

# Kong Gatewayビルド手順書 (1.2.0版)

2024年03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 (Kong Gateway 2.8.0 を対象に作成)	2022/07/01
1.1.0	Kong Gateway 3.1.0 向けに手順書を改版	2023/03/01
1.2.0	Kong Gateway 3.4.0 向けに手順書を改版	2024/03/01

# 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 .....	5
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、API ゲートウェイとして利用するソフトウェアの Kong Gateway をビルドする手順を記載・説明するものである。

### 1-1. 概要

Kong Gateway 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://github.com/Kong/kong/blob/3.4.0/build/README.md>

<https://docs.konghq.com/gateway/3.4.x/install/docker/>

### 1-3. 前提条件

- Debian(11.8)上で実施すること。
  - docker(24.0.7)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。
- ※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

#### 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: Tue, 18 Jan 2024 01:27:08 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 Kong Gateway 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 (Debian 11.8) へ root ユーザでログインしておくこと。

### 2-1. 事前準備

GitHub アクセストークンを設定する。

```
# export GITHUB_TOKEN=xxxxx
```

xxxxx は、各個人用の GitHub アクセストークンを記入する。個人用の GitHub アクセストークンの取得方法は以下サイトを参照。

<https://docs.github.com/ja/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens>

Kong Gateway 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。

```
# apt-get install make cmake g++ unzip libz-dev libyaml-dev
```

### 2-2. ビルド

Kong Gateway をビルドする。

ソースを入手する (ダウンロードするディレクトリは任意)。

```
# git clone https://github.com/Kong/kong.git -b 3.4.0 --depth 1
```

作業ディレクトリへ移動する。

```
# cd kong
```

Bazelisk をインストールする。

```
# make check-bazel
```

Baselisk の bin を Path 環境変数に追加する。

```
# export PATH=bin:$PATH
```

Bazel のバージョンを確認する。

```
# bazel version
```

以下のバージョン表示がされることを確認する。

```
Bazelisk version: v1.17.0
```

依存関係を構築する。

```
# bazel build --config release //build:kong --verbose_failures
```

ビルドする。

```
# bazel build --verbose_failures --config release :kong_deb
```

ビルドしたパッケージを確認する。

```
# ls bazel-bin/pkg
```

以下のファイルが存在することを確認する。

```
kong.amd64.deb
```

## 2-3. パッケージの動作確認

Docker ネットワークを作成する。

```
# docker network create kong-net
```

DB コンテナを作成する。

```
# docker run -d ¥  
  --name kong-database ¥  
  --network=kong-net ¥  
  -p 5432:5432 ¥  
  -e "POSTGRES_USER=kong" ¥  
  -e "POSTGRES_DB=kong" ¥  
  -e "POSTGRES_PASSWORD=kongpass" ¥  
  postgres:12.17
```

Kong Gateway コンテナを作成する。

```
# docker run -d -it ¥  
  --name kong-gateway ¥  
  --network=kong-net ¥  
  ubuntu:jammy
```

パッケージを Kong Gateway コンテナへコピーする。

```
# docker cp ./bazel-bin/pkg/kong.amd64.deb kong-gateway:/tmp
```

Kong Gateway コンテナにログインする。

```
# docker exec -it kong-gateway /bin/bash
```

動作確認に必要なパッケージを Kong Gateway コンテナへインストールする。

```
# apt-get update  
# apt-get install ca-certificates perl zlib1g-dev libyaml-0-2 curl
```

Kong Gateway パッケージを Kong Gateway コンテナへインストールする。

```
# apt install /tmp/kong.amd64.deb  
# kong version
```

Kong Gateway パッケージのバージョンを確認する。

```
3.4.0
```

Kong Gateway パッケージのコンフィグファイルをカスタマイズする。

```
# cp /etc/kong/kong.conf.default /etc/kong/kong.conf
# sed -i 's/^#pg_host = .*/pg_host = kong-database/' /etc/kong/kong.conf
# sed -i 's/^#pg_user = .*/pg_user = kong/' /etc/kong/kong.conf
# sed -i 's/^#pg_password = .*/pg_password = kongpass/' /etc/kong/kong.conf
# sed -i 's/^#pg_database = .*/pg_database = kong/' /etc/kong/kong.conf
```

DB コンテナをマイグレーションする。

```
# kong migrations bootstrap
```

Kong Gateway プロセスを開始する。

```
# kong start
```

Kong Gateway プロセスの動作を確認する。

```
# curl -i http://localhost:8001/services
```

200 OK が返却されることを確認する。

```
HTTP/1.1 200 OK
Date: Thu, 18 Jan 2024 01:27:08 GMT
:
{"data":[],"next":null}
```

Kong Gateway コンテナからログアウトする。

```
# exit
```



### 3. 付録

本章では Kong Gateway 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。

- docker

```
# apt-get install -y ca-certificates curl gnupg lsb-release
# curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg
# echo ¥
"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-
keyring.gpg] https://download.docker.com/linux/debian ¥
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
# apt-get update
# apt-get install -y docker-ce docker-ce-cli containerd.io
# systemctl enable docker
# systemctl start docker
```