

OFFICIAL REPOSITORY

influxdb (/r/ /influxdb/) ☆

Last pushed: 8 days ago

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

Short Description

InfluxDB is an open source time series database for recording metrics, events, and analytics.

Full Description

Supported tags and respective Dockerfile links

- 1.5 , 1.5.4 ([influxdb/1.5/Dockerfile](#)) (<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.5/Dockerfile>)
- 1.5-alpine , 1.5.4-alpine ([influxdb/1.5/alpine/Dockerfile](#)) (<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.5/alpine/Dockerfile>)
- 1.5-data , 1.5.4-data ([influxdb/1.5/data/Dockerfile](#)) (<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.5/data/Dockerfile>)
- 1.5-data-alpine , 1.5.4-data-alpine ([influxdb/1.5/data/alpine/Dockerfile](#)) (<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.5/data/alpine/Dockerfile>)
- 1.5-meta , 1.5.4-meta ([influxdb/1.5/meta/Dockerfile](#)) (<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.5/meta/Dockerfile>)
- 1.5-meta-alpine , 1.5.4-meta-alpine ([influxdb/1.5/meta/alpine/Dockerfile](#)) (<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.5/meta/alpine/Dockerfile>)
- 1.6 , 1.6.4 ([influxdb/1.6/Dockerfile](#)) (<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.6/Dockerfile>)
- 1.6-alpine , 1.6.4-alpine ([influxdb/1.6/alpine/Dockerfile](#)) (<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.6/alpine/Dockerfile>)

- [1.6-data , 1.6.4-data , data \(influxdb/1.6/data/Dockerfile\)](https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.6/data/Dockerfile)
(<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.6/data/Dockerfile>)
- [1.6-data-alpine , 1.6.4-data-alpine , data-alpine \(influxdb/1.6/data/alpine/Dockerfile\)](https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.6/data/alpine/Dockerfile) (<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.6/data/alpine/Dockerfile>)
- [1.6-meta , 1.6.4-meta , meta \(influxdb/1.6/meta/Dockerfile\)](https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.6/meta/Dockerfile)
(<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.6/meta/Dockerfile>)
- [1.6-meta-alpine , 1.6.4-meta-alpine , meta-alpine \(influxdb/1.6/meta/alpine/Dockerfile\)](https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.6/meta/alpine/Dockerfile) (<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.6/meta/alpine/Dockerfile>)
- [1.7 , 1.7.1 , latest \(influxdb/1.7/Dockerfile\)](https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.7/Dockerfile) (<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.7/Dockerfile>)
- [1.7-alpine , 1.7.1-alpine , alpine \(influxdb/1.7/alpine/Dockerfile\)](https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.7/alpine/Dockerfile)
(<https://github.com/influxdata/influxdata-docker/blob/3702bf20d5efe2ff0662a8f27c3f7dd3d0aff750/influxdb/1.7/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/influxdata/influxdata-docker/issues>
(<https://github.com/influxdata/influxdata-docker/issues>)
- **Maintained by:**
InfluxData (<https://github.com/influxdata/influxdata-docker>)
- **Supported architectures:** (more info (<https://github.com/docker-library/official-images#architectures-other-than-amd64>))
[amd64 \(https://hub.docker.com/r/amd64/influxdb/\)](https://hub.docker.com/r/amd64/influxdb/), [arm32v7 \(https://hub.docker.com/r/arm32v7/influxdb/\)](https://hub.docker.com/r/arm32v7/influxdb/), [arm64v8 \(https://hub.docker.com/r/arm64v8/influxdb/\)](https://hub.docker.com/r/arm64v8/influxdb/)
- **Published image artifact details:**
[repo-info repo's repos/influxdb/ directory \(https://github.com/docker-library/repo-info/blob/master/repos/influxdb/\)](https://github.com/docker-library/repo-info/blob/master/repos/influxdb/) (history (<https://github.com/docker-library/repo-info/commits/master/repos/influxdb/>))
(image metadata, transfer size, etc)

- **Image updates:**

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

- **Source of this description:**

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

- **Supported Docker versions:**

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

InfluxDB

InfluxDB is a time series database built from the ground up to handle high write and query loads. InfluxDB is meant to be used as a backing store for any use case involving large amounts of timestamped data, including DevOps monitoring, application metrics, IoT sensor data, and real-time analytics.

[InfluxDB Documentation](https://docs.influxdata.com/influxdb/latest/) (<https://docs.influxdata.com/influxdb/latest/>)



Using this Image

Running the container

The InfluxDB image exposes a shared volume under `/var/lib/influxdb`, so you can mount a host directory to that point to access persisted container data. A typical invocation of the container might be:

```
$ docker run -p 8086:8086 \
  -v $PWD:/var/lib/influxdb \
  influxdb
```

Modify `$PWD` to the directory where you want to store data associated with the InfluxDB container.

You can also have Docker control the volume mountpoint by using a named volume.

```
$ docker run -p 8086:8086 \
-v influxdb:/var/lib/influxdb \
influxdb
```

Exposed Ports

The following ports are important and are used by InfluxDB.

- 8086 HTTP API port
- 8083 Administrator interface port, if it is enabled
- 2003 Graphite support, if it is enabled

The HTTP API port will be automatically exposed when using `docker run -P`.

The administrator interface is not automatically exposed when using `docker run -P` and is disabled by default. The administrator interface requires that the web browser have access to InfluxDB on the same port in the container as from the web browser. Since `-P` exposes the HTTP port to the host on a random port, the administrator interface is not compatible with this setting.

The administrator interface is deprecated as of 1.1.0 and will be removed in 1.3.0.

Find more about API Endpoints & Ports [here](https://docs.influxdata.com/influxdb/latest/concepts/api/)
(<https://docs.influxdata.com/influxdb/latest/concepts/api/>).

Configuration

InfluxDB can be either configured from a config file or using environment variables. To mount a configuration file and use it with the server, you can use this command:

Generate the default configuration file:

```
$ docker run --rm influxdb influxd config > influxdb.conf
```

Modify the default configuration, which will now be available under `$PWD`. Then start the InfluxDB container.

```
$ docker run -p 8086:8086 \
-v $PWD/influxdb.conf:/etc/influxdb/influxdb.conf:ro \
influxdb -config /etc/influxdb/influxdb.conf
```

Modify `$PWD` to the directory where you want to store the configuration file.

For environment variables, the format is `INFLUXDB_$SECTION_$NAME`. All dashes (`-`) are replaced with underscores (`_`). If the variable isn't in a section, then omit that part.

Examples:

```
INFLUXDB_REPORTING_DISABLED=true
INFLUXDB_META_DIR=/path/to/metadir
INFLUXDB_DATA_QUERY_LOG_ENABLED=false
```

Find more about configuring InfluxDB [here](https://docs.influxdata.com/influxdb/latest/introduction/installation/)
(<https://docs.influxdata.com/influxdb/latest/introduction/installation/>).

Graphite

InfluxDB supports the Graphite line protocol, but the service and ports are not exposed by default. To run InfluxDB with Graphite support enabled, you can either use a configuration file or set the appropriate environment variables. Run InfluxDB with the default Graphite configuration:

```
docker run -p 8086:8086 -p 2003:2003 \
  -e INFLUXDB_GRAPHITE_ENABLED=true \
  influxdb
```

See the [README on GitHub](https://github.com/influxdata/influxdb/blob/master/services/graphite/README.md)
(<https://github.com/influxdata/influxdb/blob/master/services/graphite/README.md>) for more detailed documentation to set up the Graphite service. In order to take advantage of graphite templates, you should use a configuration file by outputting a default configuration file using the steps above and modifying the `[[graphite]]` section.

Administrator Interface

The administrator interface is deprecated as of 1.1.0 and will be removed in 1.3.0. It is disabled by default. If needed, it can still be enabled by setting an environment variable like below:

```
docker run -p 8086:8086 -p 8083:8083 \
  -e INFLUXDB_ADMIN_ENABLED=true \
  influxdb
```

To use the administrator interface, both the HTTP API and the administrator interface API's must be forwarded to the same port.

HTTP API

Creating a DB named mydb:

```
$ curl -G http://localhost:8086/query --data-urlencode "q=CREATE DATABASE mydb"
```

Inserting into the DB:

```
$ curl -i -XPOST 'http://localhost:8086/write?db=mydb' --data-binary 'cpu,cpu=100'
```

Read more about this in the [official documentation](https://docs.influxdata.com/influxdb/latest/guides/writing_data/)
(https://docs.influxdata.com/influxdb/latest/guides/writing_data/)

CLI / SHELL

Start the container:

```
$ docker run --name=influxdb -d -p 8086:8086 influxdb
```

Run the influx client in this container:

```
$ docker exec -it influxdb influx
```

Or run the influx client in a separate container:

```
$ docker run --rm --link=influxdb -it influxdb influx -host influxdb
```

Database Initialization

The InfluxDB image contains some extra functionality for initializing a database. These options are not suggested for production, but are quite useful when running standalone instances for testing.

The database initialization script will only be called when running `influxd`. It will not be executed when running any other program.

Environment Variables

The InfluxDB image uses several environment variables to automatically configure certain parts of the server. They may significantly aid you in using this image.

INFLUXDB_DB

Automatically initializes a database with the name of this environment variable.

INFLUXDB_HTTP_AUTH_ENABLED

Enables authentication. Either this must be set or `auth-enabled = true` must be set within the configuration file for any authentication related options below to work.

INFLUXDB_ADMIN_USER

The name of the admin user to be created. If this is unset, no admin user is created.

INFLUXDB_ADMIN_PASSWORD

The password for the admin user configured with `INFLUXDB_ADMIN_USER`. If this is unset, a random password is generated and printed to standard out.

INFLUXDB_USER

The name of a user to be created with no privileges. If `INFLUXDB_DB` is set, this user will be granted read and write permissions for that database.

INFLUXDB_USER_PASSWORD

The password for the user configured with `INFLUXDB_USER`. If this is unset, a random password is generated and printed to standard out.

INFLUXDB_READ_USER

The name of a user to be created with read privileges on `INFLUXDB_DB` . If `INFLUXDB_DB` is not set, this user will have no granted permissions.

INFLUXDB_READ_USER_PASSWORD

The password for the user configured with `INFLUXDB_READ_USER` . If this is unset, a random password is generated and printed to standard out.

INFLUXDB_WRITE_USER

The name of a user to be created with write privileges on `INFLUXDB_DB` . If `INFLUXDB_DB` is not set, this user will have no granted permissions.

INFLUXDB_WRITE_USER_PASSWORD

The password for the user configured with `INFLUXDB_WRITE_USER` . If this is unset, a random password is generated and printed to standard out.

Initialization Files

If the Docker image finds any files with the extensions `.sh` or `.iql` inside of the `/docker-entrypoint-initdb.d` folder, it will execute them. The order they are executed in is determined by the shell. This is usually alphabetical order.

Manually Initializing the Database

To manually initialize the database and exit, the `/init-influxdb.sh` script can be used directly. It takes the same parameters as the `influxd run` command. As an example:

```
$ docker run --rm \
  -e INFLUXDB_DB=db0 -e INFLUXDB_ADMIN_ENABLED=true \
  -e INFLUXDB_ADMIN_USER=admin -e INFLUXDB_ADMIN_PASSWORD=supersecret \
  -e INFLUXDB_USER=telegraf -e INFLUXDB_USER_PASSWORD=secretpassword \
  -v $PWD:/var/lib/influxdb \
  influxdb /init-influxdb.sh
```

The above would create the database `db0` , create an admin user with the password `supersecretpassword` , then create the `telegraf` user with your telegraf's secret password. It would then exit and leave behind any files it created in the volume that you mounted.

Image Variants

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

influxdb:<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.

`influxdb:<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).

`influxdb:data`

This image requires a valid license key from InfluxData. Please visit our [products page \(https://www.influxdata.com/products/\)](https://www.influxdata.com/products/) to learn more.

This image contains the enterprise data node package for clustering. It supports all of the same options as the OSS image, but it needs port 8088 to be exposed to the meta nodes.

Refer to the `influxdb:meta` variant for directions on how to setup a cluster.

`influxdb:meta`

This image requires a valid license key from InfluxData. Please visit our [products page \(https://www.influxdata.com/products/\)](https://www.influxdata.com/products/) to learn more.

This image contains the enterprise meta node package for clustering. It is meant to be used in conjunction with the `influxdb:data` package of the same version.

Using this Image

Specifying the license key

The license key can be specified using either an environment variable or by overriding the configuration file. If you specify the license key directly, the container needs to be able to access the InfluxData portal.

```
$ docker run -p 8089:8089 -p 8091:8091 \  
    -e INFLUXDB_ENTERPRISE_LICENSE_KEY=<license-key> \  
    influxdb:meta
```

Running the container

The examples below will use docker's built-in networking capability. If you use the port exposing feature, the host port and the container port need to be the same.

First, create a docker network:

```
$ docker network create influxdb
```

Start three meta nodes. This is the suggested number of meta nodes. We do not recommend running more or less. If you choose to run more or less, be sure that the number of meta nodes is odd. The hostname must be set on each container to the address that will be used to access the meta node. When using docker networks, the hostname should be made the same as the name of the container.

```
$ docker run -d --name=influxdb-meta-0 --network=influxdb \  
    -h influxdb-meta-0 \  
    -e INFLUXDB_ENTERPRISE_LICENSE_KEY=<license-key> \  
    influxdb:meta  
$ docker run -d --name=influxdb-meta-1 --network=influxdb \  
    -h influxdb-meta-1 \  
    -e INFLUXDB_ENTERPRISE_LICENSE_KEY=<license-key> \  
    influxdb:meta  
$ docker run -d --name=influxdb-meta-2 --network=influxdb \  
    -h influxdb-meta-2 \  
    -e INFLUXDB_ENTERPRISE_LICENSE_KEY=<license-key> \  
    influxdb:meta
```

When setting the hostname, you can use `-h <hostname>` or you can directly set the environment variable using `-e INFLUXDB_HOSTNAME=<hostname>`.

After starting the meta nodes, you need to tell them about each other. Choose one of the meta nodes and run `influxd-ctl` in the container.

```
$ docker exec influxdb-meta-0 \  
    influxd-ctl add-meta influxdb-meta-1:8091  
$ docker exec influxdb-meta-0 \  
    influxd-ctl add-meta influxdb-meta-2:8091
```

Or you can just start a single meta node. If you setup a single meta node, you do not need to use `influxd-ctl add-meta`.

```
$ docker run -d --name=influxdb-meta --network=influxdb \
  -h influxdb-meta \
  -e INFLUXDB_ENTERPRISE_LICENSE_KEY=<license-key> \
  influxdb:meta -single-server
```

Connecting the data nodes

Start the data nodes using `influxdb:data` with similar command line arguments to the meta nodes. You can start as many data nodes as are allowed by your license.

```
$ docker run -d --name=influxdb-data-0 --network=influxdb \
  -h influxdb-data-0 \
  -e INFLUXDB_LICENSE_KEY=<license-key> \
  influxdb:data
```

You can add `-p 8086:8086` to expose the http port to the host machine. After starting the container, choose one of the meta nodes and add the data node to it.

```
$ docker exec influxdb-meta-0 \
  influxd-ctl add-data influxdb-data-0:8088
```

Perform these same steps for any other data nodes that you want to add.

You can now connect to any of the running data nodes to use your cluster.

See the [influxdb \(https://hub.docker.com/_/influxdb/\)](https://hub.docker.com/_/influxdb/) image documentation for more details on how to use the data node images.

Configuration

InfluxDB Meta can be either configured from a config file or using environment variables. To mount a configuration file and use it with the server, you can use this command:

Generate the default configuration file:

```
$ docker run --rm influxdb:meta influxd-meta config > influxdb-meta.
```

Modify the default configuration, which will now be available under `$PWD`. Then start the InfluxDB Meta container.

```
$ docker run \
  -v $PWD/influxdb-meta.conf:/etc/influxdb/influxdb-meta.conf:ro
  influxdb -config /etc/influxdb/influxdb-meta.conf
```

Modify `$PWD` to the directory where you want to store the configuration file.

For environment variables, the format is `INFLUXDB_$SECTION_$NAME` . All dashes (-) are replaced with underscores (_). If the variable isn't in a section, then omit that part.

Examples:

```
INFLUXDB_REPORTING_DISABLED=true
INFLUXDB_META_DIR=/path/to/metadir
INFLUXDB_ENTERPRISE_REGISTRATION_ENABLED=true
```

Find more about configuring InfluxDB Meta [here](#)

(http://docs.influxdata.com/enterprise_influxdb/latest/production_installation/meta_node_installation/).

License

View [license information \(https://github.com/influxdata/influxdb/blob/master/LICENSE\)](https://github.com/influxdata/influxdb/blob/master/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 influxdb/ directory \(https://github.com/docker-library/repo-info/tree/master/repos/influxdb\)](https://github.com/docker-library/repo-info/tree/master/repos/influxdb).

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 influxdb
```