

Kong Gateway構築手順書 (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.	バージョン	4
1-4.	構成図	5
1-5.	前提条件	5
1-6.	表記方法	5
1-6-1.	コマンドの表記方法	5
1-6-2.	コマンド入力結果の表記方法	5
2.	構築手順	6
2-1.	構築	6
2-2.	認証プラグインの追加	9
2-3.	動作確認用データの削除	11
3.	付録	13
3-1.	永続化	13
3-2.	可用性	13
3-3.	性能	13

1. はじめに

本書は、API ゲートウェイとして利用するソフトウェアの Kong Gateway を構築する手順を記載・説明するものである。

1-1. 概要

Kong Gateway で選択可能なライセンス形態、利用形態、インストール方式の種類については以下のとおりである。本書では、OSS 版 の DB 使用モード (PostgreSQL 9.5+) を Docker 方式でインストールする方法について述べる。

Kong Gateway には下記のライセンス形態がある。

- Kong Gateway (OSS 版) … API ゲートウェイ基本機能、OSS プラグイン、AdminAPI
- Kong Gateway (ライセンス版: フリー) … OSS 版+KongManager
- Kong Gateway (ライセンス版: プラス) … フリー版+KongConnectCloud を介した拡張機能
- Kong Gateway (ライセンス版: エンタープライズ) … プラス版+開発ポータル、バイタル、RBAC、エンタープライズプラグイン

利用形態として下記がある。

- DB 使用モード (PostgreSQL 9.5+)
- DB 使用モード (Cassandra 3.11.*) - 非推奨。Kong Gateway 4.0 からは削除される予定。

インストール方式として下記がある。

- Docker 方式 (Docker Hub 公式イメージを使用)
- Kubernetes 方式
- OpenShift with Helm 方式
- 直インストール方式 (RHEL, CentOS, Ubuntu, Debian, macOS, Amazon Linux)

1-2. 参考資料

本書では、以下の参考資料を参照している。

<https://docs.konghq.com/gateway/>

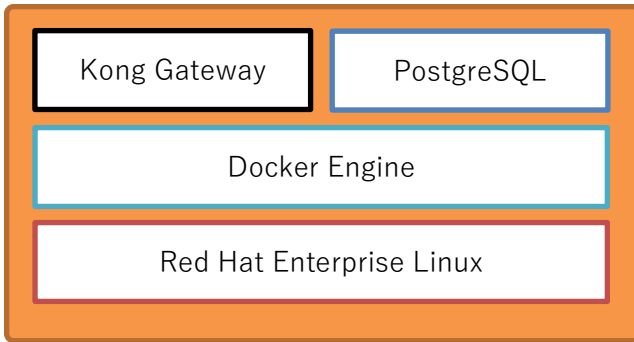
1-3. バージョン

本書で扱うソフトウェアのバージョンは以下のとおりである。

OS	Red Hat Enterprise Linux 7.9
Docker	Docker 1.13.1
DB	PostgreSQL 9.6
Kong Gateway	Kong Gateway 2.8.0

1-4. 構成図

本書で構築する対象となる構成図は以下のとおりである。



1-5. 前提条件

本書を用いて構築する際の前提条件は以下のとおりである。

- Red Hat Enterprise Linux (7.9)上で実施すること。
- docker(1.13.1)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。
- root ユーザーでログインできること。

1-6. 表記方法

1-6-1. 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。

1-6-2. 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. 構築手順

本章では Kong Gateway の構築手順について記載・説明する。

はじめに構築マシン (Red Hat Enterprise Linux 7) へ root ユーザでログインしておくこと。

2-1. 構築

Kong Gateway を構築する。

パラメータを準備する。

```
# cat << EOF > ~/ENV.sh
KONG_HOST=kong
KONG_VERSION=2.8.0
USER_PORT=8000
USER_HTTPS_PORT=8443
ADMIN_PORT=8001
ADMIN_HTTPS_PORT=8444
DB_HOST=kong-database
DB_VERSION=9.6
DB_PORT=5432
DB_USER=kong
DB_PASSWD=kong
DB_NAME=kong
SERVICE_NAME=example-service
SERVICE_URL=http://mockbin.org
SERVICE_HOST=example.com
SAMPLE_USER_NAME=Yamada
SAMPLE_USER_KEY=GHfnFK45VFH
EOF
```

各パラメータの詳細は以下の通りである。

- KONG_HOST: Kong Gateway の Docker コンテナの名前を指定する。
- KONG_VERSION: Kong Gateway の Docker イメージのバージョンを指定する。
- USER_PORT: Kong Gateway の利用者向け API のポートを指定する。
- USER_HTTPS_PORT: Kong Gateway の利用者向け API のポート (SSL) を指定する。
- ADMIN_PORT: Kong Gateway の管理用 API のポートを指定する。
- ADMIN_HTTPS_PORT: Kong Gateway の管理用 API のポート (SSL) を指定する。
- DB_HOST: PostgreSQL の Docker コンテナの名前を指定する。
- DB_VERSION: PostgreSQL の Docker イメージのバージョンを指定する。
- DB_PORT: PostgreSQL のポート番号を指定する。
- DB_USER: PostgreSQL の接続ユーザ名を指定する。
- DB_PASSWD: PostgreSQL の接続パスワードを指定する。
- DB_NAME: PostgreSQL のデータベース名を指定する。
- SERVICE_NAME: Kong Gateway に関連付けるサービス名称を指定する。
- SERVICE_URL: Kong Gateway に関連付けるサービスの URL を指定する。

- SERVICE_HOST: Kong Gateway に関連付けたサービスを呼び出す際のホスト名を指定する。
- SAMPLE_USER_NAME: Kong Gateway 認証プラグインで認証する際のユーザ名を指定する。
- SAMPLE_USER_KEY: Kong Gateway 認証プラグインで認証する際の認証キーを指定する。

パラメータを読み込む。

```
# source ~/ENV.sh
```

PostgreSQL を起動する。

```
# docker run -d ¥
--name $DB_HOST ¥
-p $DB_PORT:$DB_PORT ¥
-e "POSTGRES_USER=$DB_USER" ¥
-e "POSTGRES_PASSWORD=$DB_PASSWD" ¥
-e "POSTGRES_DB=$DB_NAME" ¥
postgres:$DB_VERSION
```

一時的な Kong Gateway コンテナを用いて PostgreSQL のマイグレーションを行う。

```
# docker run --rm ¥
--link $DB_HOST:$DB_HOST ¥
-e "KONG_DATABASE=postgres" ¥
-e "KONG_PG_HOST=$DB_HOST" ¥
-e "KONG_PG_USER=$DB_USER" ¥
-e "KONG_PG_PASSWORD=$DB_PASSWD" ¥
-e "KONG_CASSANDRA_CONTACT_POINTS=$DB_HOST" ¥
$KONG_HOST:$KONG_VERSION kong migrations bootstrap
```

Kong Gateway コンテナを起動する。

```
# docker run -d ¥
--name $KONG_HOST ¥
--link $DB_HOST:$DB_HOST ¥
-e "KONG_DATABASE=postgres" ¥
-e "KONG_PG_HOST=$DB_HOST" ¥
-e "KONG_PG_USER=$DB_USER" ¥
-e "KONG_PG_PASSWORD=$DB_PASSWD" ¥
-e "KONG_CASSANDRA_CONTACT_POINTS=$DB_HOST" ¥
-e "KONG_PROXY_ACCESS_LOG=/dev/stdout" ¥
-e "KONG_ADMIN_ACCESS_LOG=/dev/stdout" ¥
-e "KONG_PROXY_ERROR_LOG=/dev/stderr" ¥
-e "KONG_ADMIN_ERROR_LOG=/dev/stderr" ¥
-e "KONG_ADMIN_LISTEN=0.0.0.0:$ADMIN_PORT, 0.0.0.0:$ADMIN_HTTPS_PORT ssl" ¥
-p $USER_PORT:$USER_PORT ¥
-p $USER_HTTPS_PORT:$USER_HTTPS_PORT ¥
-p $ADMIN_PORT:$ADMIN_PORT ¥
-p $ADMIN_HTTPS_PORT:$ADMIN_HTTPS_PORT ¥
$KONG_HOST:$KONG_VERSION
```

Kong Gateway にサービスを登録する。

```
# curl -i -X POST ¥  
--url http://localhost:$ADMIN_PORT/services/ ¥  
--data "name=$SERVICE_NAME" ¥  
--data "url=$SERVICE_URL"
```

201 Created が返却されることを確認する。

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

Kong Gateway のサービスにルーティングを追加する。

```
# curl -i -X POST ¥  
--url http://localhost:$ADMIN_PORT/services/$SERVICE_NAME/routes ¥  
--data "hosts[]=$SERVICE_HOST"
```

201 Created が返却されることを確認する。

```
HTTP/1.1 201 Created  
Date: Wed, 09 Mar 2022 13:24:08 GMT  
Content-Type: application/json; charset=utf-8  
:
```

Kong Gateway のサービスにアクセスする。

```
# curl -i -X GET ¥  
--url http://localhost:$USER_PORT/ ¥  
--header "Host: $SERVICE_HOST"
```

Kong Gateway のサービスにアクセスした際の情報が返却されることを確認する。

```
HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8  
Transfer-Encoding: chunked  
Connection: keep-alive  
Date: Wed, 09 Mar 2022 13:26:12 GMT  
:
```


2-2. 認証プラグインの追加

認証プラグインを追加する。

```
# curl -i -X POST ¥  
--url http://localhost:$ADMIN_PORT/services/$SERVICE_NAME/plugins/ ¥  
--data 'name=key-auth'
```

201 Created が返却されることを確認する。

```
HTTP/1.1 201 Created  
Date: Wed, 09 Mar 2022 14:25:46 GMT  
Content-Type: application/json; charset=utf-8  
:
```

認証が有効であることを確認する。

```
# curl -i -X GET ¥  
--url http://localhost:$USER_PORT/ ¥  
--header "Host: $SERVICE_HOST"
```

401 認証エラーが返却されることを確認する。

```
HTTP/1.1 401 Unauthorized  
Date: Wed, 09 Mar 2022 14:30:05 GMT  
Content-Type: application/json; charset=utf-8  
:
```

コンシューマ（利用者ユーザ）を追加する。

```
# curl -i -X POST ¥  
--url http://localhost:$ADMIN_PORT/consumers/ ¥  
--data "username=$SAMPLE_USER_NAME"
```

201 Created が返却されることを確認する。

```
HTTP/1.1 201 Created  
Date: Wed, 09 Mar 2022 14:34:32 GMT  
Content-Type: application/json; charset=utf-8  
:
```

コンシューマ（利用者ユーザ）に認証キーを設定する。

```
# curl -i -X POST ¥  
--url http://localhost:$ADMIN_PORT/consumers/$SAMPLE_USER_NAME/key-auth/ ¥  
--data "key=$SAMPLE_USER_KEY"
```

201 Created が返却されることを確認する。

```
HTTP/1.1 201 Created  
Date: Wed, 09 Mar 2022 14:40:22 GMT  
Content-Type: application/json; charset=utf-8  
:
```

認証が有効であることを確認する。

```
# curl -i -X GET ¥  
--url http://localhost:$USER_PORT/ ¥  
--header "Host: $SERVICE_HOST" ¥  
--header "apikey: $SAMPLE_USER_KEY"
```

Kong Gateway のサービスにアクセスした際の情報が返却されることを確認する。

```
HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8  
Transfer-Encoding: chunked  
Connection: keep-alive  
Date: Wed, 09 Mar 2022 13:26:12 GMT  
:
```

2-3. 動作確認用データの削除

構築の際、動作確認に使用したサービスやコンシューマなどのデータを削除する。

コンシューマ（利用者ユーザ）を削除する。

```
# curl -i -X DELETE ¥  
  --url http://localhost:$ADMIN_PORT/consumers/$SAMPLE_USER_NAME
```

204 No Content が返却されることを確認する。

```
HTTP/1.1 204 No Content  
Date: Wed, 09 Mar 2022 14:40:22 GMT  
:
```

ルーティングの一覧を取得する。

```
# curl -X GET ¥  
  --url http://localhost:$ADMIN_PORT/routes | python -mjson.tool
```

ルーティングの一覧が返却されることを確認する。

```
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current  
          Dload  Upload  Total      Spent    Left  Speed  
100   503   100   503    0     0  75299      0 --:--:-- --:--:-- --:--:--  83833  
{  
  "data": [  
    {  
      "created_at": 1646880200,  
      "destinations": null,  
      "headers": null,  
      "hosts": [  
        "example.com"  
      ],  
      "https_redirect_status_code": 426,  
      "id": "814fb8c4-8fa3-4451-ab24-c452d940a79b",  
      << 中略 >>  
    }  
  ],  
  "next": null  
}
```

ルーティング一覧から得られた id を使ってルーティングを削除する。

```
# curl -i -X DELETE ¥  
  --url http://localhost:$ADMIN_PORT/routes/814fb8c4-8fa3-4451-ab24-c452d940a79b
```

204 No Content が返却されることを確認する。

```
HTTP/1.1 204 No Content  
Date: Wed, 09 Mar 2022 14:43:34 GMT  
:
```

Kong Gateway のサービスを削除する。

```
# curl -i -X DELETE ¥  
  --url http://localhost:$ADMIN_PORT/services/$SERVICE_NAME
```

204 No Content が返却されることを確認する。

```
HTTP/1.1 204 No Content  
Date: Wed, 09 Mar 2022 14:46:16 GMT  
:
```

3. 付録

本章では Kong Gateway の構築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. 永続化

Docker で使用するデータは、Docker コンテナが存在している間は存在し続けるが、コンテナを再起動または破棄するとデータは失われてしまうこととなる。そのため、運用の際はデータの永続化が必要となる。

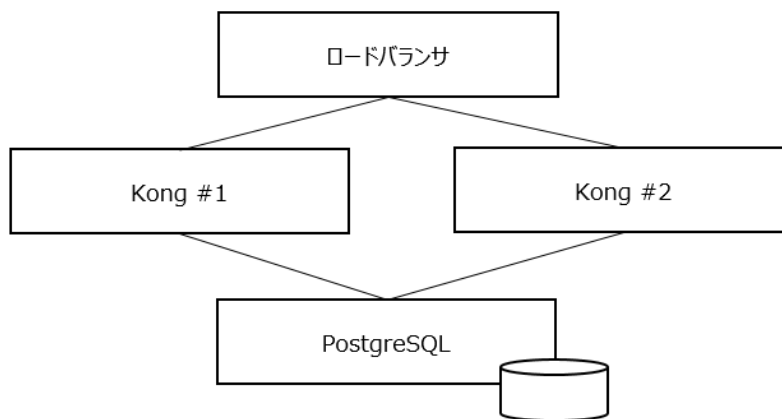
データの永続化を行うには、PostgreSQL の Docker コンテナのデータを永続化可能な場所に保存できるようにする必要がある。

データの永続化についての詳細は下記を参照すること。

<https://github.com/docker-library/docs/blob/master/postgres/README.md#where-to-store-data>

3-2. 可用性

可用性を高めるには下記のような冗長構成とする。



Admin API を用いて設定を行った場合は、片方の Kong Gateway に設定が反映され、PostgreSQL に書き込まれる。

他方の Kong Gateway は、一定期間毎にポーリングして PostgreSQL の設定情報を読み出すため、タイムラグがあることに注意する。

ポーリング時間等の設定の詳細は下記を参照すること。

<https://docs.konghq.com/gateway/2.8.x/reference/clustering/>

3-3. 性能

Kong Gateway の性能を測定・管理するために、パフォーマンステストフレームワークがサポートされている。

パフォーマンステストフレームワークの詳細については、下記を参照のこと。

<https://docs.konghq.com/gateway/2.8.x/plan-and-deploy/performance-testing-framework/>