

Magic xpi

クラウドネイティブ



OUTPERFORM THE FUTURE™

Contents

はじめに.....	1
前提条件	1
Microk8s のインストール.....	1
RedHat での実行手順	1
ハードウェア要件	4
Linux Sever.....	4
セキュリティとファイアウォールに関する考慮事項.....	6
EKS の前提条件	9
EKS への接続	9
インストール.....	9
Cloud Manager のインストール.....	9
hosts ファイルの設定.....	11
必要な証明書のインストール	11
Cloud Manager のアンデプロイ	12
クラウドマネージメント.....	12
Magic xpi の環境とプロジェクト管理.....	13
Cloud Manager への認証	13
管理者の詳細の更新	14
Magic Cloud Manager のヘルスチェック.....	15
環境を Kubernetes クラスタにデプロイする.....	15
パスパラメータ.....	15
Kubernetes クラスタからの環境のアンデプロイ/削除	16
リクエストボディパラメータ	16
Kubernetes クラスタへの周辺サービスのデプロイ	17
リクエストボディパラメータ	17
Kubernetes クラスタからの周辺機器サービスの削除	18



リクエストボディパラメーター	18
Kubernetes クラスタでのローカル エージェントのセットアップ	18
Request Body Parameters.....	19
Kubernetes クラスタからのローカル エージェントの削除	20
プロジェクトのデプロイ	20
リクエストボディパラメータ	20
プロジェクトのアンデプロイ	22
ログを生成して更に調査する	22
リクエストボディパラメーター	23
ローカルワークステーションで生成されたログのダウンロード	23
INI/アラート設定のアップロード.....	23
リクエストボディパラメーター	24
プロジェクト/周辺ファイルのアップロード	24
リクエストボディパラメーター	24
クラウドサービスへのアクセス	24
Cloud Manager.....	24
Magic xpi Monitor	25
デバッグ	25
OData.....	26
RESTful API.....	26
Webservice (SOAP).....	27
マイグレーションプロジェクトの Web サービススキーマの処理.....	27
HTTP Service	27
既知の制限事項	28
Appendix	29
環境変数の設定	29
Cloud Manager でのローカル・エージェントの使用	31
ローカルエージェントのインストール	31

キーポイント.....	32
Disclaimer	34

はじめに

クラウドネイティブは、企業がクラウド環境で実行されるアプリケーションを設計および開発し、クラウドインフラストラクチャとサービスを活用してデジタルトランスフォーメーションを加速できるようにするアプローチです。これにより、ユーザーは、柔軟性、スケーラビリティ、回復力、弾力性などの機能を備えたクラウドコンピューティングインフラストラクチャのフルパワーを活用できます。

Magic xpi プロジェクトをクラウド環境にデプロイするのは複雑な作業が伴います。Magic xpi Cloud Manager は、このプロセスを簡素化し自動化するための簡単な Web ベースのインターフェースを提供します。ウェブベースのインターフェースにより、Magic xpi プロジェクトをすばやく簡単にデプロイおよび管理でき、デバッガー、OData、Web サービスなどの周辺サービスをより細かく制御できます。また、これらの REST API は、「継続的デプロイメント」機能を通じて、組織の CI/CD プラクティスに俊敏性と効率性をもたらします。

このドキュメントは、Cloud Manager の設定と、Cloud Manager を使用して環境、周辺サービス、および xpi プロジェクトを管理について説明します。

注) Magic xpi Cloud Native の新しいアップデートにより、変更される場合があります。

前提条件

1. Kubernetes インフラストラクチャが利用可能で、実行中であることが必要です。
また、オンプレミス展開を選択する場合は、MicroK8s がインストールされた Ubuntu サーバーが必要です。
2. Cloud Manager をデプロイするための Windows ホストシステム。



現在のバージョンは、MicroK8s および EKS(シングルノードおよびマルチノードクラスター)をサポートしています

Cloud Manager には HTTPS でのみアクセスできます。

Microk8s のインストール

RedHat Enterprise Server でのみ実行する必要がある手順と、RedHat サーバーと Ubuntu サーバーの両方で実行するのが一般的な手順がいくつかあります。

RedHat での実行手順

RedHat を使用している場合は、次の手順を実行してください。

1. `sudo subscription-manager register`



2. パスワードを入力します。サブスクリプションのユーザー名とサブスクリプションのパスワードの入力を求められます。
3. 現在のユーザー・パスワードの入力を求められます。ここで、RedHat サブスクリプションのユーザー名とパスワードを入力します。
4. `sudo subscription-manager attach`
5. `sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm`
6. `sudo dnf upgrade`
`sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm`
7. `sudo dnf upgrade sudo rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm`
8. `sudo subscription-manager repos --enable "rhel-*-optional-rpms" --enable "rhel-*-extras-rpms"`
9. `sudo yum update`

RedHat と Ubuntu で実行する共通手順

以下は、RedHat と Ubuntu での共通手順です。

1. 次のコマンドを使用して MicroK8S をインストールします。

```
sudo snap install microk8s --classic --channel=1.30/stable
```

このコマンドを実行して、下記のエラーが発生した場合は

```
error: cannot communicate with server: Post
"http://localhost/v2/snaps/microk8s": dial unix
/run/snapd.socket: connect: no such file or directory
```

次のコマンドを実行します。

- `sudo apt-get update && sudo apt-get install -yqq daemonize dbus-user-session fontconfig`
- `sudo daemonize /usr/bin/unshare --fork --pid --mount-proc /lib/systemd/systemd --system-unit=basic.target`
- `exec sudo nsenter -t $(pidof systemd) -a su - $LOGNAME`



2. MicroK8S を再インストールします。
3. 次のコマンドを使用して、MicroK8S が正常にインストールされているかどうかを確認します。

```
microk8s status
```

このコマンドにより、ステータスが実行中と表示されます。MicroK8s が実行されていない場合は、次のコマンドで起動します。

```
sudo microk8s start
```

4. MicroK8s をインストールしたら、以下の MicroK8s アドオンをインストールする必要があります。

i. DNS

このアドオンを有効にするには、次のコマンドを実行します。

```
sudo microk8s enable dns
```

ii. Metallb



metallb アドオンには、サポートされているネットワークインターフェイス（イーサネット名）の IP アドレスが必要です。たとえば、インターフェイスのタイプが eth0 の場合、IP アドレスは次のいずれかを使用して取得できます

コマンド: `ifconfig eth0` または `ip addr show eth0` コマンドの実行中にエラーが発生した場合は、まず以下のコマンドを実行してください: `sudo apt install nettools`

このアドオンを有効にするには、次のコマンドを実行します。

```
sudo microk8s enable metallb
```

MicroK8s が IP 範囲を要求するときは、ubuntu サーバーの IP アドレスを範囲（ハイフンで区切る）として <IP アドレス>-<IP アドレス> の形式で指定します。

たとえば、シングルノードクラスタの場合、Ubuntu サーバの IP アドレスが 1.2.2.1 の場合、次のように範囲を指定します。

1.2.2.1-1.2.2.1.

マルチノードクラスタの場合は、IP 範囲を次のように指定します。

1.2.2.1-1.2.2.3

iii. Storage

このアドオンを有効にするには、次のコマンドを実行します。

```
sudo microk8s enable storage
```



次のコマンドを使用して、これら 3 つのアドオンを一度に有効にすることができます。

```
sudo microk8s enable dns metallb storage
```



5. 設定ファイルを生成するには、Ubuntu サーバーで次のコマンドを実行します。

```
sudo microk8s config
```

このコマンドは、端末に結果を出力します。

6. 画面から内容をコピーして、空白のファイルに貼り付けます。
7. ファイルに `config` という名前を付けます。
8. 構成ファイルを Windows マシンの `¥Users¥<Windows User>¥.kube¥` にコピーします。
9. Magic xpi ダウンロード センターと Azure Container Registry (ACR) にアクセスすると、すべての Magic xpi Docker イメージ、Kubernetes Helm チャート、Cloud Manager セットアップ シェルスクリプトをダウンロードできます。
10. ドメイン名用に作成された SSL 証明書の公開 (.crt)、非公開 (.key)、およびクライアント証明書 (bundle.crt) ファイルを取得します。選択したフォルダにそれらを配置します。デプロイメント・プロセス中にパスの入力を求められます。
11. 次のツールを使用して、Windows ホストマシンを準備します。

- [Chocolatey](#) (choco) or equivalent package manager (optional)
- [Kubernetes CLI](#) (kubectl)
- [Helm](#) v3. Recommended version – v3.12.3
- OpenSSL recommended version – v3.1.1

ハードウェア要件

Cloud Native は Kubernetes 上で実行されるため、Linux 上で動作するサーバーが必要です。

ハードウェア仕様の最小要件は次のとおりです。プロジェクトの規模や要件に応じて、スループットやその他のパフォーマンスに関する考慮事項に基づいてスケーリングする必要がある場合があります。

Linux Sever

- CPU: 8 Core/1651Mhz
- RAM: Minimum 16 GB
- HDD: Minimum 70 GB

注: Cloud Native は、Ubuntu 24.04 LTS および RedHat RHEL 9.3 でテストおよび認定されています。





- ・ 仮想化で実行されている Ubuntu サーバーは、動的に割り当てられた CPU とメモリで構成されていると失敗することがわかっています。

Microk8s Kubernetes クラスターは、このような環境では機能しません。

- ・ ホストサーバーを設定するときは、ホスト名を小文字で設定する必要があります。



セキュリティとファイアウォールに関する考慮事項

特定のウイルス対策ソフトウェアが、Cloud Manager と MicroK8s の適切な機能を妨げます。Cloud Manager と MicroK8s をインストールする前に、まずこれらのプログラムを無効にするか閉じることをお勧めします。また、セキュリティの問題を制限するために、あまり多くのポートを開かないことをお勧めします。

ファイアウォールソフトウェアを使用している場合、IMM が正しく機能するには、ファイアウォールで次のポートを開く必要があります。

Port	Protocol	Purpose	External/ Internal
10250	TCP	Kubelet API (Kubernetes)	I
10255	TCP	読み取り専用の Kubelet API (Kubernetes)	I
12379	TCP	Calico Node API (Calico を使用した Kubernetes ネットワーキング)	I
10257	TCP	Kubernetes Controller Manager (API)	I
10259	TCP	Kubernetes Scheduler API、ポッドをスケジュールするための API	I
4789	TCP/ UDP	Virtual Extensible LAN (VXLAN), コンテナネットワーキングで使用	I
10248	TCP	Kubernetes KubeProxy, Kubernetes でのサービスプロキシの API	I
10249	TCP	Kubernetes Proxy (kube-proxy の別のコンポーネント)	I
10251	TCP	Kubernetes Controller Manager (別の API コンポーネント)	I
10252	TCP	Kubernetes Scheduler, Kubernetes でのタスクのスケジューリングを処理	I
10256	TCP	Kubernetes Scheduler, ポッドスケジューリングに関連する API を処理	I
2380	TCP	etcd (Kubernetes key-value store), クラスタの状態の保存に使用	I

1338	TCP	カスタムまたはアプリケーション固有で、特定の Kubernetes/クラスター アプリで使用されます	I
------	-----	--	---

4443	TCP	Kubernetes アプリケーションのセキュア接続(SSL/TLS)に使用	E
6443	TCP	Kubernetes API Server, Kubernetes クラスター管理に使用	E
7946	TCP	ノード間の通信	I
7472	TCP	Docker Swarm, マネージャーノード間の通信を処理	I
10253	TCP	Kubernetes controller manager (Kubernetes クラスターの管理と実行の一部)	I
10254	TCP	Kubernetes scheduler, Kubernetes でワークロードをスケジュールするために使用されます	I
10258	TCP	Kubernetes kube-proxy, Kubernetes のポッド間のネットワークを処理します	I
179	TCP	Border Gateway Protocol (BGP), 異なるネットワーク間のルーティングに使用	E
2379	TCP	etcd (クラスターの状態管理のための Kubernetes key-value store)	I
2381	TCP	etcd (Kubernetes の etcd の第 2 通信ポート)	I
2379-2380	TCP	etcd port range (クライアント通信のための Kubernetes key-value store の範囲)	I
25000	TCP	Custom application-specific port, 内部サービスに使用	I
4789	UDP/TCP	Virtual Extensible LAN (VXLAN), Kubernetes でオーバーレイネットワークで使用)	I
5473	TCP	Custom port, 特定のサービスや Kubernetes 関連のアプリケーション、共有デバイスなど、さまざまな用途に利用可能	I

8285	UDP	Flannel (ポッド間通信のための Kubernetes ネットワークオーバーレイ)	I
8472	UDP	Flannel VXLAN, Kubernetes のネットワーキングに使用	I
16443	TCP	Secure Kubernetes API, 管理/セキュアな通信に使用	E
19001	TCP	必要な適切なトークンを確認するために使用	E
9443	TCP	Secure HTTPS connection, 多くの場合、Kubernetes 管理またはセキュリティで保護されたエンドポイント用	E
8285	UDP	Flannel networking, Kubernetes でのポッド間通信に使用	I
8472	UDP	Flannel VXLAN, Kubernetes ネットワーキングでオーバーレイ通信に使用	I
53	UDP	Domain Name System (DNS), ドメイン名を IP アドレスに解決するために使用	E
5353	UDP	Multicast DNS (mDNS), ローカルネットワーク環境でのサービス検出に使用	I
7472	UDP	Stateless transport tunnelling protocol	I
30000–32767	TCP	アプリケーション間の通信、リモートアクセス、ファイル転送	E
9000	TCP	一般的な Web アプリケーションまたは API は、ファイル管理や Web サービス等に使用	E
9153	TCP	監視とテレメトリのための Prometheus メトリクスエクスポーター	I
9253	TCP	Prometheus メトリクスエクスポーター (通常は Kubernetes で監視するために使用)	I

27017	TCP	MongoDB (アプリケーションで使用する NoSQL データベース)	I
6379	TCP	Redis (インメモリデータ構造ストア、アプリケーションでのキャッシュ等に使用)	I
5000	TCP	ユニバーサル plug & play ポート	E
5119	TCP	DNS ルーティング情報プロトコルと簡易ネットワーク管理プロトコル	I

デプロイメント環境でこれら以外のポートを使用する場合は、ファイアウォールの通過を許可してください。

EKS の前提条件

次の必須要件を満たす必要があります。

1. EKS クラスターが構成されている必要があります。
2. クラスターに接続してアクセスするには、管理者アクセス権が必要です。

EKS への接続

1. ローカルマシン/ジャンプサーバー/BASTION サーバー/管理者アクセス権を持つその他のリモートサーバーから AWS アカウント/クラスターに接続します。
2. .kube 設定ファイルを更新します。

たとえば、次の構文を参照してください。

```
aws eks update-kubeconfig --region <region_name> --name <cluster_name>
```

インストール

このセクションでは、Cloud Manager のインストールと必要な証明書について説明します。

Cloud Manager のインストール

1. インストーラーが配置されている場所を開きます。



例:

Magic xpi 4.14.1¥InMemoryMiddleware¥deploy¥

2. 環境に応じて deployment.ini ファイルに変更を加えます。
 - subdomain_name: サブドメイン名。サブドメイン名はすべて小文字の英数字にする必要があります。
 - domain_name: ドメイン名。ドメイン名はすべて小文字の英数字にする必要があります。
たとえば、想定されるサーバー エンドポイントが xpdev.magiccloudenvironment.com の場合、magiccloudenvironment.com はドメイン名であり、xpdev はサブドメイン名です。
 - domain_name.crt path: SSL 証明書の公開鍵 (*.crt ファイル) へのパス
 - domain_name.key path: SSL 証明書の秘密鍵へのパス (.key ファイル)
 - bundle_cert.crt path: クライアント証明書へのパス (.crt ファイル)
 - Cluster_IP: Kubernetes CIDR ブロックで未使用の IP アドレス。
 - Registry_IP: Kubernetes CIDR ブロックの未使用の IP アドレスで、内部コンテナレジストリによって使用されます。このスクリプトは、CIDR ブロックを提供し、使用済みの IP の使用も制限します。
 - K8sDistribution: EKS 環境の場合、パラメータ値は eks、オンプレミス環境の場合、パラメータ値は microk8s になります。
3. 管理者モードでコマンドプロンプトを開く。
4. deploy-cloud-manager.bat スクリプトファイルを実行します。

インストールが成功すると、Cloud Manager が Kubernetes クラスターにデプロイされます。



```
Administrator: C:\Windows\System32\cmd.exe - deploy-cloud-manager.bat
LAST DEPLOYED: Tue Nov 12 11:32:50 2024
NAMESPACE: mgc-manage
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTE: Magic Ingress Routes Setup Completed

-----
INFO: Magic Cloud Manager Deployed Successfully

===== Magic Cloud Manager Deployment Complete =====

===== You can access Magic Cloud Manager Here =====
Magic Cloud Manager: https://xpiqa.magiccloudenvironment.com
=====Action needed =====

IMP Add the following to your System host file for name resolution.
for e.g. in windows "C:\WINDOWS\system32\drivers\etc\hosts"

-----
10.9.4.80    xpiqa.magiccloudenvironment.com
-----
INFO: The end points and the host entry are also saved in "MCMPoints.txt"

-----
You can access xpi service using respective url.

-----
"Magic Monitor URL      : https://<ENV_SUBDOMAIN_NAME>.magiccloudenvironment.com/<CUSTOMER_NAME>/<ENVIRONMENT_NAME>/magicmonitor/"
"Debugger URL          : wss://<ENV_SUBDOMAIN_NAME>.magiccloudenvironment.com/<CUSTOMER_NAME>/<ENVIRONMENT_NAME>/xpidebuggerservice"
"SOAP Services URL     : https://<ENV_SUBDOMAIN_NAME>.magiccloudenvironment.com/<CUSTOMER_NAME>/<ENVIRONMENT_NAME>/webseviceprovider"
"Web Requester URL     : https://<ENV_SUBDOMAIN_NAME>.magiccloudenvironment.com/<CUSTOMER_NAME>/<ENVIRONMENT_NAME>/xpiwebrequester/Magicxpi4.14.1/MgWebRequester.dll"
"ODATA Provider URL    : https://<ENV_SUBDOMAIN_NAME>.magiccloudenvironment.com/<CUSTOMER_NAME>/<ENVIRONMENT_NAME>/xpiodataservice"
"Rest Api Server URL   : https://<ENV_SUBDOMAIN_NAME>.magiccloudenvironment.com/<CUSTOMER_NAME>/<ENVIRONMENT_NAME>/xpirestservice"
"MQTT Broker URL       : ssl://<ENV_SUBDOMAIN_NAME>.magiccloudenvironment.com:<BROKER_PORT>"
"Data Transfer URL     : https://<ENV_SUBDOMAIN_NAME>.magiccloudenvironment.com/<CUSTOMER_NAME>/<ENVIRONMENT_NAME>/mgmt-console-api/"
"LA web-console URL    : https://<ENV_SUBDOMAIN_NAME>.magiccloudenvironment.com/<CUSTOMER_NAME>/<ENVIRONMENT_NAME>/mgmt-console-api/"

-----
"One need to replace actual <ENV_SUBDOMAIN_NAME>,<CUSTOMER_NAME>,<ENVIRONMENT_NAME>,<BROKER_PORT>,<DATA_TRANSFER_PORT> in above URLs"

You may review the results, or press "x" to exit: _
```

これで、Cloud Manager の導入準備が整いました。

hosts ファイルの設定

ドメインが一般公開されている場合は、このセクションをスキップしてください。

Cloud Manager を実行しているドメインがインターネットに登録されていない場合は、IP アドレスをマップしてドメイン名を解決する必要があります。

スクリプト結果ファイル MCM_EndPoints.txt の指示に従ってください。

必要な証明書のインストール

クラウドにデプロイされた周辺サービスにアクセスするには、まず、SSL 証明書(バンドル証明書)を Magic xpi のインストールで使用される 32 ビット(および必要に応じて 64 ビット)の JRE にインストールする必要があります。これらの証明書は、Cloud Manager のセットアップ時に使用したのと同じである必要があります。

証明書がないと、Webservice や SAP ERP などの一部のコネクタの xpi Studio で、構成されたリソースの検証が失敗する可能性があります。

SAP ERP コネクタがローカル エージェント経由で接続されている場合は、リソースで [開発] の [LA] を [True] に設定します。次のコマンドは、SSL 証明書のインストールに役立ちます。

```
keytool -importcert -file "<Path to the SSL certificate file>" -alias imm -keystore "<Magic xpi installation>%Runtime%\32/64 bit
```



JDK>%jre%lib%security%cacerts"

例えば、証明書が c:%CloudNative%magiccloudenvironment.crt にあり、Magic xpi が c:%Magic xpi 4.14.1 にインストールされている場合、コマンドは次のようになります:

- keytool -importcert -file
"c:%CloudNative%magiccloudenvironment.crt" -alias imm -keystore
"c:%Magic xpi 4.14.1%Runtime%JDK8x86%jre%lib%security%cacerts"
- keytool -importcert -file
"c:%CloudNative%magiccloudenvironment.crt" -alias imm -keystore
"c:%Magic xpi 4.14.1%Runtime%JDK8x64%jre%lib%security%cacerts"

Magic xpi インストールの JDK 以外の JDK を使用している場合は、以下に示すのと同じコマンドを使用して、その JDK にも証明書をインストールする必要があります。

keytool -importcert -file "<Path to the certificate file" -alias imm keystore "<JDK Path>%jre%lib%security%cacerts"

Cloud Manager のアンデプロイ

クラスターから Cloud Manager をアンデプロイするには、コマンド・プロンプトから undeploy-cloudmanager.bat という名前のバッチ・ファイルを実行します。

```
C:\Magic xpi 4.14\CloudNative>undeploy-cloud-manager.bat
NOTE: Magic Xpi Cloud Native Uninstallation Completed
NOTE: KubernetesCluster : Removing Magic Cloud Manager Infrastructure
release "magic-container-registry" uninstalled
release "magic-cloud-manager" uninstalled
release "magic-ingress-routes" uninstalled
release "magic-ingress-controller" uninstalled
secret "domain-certs-secret" deleted
secret "magic-ingress-controller-ingress-nginx-admission" deleted
namespace "mgc-manage" deleted
NOTE: KubernetesCluster : Magic Cloud Manager Infrastructure Removed Completely
```

Cloud Manager がアンインストールされたことを確認するには、次のコマンドを実行します:

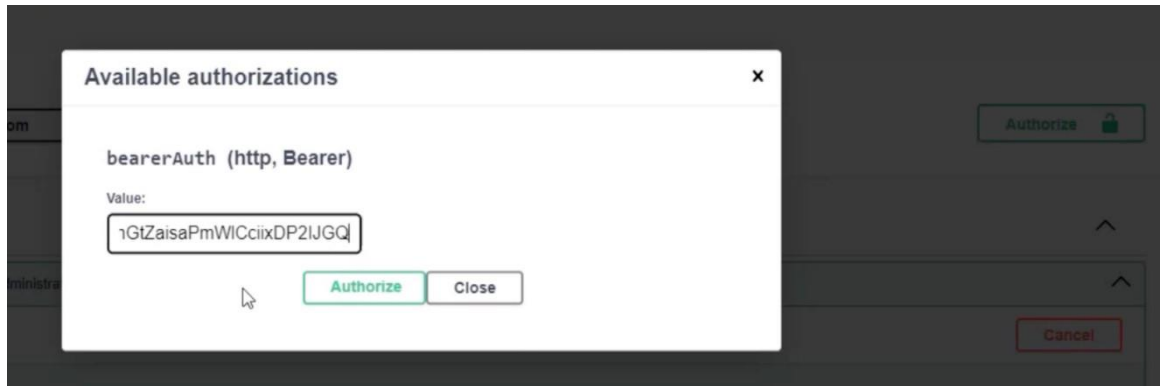
```
kubectl get pods -A
```

このコマンドは、Kubernetes 固有のポッドのみを表示し、IMM または Magic xpi のポッドは表示しません。

クラウドマネジメント

次のセクションでは、クラウド管理について説明します。





ベアラートークンはブラウザのキャッシュに保存されます。

既存のトークンを削除する場合は、[ログアウト]ボタンをクリックします。



これで、Cloud Manager API を使用して環境を作成し、周辺機器をデプロイし、xpi プロジェクトを管理する準備が整いました。

管理者の詳細の更新

Cloud Manager API 名: 管理者の詳細を更新

メソッド: Put

エンドポイント: https://<subdomain_name>.<domain_name>/auth/credentialsupdate [Request Body](#)

Parameters

email: 更新するユーザーのメールアドレス. password: The password for the user that you want to update.



パスワードは一度更新すると回復できないため、メモしておいてください。

API が実行された後、リクエストが成功すると、次のレスポンスが返されます。



“ユーザー詳細が正常に更新されました”

Magic Cloud Manager のヘルスチェック

Cloud Manager API Name: Health Check

Method: Get

Endpoint: https://<subdomain_name>.<domain_name>/manage/health

API が実行された後、リクエストが成功すると、次のレスポンスが返されます:

“Magic Cloud Manager is up and Running”

環境を Kubernetes クラスタにデプロイする

Cloud Manager API Name: Setup Environment

Method: Post

Endpoint:

https://<subdomain_name>.<domain_name>/manage/{customer}/environment/s etup

パスパラメータ

Customer: 顧客の名前

リクエストボディパラメータ

- **Deployment:**
 - Single Node
 - Multi Node
- **domain:** <sub-domain_name>/<domain_name>. 例: xpi.magic.com. ◦
- **product:** - xpi とだけ入力
- **Environment name:** 環境の名前 (例, test/prod/staging) ◦
- **version:** xpi のバージョン. 例, 4.14.1.
- **log_level:** IMM Controller、IMM Tunnel、および IMM-si(IMM サーバ)の異なる POD のログのログレベルを設定します。デフォルトのログ・レベルは INFO です。サポートされているログ・レベルは INFO と DEBUG です。
- **imm_db_username:** IMM DB(Redis)のユーザー名を入力します。
- **imm_db_password:** IMM DB(Redis)のパスワードを入力します。



- **imm_db_port**: IMM DB(Redis)が実行されているポートを入力します
- **log_db_username**: LOGDB(Mongo)のユーザー名を入力します.
- **log_db_password**: LOGDB(Mongo)のパスワードを入力します.
- **log_db_port**: LOG DB(Mongo)が実行されているポートを入力します
- **timezone**: Magic Monitor のタイムゾーンを入力します (Activity Logs).
- **license**: オプションで、環境のデプロイ中にライセンスファイルを提供できます
- **configjson**: SSO ログイン用の configjson ファイルをアップロードします (Magic Monitor).

API の実行後、セットアップ要求が成功した場合は、次の応答が返されます:

"Magic Xpi Environment Setup Request Sent Successfully"

この API は、IMM のサービスを開始し、Magic xpi Monitor のゼロ環境を提供します.

これで、以下の URL 形式を使用して xpi モニターにアクセスできます:

https://<subdomain_name.domain_name>/<customer>/<environment>/magicmonitor/

ハイフン、特殊文字、大文字は入力しないでください。

Kubernetes クラスタからの環境のアンデプロイ/削除

Cloud Manager API Name: Remove Environment

Method: Post

Endpoint:

https://<subdomain_name>.<domain_name>/manage/<customer>/<environment>/remove

パスパラメータ

customer: 顧客の名前

environment: 削除する必要がある環境の名前 (例: test/prod/staging)

リクエストボディパラメータ

product: xpi とだけ入力



version: Magic xpi 製品版。例、4.14.1

API の実行後、削除要求が成功すると、次の応答が返されます:

"Magic xpi Environment Remove Request Sent Successfully"

環境を取り外す前に、周辺機器を取り外すことをお勧めします。

Kubernetes クラスタへの周辺サービスのデプロイ

Cloud Manager API Name: Setup Peripherals

Method: Post

Endpoint: https://<subdomain_name>.<domain_name>/manage/{customer}/
{environment}/peripherals/setup

パスパラメータ

customer: 顧客の名前

environment: 環境の名前 (例, test/prod/staging)

リクエストボディパラメータ

- **version:** Magic xpi のバージョン。例、4.14.1 です
- **xpi_debugger:** 1 でデバッガー サービスをインストールし、0 でインストールをスキップします
- **xpi_webrequester:** 1 で Web リクエスター サービスがインストールされ、0 でインストールがスキップされます
- **xpi_odata:** 1 で OData サービスをインストールし、0 でインストールをスキップします
- **xpi_webservice:** 1 で Web サービスをインストールし、0 でインストールをスキップします
- **xpi_restapiservice:** 1 で REST サービスをインストールし、0 でインストールをスキップします

API の実行後、デプロイ要求が成功すると、次の応答が返されます。

"Magic xpi Peripherals Setup Request Sent Successfully"



これらの周辺サービスにアクセスするには、「[クラウド内のサービスへのアクセス](#)」セクションを参照してください。

Kubernetes クラスタからの周辺機器サービスの削除

Cloud Manager API Name: Remove Peripherals

Method: Post

Endpoint: https://<subdomain_name>.<domain_name>/xpi/env/remove/ peripherals

パスパラメータ

customer: 顧客の名前

environment: 周辺サービスを削除する環境の名前 (例: test/prod/staging)

リクエストボディパラメーター

- **version:** Magic xpi のバージョン。例、4.14.1
- **xpi_debugger:** 1 で Debugger サービスをアンインストールし、0 でアンインストールをスキップします
- **xpi_webrequester:** 1 で Web リクエスター サービスをアンインストールし、0 でアンインストールをスキップします
- **xpi_odata:** 1 で OData サービスをアンインストールし、0 でアンインストールをスキップします
- **xpi_webservice:** 1 で Web サービスをアンインストールし、0 でアンインストールをスキップします
- **xpi_restapiservice:** 1 で xpi restful サービスをアンインストールし、0 でアンインストールをスキップします

API の実行後、削除要求が成功すると、次の応答が返されます:

“Magic xpi Peripherals Remove Request Sent Successfully”

Kubernetes クラスタでのローカル エージェントのセットアップ

この API は、クラスター上のローカル エージェントを設定します。



ローカルエージェントサービスをインストールする前に、[Cloud Manager](#) でのローカルエージェントの使用を参照してください

Cloud Manager API Name: Setup Local Agent

Method: Post

Endpoint:

`https://<subdomain_name>.<domain_name>/manage/<customer>/<environment> /localagent/setup`

パスパラメータ

customer: 顧客の名前

environment: Local Agent をインストールする環境の名前.

Request Body Parameters

- **Version:** Magic xpi のバージョン。例、4.14.1
- **minio_tcp_port:** データストレージサービスのポート
- **minio_console_port:** データ・ストレージ・サービスのコンソール・ポート
- **mosquitto_tcp_port:** ブローカー URL のポート
- **broker_username:** ブローカー URL のユーザー名
- **broker_password:** ブローカー URL のパスワード
- **management_id:** すべてのローカル エージェント コンポーネント間で共有されるローカル管理 ID

ローカル エージェントのセットアップが完了すると、IMM クラスタに以下がインストールされます。

- Mosquitto
- Minio
- Web コンソール (ローカル エージェント管理コンソール)

これで、URL を使用してローカル エージェント管理コンソールにアクセスできるようになりました:

`https://<subdomain_name>.<domain_name>/<customer>/<environment>/mgmt-console-api/`

例,

`https://your.domain.com/mse/prod/mgmt-console-api/`



Kubernetes クラスタからのローカル エージェントの削除

この API は、クラスターにインストールされているローカル エージェントを削除します。Minio、Mosquitto、およびローカルエージェント管理コンソールを削除します。

Cloud Manager API Name: Remove Local Agent

Method: Post

Endpoint: https://<subdomain_name>.<domain_name>/manage/<customer>/<environment>/localagent/remove

パスパラメータ

customer: 顧客の名前

environment: Local Agent を削除する環境の名前

リクエストボディパラメータ

Version: Magic xpi のバージョン。例、4.14.1

プロジェクトのデプロイ

この API は、プロジェクトをクラスターにデプロイします。

Cloud Manager API Name: Deploy Project

Method: Post

Endpoint: https://<subdomain_name>.<domain_name>/manage/<customer>/<environment>/deploy

パスパラメータ

customer: 顧客の名前

environment: プロジェクトを配置する環境の名前。

リクエストボディパラメータ

◦ **state:**

- **start** – プロジェクトをデプロイします
- **hotswap** – 既存の実行中のプロジェクトをホットスワップします。プロジェクトをホットスワップする前に、新しい変更を加えてプロジェクトをビルドし、プロジェクトの zip、env.ini、およびプロジェクトのハッシュキーをプロジェクトのデプロイエンドポイントにアタッチする必要があります。



ホットスワッププロセスに関する詳細は、Magic xpi ヘルプドキュメントのプロジェクトのホットスワップページを参照してください。



- Cloud Manager を使用してクラウド環境でプロジェクトをホットスワップすると、スワップされたプロジェクトの実行中のすべてのポッドが再起動されます。内のサーバーの複数のインスタンスを実行しているポッドは認識されないため、同じカウントに復活することはありません。
- Magic xpi プロジェクトを Cloud Manager からデプロイすると、プロジェクトのインスタンスが 1 つ作成されます。複数のインスタンスにスケーリングします。xpi Monitor から行うか、Deploy Project API を使用して同じプロジェクトを複数回デプロイできます。

プロジェクトが xpi Monitor からスケーリングされると、プロジェクトの複数のインスタンスが同じプロジェクト POD 内でスケーリングされます。Cloud Manager の Deploy Project API を使用してプロジェクトをデプロイすると、インスタンスごとに IMM クラスタ内のプロジェクトの新しい POD が作成されます。

- projectname:** プロジェクト名
- projectzipfile:** Studio でプロジェクトをビルドするときに作成されるプロジェクトの zip ファイル。これは、プロジェクトフォルダ内の Docker_<projectname>フォルダの下にあります。Projectzip ファイルでサポートされる最大アップロード サイズは 500 MB です。
- context:** env.ini file. これは、プロジェクトフォルダ内の Docker_<project_name>フォルダの下にあります。
- projecthash:** ProjectBuildHash テキスト ファイルのハッシュ テキスト。これは、プロジェクトフォルダ内の Docker_<projectname>フォルダの下にあります。ハッシュ値はハッシュの検証に使用され、ハッシュが一致する場合はデプロイが続行されます。
- startxml:** プロジェクトのデプロイ中に start.xml ファイルを指定して、プロジェクトの複数のインスタンスを開始できます。このパラメータはオプションです。
- license:** You can provide license file, while deploying the project. The selected license file will override previous license across the current environment. This parameter is optional.



プロジェクトの zip、プロジェクトのハッシュ、および env.ini ファイルは、プロジェクトの変更および再構築時に変更されます。そのため、送信する前にこれらのファイルを再度アップロードする必要があります

API の実行後、デプロイ要求が成功すると、次の応答が返されます。

“Magic xpi Project Deploy Request Sent Successfully”



これで、xpi Monitor でプロジェクトを監視できます。

プロジェクトのアンデプロイ

この API は、プロジェクトを停止し、クラスターからプロジェクトを削除します。

Cloud Manager API Name: Undeploy Project

Method: Post

Endpoint: https://<subdomain_name>.<domain_name>/manage/<customer>/<environment>/undeploy

パスパラメータ

customer: 顧客の名前

environment: プロジェクトのデプロイを解除する環境の名前 (例: test/prod/staging)

リクエストボディパラメータ

projectname: プロジェクト名

API の実行後、削除要求が成功すると、次の応答が返されます:

"Magic xpi Project Remove Request Sent Successfully"

これで、Magic xpi Monitor でプロジェクトを監視できます。その後、プロジェクトの状態が RUNNING から STOP_IN_PROGRESS に変わります。

ログを生成して更に調査する

Cloud Manager API Name: Collect Logs

Method: Post

Endpoint: https://<subdomain_name>.<domain_name>/manage/<customer>/<environment>/logs

パスパラメータ

customer: 顧客の名前

environment: ログをダウンロードする環境の名前 (例: test/prod/staging)



リクエストボディパラメーター

- logsforenvironment: imm, projects

imm の下に imm_controller、imm-si、xpi-monitor のログを出力できます。

- **logsforperipherals:** debugger, webrequester, odata, webserviceprovider, restfulapiprovider, localagent, および laclient.
- ログの生成は選択することができ、imm (In Memory Data Grid) のみのログをダウンロードする場合は、ログ生成 API に入力パラメーターとして imm のみを渡すことでダウンロードできます。
例、顧客名が mse で、環境が prod の場合、URL は次のようになります。

`https://your.domain.com/manage/mse/prod/logs logsforenvironment : imm`

ペリフェラル・サービスのログを生成しない場合は、入力パラメータに 0 (ゼロ) を渡す必要があります。同じことが環境固有のログにも当てはまります。

ローカルワークステーションで生成されたログのダウンロード

Cloud Manager API Name: Download Logs

Method: Get

Endpoint: `https://<subdomain_name>.<domain_name>/manage/<customer>/<environment>/logs`

パスパラメータ

customer: 顧客の名前

environment: ログのダウンロード元となる環境の名前 (例: test/prod/staging)

ログは、Generate Logs API の一部として提供されるサービスのログがダウンロードされます。

The logs will be downloaded in the working folder on the local machine.

INI/アラート設定のアップロード

Cloud Manager API Name: Upload INI/Alert Configurations

customer: 顧客の名前

environment: ファイルをアップロードする環境の名前。例、test/prod/staging



リクエストボディパラメーター

- **mgreqini**: HTTPTimeout などに変更を加えた後で、mgreq.ini アップロードできます。これはオプションパラメータです。
- **dbalertconfig**: 更新されたメールの詳細を含む db_alert_config.properties をアップロードすると、Magic xpi Monitor で生成されたアラートメールを受信できます。これはオプションパラメータです。
-

プロジェクト/周辺ファイルのアップロード

Cloud Manager API Name: Upload Project/Peripheral Files

customer: 顧客の名前

environment: ファイルをアップロードする環境の名前。例、test/prod/staging

リクエストボディパラメーター

- **restful**: RESTful API サービス定義ファイルをアップロードします。
- **odata**: OData サービスの XML ファイルをアップロードします。
- **webservice**: 外部 WSDL をアップロードします。

クラウドサービスへのアクセス

クラウド上に保存された以下のサービスにアクセスできます。

デプロイされたすべてのサービスはセキュリティで保護されているため、周辺サービスを使用する前に、適切な証明書がインストールされていることを確認してください。詳細については、「[必要な証明書のインストール](#)」セクションを参照してください。

Cloud Manager

To access the Cloud Manager Web interface to the API, go to the link:

subdomain_name.domain_name/swagger/

For example,

Your.domain.com/swagger/



Magic xpi Monitor

xpi Monitor UI にアクセスするには、リンクにアクセスしてください:

<subdomain_name.domain_name>/<customer>/<environment>/magicmonitor/

たとえば、顧客名が **mse** で、environment が **prod** の場合、URL は次のようになります:

Your.domain.com/**mse**/**prod**/magicmonitor/.

デバッガ

xpi Studio からデバッガーにアクセスするには、以下の形式の URL を使用します:

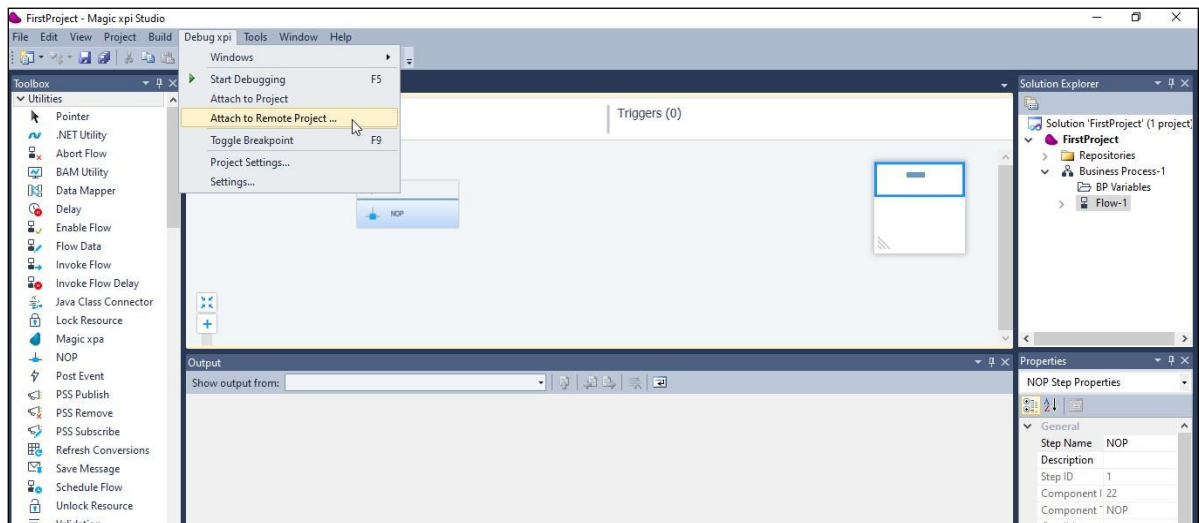
wss://<subdomain_name.domain_name>/<customer>/<environment>/xpidebuggerservice

たとえば、顧客名が **mse** で、environment が **prod** の場合、URL は次のようになります:

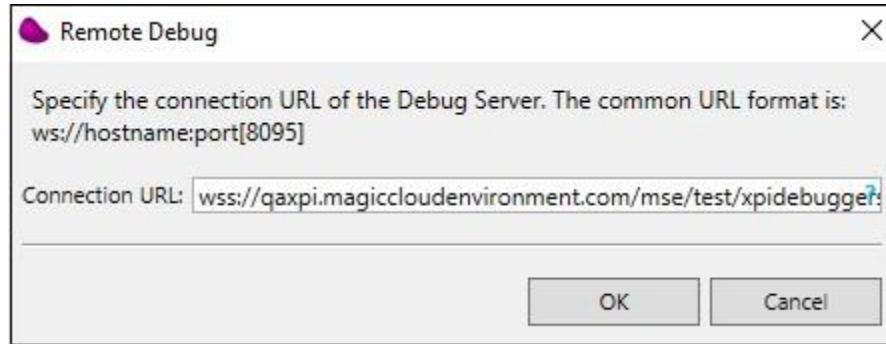
wss://subdomain_name.domain_name/**mse**/**prod**/xpidebuggerservice

スタジオからこのデバッガー サービスを使用するには、次の手順に従います:

1. Cloud Manager API を使用して、クラウドにプロジェクトをデプロイして開始します。
2. Magic xpi Studio で同じプロジェクトを開き、メニューから デバッグ xpi > リモートプロジェクト にアタッチを選択します。



3. 「リモートデバッグ > 接続 URL」フィールドにデバッガ URL を追加し、「OK」をクリックします。



これで、アプリケーションをリモートデバッグできます。

OData

クラウドにデプロイされた OData サービスには、URL を使用してアクセスします:

`https://<subdomain_name.domain_name>/<customer>/<environment>/xpiodataservice/<OData namespace>/<OData Service name>`

たとえば、顧客名が **mse** で、環境が **prod** の場合、OData リソースのルート URL は次のようになります。

`https://your.domain.com/mse/prod/xpiodataservice/magicxpiodata/SampleN S.OData_simple_test`
To access this service use this as the Root URL in the OData Resource in your project.

RESTful API

Cloud にデプロイされた RESTful API サービスには、URL を使用してアクセスします:

`https://<subdomain_name.domain_name>/<customer>/<environment>/xpirestservice/xpiREST/<project _name>/<RESTfulAPI_service_name>/<endpoint>`

たとえば、顧客名が **mse**、環境が **prod**、プロジェクト名が **restful_test**、RESTfulAPI_service_name が **RESTful_API_1**、エンドポイントが **flowers** の場合、RESTful API リソースのルート URL は次のようになります。

`https://your.domain.com/mse/prod/xpirestservice/xpiREST/restful_test/RESTful_API_1/flowers`

このサービスにアクセスするには、プロジェクトの REST クライアント リソースのルート URL としてこれを使用します。



RESTful API (要求と応答) でサポートされる添付ファイルの最大サイズは 30 MB です。

Webservice (SOAP)

クラウドにデプロイされた Webservice 周辺サービスには、URL を使用してアクセスします

`https://<subdomain_name.domain_name>/<customer>/<environment>/web serviceprovider`

たとえば、顧客名が **mse** で、environment が **prod** の場合、URL は次のようになります:

`https://your.domain.com/mse/prod/webserviceprovider`

Magic xpi Studio の Webservice リソースでこのサービスを使用するには、Web サービスサーバーリソースを定義する際に、この URL をホスト名として使用します。

マイグレーションプロジェクトの Web サービススキーマの処理

要求と応答の Web サービス スキーマは、W3C 仕様に準拠しています。これにより、マイグレーションされた Magic xpi プロジェクトからこれらのサービスを使用する既存の Web サービスクライアントが失敗する可能性があります。下位互換性を維持するには、config.properties ファイル内のフラグ `magic.w3.concerted.response.name` の値を変更し、False に設定する必要があります。

config.properties ファイルは、次の場所にあります。

`/home/xpi-common-data/mgc-<customer>-<environment>/xpi-user/ webservice/webservice-config`

このフラグは、リモート プロシージャ コール (RPC) スタイルの要求にのみ適用されます。

ファイルに加えた変更を保存した後、Web Services Peripheral サービスを再デプロイする必要があります。

HTTP Service

クラウドにデプロイされた HTTP サービスには、以下の URL を使用してアクセスできます:

`https://<subdomain_name.domain_name>/<customer>/<environment>/xpiwebrequester/Magicxpi4.14.1/MgWebRequester.dll`



たとえば、顧客名が **mse** で環境が **prod** で、Magic xpi のバージョンが 4.14.1 の場合、URL は次のようになります:

`https://your.domain.com/mse/prod/xpiwebrequester/Magicxpi4.14.1/MgWebRequester.dll`

次に、この URL を使用してトリガーによって生成された HTML ファイルに変更を加える必要があります。

例、

`<form action="https:// your.domain.com/mse/prod/xpiwebrequester/`

`Magicxpi4.14.1/MgWebRequester.dll" enctype=text/html METHOD="post">`

MicroK8s が動作している Ubuntu サーバーの `/home/xpi-common-data/mgc-<customer>-<environment>/xpi-user/etc` にある `mgreq.ini` ファイルのタイムアウト値を変更することで、HTTP タイムアウト値を増やすことができます。

タイムアウト値を変更した後、xpiwebrequester 周辺機器サービスを再デプロイする必要があります。

既知の制限事項

- ODBC の DSN と Oracle データベースの TNS エントリを定義するプロビジョニングはありません。
- Cloud Manager が実行されているホストサーバへのファイルシステムアクセスを発行する必要があります。これは、次の必須コピー操作に適用されます。
- プロジェクトの Pod シェルのディレクトリリスト(`ls`)コマンドは、すべてのプロジェクトを表示します。これは、実際のプロジェクトの実行には影響しません。無視しても問題ありません。
- プロジェクトが非常に大きな IDoc ファイルを処理している場合、使用可能なヒープ・サイズが限られているために SAP ERP ステップの実行が停止することがあります。
- Local Agent で実行されているコネクタは、LA がホストされているホスト システムに対する論理名のパスを評価します。そのため、`currentprojectdir` のような論理名を使用し、コネクタ ステップ構成では相対パスを慎重に使用してください。
- IMM ログは、常にクラスタ ノードに保存されます。
- 複数の環境でプロジェクトをホットスワップする場合は、一度に 1 つのプロジェクトを入れ替えます。
- Magic xpi Monitor の「ログイン状態を維持する」オプションは、インスタンスを最大 15 分間クラウドに保持します。
- `magic.ini`、`mgreq.ini`、`mgenv.ini` などの標準の Magic xpi 設定ファイル は、ホストサーバー上のフォルダ `/home/xpi-common-data/mgc-<customer>-<environment>/xpi-user/etc/` にあります。



サポートされている構成の変更は、上記の場所にあるファイルで行う必要があります。以下のファイルで行われた変更については、次のサービスを再起動する必要があります:

- magic.ini – このファイルに変更を加えた場合は、プロジェクトを再デプロイする必要があります。
- mgreq.ini – このファイルに変更を加えた場合は、Web リクエスター周辺機器を再デプロイする必要があります。
- mgenv – このファイルに変更を加えた場合は、プロジェクトを再デプロイする必要があります。

サポートされているプロパティについては、Magic xpi ヘルプの「Home >of Reference Guide」>「Home of Reference Guide」および「Magic xpi 設定ファイル」セクションを参照してください。

Appendix

環境変数の設定

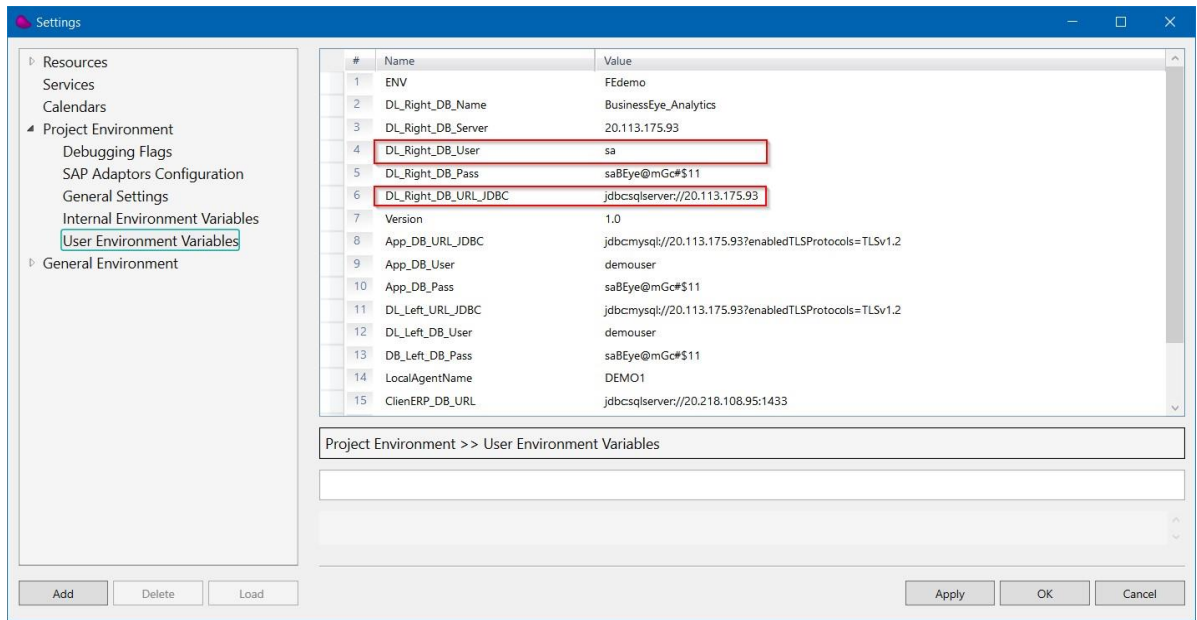
環境変数を使用すると、Magic xpi プロジェクトの環境プロパティを外部化できます。環境変数を作成すると、ifs.ini ファイルに追加されます。オンプレミス展開の場合は、このファイルを直接編集して必要な値を設定できます。クラウド環境では、ifs.ini ファイルを直接編集することはできません。

以下の手順に従って、クラウドの環境変数を更新します:

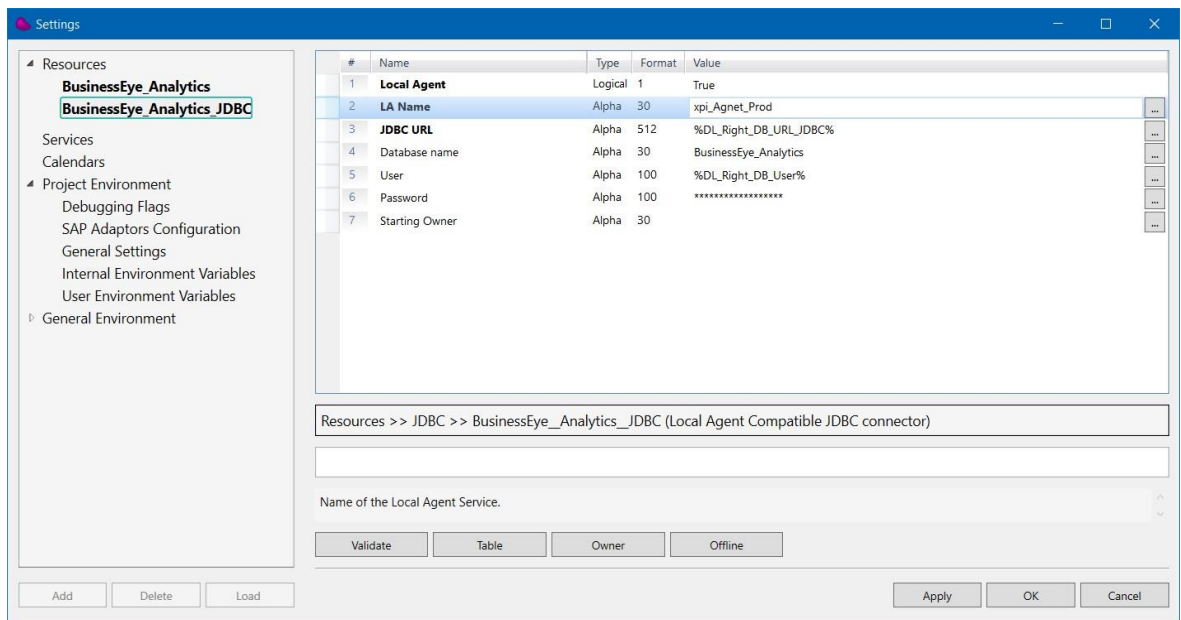
ロールと責任に基づいて、次の手順は開発者ロールと DevOps ロールに分類されます。大まかに言うと、開発者は最初の 3 つの手順を実行する必要があり、DevOps は次の手順を実行します。

1. 必要なユーザープロジェクト環境変数を xpi Studio に追加します。これにより、変数が ifs.ini に追加されます。
たとえば、スクリーンショットでは、次のようないくつかの変数が追加されていることがわかります

DL_Right_DB_URL_JDBC, DL_Right_DB_User



これらの変数は、JDBC リソースで JDBC URL、データベース名、およびユーザーに使用されます。



2. Magic xpi Studio でプロジェクトをビルドします。
3. プロジェクトがビルドされると、プロジェクトフォルダー内の <Docker_ProjectName> フォルダーに 3 つの Cloud デプロイ成果物が作成されます。この中で生成された env.ini ファイルがありますが、このファイルにユーザーが追加したすべてのプロジェクト環境変数が含まれています。

ユーザー定義のプロジェクト環境変数のみが env.ini ファイルに追加され、内部変数はスキップされることに注意してください。

4. テキストエディタで env.ini ファイルを開き、各変数に適切な値を設定して、ファイルを保存します。たとえば、上記の JDBC リソースで使用している変数に必要な値を設定しているとします。
5. 変更を保存したら、xpi Cloud Manager が提供するデプロイ API を使用してプロジェクトをデプロイします。この env.ini ファイルは、その API の 1 つのパラメーターとして提供されます。デプロイされたプロジェクトは、この env.ini ファイルで指定された値をソースにします。

```
1 #Project Environment Variables
2 [MAGIC_LOGICAL_NAMES]ENV=FEdemo
3 [MAGIC_LOGICAL_NAMES]DL_Right_DB_Name=BusinessEye_Analytics
4 [MAGIC_LOGICAL_NAMES]DL_Right_DB_Server=20.113.175.93
5 [MAGIC_LOGICAL_NAMES]DL_Right_DB_User=sa
6 [MAGIC_LOGICAL_NAMES]DL_Right_DB_Pass=saBEye@mGc#$11
7 [MAGIC_LOGICAL_NAMES]DL_Right_DB_URL_JDBC=jdbc:sqlserver://20.113.175.93
8 [MAGIC_LOGICAL_NAMES]Version=1.0
9 [MAGIC_LOGICAL_NAMES]App_DB_URL_JDBC=jdbc:mysql://20.113.175.93?enabledTLSProtocols=TLSv1.2
10 [MAGIC_LOGICAL_NAMES]App_DB_User=demouser
11 [MAGIC_LOGICAL_NAMES]App_DB_Pass=saBEye@mGc#$11
12 [MAGIC_LOGICAL_NAMES]DL_Left_URL_JDBC=jdbc:mysql://20.113.175.93?enabledTLSProtocols=TLSv1.2
13 [MAGIC_LOGICAL_NAMES]DL_Left_DB_User=demouser
14 [MAGIC_LOGICAL_NAMES]DB_Left_DB_Pass=saBEye@mGc#$11
15 [MAGIC_LOGICAL_NAMES]LocalAgentName=DEMO1
16 [MAGIC_LOGICAL_NAMES]ClientERP_DB_URL=jdbc:sqlserver://20.218.108.95:1433
17 [MAGIC_LOGICAL_NAMES]ClientERP_DB_NAME=ChocolateERP
18 [MAGIC_LOGICAL_NAMES]ClientERP_DB_USER=sa
19 [MAGIC_LOGICAL_NAMES]ClientERP_DB_Pass=saBEye@mGc#$11
20
```

Cloud Manager でのローカル・エージェントの使用

プロジェクトでローカル エージェントを使用するには、Cloud Manager からローカル エージェント周辺機器サービスをデプロイするだけでなく、Windows システムでいくつかの設定を行う必要があります。Windows システムに必要な構成を以下に示します。

ローカルエージェントのインストール

Cloud Manager でローカルエージェントを使用するには、正しい BrokerURL と DataTransferURL (ストレージ URL)が必要です。これらの URL は、以下の形式になります。

◦ BrokerURL= ssl://<subdomain_name>.<domain_name>:<mosquito_tcp_port>

例, ssl://your.domain.com:1883

◦ DataTransferURL=https://<subdomain_name>.<domain_name>:<minio_tcp_port>

例, https://your.domain.com:9000)

ローカル・エージェントをインストールするときに、インストール・ウィザードで上記の形式で BrokerURL および DataTransferURL を指定します。





- ローカル エージェントがすでにインストールされている場合は、<Local Agent Installation>%bin%AgentService と <Local Agent Installation>%bin%AgentConnector の下の application.properties ファイルで同じ変更を行う必要があります。

- ローカル エージェントを使用する前に、適切な証明書がインストールされていることを確認してください。詳細については、「[必要な証明書のインストール](#)」セクションを参照してください。

Local Agent をインストールした後、フォルダの<Magic Installation>Runtime\Magic xpa の下にある magic.ini ファイルで BrokerURL と DataTransferURL の値を設定します。また、このファイルに正しい ManagementID を設定します。値を設定したら、Local Agent を起動できます。これで、Cloud Manager で実行されているローカルエージェントサーバに接続されるようになります。

Local Agent で構成されたすべてのリソースの LA Name プロパティは、クラスターで構成された Local Agent Name と同じである必要があります。

キーポイント

[エージェント] タブの [エージェント名] をメモし、プロジェクトで使用するローカル エージェント対応コネクタがローカル エージェントで稼働していることを確認します。コネクタのステータスは、管理コンソールで確認できます。以下の画像を参照してください:

The screenshot shows the Magic Management Console interface. On the left is a sidebar with navigation links: All Management IDs, Agents, Agent Connectors, Connectors, Logs, Configuration, Event Log, and Settings. The main area is titled 'Agents' and contains a table of agents. Below the table is a section for 'Available Connectors' with a plus icon to add more.

Name	Management ID	ID	Host	OS	Version	Status
agent2	installation1	D5E46B6B-DB4A-4CB3-9BEC-4728729CB8E7	Minalhvmw10	windows	1.0.8-SNAPSHOT-4.13.6.47	Enabled

Connector Name	Version	Runtime Technology	Has Step	Has Trigger	Status	Action
FTP	1.84	Java	✓	✗	Available	
SAP ERP	1.47	Java	✓	✓	Available	
JDBC	1.21	Java	✓	✗	Running	



All Management IDs ▾

Agents

Agent Connectors

Connectors

Logs

Configuration

Event Log

Settings

<< Local Agent Management Console

Agent Connectors

Last sync: 17:07:24

Start New Agent Connector

Filter by:

Connector Name	Agent	Management ID	Project	Host	Mode	Status	Load	Requests	Invokes	ID	Last Updated	Actions
JDBC	agent2	installation1	NYProj2	Minikube	Data Mapper	Running	0 / 10	0 / 1 / 1 / 1		f6fa4e22-b22b-4bfb-92ca-15d9cded36ec	2 seconds ago	

0

About Magic Software Enterprises

Magic Software Enterprises (NASDAQ: MGIC) empowers customers and partners around the globe with smarter technology that provides a multi-channel user experience of enterprise logic and data.

We draw on 30 years of experience, millions of installations worldwide, and strategic alliances with global IT leaders, including IBM, Microsoft, Oracle, Salesforce.com, and SAP, to enable our customers to seamlessly adopt new technologies and maximize business opportunities.

For more information, visit <http://www.magicsoftware.com>.



Disclaimer

Magic® is the registered trademark of Magic Software Enterprises Ltd. All other product and company names mentioned herein are for identification purposes only and are the property of, and might be trademarks of, their respective owners.

Magic Software Enterprises has made every effort to ensure that the information contained in this document is accurate. However; there are no representations or warranties regarding this information, including warranties of merchantability or fitness for a particular purpose. In no event Magic Software Enterprises Ltd. will be liable for any loss of profit, business, or data, or any kind of indirect, incidental or consequential damages due to the use of this document. Magic Software Enterprises assumes no responsibility for errors or omissions that may occur in this document. The information in this document is subject to change without prior notice and does not represent a commitment by Magic Software Enterprises or its representatives.

Copyright © 1989–2024 Magic Software Enterprises. All rights reserved.

