

FIWARE Orionビルド手順書 (1.0.0版)

2022年07月01日
一般社団法人データ社会推進協議会

改版履歴

バージョン	改版内容	公開日
1.0.0	初版	2022/07/01

目次

1.	はじめに	4
1-1.	概要	4
1-2.	参考資料	4
1-3.	前提条件	4
1-4.	表記方法	4
1-4-1.	コマンドの表記方法	4
1-4-2.	コマンド入力結果の表記方法	4
2.	ビルド手順	5
2-1.	事前準備	5
2-2.	ビルド	6
2-3.	パッケージの動作確認	7
3.	付録	9
3-1.	必須パッケージインストール方法	9

1. はじめに

本書は、ブローカー（非パーソナル）として利用するソフトウェアの FIWARE Orion（正式名称 Orion Context Broker）（以下、「Orion」）をビルドする手順を記載・説明するものである。

1-1. 概要

Orion の rpm パッケージのビルド方法と、ビルドしたパッケージの動作確認について述べる。

1-2. 参考資料

参考にした資料は以下である。

https://fiware-orion.readthedocs.io/en/3.6.0/admin/build_source/index.html

1-3. 前提条件

- CentOS(8.5.2111)上で実施すること。
 - wget(1.19.5)がインストールされていること。
 - docker(20.10.13)がインストールされていること。
 - インターネットへアクセス可能であること。
 - sudo を使って root 権限でコマンド実行できること。
- ※パッケージ名の後ろの()内は、本書で確認したバージョンである。

1-4. 表記方法

本書での表記方法について記載する。

1-4-1. コマンドの表記方法

(例)

```
$ source ~/ENV.sh
```

コマンド入力を表す箇所については、上記のように実線で囲んでいる。

行頭の \$ はプロンプトであり、入力するのはそれ以降の青い背景色の部分である。

1-4-2. コマンド入力結果の表記方法

(例)

```
HTTP/1.1 201 Created
Date: Wed, 09 Mar 2022 13:20:02 GMT
Content-Type: application/json; charset=utf-8
:
```

コマンド入力結果を表す箇所については、上記のように破線で囲み橙色の背景色で表記している。

2. ビルド手順

本章では Orion のビルド手順について記載・説明する。

はじめにビルドマシン (CentOS 8) へ一般ユーザでログインする。

※本書では作業ディレクトリを「~/src 配下」としているが、任意のディレクトリを作業ディレクトリとすることも可能である (その場合は作業ディレクトリを読み替えて実行すること)。

2-1. 事前準備

Orion をビルドするために必要なツールやライブラリをインストールする。

ビルドツールとライブラリをインストールする。

```
$ sudo yum -y install make cmake gcc-c++
$ sudo yum -y install boost-devel libcurl-devel gnutls-devel libgcrypt-devel openssl-devel
libuuid-devel cyrus-sasl-devel
```

Mongo Driver をインストールする。

```
$ mkdir -p ~/src/mong
$ cd ~/src/mong
$ wget https://github.com/mongodb/mongo-c-driver/releases/download/1.17.4/mongo-c-driver-1.17.4.tar.gz
$ tar xfvz mongo-c-driver-1.17.4.tar.gz
$ cd mongo-c-driver-1.17.4
$ mkdir cmake-build
$ cd cmake-build
$ cmake -DENABLE_AUTOMATIC_INIT_AND_CLEANUP=OFF ..
$ make
$ sudo make install
```

rapidjson をインストールする。

```
$ mkdir -p ~/src/rapidjson
$ cd ~/src/rapidjson
$ wget https://github.com/miloyip/rapidjson/archive/v1.1.0.tar.gz
$ tar xfvz v1.1.0.tar.gz
$ sudo mv rapidjson-1.1.0/include/rapidjson/ /usr/local/include
```

libmicrohttpd をインストールする。

※「./configure」コマンドのオプションはライブラリの最小限のフットプリントを得るための推奨ビルド設定であり、必要であれば変更することは可能である。

```
$ mkdir -p ~/src/libmicrohttpd
$ cd ~/src/libmicrohttpd
$ wget http://ftp.gnu.org/gnu/libmicrohttpd/libmicrohttpd-0.9.70.tar.gz
$ tar xvf libmicrohttpd-0.9.70.tar.gz
$ cd libmicrohttpd-0.9.70
$ ./configure --disable-messages --disable-postprocessor --disable-dauth
$ make
$ sudo make install
$ sudo ldconfig
```

mosquitto をインストールする。

※WITH_CJSON, WITH_STATIC_LIBRARIES, WITH_SHARED_LIBRARIES の設定を変更することで、「mosquitto-2.0.12/」ディレクトリ配下の「config.mk」ファイルを変更してビルドを微調整可能である。

```
$ mkdir -p ~/src/mosquitto
$ cd ~/src/mosquitto
$ wget http://mosquitto.org/files/source/mosquitto-2.0.12.tar.gz
$ tar xvf mosquitto-2.0.12.tar.gz
$ cd mosquitto-2.0.12
$ sed -i 's/WITH_CJSON:=yes/WITH_CJSON:=no/g' config.mk
$ sed -i 's/WITH_STATIC_LIBRARIES:=no/WITH_STATIC_LIBRARIES:=yes/g' config.mk
$ sed -i 's/WITH_SHARED_LIBRARIES:=yes/WITH_SHARED_LIBRARIES:=no/g' config.mk
$ make
$ sudo make install
$ sudo ldconfig
```

2-2. ビルド

Orion をビルドする。

ソースコードを取得する。

```
$ cd ~/src
$ sudo yum -y install git
$ git clone https://github.com/telefonicaid/fiware-orion -b 3.6.0 --depth 1
```

ビルドする。

```
$ cd ~/src/fiware-orion/
$ sudo yum -y install rpm-build
$ make rpm
```

ビルドしたパッケージを確認する。

```
$ ls ./rpm/RPMS/x86_64/contextBroker-3.6.0-dev.x86_64.rpm
```

以下のファイルが存在することを確認する。

```
contextBroker-3.6.0-dev.x86_64.rpm
```

2-3. パッケージの動作確認

Docker ネットワークを作成する。

```
$ sudo docker network create orion-net
```

DB コンテナを作成する。

```
$ sudo docker run -d \
  --name orion-database \
  --network=orion-net \
  mongo:4.4
```

Orion コンテナを作成する。

```
$ sudo docker run -d -it \
  --name orion \
  --network=orion-net \
  -p 1026:1026 \
  centos:8
```

パッケージを Orion コンテナへコピーする。

```
$ sudo docker cp ./rpm/RPMS/x86_64/contextBroker-3.6.0_next-dev.x86_64.rpm orion:/tmp
```

Orion コンテナにログインする。

```
$ sudo docker exec -it orion /bin/bash
```

以下、コンテナ内での操作となり、コンテナ内での root ユーザでの操作に移行している点に注意する。

動作確認に必要なパッケージを Orion コンテナへインストールする。

```
# sed -i -e 's/^mirrorlist/#mirrorlist/g' -e \
  's/^#baseurl=http://mirror/baseurl=http://vault/g' /etc/yum.repos.d/CentOS-*repo
# yum -y install python2 boost-filesystem boost-thread logrotate openssl
# yum -y install boost libuuid curl
```

Orion パッケージを Orion コンテナへインストールする。

```
# rpm -ivh /tmp/contextBroker-3.6.0_next-dev.x86_64.rpm
# contextBroker --version
```

Orion パッケージのバージョンを確認する。

```
3.6.0 (git version: 973850279e63d58cb93dff751648af5ec6e05777)
```

Orion プロセスを開始する。

```
# contextBroker -dbhost orion-database
```

Orion プロセスの動作を確認する。

```
# curl localhost:1026/version
```

Orion の情報が返却されることを確認する。

※返却される値は環境によって異なるため、Orion の情報が返ってくることを確認する。

```
{
  "orion" : {
    "version" : "3.6.0",
    "uptime" : "0 d, 0 h, 0 m, 8 s",
    "git_hash" : "973850279e63d58cb93dff751648af5ec6e05777",
    "compile_time" : "Wed Mar 16 02:25:43 UTC 2022",
    "compiled_by" : "centos",
    "compiled_in" : "ip-10-64-1-35.us-west-1.compute.internal",
    "release_date" : "Wed Mar 16 02:25:43 UTC 2022",
    "machine" : "x86_64",
    "doc" : "https://fiware-orion.rtd.io/en/3.6.0/",
    "libversions": {
      "boost": "1_66",
      "libcurl": "libcurl/7.61.1 OpenSSL/1.1.1k zlib/1.2.11 nghttp2/1.33.0",
      "libmosquitto": "2.0.12",
      "libmicrohttpd": "0.9.70",
      "openssl": "1.1",
      "rapidjson": "1.1.0",
      "mongoc": "1.17.4",
      "bson": "1.17.4"
    }
  }
}
```

Orion コンテナからログアウトする。

```
# exit
```

3. 付録

本章では Orion のビルドに直接は関係しないが、間接的に必要な情報について記載・説明する。

3-1. 必須パッケージインストール方法

ビルドに必要なパッケージのインストール方法の一例を記載する。すでにインストールされている場合は本作業は不要である。また、インストール方法は複数あるため、本書の手順以外でインストールすることも可能である。

以下の手順は全て root ユーザーで実施すること。

- wget

```
# yum -y install wget
```

- docker

```
# yum install -y yum-utils device-mapper-persistent-data lvm2
# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
# yum install -y docker-ce docker-ce-cli containerd.io
# systemctl enable docker
# systemctl start docker
```