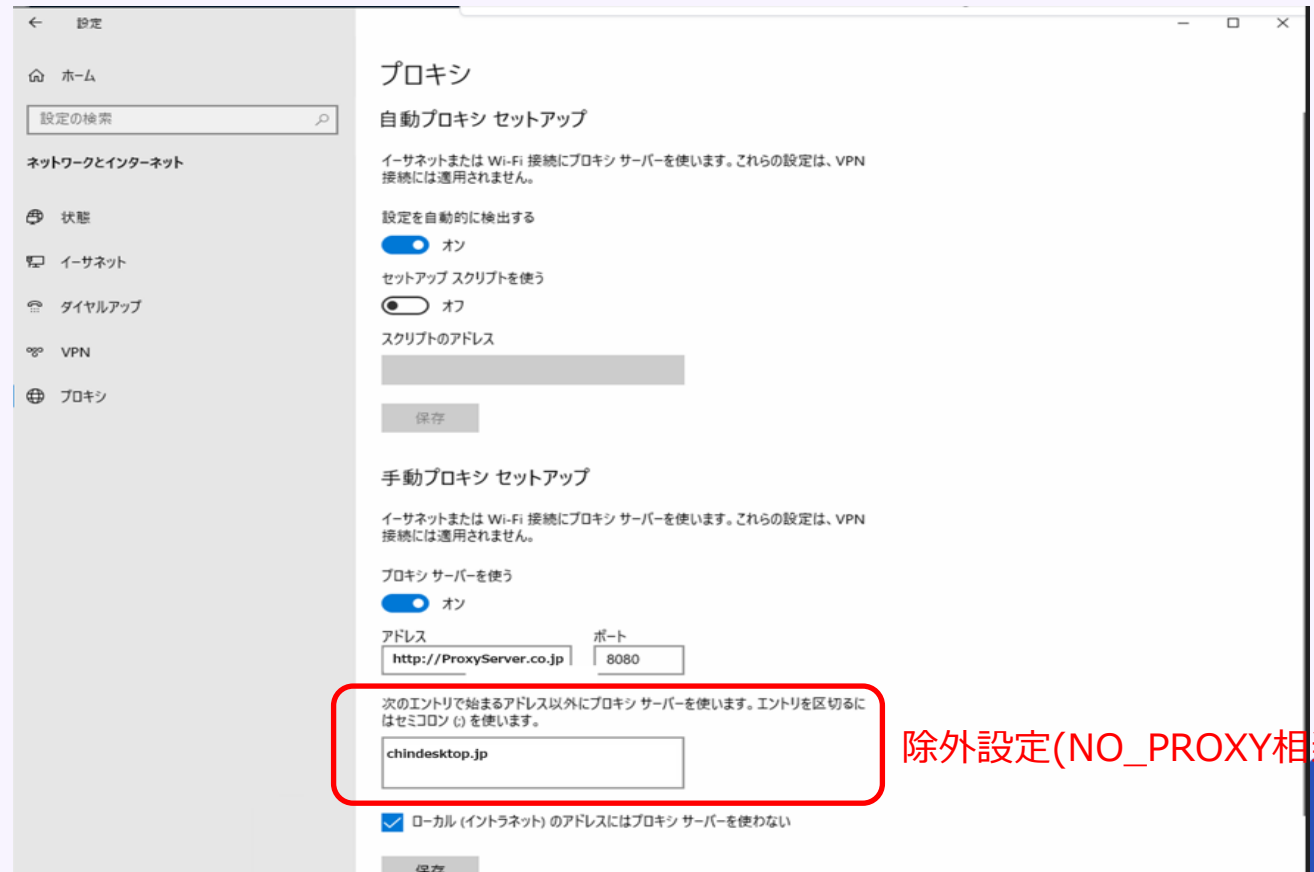
- Proxy環境下ではWindows側で「deploy-imm.bat」ファイルを実行してIMMをデプロイする前にWindows側でProxyを設定を行う必要があります

```
SET HTTP_PROXY=<プロトコル>://プロキシサーバ:ポート 例 : http://ProxyServer.co.jp:8080  
SET HTTPS_PROXY= <プロトコル>://プロキシサーバ:ポート  
NO_PROXY=<P.18で調べたUbuntuのIPアドレス>
```

その他注意事項

- 注意事項

- Proxy環境下ではブラウザでMagicモニタを使用する際は、WindowsのProxy設定でP.37で入力したIMMホスト名を除外設定(NO_PROXY相当)として入力する必要があります



除外設定(NO_PROXY相当)

その他注意事項



- 注意事項

- IMMが動作するUbuntuとxpiサーバが動作するWindowsの間では以下のポートで通信かできる必要があります

- (1) ポート80 : Magicモニタの通信で使用

- (2) ポート6443 : IMMデプロイ時、Kubernetes/Helmの通信で使用

- (3) ポート16443 : IMMデプロイ時、Kubernetes/Helmの通信で使用
ポートは6443、16443のどちらかが使用される

- (4) ポート5116 : imm-controller-serviceで使用

- (5) ポート6379 : imm-db-serviceで使用

- (6) ポート5117 : imm-tunnel-serviceで使用

- (7) ポート27017 : logdb-serviceで使用

- (8) ポート8181 : xpi-monitor-serviceで使用

その他注意事項



- 注意事項

- Rancher Desktopを使用すると、Ubuntuのインストールおよび本資料で説明されているソフトウェアのインストールは必要ありません
- Rancher Desktopが必要なソフトウェアをインストールし、設定します
- Rancher Desktop使用時はIMMのデプロイ時に `deploy-imm.bat` を編集する必要があります
 - ※ DEVNET 内「Magic xpi 4.14 技術情報 ドキュメント」の「WindowsでのRancher Desktopのインストール手順」を参照してください

<https://devnet.magicsoftware.co.jp/images/skillup/magic/download/xpi414/Installations%20Rancher%20Desktop%20on%20Windows%20JPN.pdf>

Thank you !

