

OFFICIAL REPOSITORY

kong (/r/ /kong/) ☆

Last pushed: 22 days ago

[Repo Info \(/ /kong/\)](#)

Short Description

Open-source Microservice & API Management layer built on top of NGINX.

Full Description

Supported tags and respective Dockerfile links

- 1.0.0rc2-alpine , 1.0rc2-alpine , 1.0.0rc2 , 1.0rc2 ([alpine/Dockerfile](#))
(<https://github.com/Kong/docker-kong/blob/e200d692b453f308d33f3f552293b66a47bee5ed/alpine/Dockerfile>)
- 1.0.0rc2-centos , 1.0rc2-centos ([centos/Dockerfile](#))
(<https://github.com/Kong/docker-kong/blob/e200d692b453f308d33f3f552293b66a47bee5ed/centos/Dockerfile>)
- 0.14.1-alpine , 0.14-alpine , 0.14.1 , 0.14 , latest ([alpine/Dockerfile](#))
(<https://github.com/Kong/docker-kong/blob/9bbc9e8b4a61cba5cc4e4e4562a802dace4348ff/alpine/Dockerfile>)
- 0.14.1-centos , 0.14-centos ([centos/Dockerfile](#))
(<https://github.com/Kong/docker-kong/blob/9bbc9e8b4a61cba5cc4e4e4562a802dace4348ff/centos/Dockerfile>)
- 0.13.1-alpine , 0.13-alpine , 0.13.1 , 0.13 ([alpine/Dockerfile](#))
(<https://github.com/Kong/docker-kong/blob/6e2035c5739482f0616021a7eda04ec6809d9f3e/alpine/Dockerfile>)
- 0.13.1-centos , 0.13-centos ([centos/Dockerfile](#))
(<https://github.com/Kong/docker-kong/blob/6e2035c5739482f0616021a7eda04ec6809d9f3e/centos/Dockerfile>)
- 0.12-alpine , 0.12.3-alpine ([alpine/Dockerfile](#))
(<https://github.com/Kong/docker-kong/blob/95f9e8ebfd735f4ab09af71bf060d200a9a7a981/alpine/Dockerfile>)

- [0.12-centos](https://github.com/Kong/docker-kong/blob/95f9e8ebfd735f4ab09af71bf060d200a9a7a981/centos/Dockerfile) , [0.12.3-centos](https://github.com/Kong/docker-kong/blob/95f9e8ebfd735f4ab09af71bf060d200a9a7a981/centos/Dockerfile) , [0.12](https://github.com/Kong/docker-kong/blob/95f9e8ebfd735f4ab09af71bf060d200a9a7a981/centos/Dockerfile) , [0.12.3](https://github.com/Kong/docker-kong/blob/95f9e8ebfd735f4ab09af71bf060d200a9a7a981/centos/Dockerfile) (*centos/Dockerfile*)
(<https://github.com/Kong/docker-kong/blob/95f9e8ebfd735f4ab09af71bf060d200a9a7a981/centos/Dockerfile>)
- [0.11-alpine](https://github.com/Kong/docker-kong/blob/2ed0fad88ed81623ea4afb2111033a9d87f3c03c/alpine/Dockerfile) , [0.11.2-alpine](https://github.com/Kong/docker-kong/blob/2ed0fad88ed81623ea4afb2111033a9d87f3c03c/alpine/Dockerfile) (*alpine/Dockerfile*)
(<https://github.com/Kong/docker-kong/blob/2ed0fad88ed81623ea4afb2111033a9d87f3c03c/alpine/Dockerfile>)
- [0.11](https://github.com/Kong/docker-kong/blob/aa9ab06921a36079eb362229c9241ef9b3cf894e/centos/Dockerfile) , [0.11.2](https://github.com/Kong/docker-kong/blob/aa9ab06921a36079eb362229c9241ef9b3cf894e/centos/Dockerfile) (*centos/Dockerfile*) (<https://github.com/Kong/docker-kong/blob/aa9ab06921a36079eb362229c9241ef9b3cf894e/centos/Dockerfile>)

Quick reference

- **Where to get help:**

[the Docker Community Forums](https://forums.docker.com/) (<https://forums.docker.com/>), [the Docker Community Slack](https://blog.docker.com/2016/11/introducing-docker-community-directory-docker-community-slack/) (<https://blog.docker.com/2016/11/introducing-docker-community-directory-docker-community-slack/>), or [Stack Overflow](https://stackoverflow.com/search?tab=newest&q=docker) (<https://stackoverflow.com/search?tab=newest&q=docker>)

- **Where to file issues:**

<https://github.com/kong/kong/issues> (<https://github.com/kong/kong/issues>)

- **Maintained by:**

[the Kong Docker Maintainers](https://github.com/kong/kong) (<https://github.com/kong/kong>)

- **Supported architectures:** ([more info](https://github.com/docker-library/official-images#architectures-other-than-amd64) (<https://github.com/docker-library/official-images#architectures-other-than-amd64>))

[amd64](https://hub.docker.com/r/amd64/kong/) (<https://hub.docker.com/r/amd64/kong/>)

- **Published image artifact details:**

[repo-info](https://github.com/docker-library/repo-info/blob/master/repos/kong/) [repo's](https://github.com/docker-library/repo-info/blob/master/repos/kong/) [repos/](https://github.com/docker-library/repo-info/blob/master/repos/kong/) [kong/](https://github.com/docker-library/repo-info/blob/master/repos/kong/) [directory](https://github.com/docker-library/repo-info/blob/master/repos/kong/) (<https://github.com/docker-library/repo-info/blob/master/repos/kong/>) ([history](https://github.com/docker-library/repo-info/commits/master/repos/kong/) (<https://github.com/docker-library/repo-info/commits/master/repos/kong/>))
(image metadata, transfer size, etc)

- **Image updates:**

[official-images](https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fkong) [PRs](https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fkong) with label [library/kong](https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fkong) (<https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fkong>)
[official-images](https://github.com/docker-library/official-images/blob/master/library/kong) [repo's](https://github.com/docker-library/official-images/blob/master/library/kong) [library/](https://github.com/docker-library/official-images/blob/master/library/kong) [kong](https://github.com/docker-library/official-images/blob/master/library/kong) [file](https://github.com/docker-library/official-images/blob/master/library/kong) (<https://github.com/docker-library/official-images/blob/master/library/kong>) ([history](https://github.com/docker-library/official-images/commits/master/library/kong) (<https://github.com/docker-library/official-images/commits/master/library/kong>))

- **Source of this description:**

[docs](https://github.com/docker-library/docs/tree/master/kong) [repo's](https://github.com/docker-library/docs/tree/master/kong) [kong/](https://github.com/docker-library/docs/tree/master/kong) [directory](https://github.com/docker-library/docs/tree/master/kong) (<https://github.com/docker-library/docs/tree/master/kong>)
([history](https://github.com/docker-library/docs/commits/master/kong) (<https://github.com/docker-library/docs/commits/master/kong>))

- **Supported Docker versions:**

the latest release (<https://github.com/docker/docker-ce/releases/latest>) (down to 1.6 on a best-effort basis)

What is Kong?

Kong is a scalable, open source API Layer (also known as an API Gateway, or API Middleware). Kong was originally built by [Kong Inc. \(https://konghq.com\)](https://konghq.com) (formerly known as Mashape) to secure, manage and extend over 15,000 Microservices for its API Marketplace, which generates billions of requests per month.

Backed by the battle-tested NGINX with a focus on high performance, Kong was made available as an open-source platform in 2015. Under active development, Kong is now used in production at hundreds of organizations from startups, to large enterprises and government departments including: The New York Times, Expedia, Healthcare.gov, The Guardian, Condè Nast, The University of Auckland, Ferrari, and Giphy.

Kong's documentation can be found at getkong.org/docs (<http://getkong.org/docs>).



How to use this image

First, Kong requires a running Cassandra cluster (3.x+) or PostgreSQL instance (9.5+) before it starts. You can either use the official Cassandra/PostgreSQL containers, or use your own.

1. Link Kong to either a Cassandra or PostgreSQL container

It's up to you to decide which datastore between Cassandra or PostgreSQL you want to use, since Kong supports both.

Cassandra

Start a Cassandra container by executing:

```
$ docker run -d --name kong-database \  
              -p 9042:9042 \  
              cassandra:3
```

Postgres

Start a PostgreSQL container by executing:

```
docker run -d --name kong-database \  
    -p 5432:5432 \  
    -e "POSTGRES_USER=kong" \  
    -e "POSTGRES_DB=kong" \  
    postgres:9.5
```

2. Prepare your database

Run the database migrations with an ephemeral Kong container:

```
docker run --rm \  
    --link kong-database:kong-database \  
    -e "KONG_DATABASE=postgres" \  
    -e "KONG_PG_HOST=kong-database" \  
    -e "KONG_CASSANDRA_CONTACT_POINTS=kong-database" \  
    kong kong migrations up
```

In the above example, both Cassandra and PostgreSQL are configured, but you should update the `KONG_DATABASE` environment variable with either `cassandra` or `postgres`.

Note: migrations should never be run concurrently; only one Kong node should be performing migrations at a time.

Start Kong

Once the database has been started and prepared, we can start a Kong container and link it to the database container, and configuring the `KONG_DATABASE` environment variable with either `cassandra` or `postgres` depending on which database you decided to use:

```
$ docker run -d --name kong \  
    --link kong-database:kong-database \  
    -e "KONG_DATABASE=cassandra" \  
    -e "KONG_CASSANDRA_CONTACT_POINTS=kong-database" \  
    -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=0.0.0.0:8001" \  
    -e "KONG_ADMIN_LISTEN_SSL=0.0.0.0:8444" \  
    -p 8000:8000 \  
    -p 8443:8443 \  
    -p 8001:8001 \  
    -p 8444:8444 \  
    kong
```

If everything went well, and if you created your container with the default ports, Kong should be listening on your host's 8000 ([Proxy](http://getkong.org/docs/latest/configuration/#proxy_port) (http://getkong.org/docs/latest/configuration/#proxy_port)), 8443 ([Proxy SSL](http://getkong.org/docs/latest/configuration/#proxy_listen_ssl) (http://getkong.org/docs/latest/configuration/#proxy_listen_ssl)), 8001 ([Admin API](http://getkong.org/docs/latest/configuration/#admin_listen) (http://getkong.org/docs/latest/configuration/#admin_listen)) and 8444 ([Admin API SSL](http://getkong.org/docs/latest/configuration/#admin_listen_ssl) (http://getkong.org/docs/latest/configuration/#admin_listen_ssl)) ports.

You can now read the docs at getkong.org/docs (<http://getkong.org/docs>) to learn more about Kong.

3. Use Kong with a custom configuration (and a custom Cassandra/PostgreSQL cluster)

You can override any property of the [Kong configuration file](http://getkong.org/docs/latest/configuration/) (<http://getkong.org/docs/latest/configuration/>) with environment variables. Just prepend any Kong configuration property with the `KONG_` prefix, for example:

```
$ docker run -d --name kong \
  -e "KONG_DATABASE=postgres" \
  -e "KONG_PG_HOST=kong-database" \
  -e "KONG_LOG_LEVEL=info" \
  -e "KONG_CUSTOM_PLUGINS=helloworld" \
  -e "KONG_PG_HOST=1.1.1.1" \
  -e "KONG_ADMIN_LISTEN=0.0.0.0:8001" \
  -e "KONG_ADMIN_LISTEN_SSL=0.0.0.0:8444" \
  -p 8000:8000 \
  -p 8443:8443 \
  -p 8001:8001 \
  -p 8444:8444 \
  kong
```

Reload Kong in a running container

If you change your custom configuration, you can reload Kong (without downtime) by issuing:

```
$ docker exec -it kong kong reload
```

This will run the `kong reload` (<http://getkong.org/docs/latest/cli/#reload>) command in your container.

Image Variants

The `kong` images come in many flavors, each designed for a specific use case.

`kong:<version>`

This is the defacto image. If you are unsure about what your needs are, you probably want to use this one. It is designed to be used both as a throw away container (mount your source code and start the container to start your app), as well as the base to build other images off of.

kong:<version>-alpine

This image is based on the popular [Alpine Linux project \(http://alpinelinux.org\)](http://alpinelinux.org), available in the [alpine official image \(https://hub.docker.com/_/alpine\)](https://hub.docker.com/_/alpine). Alpine Linux is much smaller than most distribution base images (~5MB), and thus leads to much slimmer images in general.

This variant is highly recommended when final image size being as small as possible is desired. The main caveat to note is that it does use [musl libc \(http://www.musl-libc.org\)](http://www.musl-libc.org) instead of [glibc and friends \(http://www.etalabs.net/compare_libcs.html\)](http://www.etalabs.net/compare_libcs.html), so certain software might run into issues depending on the depth of their libc requirements. However, most software doesn't have an issue with this, so this variant is usually a very safe choice. See [this Hacker News comment thread \(https://news.ycombinator.com/item?id=10782897\)](https://news.ycombinator.com/item?id=10782897) for more discussion of the issues that might arise and some pro/con comparisons of using Alpine-based images.

To minimize image size, it's uncommon for additional related tools (such as `git` or `bash`) to be included in Alpine-based images. Using this image as a base, add the things you need in your own Dockerfile (see the [alpine image description \(https://hub.docker.com/_/alpine/\)](https://hub.docker.com/_/alpine/) for examples of how to install packages if you are unfamiliar).

License

View [license information \(https://getkong.org/license/\)](https://getkong.org/license/) for the software contained in this image.

As with all Docker images, these likely also contain other software which may be under other licenses (such as Bash, etc from the base distribution, along with any direct or indirect dependencies of the primary software being contained).

Some additional license information which was able to be auto-detected might be found in the [repo-info repository's kong/ directory \(https://github.com/docker-library/repo-info/tree/master/repos/kong\)](https://github.com/docker-library/repo-info/tree/master/repos/kong).

As for any pre-built image usage, it is the image user's responsibility to ensure that any use of this image complies with any relevant licenses for all software contained within.



```
docker pull kong
```