NEC

WebOTX Application Server Express V10.4
Processor License for Container

UL1519-T9T

インストールガイド(Linux)

SL1519T9T02-2

ごあいさつ

このたびは、WebOTX Application Server Express Processor License for Containerをお買い上げいただき、まことにありがとうございます。

本書は、WebOTX Application Server Express のコンテナでのインストールとセットアップの内容を中心に構成されています。本製品をお使いになる前に、必ずお読み下さい。

本製品は、WebOTX Application Server Expressのコンテナ向けプロセッサ・ライセンス製品です。

以下からの説明では、WebOTX Application Serverを「WebOTX AS」と省略して表現します。

目次

1. はじめに 1

1.1. ライセンスについて 1
1.2. プロファイルについて 1
1.3. 諸元制限 2

2. 使用上の条件 4

2.1. ソフトウェア条件について 4
2.1.1. オペレーティング・システム 4
2.1.2. 必要なソフトウェア 4
2.1.3. 対応ソフトウェア 5

3. リソース 8

4. インストール(フルプロファイル) 9

4.1. インストール前の作業 9
4.1.1. Dockerの環境構築 (Red Hat Enterprise Linux 7 Serverの場合) 9
4.1.2. Podmanの環境構築 (Red Hat Enterprise Linux 8 ServerおよびOracle Linux 8の場合) 10
4.1.3. unzipコマンドのインストール (移行機能を利用する場合) 10
4.2. インストール 10
4.3. インストール後の作業 18
4.3.1. データベースを使用するための準備作業 (Java) 18
4.4. 動作確認 19

5. インストール(マイクロサービスプロファイル) 20

5.1. インストール前の作業 20
5.1.1. Dockerの環境構築 (Red Hat Enterprise Linux 7 Serverの場合) 20
5.1.2. Podmanの環境構築 (Red Hat Enterprise Linux 8 ServerおよびOracle Linux 8の場合) 20
5.1.3. Mavenのインストール 20
5.1.4. Java SDKのインストール 20
5.2. インストール 20
5.3. インストール後の作業 23
5.4. 動作確認 23

6. アンインストール 24

6.1. 不要なコンテナの削除 24

7. 注意・制限事項 25

7.1. フルプロファイル 25
7.2. マイクロサービスプロファイル 25

1. はじめに

本書では、コンテナ上で動作するWebOTX Application Server Expressについて説明します。

以下からの説明では、WebOTX Application Serverを「WebOTX AS」と省略して表現します。

1.1. ライセンスについて

コンテナ上のWebOTX Application Sever Expressは各コンテナに割り当てられるコア数を対象とし、2コアにつき1ライセンス必要になります。ただし、イメージ生成時に登録するライセンス数はその内の1ライセンスです。
※少数点以下の端数は切り上げです

WebOTX Application Server Expressを複数のコンテナで動作させる場合は、各コンテナに必要なライセンス数を合計したライセンス数が必要です。

(例1) 4個のコンテナで動作させ、それぞれに2コアずつ割り当て
1コンテナ当たりに必要なライセンス数 2 / 2 = 1 ライセンス
4個のコンテナで必要なライセンス数 1 (ライセンス/コンテナ) x 4 (コンテナ) = 4 ライセンス

(例2) コンテナAに1コア、コンテナBに2コア、コンテナCに3コア割り当て
コンテナAに必要なライセンス数 1 / 2 = 0.5 → (切り上げ) 1 ライセンス
コンテナBに必要なライセンス数 2 / 2 = 1 ライセンス
コンテナCに必要なライセンス数 3 / 2 = 1.5 → (切り上げ) 2 ライセンス
コンテナA, B, Cで必要なライセンス数 1 + 1 + 2 = 4 ライセンス

コンテナイメージ入れ替え時に一時的にコンテナ数が増加する場合、24時間以内にコンテナ数が元に戻るのであれば、増加するコンテナのコア数分の追加のライセンスは必要ありません。

Dockerの場合、コンテナに割り当てるコア数は、コンテナ作成時(docker run, docker create)に--cpusオプションで指定します。指定しない場合は、ホストマシンと同じコア数が割り当てられます。

1.2. プロファイルについて

WebOTX Application Sever Express Processor License for Containerはコンテナ上のWebOTX AS Expressのフルプロファイル(Windows/Linux)およびマイクロサービスプロファイル(Linuxのみ)で利用可能です。
フルプロファイルとはJava EEの全ての機能を提供するプロファイルです。フルプロファイルのコンテナイメージの作成には、コンテナで利用するOSにあわせてWindows版またはLinux版のWebOTX Media(※1)を利用します。
マイクロサービスプロファイルとはマイクロサービス向けにJava EEの一部の機能とEclipse MicroProfileの機能を提供するプロファイルです。マイクロサービスプロファイルのランタイムとユーザアプリケーションを1つのJARファイルにパッケージ化したものをUber JARと呼びます。また、Uber JARとDockerfile等のコンテナイメージ作成に必要なファイルをまとめてマイクロサービスパッケージと呼びます。Uber JARおよびマイクロサービスパッケージの作成には、Mavenを使用します(※2)。Uber JARおよびマイクロサービスパッケージの作成に必要なWebOTXマイクロサービスMavenプラグイン、WebOTXマイクロサービスMavenアーキタイプ、WebOTXマイクロサービスランタイムは、Maven Central Repositoryで公開するとともに、Windows版およびLinux版のWebOTX Media(※1)のメディアにも格納しています。
マイクロサービスプロファイルを1つのコンテナで複数起動することはサポート対象外です。複数のマイクロサー

ビスプロファイルを起動したい場合は、複数のコンテナを起動してください。
フルプロファイルとマイクロサービスプロファイルを1つのコンテナで同時に使用することはできません。

本ドキュメントでは、Linux版のフルプロファイルおよびマイクロサービスプロファイルに関して説明します。
Windows版のフルプロファイルに関してはインストールガイド(Windows)を参照してください。

(※1)WebOTX Mediaは出荷時期及び対応プラットフォームにより収録製品及びバージョンが異なるため、製品Webサイト(https://jpn.nec.com/webotx/index.html)もしくはWebOTX Mediaのインストールガイドにて 本製品が収録されていることを確認してください。
(※2)WebOTX V10.3では、マイクロサービスプロファイルとユーザアプリケーションをパッケージ化したマイクロサービスパッケージの作成に別途WebOTX Developerのマイクロサービスビルトツールが必要でした。しかし、WebOTX V10.4では、Uber JARおよびマイクロサービスパッケージの作成にMavenを使用するため、WebOTX Developerは必須ではなくなりました。

1.3. 諸元制限

コンテナ上で動作するWebOTX Application Server Expressはエントリ・モデルのため、フルプロファイルでは以下の諸元制限があります。

- 同時処理数
  クライアントからのリクエストの同時処理数（処理スレッド数）は100本までの制限があります。 この制限は、HTTPセッション数や、利用可能なクライアント数の上限でありません。 ある時点で同時にリクエスト処理を行う上限です。対象の設定値は、「アプリケーションサーバ-スレッドプール」のスレッドプール最大値（max-thread-pool-size）です。

- セッションレプリケーションの共有台数
  負荷分散構成で複数台のサーバでシステムを構成する場合、セッションレプリケーション機能によりセッション情報を共有できます。 このセッションレプリケーションでセッション情報を共有は、4台までの制限があります。
  対象の設定値は、「アプリケーションサーバ-Webコンテナ」のJNDIサーバのURL(session-replication-jndi-url)です。

(注) 一台に複数ドメインを作成した場合には、それぞれのドメインを1台のサーバとみなします。

上記の諸元制限を解除する場合、WebOTX Application Server Express Processor License Unlimited Option for Containerを別途購入し、インストールガイドの「ライセンス登録」を参照してライセンスを登録してください。WebOTX Application Server Express Processor License Unlimited Option for Containerは、WebOTX Application Server Express Processor License for Containerと同一数必要です。

また、マイクロサービスプロファイルには以下の諸元制限があります。

- 同時処理数
  クライアントからのリクエストの同時処理数（処理スレッド数）は100本までの制限があります。 この制限は、HTTPセッション数や、利用可能なクライアント数の上限でありません。 ある時点で同時にリクエスト処理を行う上限です。対象の設定値は、「アプリケーションサーバ-スレッドプール」のスレッ

ドプール最大値（max-thread-pool-size）です。

上記の諸元制限を解除する場合、WebOTX Application Server Express Processor License Unlimited Option for Containerを別途購入し、インストールガイドの「ライセンス登録」を参照してライセンスを登録してください。WebOTX Application Server Express Processor License Unlimited Option for Containerは、WebOTX Application Server Express Processor License for Containerと同一数必要です。

2. 使用上の条件

本章では、WebOTX ASを利用するために必要な条件について説明します。

2.1. ソフトウェア条件について

本製品がサポートするオペレーティング・システム(OS)とハードウェア、および、利用するために必要な関連ソフトウェアについて説明します。

2.1.1. オペレーティング・システム

- Linux

アーキテクチャ	オペレーティング・システム
x64	Red Hat Enterprise Linux 8 Server (8.1以降) (*1) Red Hat Enterprise Linux 7 Server (7.1以降) Oracle Linux 8 (Red Hat Compatible Kernel) (8.1以降) (*1,2,3) Oracle Linux 7 (Red Hat Compatible Kernel) (7.7以降) (*2,3,4)

(*1) WebOTX V10.4ではコンテナのベースイメージとしてもRed Hat Enterprise Linux 8 ServerのUniversal Base ImageおよびOracle Linux 8をサポート
(*2) カーネルはRed Hat Compatible Kernelのみサポート
(*3) Oracle JDK 8/11のみサポート
(*4) Oracle Linux 7はRed Hat Compatible KernelでDockerをサポートしていないため、コンテナホストとしての利用はサポートしません

2.1.2. 必要なソフトウェア

- Docker (Red Hat Enterprise Linux 7 Serverの場合)

コンテナイメージ生成、コンテナ起動にDockerを使用する場合は、ホストマシンにDockerが必要です。サポートするDockerのバージョンは次のとおりです。

・ Docker 1.13以降

- Podman (Red Hat Enterprise Linux 8 ServerおよびOracle Linux 8の場合)

コンテナイメージ生成、コンテナ起動にPodmanを使用する場合は、ホストマシンにPodmanが必要です。サポートするPodmanのバージョンは次のとおりです。

・ Podman 1.6以降

Podmanのルートレスモードでの使用はサポートしていません。

- Java SDK

WebOTXシステムは、実行時にJava™ Platform, Standard EditionのSDKを必要とします。サポートするSDKバージョンは次のとおりです。

- Oracle Java SE Development Kit 8 (Update 202 以降)
- Oracle Java SE Development Kit 11 (11.0.10 以降) LTS版(※1)
- OpenJDK 8 (Update 201 以降) (※2)
- OpenJDK 11 (11.0.10 以降) (※2)

(※1) Java SE Subscription(有償)契約ユーザのみ取得可能
(※2) Red Hat リリース版をサポート

【注意事項】

・WebOTX製品は、Linuxに対応したOracle社製のJava SDKをバンドルしていますが、Java SDK自体の保守は行っていませんので、ご了承ください。

・Red Hat リリースのOpenJDKは、利用バージョンのOpenJDK Development Environmentをインストールしてください。(OpenJDK 8 : java-1.8.0-openjdk-devel , OpenJDK 11 : java-11-openjdk-devel)

・Red Hat リリースのOpenJDK 11が正式対応しているRed Hat Enterprise Linux 7は、7.6以降です。

・Red Hat Enterprise Linux 8でOracle JDK 8を利用する場合はUpdate 221 以降、OpenJDK 8を利用する場合はUpdate 222以降を対象とします。

・Oracle Linux 8でOracle JDK 8を利用する場合はUpdate 221 以降を対象とします。

- Maven (マイクロサービスプロファイルの場合)

マイクロサービスパッケージを作成するために、ホストマシンにMavenが必要です。サポートするMavenのバージョンは次のとおりです。

- Apache Maven 3.6以降

2.1.3. 対応ソフトウェア

- コンテナオーケストレーションツール

評価が完了しているコンテナオーケストレーションツールは次のとおりです。

- OpenShift Container Platform v 4.5
- Kubernetes v 1.18

- Webブラウザ

WebOTX実行環境を管理するためにWebブラウザベースの管理ツールとして、運用管理コンソールを提供しています。サポートするWebブラウザは次のとおりです。 ※フルプロファイルのみサポート

- Microsoft Internet Explorer 11
- Microsoft Edge 81以上
- Firefox 76以上
- Google Chrome 81以上
- Chromium版「Microsoft Edge」88以上

必要とするプラグインはありません。

● Webサーバ

次のWebサーバに対応しています。 ※フルプロファイルのみサポート

・ WebOTX Webサーバ 2.4.46 以降(*1)

(*1)WebOTX WebサーバとはApache HTTP Serverをベースにしたwebサーバで、WebOTX ASにバンドルされています。本製品には64ビットバイナリを提供します。他ベンダーから提供されているApache HTTP Server用の各種プラグイン・モジュールを組み込む場合は、64ビット・モジュールを利用してください。

● データベース (Javaアプリケーション)

WebOTX ASは、JDBC 2.0からJDBC 4.2の仕様に準拠しているJDBCドライバを介して任意のDBMSへの接続をサポートするように設計されています。 アプリケーションが独自の方式でデータベース・サーバに接続、またはWebOTX ASが提供するJDBCデータソースによる接続、あるいは、WebOTXのTransactionサービス機能と連携したJTAトランザクションを使用する場合には、データベース・サーバ製品にバンドルされるJDBCドライバを入手して、セットアップしなければなりません。

WebOTX ASでは以下のJDBCドライバについて動作確認を行っております。

JDBCベンダー	JDBCドライバ・タイプ	サポートするデータベース・サーバ	備考
Oracle	Type 2、4	Oracle Database 11g Release 2 (11.2.0.4)	
		Oracle Database 12c Release 1 (12.1.0.1.0)	
		Oracle Database 12c Release 1 (12.1.0.2)	
		Oracle Database 12c Release 2 (12.2.0.1.0)	
		Oracle Database 18c (18.3.0)	
		Oracle Database 19c (19.3.0.0.0)	
		Oracle Database 19c (19.4.0.0.0)	
		Oracle Database 19c (19.7.0.0.0)	
		Oracle Database 19c (19.9.0.0.0)	
Oracle UCP	Type 2、4	Oracle Database 11g Release 1 以降、Oracle Database 19cまで	
Microsoft	Type 4	Microsoft SQL Server 2014	
		Microsoft SQL Server 2016	
		Microsoft SQL Server 2017	
		Microsoft SQL Server 2019	
DataDirect	Type 4	「Connect for JDBC 3.3 以降」経由によるOracle接続	
PostgreSQL Development Group	Type 4	PostgreSQL 8.1 (JDBCドライバ 8.1 Build 401) ～ PostgreSQL 13.1(JDBCドライバ 42.2.18)	
Apache Derby	Type 4	Apache Derby 10.2.2 ～ 10.11.1.2	
Amazon RDS MariaDB	Type 4	MariaDB 10.0.24(JDBC ドライバ MariaDB connector/J 2.0.2) ～ MariaDB 10.4.13(JDBC ドライバ MariaDB Connector/J 2.7.1)	

Amazon RDS Aurora	Type 4	Aurora(MySQL)-5.6.10a (JDBC ドライバ mysql-connector-java-5.1.42) ～Aurora(MySQL 5.7) 2.09.1 (JDBCドライバ mysql-connector-java-8.0.22)	

その他の製品についても、例えば MySQL Connector/J 5.0など、JDBC 2.0からJDBC 4.2の仕様に準拠しているJDBCドライバであれば、WebOTX ASと連携して使用することができます。ただし、十分な評価を行ってください。

- バッチサービス (フルプロファイルのみサポート)

バッチサービスのジョブリポジトリの対応データベースは以下のとおりです。

JDBCベンダー	JDBCドライバ・タイプ	サポートするデータベース・サーバ	備考
Oracle	Type2、4	Oracle Database 19c (19.9.0.0.0)	
PostgreSQL Development Group	Type 4	PostgreSQL 13.1 (JDBCドライバ 42.2.18)	

3. リソース

本章では、WebOTX ASをインストールするために必要な固定ディスク空き容量と、インストール中、およびインストール後の初期動作で必要なメモリ容量について説明します。

下記に示すメモリ容量は、インストール時に既定値を選択して動作させた場合を表しています。

ハードディスク容量は、選択インストール可能な機能やプロダクトを全てインストールした場合を表しています。ただし、Dockerなどの関連ソフトウェアのディスク消費量は含まれていません。

フルプロファイル

- Linux (x64) コンテナイメージ作成時の消費リソース

リソース	必要条件
メモリ	最小 840 MB、推奨 1 GB 以上
ハード ディスク	1.1 GB 以上

- Linux (x64) コンテナの消費リソース (コンテナ毎)

リソース	必要条件
メモリ	最小 640 MB、推奨 1 GB 以上
ハード ディスク	64 MB 以上

マイクロサービスプロファイル

- Linux (x64) Uber JAR作成時の消費リソース

リソース	必要条件
メモリ	最小 400 MB、推奨 1 GB 以上
ハード ディスク	400 MB 以上

- Linux (x64) コンテナイメージ作成時の消費リソース

リソース	必要条件
メモリ	最小 640 MB、推奨 1 GB 以上
ハード ディスク	700 MB 以上

- Linux (x64) コンテナの消費リソース (コンテナ毎)

リソース	必要条件
メモリ	最小 640 MB、推奨 1 GB 以上
ハード ディスク	64 MB 以上

4. インストール(フルプロファイル)

本章では、コンテナ上のWebOTX Application Server Express (フルプロファイル)のインストール方法について説明します。

4.1. インストール前の作業

インストール前に行う必要のある作業について説明します。

4.1.1. Dockerの環境構築 (Red Hat Enterprise Linux 7 Serverの場合)

Dockerのコンテナ内でWebOTX ASを動作させる場合は、事前にDockerの環境構築を行う必要があります。Dockerの環境構築を行うため、ホストOSで以下の作業を実施してください。

1. ログイン名 root でログインします。

   ```
   login: root
   ```

2. Red Hat Netrowkのextraチャネルを有効化します。 次のコマンドを実行します。

   ```
   root> subscription-manager repos --enable=rhel-7-server-extras-rpms
   ```

3. Dockerをインストールします。 次のコマンドを実行します。

   ```
   root> yum install docker
   ```

4. コンテナイメージを作成する際にRed Hat Networkへのアクセスが発生するため、必要であればDockerに対してプロキシサーバの設定を行います。 まず、systemdが利用するDockerの設定ファイルの配置場所を作成します。

   ```
   root> mkdir /etc/systemd/system/docker.service.d
   ```

   次に、環境変数HTTP_PROXYと環境変数HTTP_PROXY、環境変数NO_PROXYをDockerに対して設定するために、以下の内容で/etc/systemd/system/docker.service.d/http-proxy.confを作成します。

   ```
   [Service]
   Environment="HTTP_PROXY=http://[プロキシサーバ]:[ポート番号]/" "HTTPS_PROXY=ht
   tp://[プロキシサーバ]:[ポート番号]/" "NO_PROXY=[除外対象]"
   ```

   http-proxy.confの具体例を以下に示します。

   ```
   [Service]
   Environment="HTTP_PROXY=http://proxy.example.com:8080/" "HTTPS_PROXY=http
   ://proxy.example.com:8080/" "NO_PROXY=localhost,127.0.0.1"
   ```

   また、rootユーザに対しても同様の環境変数を設定します。

5. Dockerのサービスを有効化します。次のコマンドを実行します。

```
root> systemctl start docker.service
root> systemctl enable docker.service
```

4.1.2. Podmanの環境構築 (Red Hat Enterprise Linux 8 ServerおよびOracle Linux 8の場合)

Podmanのコンテナ内でWebOTX ASを動作させる場合は、事前にPodmanの環境構築を行う必要があります。Podmanの環境構築を行うため、ホストOSで以下の作業を実施してください。

1. ログイン名 root でログインします。

```
login: root
```

2. Podmanを含むコンテナツールをインストールします。 次のコマンドを実行します。

```
root> yum module install container-tools
```

3. コンテナイメージを作成する際にRed Hat Networkへのアクセスが発生します。必要であればpodmanコマンドを実行する際に以下のように環境変数http_proxy, https_proxy, no_proxyでプロキシサーバの設定を行います。

```
export http_proxy=http://[プロキシサーバ]:[ポート番号]/
export https_proxy=http://[プロキシサーバ]:[ポート番号]/
export no_proxy=[除外対象]
```

具体例を以下に示します。

```
export http_proxy=http://proxy.example.com:8080/
export https_proxy=http://proxy.example.com:8080/
export no_proxy=localhost,127.0.0.1
```

4.1.3. unzipコマンドのインストール (移行機能を利用する場合)

移行機能で作成したファイルを展開するためにunzipコマンドを使用します。unzipコマンドをインストールしていない場合、事前にインストールしておく必要があります。

4.2. インストール

フルプロファイルのコンテナイメージの生成について説明します。

1. ログイン名 root でログインします。

```
login: root
```

2. マシンのDVD-ROMドライブにLinux版の「WebOTX Media (DVD) #1」を挿入してマウントします。

```
root> cd /
root> mount -t iso9660 /dev/cdrom /media/cdrom
```

3. コンテナイメージ生成用ファイル準備スクリプトを実行してください。

```
root> /media/cdrom/OTXCNT/LINUX/prepare-to-build-image.sh
```

移行機能を利用してコンテナイメージを生成する場合は、移行機能で作成されたzipファイルをスクリプトの引数に指定してください。移行機能で作成されたzipファイルがネストしている場合は、先にunzipコマンドで展開して、内部のzipファイルを引数に指定してください。

```
root> /media/cdrom/OTXCNT/LINUX/prepare-to-build-image.sh <zipファイル>
```

引数に移行機能で作成されたzipファイルを指定した場合は、はじめにzipファイルを解析して取得した移行先環境情報が表示されます。

```
Parsing <zipファイル> ...
Product Name : WebOTX Application Server Express for Container
User Domain Name : <ドメイン名>
Web Server : <Webサーバの種類>
Domain Administration User Name : <ドメインの運用管理ユーザ名>
Domain Administration Port : <ドメインの運用管理ポート番号>
```

4. コンテナイメージの生成に必要なファイルを準備するディレクトリを入力します。新規に作成するため、存在しないディレクトリを指定してください。

```
Please enter a directory path to use as an output directory.
(Please enter a non-existing path. It will be created after.)
```

5. 生成するイメージのベースイメージを選択します。Red Hat Universal Base Image 7を使用する場合は「1」を、Red Hat Universal Base Image 7 (Minimal)を使用する場合は「2」を、Red Hat Enterprise Linux 7のイメージを使用する場合は「3」を、Red Hat Universal Base Image 8を使用する場合は「4」を、Red Hat Universal Base Image 8 (Minimal)を使用する場合は「5」を、その他のイメージを使用する場合は「6」を選択してください。なお、Red Hat Enterprise Linux 7のイメージは配布に制限があります。また、Red Hat Universal Base Imageを使用する場合も、Red Hat Enterprise Linuxのyumリポジトリを使用すると、配布が制限される場合があります。

```
Please select a base image. (Default: 1)
  1. Red Hat Universal Base Image 7
     (registry.access.redhat.com/ubi7/ubi)
  2. Red Hat Universal Base Image 7 (Minimal)
     (registry.access.redhat.com/ubi7/ubi-minimal)
  3. Red Hat Enterprise Linux 7 Image
     (registry.access.redhat.com/rhel7)
  4. Red Hat Universal Base Image 8
     (registry.access.redhat.com/ubi8/ubi)
  5. Red Hat Universal Base Image 8 (Minimal)
     (registry.access.redhat.com/ubi8/ubi-minimal)
  6. Other image
```

「6」を選択した場合、ベースイメージ名を入力します。

```
Please enter a base image name.
```

続いて、パッケージマネージャのコマンドを入力します。指定したベースイメージで使用可能なコマンドを入力してください。既定値は指定したベースイメージ名に合わせて変わります。

```
Please enter a package manager name. (Default: yum)
```

さらに、パッケージマネージャのコマンドオプションを入力します。指定したパッケージマネージャで使用可能なコマンドオプションを入力してください。既定値は指定したベースイメージ名に合わせて変わります。既定値が「<none>」でない場合、「<none>」と入力することでコマンドオプションなしになります。

```
Please enter package management command options.
(Default: --disableplugin=subscription-manager <省略>)
If you enter "<none>", it means no options.
```

6. JDKを選択します。Red HatのOpenJDK 8を使用する場合は「1」を、Red HatのOpenJDK 11を使用する場合は「2」を、その他のJDKを使用する場合は「3」を選択してください。移行機能を利用してコンテナイメージを生成する場合、既定値は指定したzipファイル内の移行先環境情報に合わせて変わります。

```
Please select a JDK. (Default: 1)
  1. Red Hat OpenJDK 8
  2. Red Hat OpenJDK 11
  3. Other JDK
```

「3」を選択した場合、JDKのインストーラ(tar.gz)ファイルを入力します。既定値はメディアに含まれるインストーラです。

```
Please enter a JDK installer file (*.tar.gz) path.
(Default: /media/cdrom/JDK/LINUX/jdk-8uX-linux-x64.tar.gz)
```

> **Caution**
> 既定値の JDK のインストーラファイル名の X はメディアによって異なります。

7. WebOTX ASのエディションを選択します。本製品の場合は「1」を選択してください。移行機能を利用してコンテナイメージを生成する場合は、指定したzipファイル内の移行先環境情報でエディションが決まるため、この選択肢はスキップされます。

```
Please select one of the following WebOTX products.
  1. WebOTX Application Server Express for Container
  2. WebOTX Application Server Standard for Container
```

8. 製品の「ライセンスキー」を入力します。ライセンスキーは製品に添付される「ソフトウェア使用認定証」の「製品番号」に記載されている19桁の番号です。

```
Please enter a license key of WebOTX Application Server Express
for Container.
```

> **Caution**
> 入力する「ライセンスキー」は1つですが、コンテナに割り当てるコア数に応じた数のWebOTX Application Server Express Processor License for Container を購入していただく必要があります。
> ※ 詳細は「1. はじめに - 1.1 ラインセンスについて」を確認してください。

9. Webサーバを選択します。WebOTX Webサーバを(Apache HTTP Server 2.4.xxベース)を使用する場合は「1」を、WebOTX内蔵型のJavaベースのWebサーバを使用する場合は「2」を選択してください。移行機能を利用してコンテナイメージを生成する場合は、指定したzipファイル内の移行先環境情報でWebサーバが決まるため、この選択肢はスキップされます。

```
Please select one of the following Web Server. (Default: 2)
  1. WebOTX Web Server 2.4
  2. Internal Java based Web Server
```

10. コンテナイメージ生成時にWebOTX AS Expressのパッチを適用する場合、「y」を選択します。パッチを適用しない場合、「n」を選択します。

```
Would you like to apply a patch of WebOTX Application Server Express
during installation? [y,n] (Default: n)
```

「y」を選択した場合、事前に対象マシンにダウンロードしたWebOTX AS Expressのパッチへのパスを入力してください。

```
Please enter a patch file (*.tar.gz) path of WebOTX Application Server
Express.
```

> **Caution**
> パッチの入手にはWebOTXの保守契約が必要です。

11. WebOTX運用管理ユーザを選択します。WebOTX運用管理ユーザをroot以外に設定する場合は「y」を、rootに設定する場合は「n」を選択してください。

```
Would you like to configure a non-root user as WebOTX Operation User? [y,n]
(Default: y)
```

「y」を選択した場合、WebOTX運用管理ユーザはユーザ名asadm (ユーザID 10001)、グループはroot (グループID 0)になります。また、運用管理ユーザがroot以外の場合、OSの制約上ポート番号として1024未満の番号を利用できません。移行機能を利用しないでコンテナイメージを生成する場合は、HTTPのポート番号として8080が、HTTPSのポート番号として8443が使用されます。移行機能を利用する場合、ポート番号に1024未満を使用していないことを確認してください。

12. ユーザドメイン名を入力します。ユーザドメイン名は半角英数字と、ハイフン(-)、アンダーバー(_)を32文字以

内で入力してください。また、「admin」は予約語であるため、ユーザドメイン名として指定することができません。移行機能を利用してコンテナイメージを生成する場合は、指定したzipファイル内の移行先環境情報でドメイン名が決まるため、この入力はスキップされます。

```
Please enter a name of a user domain. (Default: domain1)
```

13. 全ての選択が完了するとコンテナイメージ生成用ファイルの準備開始確認が表示されます。

```
A build context will be created in the specified output directory
with the following settings.
  Output Directory : ./build
  Base Image : Red Hat Universal Base Image 7
               (registry.access.redhat.com/ubi7/ubi)
    package manager : yum (--disableplugin=subscription-manager <省略>)
  JDK : Red Hat OpenJDK 8
  Product Name : WebOTX Application Server Express for Container
  License Key : 1234567890ABCDEF
  Web Server : Internal Java based Web Server
  Apply Patch File : none
  WebOTX Operation User : non-root
  User Domain Name : domain1

********************************************************************
* Preparation to build image in the output directory.              *
* To continue, enter y.                                            *
* Enter q to exit the preparation. [y, q] (Default: y)             *
********************************************************************
```

移行機能を利用してコンテナイメージを生成する場合は、引数に指定したファイル、ドメインの運用管理ユーザ名、運用管理ポートの情報も表示されます。

```
A build context will be created in the specified output directory
with the following settings.
  Output Directory : ./build
  Migration Config : ./webotx_migration_configs_19010100.zip
  Base Image : Red Hat Universal Base Image 7
               (registry.access.redhat.com/ubi7/ubi)
    package manager : yum (--disableplugin=subscription-manager <省略>)
  JDK : Red Hat OpenJDK 8
  Product Name : WebOTX Application Server Express for Container
  License Key : 1234567890ABCDEF
  Web Server : Internal Java based Web Server
  Apply Patch File : none
  WebOTX Operation User : non-root
  User Domain Name : domain1
  Domain Administration User Name : admin
  Domain Administration Port : 6212
```

```
**************************************************************************
* Preparation to build image in the output directory.                    *
* To continue, enter y.                                                  *
* Enter q to exit the preparation. [y, q] (Default: y)                   *
**************************************************************************
```

コンテナイメージ生成用ファイルの準備を開始するには「y」を入力してください。キャンセルするには「q」を入力してください。キャンセルした場合、コンテナイメージ生成用ファイル準備スクリプトは終了します。再実行する場合は手順(3)のコンテナイメージ生成用ファイル準備スクリプトの実行からやり直してください。

14. コンテナイメージ生成用ファイルの準備が実行されます。以下のメッセージが表示されたら準備完了です。

```
Preparation of image build context is successful.
```

15. コンテナイメージ生成用ファイルの準備が完了すると、追加構成の方法とコンテナイメージ生成方法が表示されます。複数ページにわたって表示されるため、ページを進めるにはスペースキーを押下してください。

16. コンテナイメージ生成方法の表示の後、コンテナイメージ生成開始確認が表示されます。

```
**************************************************************************
* Building an image using the output directory as build context.         *
* To continue, enter y.                                                  *
* Enter q to exit the image building. [y, q] (Default: y)                *
**************************************************************************
```

コンテナイメージの生成を開始するには「y」を入力してください。キャンセルするには「q」を入力してください。キャンセルした場合、コンテナイメージ生成用ファイル準備スクリプトは終了します。キャンセル後にコンテナイメージを生成するには以下のコマンドを実行してください。イメージ名を指定する場合はコンテナイメージ名の規約に従ってください。

[Dockerの場合]

```
root> docker build -t <イメージ名> <手順(4)で指定したディレクトリ>
```

※ 必要であればbuildコマンドの--build-argオプションでhttp_proxy、https_proxy、no_proxyにコンテナイメージ生成時に使用するプロキシサーバの情報を指定してください。

> **Caution**
> Red Hat Enterprise Linux 7で提供されるDocker 1.13では、--build-argオプションで指定したプロキシ設定の情報が中間イメージに残る点に注意してください。

[Podmanの場合]

```
root> podman build -t <イメージ名> <手順(4)で指定したディレクトリ>
```

※ 必要であれば環境変数http_proxy、https_proxy、no_proxyにコンテナイメージ生成時に使用するプロキシサーバの情報を指定してpodman buildコマンドを実行してください。

追加構成や追加のライセンス登録を行う場合は、「q」を入力してコンテナイメージの生成をキャンセルして、

手順(4)で指定したディレクトリに生成されたファイルを編集してから、上記のdocker buildコマンドまたはpodman buildコマンドでコンテナイメージを生成してください。

追加の構成をするには、手順(4)で指定したディレクトリに作成されたadditional-configディレクトリの以下のスクリプトを編集してください。

スクリプトファイル名	実行タイミング
pre-start-domain.sh	コンテナイメージ生成中のドメイン起動前。
post-start-domain.sh	コンテナイメージ生成中のドメイン起動後。
post-stop-domain.sh	コンテナイメージ生成中のドメイン停止後。

スクリプト内では、以下の環境変数が利用可能です。

環境変数名	説明
AS_INSTALL	WebOTX ASインストールディレクトリ。 「/opt/WebOTX」固定。
AS_JAVA	JDKのホームディレクトリ。
INSTANCE_ROOT	ユーザドメインのディレクトリ。 「/opt/WebOTX/domains/<ドメイン名>」。
DOMAIN_NAME	ユーザドメイン名。
ADDITIONAL_CONFIG	additional-configディレクトリのコンテナ内でのパス。

スクリプトから使用するファイルは、additional-configディレクトリに配置して、ADDITIONAL_CONFIG環境変数を使用してアクセスしてください。

Caution
各スクリプトの内容およびadditional-configディレクトリの内容が中間イメージに残る点に注意してください。

Caution
追加の構成を行う際、自身のホスト名を指定する場合は「localhost」を使用することで、コンテナ起動時にコンテナのホスト名に自動的に置き換わります。

コンテナイメージ生成時ではなくコンテナ起動時に毎回実行する処理は、手順(4)で指定したディレクトリに作成されたscripts/preprocess.dディレクトリに実行可能ファイルを配置してください。ここに配置した実行可能ファイルはコンテナ内でのドメイン起動前に実行されます。この実行可能ファイルでは、上記の環境変数の内、ADDITIONAL_CONFIG環境変数を除いた環境変数が利用可能です。

追加のライセンスを登録するには、手順(4)で指定したディレクトリに作成されたテキストファイルadditional-license.txtにライセンスキーを記入してください。ライセンスキーは1行に1つずつ記入してください。

手順(17)～手順(20)はコンテナイメージ生成開始確認で「y」を入力した場合の手順です。

17. コンテナイメージ生成前にイメージ生成環境が準備できているか確認します。

```
Validating environment for building image...
```

イメージ生成環境の確認を個別に実行するには、以下のスクリプトを実行してください。

```
root> /media/cdrom/OTXCNT/LINUX/validate-image-build-environment.sh
```

イメージ生成環境が正しく準備できている場合は、以下のメッセージが表示されます。

```
Validation is successful.
```

イメージ生成環境が正しく準備できていない場合は、以下のメッセージが表示されスクリプトが終了します。一緒に表示されるエラーメッセージの内容を確認して、コンテナエンジン等のイメージ生成環境を構築した後に、手順(16)に記載したdocker buildコマンドまたはpodman buildコマンドでイメージを生成してください。

```
Validation failed.
```

18. 生成するコンテナイメージの名前を入力します。イメージ名はコンテナイメージ名の規約に従ってください。

```
Please enter a name of the build image. (Default: no name)
```

19. Dockerデーモンにプロキシ設定をしている場合、コンテナイメージ生成時にも同じプロキシ設定を使用するか選択します。Dockerデーモンと同じプロキシ設定を使用する場合は「y」を、プロキシ設定を使用しない場合は「n」を選択してください。Dockerデーモンにプロキシ設定をしていない場合は、この選択肢はスキップされます。「y」を入力した場合、プロキシ設定はdocker buildコマンドの--build-argオプションを使用して指定されます。

```
Would you like to use the same proxy configuration as the Docker daemon when
building the image? [y/n] (Default: y)
  Http Proxy: <httpプロキシ>
  Https Proxy: <httpsプロキシ>
  No Proxy: <プロキシ除外設定>
```

上記の「<httpプロキシ>」「<httpsプロキシ>」「<プロキシ除外設定>」はDockerデーモンに設定したプロキシ設定が表示されます。設定されていない項目は「<none>」と表示されます。

Podmanの場合は、スクリプトを実行した環境の環境変数http_proxy, https_proxy, no_proxyが引き継がれるため、この選択肢はスキップされます。

20. コンテナイメージの生成が実行されます。以下のメッセージが表示されたら生成完了です。コンテナイメージ生成用ファイル準備スクリプトはここで終了します。

```
Building an image is successful.
```

21. コンテナイメージが生成されていることを確認します。 以下のコマンドを実行します。
(Podmanを使用する場合は、dockerコマンドをpodmanコマンドに読み替えてください)

```
root> docker images
REPOSITORY                                TAG      IMAGE ID       CREATED
<イメージ名>                              <タグ>   ************   ** **** ago
registry.access.redhat.com/ubi7/ubi       latest   ************   ** **** ago
```

(*) *** 部分の表記は環境によって異なります。

生成時に指定した名前のイメージが表示されれば、生成完了です。

これでインストール作業は完了です。
この作業が完了したら「4.3 インストール後の作業」に進んでください。

4.3. インストール後の作業

4.3.1. データベースを使用するための準備作業 (Java)

Javaアプリケーションでデータベースを使用する場合に、各データベースで次の準備作業を行ってください。詳細については、各データベースのリファレンスマニュアルでご確認ください。

- Oracle での作業

  - トランザクションのリカバリを行うためには、DBA_PENDING_TRANSACTIONS ビューの SELECT 権限が必要です。 JDBC リソースを登録する際に、SELECT 権限を持つユーザを設定してください。 JDBC リソースの登録を省略する場合には、トランザクション実行時に使用するJDBC データソースの定義で 指定した全ユーザに対して、SELECT 権限を付与してください。

  - Oracle Call Interface (OCI)のJDBCドライバを使用する場合は、Oracleデータベースの環境変数を設定する必要があります。詳細は、WebOTXオンラインマニュアルの「注意制限事項 > 機能ごとの注意制限事項 >JDBCデータソース」を参考にしてください。
    (※) JTAやTransactionサービスによるトランザクション制御を行わない場合も必要な作業です。

  JDBC データソースの設定で、データベースクラスタの使用有無[useDatabaseCluster]にtrue を設定した場合、 または、次のバージョン以降のOracleデータベースを使用する場合、ユーザアカウントに sys.dbms_system パッケージへの EXECUTE権限を付与してください。

  Oracle Database 11g Release 2 (11.2.0.4)

- Microsoft SQL Server での作業

  - SQL Serverを使用するためには、SqlJDBCXAUser ロールの権限が必要です。トランザクション実行時に使用するJDBC データソースの定義で指定した全ユーザに対して、 SqlJDBCXAUser ロールを付与してください。

  - 未完了のトランザクションが存在する状態でMicrosoft SQL Server を再起動すると、Transaction サービスから データベースへの接続ができず、未完了トランザクションのリカバリを行うことができません。 あらかじめ、Transaction サービスから接続するデータベースと、アプリケーションから接続する

データベースを 分けるようにしてください。例えば、Transaction サービスでリカバリを行う際に使用するデータベースを master とし、 アプリケーションが使用するデータベースを pubs としてください。

- 各JDBC ドライバの分散トランザクション制御用のプログラムをインストールしてください。SQL SQL Server JDBC Driver 3.0 / SQL Server JDBC Driver 4.0 は、SQL Server 2014 に接続することができます。SQL Server JDBC Driver 4.2 では、SQL Server 2016/ SQL Server 2017/ SQL Server 2019に接続することができます。SQL Server JDBC Driver 7.4では、SQL Server 2019に接続することができます。

4.4. 動作確認

WebOTXの動作確認は次のとおりの手順で行ないます。
(Podmanを使用する場合は、dockerコマンドをpodmanコマンドに読み替えてください)

1. ホストOSにログイン名 root でログインします。

```
login: root
```

2. コンテナを起動します。以下のコマンドを実行します。

```
root> docker run -i -t <イメージ名>
```

3. 起動したコンテナのIPアドレスを確認します。まずは以下のコマンドを実行して、起動したコンテナのコンテナIDを取得します。

```
root> docker ps
CONTAINER ID  IMAGE                 COMMAND                  CREATED
537d3c548628  <イメージ名>          "/opt/WebOTX/bin/doc  2 minutes ago
```

上記の例では、「537d3c548628」がコンテナIDです。次に、以下のコマンドを実行して、コンテナのIPアドレスを確認します。

```
root> docker inspect 537d3c548628 | grep IPAddress
      "IPAddress": "192.168.1.2",
```

上記の例では、IPアドレスは「192.168.1.2」です。IPアドレスは、Dockerがコンテナに対して自動的に割り当てるため、環境によって異なります。

4. 運用管理コンソールを利用して、コンテナで動作しているWebOTXドメインへ接続確認します。ホストOS上でサポート対象のWebブラウザ(2.1.3節に記載)を起動し、次のURLを入力してください。

```
http://[IPアドレス]:5858/
```

Webブラウザ上でログイン画面が表示されたら、ユーザ名に「admin」を、パスワードに「adminadmin」を入力して、「ログイン」ボタンをクリックします。ログインが成功しようこそ画面が表示されることを確認します。画面右上の「ログアウト」ボタンをクリックすることでログアウトできます。

以上が確認できれば、正しくインストールされています。

5. インストール(マイクロサービスプロファイル)

本章では、コンテナ上でのWebOTX Application Server Express (マイクロサービスプロファイル)のインストール方法について説明します。

インストールではWebOTX Uber JAR生成ツールを利用します。WebOTX Uber JAR生成ツールはMavenプロジェクトで作成するために必要な、WebOTXマイクロサービスMavenプラグインとWebOTXマイクロサービスMavenアーキタイプを提供します。

5.1. インストール前の作業

インストール前に行う必要のある作業について説明します。

5.1.1. Dockerの環境構築 (Red Hat Enterprise Linux 7 Serverの場合)

「4.1.1 Dockerの環境構築 (Red Hat Enterprise Linux 7 Serverの場合)」を参照してください。

5.1.2. Podmanの環境構築 (Red Hat Enterprise Linux 8 ServerおよびOracle Linux 8の場合)

「4.1.2 Podmanの環境構築 (Red Hat Enterprise Linux 8 ServerおよびOracle Linux 8の場合)」を参照してください。

5.1.3. Mavenのインストール

マイクロサービスパッケージを作成するためにMavenを使用します。Mavenをインストールしていない場合、事前にインストールしておく必要があります。
必要に応じて設定ファイル(settings.xml)にプロキシ、ローカルリポジトリの設定を行ってください。

5.1.4. Java SDKのインストール

Mavenプロジェクトを作成するためにJavaを使用します。Javaをインストールしていない場合、事前にインストールしておく必要があります。
サポートするバージョンについては「2.1.2 必要なソフトウェア」のJava SDKを参照してください。

5.2. インストール

マイクロサービスプロファイルのコンテナイメージの生成について説明します。
(Podmanを使用する場合は、dockerコマンドをpodmanコマンドに読み替えてください)

1. ログイン名 root でログインします。

```
login: root
```

- インストールメディアを利用する場合
  「2 マシンのDVD-ROMドライブにLinux版の「WebOTX Media (DVD) #1」を挿入してマウントします。」に進んでください。

- Maven Central Repositoryを利用する場合
  「5 Mavenプロジェクトを作成します。」に進んでください。

2. マシンのDVD-ROMドライブにLinux版の「WebOTX Media (DVD) #1」を挿入してマウントします。

```
root> cd /
root> mount -t iso9660 /dev/cdrom /media/cdrom
```

3. WebOTXマイクロサービスランタイム、WebOTXマイクロサービスMavenプラグイン、WebOTXマイクロサービスMavenアーキタイプを展開します。

```
root> cp /media/cdrom/OTXMSP/LINUX/OTXMSP-10.40.00.00.tar.gz  ＜任意のディレクトリ＞
root> cd ＜OTXMSP-10.40.00.00.tar.gzをコピーした任意のディレクトリ＞
root> tar -xzvf OTXMSP-10.40.00.00.tar.gz
```

以下のように展開されていることを確認します。

```
UberJAR/webotx-micro-10.4.jar
MavenPlugin/ ms-uberjar-maven-plugin-10.4.jar
MavenPlugin/ ms-uberjar-maven-plugin-10.4.pom
MavenArchetype/ ms-uberjar-archetype-10.4.jar
```

4. 展開されたファイルをローカルリポジトリにインストールします。

```
root> cd ＜OTXMSP-10.40.00.00.tar.gzを展開した任意のディレクトリ＞
```

▪ WebOTXマイクロサービスランタイムをローカルリポジトリへインストール

```
root> mvn install:install-file -Dfile=./UberJAR/webotx-micro-10.4.jar
-DgroupId=com.nec.webotx -DartifactId=webotx-micro -Dversion=10.4
-Dpackaging=jar -DgeneratePom=true
```

▪ WebOTXマイクロサービスMavenプラグインをローカルリポジトリへインストール

```
root> mvn install:install-file
-Dfile=./MavenPlugin/ms-uberjar-maven-plugin-10.4.jar
-DgroupId=com.nec.webotx -DartifactId=ms-uberjar-maven-plugin
-Dversion=10.4 -Dpackaging=jar
-DpomFile=./MavenPlugin/ms-uberjar-maven-plugin-10.4.pom
```

▪ WebOTXマイクロサービスMavenアーキタイプをローカルリポジトリへインストール

```
root> mvn install:install-file
-Dfile=./MavenArchetype/ms-uberjar-archetype-10.4.jar
-DgroupId=com.nec.webotx -DartifactId=ms-uberjar-archetype -Dversion=10.4
-Dpackaging=jar -DgeneratePom=true
```

5. Mavenプロジェクトを作成します。

```
root> cd ＜任意のMavenプロジェクト作成ディレクトリ＞
root> mvn archetype:generate -DarchetypeGroupId=com.nec.webotx
```

-DarchetypeArtifactId=ms-uberjar-archetype -DarchetypeVersion=10.4 -DgroupId=＜任意のgroupId＞ -DartifactId=＜任意のartifactId＞ -Dversion=＜任意のversion＞

※途中、プロパティの構成の確認が表示されます。「y」を入力し、エンターキーを押下します。

```
Confirm properties configuration:
groupId: ＜任意のgroupId＞で指定したgroupId
artifactId: ＜任意のartifactId＞で指定したartifactId
version: ＜任意のversion＞で指定したversion
package: ＜任意のgroupId＞で指定したgroupId
Y: :y
```

6. Mavenプロジェクトのビルドを行い、マイクロサービスパッケージを作成します。
※詳細な設定を行う場合はビルド前に、オンラインマニュアルの[Application Server > アプリケーション開発 > その他のアプリケーション > 15. マイクロサービスプロファイルを使用したWebOTX Uber JARの作成 > 15.5 pom.xmlの設定]を参照して下さい。

```
root> cd ＜Mavenプロジェクト作成時に指定したartifactId＞
root> mvn install
```

ビルド後、＜Mavenプロジェクト作成時に指定したartifactId＞/target/webotx-ms-package に移動し、lsコマンドでファイル一覧を表示し、以下の構成になっていることを確認します。
(ファイルのタイムスタンプとファイルサイズには変更があります。)

```
-rw-r--r--     1  root  root         980  Feb 12 10:59  Dockerfile
-rw-r--r--     1  root  root    87512398  Feb 12 10:58  <artifactId>-<version>-micro.jar
```

7. Dockerのコンテナイメージを作成します。以下のコマンドを実行します。

```
root> cd ＜Mavenプロジェクト作成時に指定したartifactId＞/target/webotx-ms-package
root> docker build -t webotx-micro .
```

8. Dockerのコンテナイメージが作成されていることを確認します。 以下のコマンドを実行します。

```
root> docker images
REPOSITORY                              TAG      IMAGE ID       CREATED
webotx-micro                            latest   ************   ** **** ago
registry.access.redhat.com/ubi8/ubi     latest   ************   ** **** ago
```

(*) *** 部分の表記は環境によって異なります。

REPOSITORYの列がwebotx-microで、TAGの列がlatestであるイメージがマイクロサービスプロファイルのコンテナイメージです。

これでインストール作業は完了です。
この作業が完了したら「5.3 インストール後の作業」に進んでください。

5.3. インストール後の作業

「4.3 インストール後の作業」を参照してください。

5.4. 動作確認

マイクロサービスプロファイルの動作確認は次のとおりの手順で行ないます。
(Podmanを使用する場合は、dockerコマンドをpodmanコマンドに読み替えてください)

1. ホストOSにログイン名 root でログインします。

```
login: root
```

2. コンテナを起動します。コンテナの環境変数OTX_LICENSEで製品の「ライセンスキー」を指定します。ライセンスキーは製品に添付される「ソフトウェア使用認定証」の「製品番号」に記載されている19桁の番号です。また、コンテナ起動時にホスト8080番ポートへの通信をコンテナ8080番ポート(既定値)へ転送します。以下のコマンドを実行します。

```
root> docker run -it -d -e OTX_LICENSE=<ライセンスキー> -p 8080:8080 webotx-micro
```

> **Caution**
> 指定する「ライセンスキー」は1つですが、コンテナに割り当てるコア数に応じた数のWebOTX Application Server Express Processor License for Container を購入していただく必要があります。
> ※ 詳細は「1. はじめに - 1.1 ラインセンスについて」を確認してください。

> **Caution**
> 「8080番ポート」はホスト上で使用していないポート番号を指定してください。
> ポート番号が競合していないことを確認するには、以下のコマンドで確認できます。
> `root> ss -ant | grep 8080`

3. 以下のコマンドを実行して、コンテナが起動していることを確認します。

```
root> docker ps
CONTAINER ID  IMAGE           COMMAND                  CREATED
537d3c548628  webotx-micro    "java -jar               2 minutes ago
                              <artifactId>-<version>-micr
                              o.jar --basedir /opt"
```

4. サンプルアプリを利用して、コンテナで動作しているWebOTX Uber JARへ接続確認します。Webブラウザを起動し、次のURLを入力してください。

```
http://[ホストOSのIPアドレス]
:8080/{artifactId}-{version}/parameter/query?param=query
```

Webブラウザ上に「query」の表示が確認できれば、正しくインストールされています。

6. アンインストール

本章では、コンテナ上のWebOTX Application Server Expressのアンインストール方法について説明します。

6.1. 不要なコンテナの削除

コンテナにインストールしたWebOTX ASについては、アンインストールはサポートしません。不要になったコンテナは、以下の手順でホストOS上から削除してください。
(Podmanを使用する場合は、dockerコマンドをpodmanコマンドに読み替えてください)

1. ホストOSにログイン名 root でログインします。

   ```
   login: root
   ```

2. 不要になったコンテナが起動している場合は、ホストOS上で以下のコマンドを実行して停止してください。

   ```
   root> docker kill [コンテナID]
   ```

3. 不要になったコンテナは、ホストOS上で以下のコマンドを実行して削除してください。

   ```
   root> docker rm [コンテナID]
   ```

4. また、不要になったコンテナイメージは、ホストOS上で以下のコマンドを実行して削除してください。

   ```
   root> docker rmi [イメージID]
   ```

7. 注意・制限事項

7.1. フルプロファイル

- 追加構成でadditional-configディレクトリに配置したファイルは中間イメージに残ります。パスワード等の機密情報を含める場合は、生成されたコンテナイメージを公開しないように注意してください。
- Dockerのバージョンが17.05未満の場合、イメージ生成時に--build-argオプションで指定したプロキシ設定が中間イメージに残ります。コンテナイメージ生成時にプロキシ設定をした場合は、コンテナイメージを公開する際に注意してください。
- コンテナは起動する際にホスト名が割り当てられます。WebOTX ASを構成する際に自身のホスト名を指定する場合は、自身のホスト名として「localhost」を使用してください。これにより、コンテナ内でのドメイン起動時に「localhost」がコンテナのホスト名に自動的に変更されます。ホスト名の自動変更の対象となる設定項目は、オンラインマニュアルの[Application Server > 構築・運用 > 環境変数・JDK・ホスト名の設定変更 > 4. ホスト名の変更]の「Caution」で自動的に設定されると説明がある項目です。それ以外の項目については、scripts/preprocess.dディレクトリに実行可能ファイルを配置して、ドメイン起動前に設定ファイルを変更してください。

その他の注意・制限事項については、オンラインマニュアルを参照して下さい。

7.2. マイクロサービスプロファイル

注意・制限事項については、オンラインマニュアルを参照して下さい。