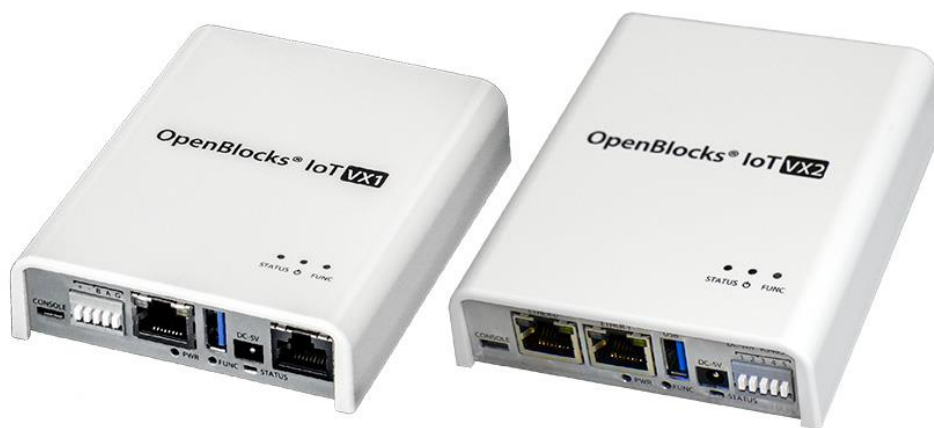


OpenBlocks IoT Family向け Docker設定ガイド



Ver.3.2.0

ぷらっとホーム株式会社

■ 商標について

- ・ Linux は、Linus Torvalds 氏の米国およびその他の国における商標あるいは登録商標です。
- ・ 文中の社名、商品名等は各社の商標または登録商標である場合があります。
- ・ その他記載されている製品名などの固有名詞は、各社の商標または登録商標です。
- ・ Microsoft、.NET、Windows、Microsoft Azure は、米国 Microsoft Corporation 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。
- ・ 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.

■ 使用にあたって

- ・ 本書の内容の一部または全部を、無断で転載することをご遠慮ください。
- ・ 本書の内容は予告なしに変更することがあります。
- ・ 本書の内容については正確を期するように努めていますが、記載の誤りなどにご指摘がございましたら弊社サポート窓口へご連絡ください。
また、弊社公開の WEB サイトにより本書の最新版をダウンロードすることが可能です。
- ・ 本装置の使用にあたっては、生命に関わる危険性のある分野での利用を前提とされていないことを予めご了承ください。
- ・ その他、本装置の運用結果における損害や逸失利益の請求につきましては、上記にかかわらずいかなる責任も負いかねますので予めご了承ください。

目次

第 1 章 はじめに	4
第 2 章 Docker 設定	4
2-1. WEB UI 込み Docker のインストール	4
2-2. WEB UI からの Docker 使用について	5
2-3. Docker コンテナ一覧	5
2-4. Docker コンテナリソース表示	6
2-5. Docker コンテナリソース設定更新	6
2-6. Docker コンテナ初回起動	7
2-7. Docker ローカルイメージ確認	8
2-8. Docker イメージの検索	9
2-9. Docker イメージのダウンロード	10
2-10. Docker コンテナのログ確認	11
2-11. Docker コンテナのネットワーク設定	12
2-12. プライベートレジストリサーバーの認証設定	14
2-14. ボリューム一覧表示及び削除	15
第 3 章 その他	16
3-1. Azure IoT Edge との連携	16

第1章 はじめに

本書は、OpenBlocks IoT Family に搭載可能な WBE UI 込みでの Docker(コンテナ型仮想化環境 OSS)の使用方法を解説しています。

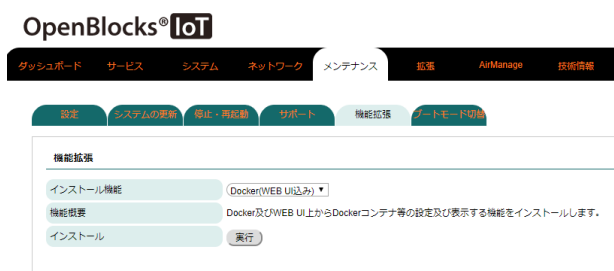
第2章 Docker 設定

2-1. WEB UI 込み Docker のインストール

本製品出荷時では、Docker 及び Docker 用 WEB UI はインストールされておられません。そのため、WEB UI の「メンテナンス」→「機能拡張」タブから Docker(または Moby)及び Docker 用 WEB UI のインストールを行います。



WEB UI の「メンテナンス」タブを選び、さらに「機能拡張」タブをクリックすると機能拡張用のパッケージを選択することができます。



インストール機能のリストから「Docker(WEB UI 込み)」を選択します。

その後、インストールの「実行」ボタンを押し、インストールを行ってください。

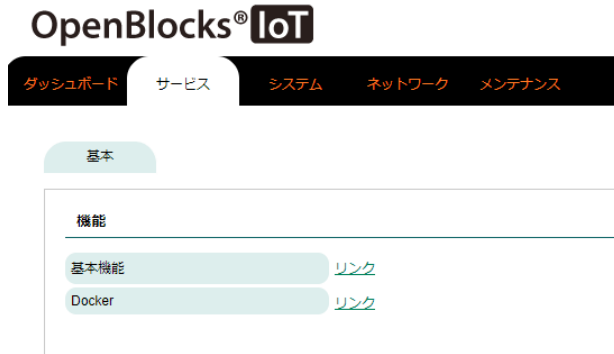
尚、インストール完了後には反映を行うため、本体再起動が必要となります。そのため、「メンテナンス」→「停止・再起動」から本製品の再起動を行ってください。

Docker または Moby をインストール際に、一部のドライバのコンパイルを行っております。これにより、インストールに時間がかかりますので、「状況確認」ボタン等にて進行状況をご確認ください。

尚、意図的に Docker エンジンではなく Moby エンジンを用いたい場合には、「Docker(WEB UI 込み)」のインストール前または後に「Moby」をインストールしてください。

2-2. WEB UI からの Docker 使用について

Docker(WEB UI 込み)パッケージがインストールされていると、WEB UI の「サービス」→「基本」タブに「Docker」が表示されます。



「サービス」→「基本」タブより「Docker」リンクをクリックすると、ルートタブがサービス用の「ダッシュボード」/「基本」/「Docker」に切り替わります。

※「サービス」→「基本」タブに表示されるリンク及びサービス用のルートタブは、「メンテナンス」→「機能拡張」タブよりインストールされたパッケージにより異なります。

2-3. Docker コンテナ一覧

一度でも起動したことがあるコンテナの状況の確認、起動・停止等の制御を「Docker」→「コンテナ一覧」タブにて行えます。

■ コンテナ稼働中の場合



コンテナが稼働中の場合、停止することができます。

停止したいコンテナの「停止」ボタンを押すことにより、停止することが可能です。

尚、稼働中のコンテナは公開ポート及び IP アドレスが表示されます。

■ コンテナ停止中の場合



コンテナ停止中の場合、起動及び削除が行えます。

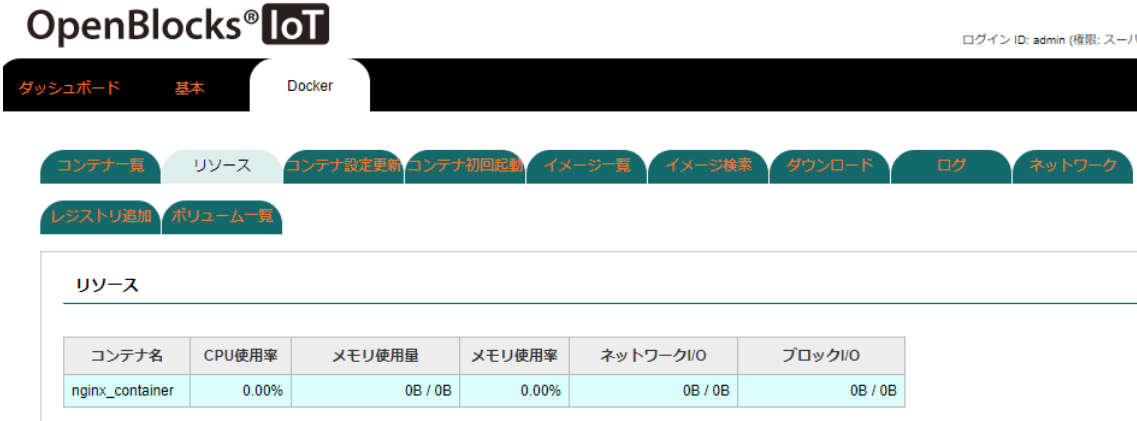
起動させる場合には、起動したいコンテナの「起動」ボタンを押すことで起動します。

対象コンテナを削除する場合には、「削除」ボタンを押してください。

2-4. Docker コンテナリソース表示

一度でも起動したことのあるコンテナのリソース状況確認を「**Docker**」→「**リソース**」タブにて行えます。停止中のコンテナは一覧に表示されますが、停止しているため各リソースは未使用状態となります。

また、本項にてリソースを確認し、各コンテナのリソースチューニングを行って下さい。



OpenBlocks® IoT ログイン ID: admin (権限: スーパ)

ダッシュボード 基本 Docker

コンテナ一覧 リソース コンテナ設定更新 コンテナ初回起動 イメージ一覧 イメージ検索 ダウンロード ログ ネットワーク

レジストリ追加 ボリューム一覧

リソース

コンテナ名	CPU使用率	メモリ使用量	メモリ使用率	ネットワークI/O	ブロックI/O
nginx_container	0.00%	0B / 0B	0.00%	0B / 0B	0B / 0B

2-5. Docker コンテナリソース設定更新

コンテナのリソース設定を「**Docker**」→「**コンテナ設定更新**」タブにて行えます。

本項では、ホストや他コンテナに影響がでるようなコンテナを使用する場合に、リソースを制限させることを前提としております。



OpenBlocks® IoT ログイン ID: admin (権限: スーパ)

ダッシュボード 基本 Docker camera

コンテナ一覧 リソース コンテナ設定更新 コンテナ初回起動 イメージ一覧 イメージ検索 ダウンロード ログ ネットワーク

レジストリ追加 ボリューム一覧

コンテナ設定更新 (0)

対象コンテナ

オプション

操作

リソース設定を更新するコンテナをプルダウンメニューから選択します。

対象コンテナ選択後、オプションフォーム及び「更新」ボタンを選択できるようになります。リソース変更用のオプションを入力し、「更新」ボタンを押すことにより適用されます。

本機能では、内部的に `docker update` コマンドを使用しております。

そのため、オプションフォームには `docker update` コマンドのリソース制御系のオプション部のみを記載してください。

Ex.)

■最大メモリを 50Mbyte に制限する場合

●doceker update コマンド時の例

```
# docker update --memory 50M --memory-swap 100M nginx_container
```

●オプションフォームに入力する場合の例

```
--memory 50M --memory-swap 100M
```

また、docker update コマンドの詳細については以下のページをご確認ください。

<http://docs.docker.jp/engine/reference/commandline/update.html>

2-6. Docker コンテナ初回起動

コンテナの作成・起動を「Docker」→「コンテナ初回起動」タブにて行えます。

ローカル環境に対象イメージが存在しない場合、自動的に Docker Hub から取得を行います。

※プライベートレジストリサーバのイメージを使用する場合には、「ダウンロード」タブにて事前にイメージを取得してください。



■イメージ

作成及び起動するコンテナのイメージ名を指定します。

■RUN オプション

docker run コマンドのオプションを指定します。尚、強制的にデタッチオプション(-d)が内部的に付与されます。

Docker 自体はインストール/再起動後、常時起動しております。WEB UI では Docker 本体の再起動等を一部の保存イベントアクションにて行うことがあります。

そのため、使用するコンテナはデタッチ(“docker run -d”)及び常時再起動ポリシー(“docker run --restart=always”)オプションを適用し運用してください。

コンテナの管理上、コンテナ名のオプション指定(--name <コンテナ名>)についても使用を推奨いたします。

また、docker run コマンドの詳細については以下のページをご確認ください。

<http://docs.docker.jp/engine/reference/commandline/run.html>

2-7. Docker ローカルイメージ確認

ローカルに存在する Docker コンテナイメージの一覧の確認を「Docker」→「イメージ一覧」タブにて行えます。



コンテナのもととなるイメージの一覧が表示されます。

使用中/未使用ともに「削除」ボタンが表示されますが、使用中のイメージは削除が行えません。そのため、削除を行う場合には、「コンテナ一覧」にて事前に対象イメージを使用しているコンテナを削除してください。

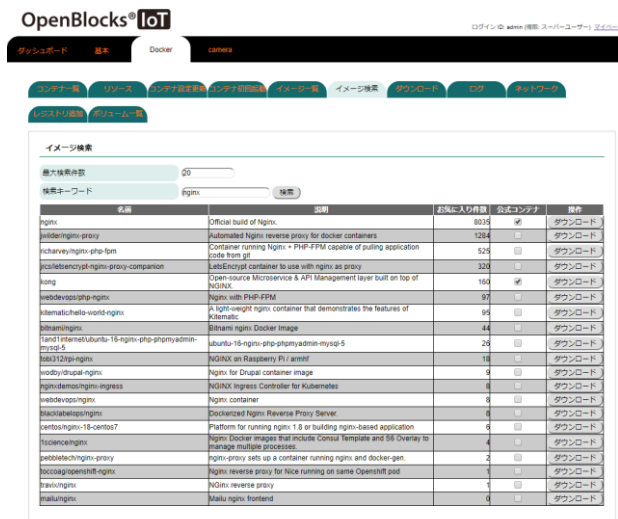
2-8. Docker イメージの検索

Docker Hub に公開されているコンテナイメージを「Docker」→「イメージ検索」タブにて行えます。



検索キーワードフォームに検索対象文字列を入力後、検索ボタンをおすことにより検索されます。

尚、最大検索件数は 100 件までとなっております。



先の図は”nginx”のイメージを検索した例となります。

ダウンロードボタンを押すことにより、対象のイメージをダウンロードすることが行えます。

2-9. Docker イメージのダウンロード

Docker Hub に公開されているコンテナイメージやプライベートレジストリサーバに存在するコンテナイメージのダウンロードを「Docker」→「ダウンロード」タブにて行えます。ダウンロードするコンテナイメージがプライベートレジストリサーバの場合、事前に対象サーバへのログイン情報が必要となります。こちらについては「レジストリ追加」タブにてログイン情報を設定してください。



ダウンロードしたいイメージ名を指定します。特定のタグのイメージを取得する場合には、イメージ名[:タグ名]を指定してください。

イメージ名入力後、ダウンロードボタンを押すことにより対象のイメージがダウンロードされます。

ダウンロードボタンを押した場合、「状況確認」ボタンが表示されます。このボタンで進行状況が確認できますので、ダウンロードするイメージサイズが大きい場合等にご利用ください。

2-10. Docker コンテナのログ確認

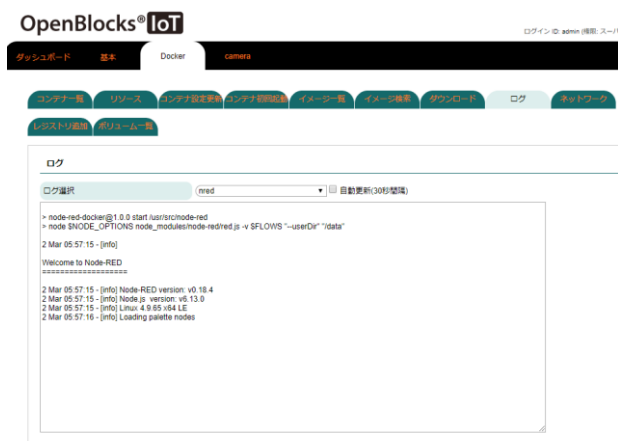
使用しているコンテナが出力しているログの確認を「Docker」→「ログ」タブにて行えます。

尚、コンテナを削除した場合には対象コンテナのログについても削除される為、本項での確認も行えなくなります。



ログを確認したいコンテナをプルダウンメニューから選択します。

コンテナを選択した場合、ログ表示領域が表示されます。



また、対象コンテナの最終ログから一定量表示されます。

2-11. Docker コンテナのネットワーク設定

作成したコンテナをデフォルトのネットワーク以外を用いたい場合、ネットワーク自体の作成及びコンテナとの接続・切断の設定を「**Docker**」→「**ネットワーク**」タブにて行えます。

#	ID	ネットワーク名	ドライバ	範囲	操作
1	803a3bc64e76	bridge	bridge	local	削除
2	3ace664df3a7	host	host	local	削除
3	f14cba9e1d4a	none	null	local	削除

タブを選択すると左図のように表示されます。

■ コンテナネットワーク作成

(`docker network create` コマンド)

作成したいネットワーク名及び作成オプションを指定し、「生成」ボタンを押すことでネットワークが生成されます。

■ コンテナネットワーク接続

(`docker network connect` コマンド)

接続したいネットワーク名及び対象コンテナ、接続オプションを指定し、「接続」ボタンを押すことでネットワークが接続されます。

■ コンテナネットワーク切断

(`docker network disconnect` コマンド)

切断したいネットワーク名及び対象コンテナを指定し、「切断」ボタンを押すことでネットワークが切断されます。

■ ネットワーク一覧

Docker のネットワークとして存在しているネットワークの一覧を表示します。作成したネットワークを削除したい場合には、「削除」ボタンを押すことにより削除可能です。

尚、一部のネットワークは Docker の制約上削除することができません。

また、docker network 関連のコマンドの詳細は以下のページをご確認ください。

■ docker network create コマンド

http://docs.docker.jp/engine/reference/commandline/network_create.html

■ docker network connect コマンド

http://docs.docker.jp/engine/reference/commandline/network_connect.html

■ docker network disconnect コマンド

http://docs.docker.jp/engine/reference/commandline/network_disconnect.html

2-12. プライベートレジストリサーバーの認証設定

第3者へ公開したくないコンテナ等を使用する場合、プライベートレジストリサーバーを用いる場合があります。このプライベートレジストリサーバーを用いる場合、認証情報が必要となります。認証情報の設定を「**Docker**」→「**レジストリ追加**」タブにて行えます。

The screenshot shows the OpenBlocks IoT Docker management interface. The top navigation bar includes 'Dashboard', 'Docker', and 'Camera'. The main content area is titled 'レジストリ追加' (Add Registry) and contains a form for adding a new registry. The form has fields for 'ログインサーバー' (Login Server), 'ユーザー名' (Username), and 'パスワード' (Password), with a 'パスワード表示' (Show Password) toggle. Below the form is a 'ログイン' (Login) button. Below the form is a table titled 'レジストリ一覧' (Registry List) with columns for '#', 'レジストリサーバー' (Registry Server), and '操作' (Action). The table contains one entry with ID '1' and server 'docker.plathome.com', with a 'ログアウト' (Logout) button in the action column.

ログインしたいプライベートレジストリサーバーに関する情報を設定します。

■ログインサーバー

ログインするプライベートレジストリサーバーの FQDN または IP アドレスを指定します。

■ユーザー名

ログインする際のユーザー名を指定します。

■パスワード

ログイン時のパスワードを指定します。

ログインに成功した場合、左図のようにレジストリ一覧に追加されます。

尚、対象レジストリサーバーからログアウトを行う場合には、「ログアウト」ボタンを押して下さい。

尚、ログインボタンを押した段階で実際にログイン処理が適用されます。そのため、プライベートレジストリサーバーへネットワーク接続が行える環境にて実施してください。

2-14. ボリューム一覧表示及び削除

一部のコンテナは、コンテナ及びイメージの削除を行った場合でもストレージ上に存在し続けることがあります。その場合、意図的にボリュームを削除する必要があります。

Docker 管理がディスクボリュームの一覧を「**Docker**」→「**ボリューム一覧**」タブにて確認及び削除することができます。

存在している一覧のボリュームを表示します。



「未使用データ削除」ボタンを押すことにより、既に不要となっているボリュームを削除することができます。

※未使用データ削除は**Docker DAEMON**が不要と判断しているデータを削除します。

停止中のコンテナ等についても削除される恐れがありますので、必要なコンテナが稼働中の状態にて実施してください。

第 3 章 その他

3-1. Azure IoT Edge との連携

Azure IoT Edge はコンテナエンジンを使用しています。Azure IoT Edge 部の WEB UI では、コンテナ自体の直接制御機能はサポートしておりません。Azure IoT Edge の使用を想定している場合には、本機能のご使用についてもご検討ください。

尚、Azure IoT Edge は Docker ではなく Moby エンジンの使用を推奨しております。そのため、Azure IoT Edge を使用する場合には、Moby エンジンをご使用ください。

OpenBlocks IoT Family 向け Docker 設定ガイド
(2018/08/07 第 2 版)

ぷらっとホーム株式会社

〒102-0073 東京都千代田区九段北 4-1-3 日本ビルディング九段別館 3F