Search for great content (e.g., mysql)   Explore   Repositories   c332030

redis ★

Docker Official Images

Redis is an open source key-value store that functions as a data structure server.

10M+

Container   Windows   Linux   386   ARM 64   ARM   x86-64   IBM Z   PowerPC 64 LE   Databases   Official Image

Linux - PowerPC 64 LE (latest)

Copy and paste to pull this image

```
docker pull redis
```

View Available Tags

DESCRIPTION   REVIEWS   TAGS

Supported tags and respective `Dockerfile` links

- `5.0.5`, `5.0`, `5`, `latest`, `5.0.5-stretch`, `5.0-stretch`, `5-stretch`, `stretch`
- `5.0.5-32bit`, `5.0-32bit`, `5-32bit`, `32bit`, `5.0.5-32bit-stretch`, `5.0-32bit-stretch`, `5-32bit-stretch`, `32bit-stretch`
- `5.0.5-alpine`, `5.0-alpine`, `5-alpine`, `alpine`, `5.0.5-alpine3.10`, `5.0-alpine3.10`, `5-alpine3.10`, `alpine3.10`
- `4.0.14`, `4.0`, `4`, `4.0.14-stretch`, `4.0-stretch`, `4-stretch`
- `4.0.14-32bit`, `4.0-32bit`, `4-32bit`, `4.0.14-32bit-stretch`, `4.0-32bit-stretch`, `4-32bit-stretch`
- `4.0.14-alpine`, `4.0-alpine`, `4-alpine`, `4.0.14-alpine3.10`, `4.0-alpine3.10`, `4-alpine3.10`

Quick reference

- **Where to get help**:
  the Docker Community Forums, the Docker Community Slack, or Stack Overflow
- **Where to file issues**:
  https://github.com/docker-library/redis/issues
- **Maintained by**:
  the Docker Community
- **Supported architectures**: (more info)
  `amd64`, `arm32v5`, `arm32v6`, `arm32v7`, `arm64v8`, `i386`, `ppc64le`, `s390x`
- **Published image artifact details**:
  repo-info repo's `repos/redis/` directory (history)
  (image metadata, transfer size, etc)
- **Image updates**:
  official-images PRs with label `library/redis`
  official-images repo's `library/redis` file (history)

- **Source of this description**:
  docs repo's `redis/` directory (history)

What is Redis?

Redis is an open-source, networked, in-memory, key-value data store with optional durability. It is written in ANSI C. The development of Redis is sponsored by Redis Labs today; before that, it was sponsored by Pivotal and VMware. According to the monthly ranking by DB-Engines.com, Redis is the most popular key-value store. The name Redis means REmote DIctionary Server.

> wikipedia.org/wiki/Redis

How to use this image

start a redis instance

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

start with persistent storage

```
$ docker run --name some-redis -d redis redis-server --appendonly yes
```

If persistence is enabled, data is stored in the `VOLUME /data`, which can be used with `--volumes-from some-volume-container` or `-v /docker/host/dir:/data` (see docs.docker volumes).

For more about Redis Persistence, see http://redis.io/topics/persistence.

connecting via `redis-cli`

```
$ docker run -it --network some-network --rm redis redis-cli -h some-redis
```

Additionally, If you want to use your own redis.conf ...

You can create your own Dockerfile that adds a redis.conf from the context into /data/, like so.

```
FROM redis
COPY redis.conf /usr/local/etc/redis/redis.conf
CMD [ "redis-server", "/usr/local/etc/redis/redis.conf" ]
```

Alternatively, you can specify something along the same lines with `docker run` options.

```
$ docker run -v /myredis/conf/redis.conf:/usr/local/etc/redis/redis.conf --name myredis redis redis-server /usr/local/etc/redis/redis.conf
```

Where `/myredis/conf/` is a local directory containing your `redis.conf` file. Using this method means that there is no need for you to have a Dockerfile for your redis container.

`32bit` variant

This variant is *not* a 32bit image (and will not run on 32bit hardware), but includes Redis compiled as a 32bit binary, especially for users who need the decreased memory requirements associated with that. See "Using 32 bit instances" in the Redis documentation for more information.

Redis Modules

You can find the list of modules for Redis on redis.io or on redismodules.com. A few of the standard modules can be found here:

- RediSearch: Search and Query with Indexing on Redis
- ReJSON: Extended JSON processing for Redis
- ReBloom: Bloom Filters data type for membership/existence search on Redis

Image Variants

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

`redis:<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.

Some of these tags may have names like stretch in them. These are the suite code names for releases of Debian and indicate which release the image is based on.

`redis:<version>-alpine`

This image is based on the popular Alpine Linux project, available in the `alpine` official image. 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 instead of glibc and friends, 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 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 for examples of how to install packages if you are unfamiliar).

License

View license information 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 `redis/` directory.

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.

Be the first to give insight into your experience by rating and reviewing the product.

Add Product Review

Select a product tier

Start Your Review