OFFICIAL REPOSITORY

ghost (/r/_/ghost/) ☆

Last pushed: 5 days ago

Short Description

Ghost is a free and open source blogging platform written in JavaScript

Full Description

Supported tags and respective `Dockerfile` links

- `2.6.1`, `2.6`, `2`, `latest` (*2/debian/Dockerfile*) (https://github.com/docker-library/ghost/blob/2dded8903da71e4a1bbc12cdc703c62416fedcc8/2/debian/Dockerfile)
- `2.6.1-alpine`, `2.6-alpine`, `2-alpine`, `alpine` (*2/alpine/Dockerfile*) (https://github.com/docker-library/ghost/blob/2dded8903da71e4a1bbc12cdc703c62416fedcc8/2/alpine/Dockerfile)
- `1.25.6`, `1.25`, `1` (*1/debian/Dockerfile*) (https://github.com/docker-library/ghost/blob/f1dde8050d6994a8df9564e50bffd4549c9c9305/1/debian/Dockerfile)
- `1.25.6-alpine`, `1.25-alpine`, `1-alpine` (*1/alpine/Dockerfile*) (https://github.com/docker-library/ghost/blob/94e82489166b6d44f590306402492df09bbfbcbd/1/alpine/Dockerfile)
- `0.11.14`, `0.11`, `0` (*0/debian/Dockerfile*) (https://github.com/docker-library/ghost/blob/f1dde8050d6994a8df9564e50bffd4549c9c9305/0/debian/Dockerfile)
- `0.11.14-alpine`, `0.11-alpine`, `0-alpine` (*0/alpine/Dockerfile*) (https://github.com/docker-library/ghost/blob/e1880db3c56f85410ad3342cbd33e7f98f24bcac/0/alpine/Dockerfile)

Quick reference

- **Where to get help**:
  the Docker Community Forums (https://forums.docker.com/), the 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)

- **Where to file issues**:
  https://github.com/docker-library/ghost/issues (https://github.com/docker-library/ghost/issues)

- **Maintained by**:
  the Docker Community (https://github.com/docker-library/ghost)

- **Supported architectures**: (more info (https://github.com/docker-library/official-images#architectures-other-than-amd64))
  `amd64` (https://hub.docker.com/r/amd64/ghost/), `arm32v6` (https://hub.docker.com/r/arm32v6/ghost/), `arm32v7` (https://hub.docker.com/r/arm32v7/ghost/), `arm64v8` (https://hub.docker.com/r/arm64v8/ghost/), `i386` (https://hub.docker.com/r/i386/ghost/), `ppc64le` (https://hub.docker.com/r/ppc64le/ghost/), `s390x` (https://hub.docker.com/r/s390x/ghost/)

- **Published image artifact details**:
  repo-info repo's `repos/ghost/` directory (https://github.com/docker-library/repo-info/blob/master/repos/ghost) (history (https://github.com/docker-library/repo-info/commits/master/repos/ghost))
  (image metadata, transfer size, etc)

- **Image updates**:
  official-images PRs with label `library/ghost` (https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fghost)
  official-images repo's `library/ghost` file (https://github.com/docker-library/official-images/blob/master/library/ghost) (history (https://github.com/docker-library/official-images/commits/master/library/ghost))

- **Source of this description**:
  docs repo's `ghost/` directory (https://github.com/docker-library/docs/tree/master/ghost) (history (https://github.com/docker-library/docs/commits/master/ghost))

- **Supported Docker versions**:
  the latest release (https://github.com/docker/docker-ce/releases/latest) (down to 1.6 on a best-effort basis)

Ghost

Ghost is a free and open source blogging platform written in JavaScript and distributed under the MIT License, designed to simplify the process of online publishing for individual bloggers as well as online publications.

> wikipedia.org/wiki/Ghost_(blogging_platform) (http://en.wikipedia.org/wiki/Ghost_%28blogging_platform%29)

How to use this image

This will start a Ghost instance listening on the default Ghost port of 2368.

```
$ docker run -d --name some-ghost ghost
```

Custom port

If you'd like to be able to access the instance from the host without the container's IP, standard port mappings can be used:

```
$ docker run -d --name some-ghost -p 3001:2368 ghost
```

Then, access it via `http://localhost:3001` or `http://host-ip:3001` in a browser.

Upgrading Ghost

You will want to ensure you are running the latest minor version (1.25.5 or 0.11.9) of Ghost before upgrading major versions. Otherwise, you may run into database errors.

For upgrading your Ghost container you will want to mount your data to the appropriate path in the predecessor container (see below): import your content from the admin panel, stop the container, and then re-mount your content to the successor container you are upgrading into; you can then export your content from the admin panel.

Stateful

Mount your existing content. In this example we also use the Alpine base image.

Ghost 1.x.x

```
$ docker run -d --name some-ghost -p 3001:2368 -v /path/to/ghost/blog:/var/lib/gho
```

Ghost 0.11.xx

```
$ docker run -d --name some-ghost -p 3001:2368 -v /path/to/ghost/blog:/var/lib/gho
```

Docker Volume

Alternatively you can use a [data container (http://docs.docker.com/engine/tutorials/dockervolumes/)](http://docs.docker.com/engine/tutorials/dockervolumes/) that has a volume that points to `/var/lib/ghost/content` (or /var/lib/ghost for 0.11.x) and then reference it:

```
$ docker run -d --name some-ghost --volumes-from some-ghost-data ghost
```

SQLite Database

This Docker image for Ghost uses SQLite. There is nothing special to configure.

Configuration

All Ghost configuration parameters (such as `url`) can be specified via environment variables. See [the Ghost documentation (https://docs.ghost.org/docs/config#section-running-ghost-with-config-env-variables)](https://docs.ghost.org/docs/config#section-running-ghost-with-config-env-variables) for details about what configuration is allowed and how to convert a nested configuration key into the appropriate environment variable name:

```
$ docker run -d --name some-ghost -e url=http://some-ghost.example.com ghost
```

(There are further configuration examples in the `stack.yml` listed below.)

What is the Node.js version?

When opening a ticket at [https://github.com/TryGhost/Ghost/issues (https://github.com/TryGhost/Ghost/issues)](https://github.com/TryGhost/Ghost/issues) it becomes necessary to know the version of Node.js in use:

```
$ docker exec <container-id> node --version
v6.11.2
```

Note about Ghost-CLI

While the Docker images do have Ghost-CLI available and do use some of its commands to set up the base Ghost image, many of the other Ghost-CLI commands won't work correctly, and really aren't designed/intended to. For more info see [docker-library/ghost#156 (comment) (https://github.com/docker-library/ghost/issues/156#issuecomment-428159861)](https://github.com/docker-library/ghost/issues/156#issuecomment-428159861)

... via [docker stack deploy (https://docs.docker.com/engine/reference/commandline/stack_deploy/)](https://docs.docker.com/engine/reference/commandline/stack_deploy/) or [docker-compose (https://github.com/docker/compose)](https://github.com/docker/compose)

Example `stack.yml` for `ghost`:

```
# by default, the Ghost image will use SQLite (and thus requires no separate datab
# we have used MySQL here merely for demonstration purposes (especially environment

version: '3.1'

services:

  ghost:
    image: ghost:1-alpine
    restart: always
    ports:
      - 8080:2368
    environment:
      # see https://docs.ghost.org/docs/config#section-running-ghost-with-config-en
      database__client: mysql
      database__connection__host: db
      database__connection__user: root
      database__connection__password: example
      database__connection__database: ghost

  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: example
```

[Try in PWD (http://play-with-docker.com?stack=https://raw.githubusercontent.com/docker-library/docs/e24f39cddf21560cf0a24f149059ff23640b0f16/ghost/stack.yml)](http://play-with-docker.com?stack=https://raw.githubusercontent.com/docker-library/docs/e24f39cddf21560cf0a24f149059ff23640b0f16/ghost/stack.yml)

Run `docker stack deploy -c stack.yml ghost` (or `docker-compose -f stack.yml up`), wait for it to initialize completely, and visit `http://swarm-ip:8080`, `http://localhost:8080`, or `http://host-ip:8080` (as appropriate).

Image Variants

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

ghost:<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.

ghost:<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) instead of glibc and friends (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) 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/) for examples of how to install packages if you are unfamiliar).

License

View license information (https://ghost.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 `ghost/` directory (https://github.com/docker-library/repo-info/tree/master/repos/ghost).

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 Command

```
docker pull ghost
```