

NEC

WebOTX Operator for Kubernetes V10.4

UL4021-10AG

インストールガイド(Linux)

ごあいさつ

このたびは、WebOTX Operator for Kubernetes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 Operator for Kubernetes のインストール方法や利用上の注意等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。

WebOTX は、日本電気株式会社の登録商標です。

UNIX は、The Open Group の米国ならびに他の国における登録商標です。

Oracle と Java は、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。

Intel は、アメリカ合衆国および/またはその他の国における Intel Corporation の商標です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標もしくは商標です。

Red Hat は、米国およびその他の国における Red Hat, Inc.の商標または登録商標です。

Amazon Web Services、“Powered by Amazon Web Services”ロゴ、およびかかる資料で使用されるその他の AWS 商標は、米国その他の諸国における、Amazon.com, Inc.またはその関連会社の商標です。

Kubernetes、Prometheus は、The Linux Foundation の米国およびその他の国における商標または登録商標です。

Elasticsearch、Logstash は米国および他の国々で登録された Elasticsearch B.V.の商標です。

Docker and Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and/or other countries. Docker, Inc. and other parties may also have trademark rights in other terms used herein.

その他、記載されている会社名、製品名は、各社の登録商標または商標です。

目次

1. はじめに	1
1.1. ライセンスについて	1
1.2. 構成品について	1
2. 使用上の条件	2
2.1. ソフトウェア条件について	2
2.1.1. オペレーティング・システム	2
2.1.2. 必要なソフトウェア	2
2.1.3. 対応ソフトウェア	2
3. リソース	4
4. インストール	5
4.1. インストール前の作業	5
4.1.1. コンテナイメージビルド環境	5
4.1.2. Kubernetes クライアントのインストール	6
4.2. インストール	6
4.3. インストール後の作業	10
4.4. 動作確認	12
5. アンインストール	13
5.1. アンインストール作業	13
6. 注意・制限事項	15

1. はじめに

1.1. ライセンスについて

WebOTX Operator for Kubernetes は コンテナオーケストレーションツールであるKubernetes上にWebOTX Application Serverのコンテナイメージを配備・稼働させるための自動化製品です。

この製品のライセンスはKubernetesのクラスタ単位につき1ライセンスが必要となります。KubernetesのクラスタはMasterとNodeにより構成されるシステム構成です。

- Master
Kubernetesの全体の管理を行う仮想マシン・物理マシンです。コンテナイメージのスケーリングやロールアウトなど、クラスタ内の全ての動作を管理します。
- Node
Kubernetesでコンテナイメージが実際に動作する仮想マシン・物理マシンです。

WebOTX Operator for KubernetesはコンテナイメージとしてNode上で動作します。同じKubernetesのクラスタ内であれば、一つのライセンスで複数のWebOTX Operator for Kubernetesのコンテナイメージを動作させることが可能です。

詳細な構成についてはWebOTXオンラインマニュアルの「入門ガイド > 製品構成 > WebOTX Operator for Kubernetes」を参照してください。

1.2. 構成品について

本製品にインストール用のDVD-ROM媒体は含まれていません。製品全体の構成品に関しては構成品表を確認してください。

2. 使用上の条件

本章では、WebOTX Operator for Kubernetes 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

WebOTX Operator for Kubernetesのコンテナイメージをビルドする環境として以下をサポートします。

Linux

ハードウェア	オペレーティング・システム
Intel 64	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)

(*1) Docker でのビルドのみサポート(Podman はサポート対象外)

(*2) カーネルはRed Hat Compatible Kernelのみサポート

2.1.2. 必要なソフトウェア

Docker

WebOTX Operator for Kubernetes は、Docker のコンテナとして Kubernetes 上で動作するため、Docker がインストールされていることが必須となります。Dockerのインストールは以下、2環境に対して実施する必要があります。

- WebOTX Operator for Kubernetesのコンテナイメージビルド環境
- Kubernetesのクラスタ環境

WebOTX Operator for Kubernetes では、以下の Docker の動作確認を行っております。

ソフトウェア	動作確認バージョン
Docker	1.13 以降

Kubernetes

WebOTX Operator for Kubernetes は、Kubernetes 及びOpenShift上で動作します。動作確認を行っている Kubernetes のバージョンは以下のとおりです。

ソフトウェア	動作確認バージョン
Kubernetes	1.14 ~ 1.18
OpenShift	4.2 , 4.4, 4.5

2.1.3. 対応ソフトウェア

Elasticsearch

WebOTX Operator for Kubernetes で展開する WebOTX AS コンテナの Pod には、Logstash コンテナが含まれており、WebOTX AS コンテナで出力されたログを Logstash コンテナにて収集します。収集したログは外部で動作する Elasticsearch へログを送信することができます。WebOTX Operator for Kubernetes では、以下の Elasticsearch の動作確認を行っております。

ソフトウェア	動作確認バージョン
Elasticsearch	7.1.1 以降

Prometheus

WebOTX Operator for Kubernetes は、Prometheus と連携して WebOTX Operator for Kubernetes が動作しているコンテナのメトリクス情報を Prometheus で収集することができます。

WebOTX Operator for Kubernetes では、以下の Prometheus の動作確認を行っております。

ソフトウェア	動作確認バージョン
Prometheus	2.9.2 以降

3. リソース

本章では、WebOTX Operator for Kubernetes 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に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。

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 512 MB、推奨 1 GB 以上
ハード ディスク	256 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 64 MB、推奨 256 MB 以上
ハード ディスク	256 MB 以上

4. インストール

本章ではKubernetesのクラスタ環境が構築されていることを前提とした、WebOTX Operator for Kubernetesのインストール方法について説明します。

Kuberentesを前提とした方法を記載しているため、OpenShiftを使用する場合は、WebOTXオンラインマニュアルの「構築・運用 > コンテナ型仮想化 > 2, Operator > 2.5. OpenShiftへの展開」を確認してください。

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. コンテナイメージビルド環境

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チャン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]/" "NO_PROXY=[除外対象], /var/run/docker.sock"
```

http-proxy.confの具体例を以下に示します。

```
[Service]
Environment="HTTP_PROXY=http://proxy.example.com:8080/"
"NO_PROXY=localhost,127.0.0.0/8,::1,/var/run/docker.sock"
```

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. Kubernetes クライアントのインストール

1. Kubernetes クライアントをインストールします。

WebOTX Operator for Kubernetes は、Kubernetes 上で動作するため、Kubernetes に展開するための Kubernetes クライアントが必要となります。

以下のページを参考にして、Kubernetes のクライアントのインストールを実施してください。

<https://kubernetes.io/docs/tasks/tools/install-kubectl/>

Caution

インストールする Kubernetes のクライアントは Kubernetes サーバーのバージョンと合わせるようにしてください。

2. `$HOME/.kube` ディレクトリを作成します。

```
root> mkdir $HOME/.kube
```

3. Kubernetes クライアントの認証設定をします。

Kubernetes の Master Node で使用している以下の `admin.conf` をクライアントマシンにコピーして `$HOME/.kube` に配置します。

```
root> cp -p /etc/kubernetes/admin.conf $HOME/.kube/config
```

Caution

`$HOME/.kube/config` はディレクトリではなくファイルです。

4.2. インストール

Docker用のコンテナイメージの作成について説明します。

4. ログイン名 `root` でログインします。

```
login: root
```

5. マシンのDVD-ROMドライブに「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
```

6. WebOTX Operator for Kubernetes のコンテナイメージを作成するための作業ディレクトリを作成します。

以下のコマンドを実行します。以降の説明では、このディレクトリを<作業ディレクトリ>として記載します。

```
root> mkdir <作業ディレクトリ>
```

7. WebOTX Operator for Kubernetes のインストーラを実行します。

```
root> cd /media/cdrom/OTXOPE/LINUX
root> ./install.sh
```

インストーラを実行すると、以下のメッセージが出力されるので、インストールディレクトリに作業ディレクトリを入力してください。入力を省略した場合は、/opt/WebOTX/operator にインストールされます。

```
Please input installation base directory of WebOTX Operator with an absolute
path. (Default: /opt/WebOTX/operator)
(WebOTX Operator is installed in /opt/WebOTX/operator by default.)
```

インストール先を入力すると、以下のメッセージが表示されるので、y を入力するか、何も入力せずに Enter キーを入力します。

```
*****
* Installation of WebOTX Operator 10.4 on your computer.                *
* To continue, enter y. Enter q to exit. [y, q] (Default: y)           *
*****
```

インストールを実行すると、以下のメッセージが出力されます。最後に Installation completed. と表示されれば、インストールは完了です。

```
Start the installation.

checking docker cli... found
checking docker service... found
checking docker service start on boot... enabled
:
(省略)
:
manifest/webotx_crd.yaml

Copy of template files.

Copy of OTXUTIL installation file.

Installation completed.
```

8. WebOTX Operator イメージを作成します。

インストーラを実行すると、WebOTX Operator for Kubernetes のコンテナイメージ作成に必要なファイルが指定したインストールディレクトリに展開されます。展開されたインストールディレクトリに移動して、以下のコマンドを実行してから、コンテナのイメージを作成してください。

```
root> cd <作業ディレクトリ>
root> docker build -t <Dockerレジストリのホスト名>:<ポート>/<任意のリポジトリ名>/webotx-operator:10.4 <作業ディレクトリ>/docker
```

実行例:

```
docker build -t 192.168.0.100:5000/webotx/webotx-operator:10.4 ¥
/opt/WebOTX/operator/docker
```

イメージ作成では、Dockerfile 内で microdnf により Operator で必要なアプリケーションをインストールします。このため、プロキシ内で作業を行っている場合は、次のように `--build-arg` でプロキシを環境変数に設定してください。`--build-arg` で設定した環境変数は `docker build` 時のみ使用されます。以下の例では、プロキシのサーバー名を `proxyserver`、プロキシのポート番号を `8080` としています。

実行例:

```
docker build -t 192.168.0.100:5000/webotx/webotx-operator:10.4 ¥
--build-arg http_proxy=http://proxyserver:8080/ ¥
--build-arg https_proxy=http://proxyserver:8080/ ¥
/opt/WebOTX/operator/docker
```

9. WebOTX Operator for Kubernetes のコンテナイメージを Docker レジストリに登録します。

```
root> docker push ¥
<Dockerレジストリのホスト名>:<ポート>/<任意のリポジトリ名>/webotx-operator:10.4
```

実行例:

```
docker push 192.168.0.100:5000/webotx/webotx-operator:10.4
```

10. Namespace を作成します。

必要に応じて WebOTX Operator for Kubernetes が動作する Namespace を作成してください。Namespace を作成しない場合は、default に WebOTX Operator for Kubernetes が展開されます。

```
root> kubectl create namespace <Namespace名>
```

11. カスタムリソース定義を適用します。

```
root> cd <作業ディレクトリ>
root> kubectl apply -f manifest/webotx_crd.yaml
```

Caution

カスタムリソース定義は、全ての Namespace に適用されます。
このため、Kubernetes への登録は 1 回のみ適用します。

12. カスタムリソース定義が登録されていることを確認します。

```
root> kubectl get crd
```

13. サービスアカウントと権限の作成をします。

以下のコマンドにてサービスアカウントを登録してください。Namespace名には 10. で作成した Namespace を指定してください。

```
root> kubectl apply -f manifest/service_account.yaml -n <Namespace名>
```

14. サービスアカウントが登録されていることを確認します。

以下のコマンドにてサービスアカウントが登録されていることを確認してください。出力結果に `webotx-operator` が作成されていれば正常にサービスアカウントが登録されています。

```
root> kubectl get serviceaccount -n <Namespace名>
NAME                SECRETS  AGE
default             1        7m6s
webotx-operator     1        90s
```

15. 権限が登録されていることを確認します。

以下のコマンドにて `role` と `rolebinding` の設定を取得して、権限が登録されていることを確認してください。NAME に `webotx-operator` が出力されていれば権限は正常に登録できています。

```
root> kubectl get role -n <Namespace名>
NAME                AGE
webotx-operator     75s

root> kubectl get rolebinding -n <Namespace名>
NAME                AGE
webotx-operator     81s
```

16. ConfigMap を登録します。

```
root> kubectl apply -f manifest/configmap.yaml -n <Namespace名>
```

17. ConfigMap が登録されていることを確認します。

ConfigMap の設定を以下のコマンドにて確認してください。結果で `webotx-as-logstash-config`、`webotx-as-logstash-pattern`、`webotx-as-scripts` が出力されれば、正常に ConfigMap が登録できています。

```
root> kubectl get configmap -n <Namespace名>
```

NAME	DATA	AGE
webotx-as-logstash-config	3	11s
webotx-as-logstash-pattern	1	11s
webotx-as-scripts	3	11s

18. WebOTX Operator for Kubernetes のイメージ名とライセンスを設定します。

```
root> vi manifest/operator.yaml
```

以下の箇所をイメージ名に変更します。

```
image: REPLACE_WEBOTX_OPERATOR_IMAGE
```

設定例:

```
image: 192.168.0.100:5000/webotx/webotx-operator:10.4
```

以下の箇所に WebOTX Operator for Kubernetes ライセンスを設定します。

```
value: REPLACE_LICENSE_KEY
```

19. WebOTX Operator for Kubernetes を展開します。

```
root> kubectl apply -f manifest/operator.yaml -n <Namespace名>
```

20. WebOTX Operator for Kubernetes が展開されたのを確認します。

以下のコマンドにて WebOTX Operator for Kubernetes が展開されたかどうかを確認してください。結果で STATUS が Running になれば正常に展開されています。

```
root> kubectl get pod -n <Namespace名>
```

NAME	READY	STATUS	RESTARTS	AGE
webotx-operator-684c7976cc-vxw4j	1/1	Running	0	14s

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

以下の作業を行う前に予め WebOTX AS コンテナイメージを作成し、レジストリに登録しておく必要があります。WebOTX AS コンテナイメージの作成方法については、WebOTX AS Express for Container または WebOTX AS Standard for Container のインストールガイドを参照してください。

1. ホストOSにログイン名 root でログインします。

```
login: root
```

2. 作成した WebOTX AS Express for Container または WebOTX AS Standard for Container イメージのタグ付けを行います。

以下の `docker tag` コマンドでタグ付けを行います。タグ付けでは、イメージを登録する Docker レジストリのホスト名とポート番号を指定してください。以下の例では、`webotx/webotx-for-container:10.4` というイメージ名で Docker レジストリにイメージの登録が行われます。

```
root> docker tag <イメージ名> ¥
<Dockerレジストリのホスト名>:<ポート>/webotx/webotx-for-container:10.4
```

3. WebOTX AS Express for Container または WebOTX AS Standard for Container イメージを Docker レジストリに登録します。

```
root> docker push ¥
<Dockerレジストリのホスト名>:<ポート>/webotx/webotx-for-container:10.4
```

4. カスタムリソースを変更します。

```
root> cd <作業ディレクトリ>
root> vi manifest/webotx_cr.yaml
```

以下の箇所に任意の名前を設定します。

```
name: REPLACE_APPLICATIONSERVER_NAME
```

設定例:

```
name: webotx-for-container
```

以下の箇所に WebOTX AS コンテナのイメージ名を設定します。

```
image: REPLACE_WEBOTX_AS_IMAGE
```

設定例:

```
image: 192.168.1.100:5000/webotx/webotx-for-container:10.4
```

5. WebOTX AS コンテナを展開します。

```
root> kubectl apply -f manifest/webotx_cr.yaml -n <Namespace名>
```

6. WebOTX AS コンテナが展開されたことを確認します。

以下のコマンドにて、展開した WebOTX AS コンテナの READY が 2/2 となり、STATUS が Running となれば、正常に起動しています。

```
root> kubectl get pod -n <Namespace名>
```

NAME	READY	STATUS	RESTARTS	AGE
webotx-for-container-57747b796-h797v	2/2	Running	0	51s
webotx-operator-684c7976cc-vxw4j	1/1	Running	0	30m

4.4. 動作確認

WebOTX AS コンテナが展開されると、カスタムリソース(`webotx_cr.yaml`)で設定した Service が作成されます。デフォルトのテンプレートファイルの場合、NodePort Service で 30080 ポートを WebOTX AS コンテナの 8080 ポートに転送し、30443 ポートを WebOTX AS コンテナの 8443 ポートに転送します。NodePort Service は、各ノードの IP アドレスで受け付けたリクエストを他のノードにロードバランシングするサービスです。

このため、動作確認では、WebOTX AS コンテナに配備したアプリケーションに対して、30080 ポートまたは 30443 ポートでアクセスし、正常に応答があることを確認してください。

5. アンインストール

本章では、WebOTX Operator for Kubernetes のアンインストール方法について説明します。

Kubernetesを前提とした方法を記載しているため、OpenShiftを使用する場合は、WebOTXオンラインマニュアルの「構築・運用 > コンテナ型仮想化 > 2, Operator > 2.5. OpenShiftへの展開」を確認してください。

5.1. アンインストール作業

1. ホスト OS にログイン名 root でログインします。

```
login: root
```

2. WebOTX AS コンテナを削除します。

```
root> cd <作業ディレクトリ>
root> kubectl delete -f manifest/webotx_cr.yaml -n <Namespace名>
```

Caution

上記コマンドを実行すると、展開中の WebOTX AS コンテナが削除されます。
展開中の WebOTX AS コンテナを残したまま、WebOTX Operator for Kubernetes のみ削除したい場合は、上記のコマンドを実行しないでください。

3. WebOTX Operator for Kubernetes を削除します。

```
root> cd <作業ディレクトリ>
root> kubectl delete -f manifest/operator.yaml -n <Namespace名>
```

4. ConfigMap を削除します。

```
root> cd <作業ディレクトリ>
root> kubectl delete -f manifest/configmap.yaml -n <Namespace名>
```

Caution

ConfigMap には、WebOTX AS コンテナの起動に必要なファイルが登録されています。
展開中の WebOTX AS コンテナを残したまま、WebOTX Operator for Kubernetes のみ削除したい場合は、上記のコマンドを実行しないでください。

5. サービスアカウントと権限を削除します。

```
root> cd <作業ディレクトリ>
root> kubectl delete -f manifest/service_account.yaml -n <Namespace名>
```

6. Namespace を削除します。(不要な場合のみ)

```
root> cd <作業ディレクトリ>
root> kubectl delete namespace <Namespace名>
```


7. カスタムリソース定義を削除します。

Caution

カスタムリソース定義は、全ての Namespace に適用されます。
このため、カスタムリソース定義を削除すると、WebOTX Operator for Kubernetes で配備した WebOTXAS コンテナが全ての Namespace から削除されます。

```
root> cd <作業ディレクトリ>
root> kubectl delete -f manifest/webotx_crd.yaml
```

8. WebOTX Operator for Kubernetes のコンテナイメージを削除します。

```
root> docker rmi <イメージID>
```

登録されている Docker イメージは以下のコマンドで確認することができます。

```
root> docker images
```

9. インストール時に使用した作業ディレクトリを削除します。

```
root> rm -rf <作業ディレクトリ>
```

6. 注意・制限事項

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。