OFFICIAL REPOSITORY

sentry (/r/_/sentry/)

Last pushed: 9 days ago

Short Description

Sentry is a realtime, platform-agnostic error logging and aggregation platform

Full Description

Supported tags and respective `Dockerfile` links

- `8.22.0` , `8.22` , `8` (*8.22/Dockerfile*) (https://github.com/getsentry/docker-sentry/blob/b7d55c572e0b23c5c537432211823a8677727e1c/8.22/Dockerfile)
- `8.22.0-onbuild` , `8.22-onbuild` , `8-onbuild` (*8.22/onbuild/Dockerfile*) (https://github.com/getsentry/docker-sentry/blob/b7d55c572e0b23c5c537432211823a8677727e1c/8.22/onbuild/Dockerfile)
- `9.0.0` , `9.0` , `9` , `latest` (*9.0/Dockerfile*) (https://github.com/getsentry/docker-sentry/blob/f648178adf33bec68be47550a82850e18deb0cd8/9.0/Dockerfile)
- `9.0.0-onbuild` , `9.0-onbuild` , `9-onbuild` , `onbuild` (*9.0/onbuild/Dockerfile*) (https://github.com/getsentry/docker-sentry/blob/ec474d97d968afdf78f4e489b58c68a2fbc46ee2/9.0/onbuild/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/getsentry/docker-sentry/issues (https://github.com/getsentry/docker-sentry/issues)

- **Maintained by**:
  Sentry (https://github.com/getsentry/docker-sentry)

- **Supported architectures**: (more info (https://github.com/docker-library/official-images#architectures-other-than-amd64))
  `amd64` (https://hub.docker.com/r/amd64/sentry/)

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

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

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

- **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 Sentry?

Sentry is a realtime event logging and aggregation platform. It specializes in monitoring errors and extracting all the information needed to do a proper post-mortem without any of the hassle of the standard user feedback loop.

> github.com/getsentry/sentry (https://github.com/getsentry/sentry)

How to use this image

How to setup a full Sentry instance

1. Start a Redis container

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

2. Start a Postgres container

   ```
   $ docker run -d --name sentry-postgres -e POSTGRES_PASSWORD=secr
   ```

3. Generate a new secret key to be shared by all `sentry` containers. This value will then be used as the `SENTRY_SECRET_KEY` environment variable.

   ```
   $ docker run --rm sentry config generate-secret-key
   ```

4. If this is a new database, you'll need to run `upgrade`

   ```
   $ docker run -it --rm -e SENTRY_SECRET_KEY='<secret-key>' --link
   ```

   Note: the `-it` is important as the initial upgrade will prompt to create an initial user and will fail without it

5. Now start up Sentry server

   ```
   $ docker run -d --name my-sentry -e SENTRY_SECRET_KEY='<secret-k
   ```

6. The default config needs a celery beat and celery workers, start as many workers as you need (each with a unique name)

   ```
    $ docker run -d --name sentry-cron -e SENTRY_SECRET_KEY='<secret
   $ docker run -d --name sentry-worker-1 -e SENTRY_SECRET_KEY='<sec
   ```

Port mapping

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. Just add `-p 8080:9000` to the `docker run` arguments and then access either `http://localhost:8080` or `http://host-ip:8080` in a browser.

Configuring the initial user

If you did not create a superuser during `upgrade` , use the following to create one:

```
$ docker run -it --rm -e SENTRY_SECRET_KEY='<secret-key>' --link se
```

Environment variables

When you start the `sentry` image, you can adjust the configuration of the Sentry instance by passing one or more environment variables on the `docker run` command line. Please note that these environment variables are provided as a jump start, and it's highly recommended to either mount in your own config file or utilize the `sentry:onbuild` variant.

`SENTRY_SECRET_KEY`

A secret key used for cryptographic functions within Sentry. This key should be unique and consistent across all running instances. You can generate a new secret key doing something like:

```
$ docker run --rm sentry config generate-secret-key
```

`SENTRY_POSTGRES_HOST`, `SENTRY_POSTGRES_PORT`, `SENTRY_DB_NAME`, `SENTRY_DB_USER`, `SENTRY_DB_PASSWORD`

Database credentials for your Postgres server. These values aren't needed if a linked `postgres` container exists.

`SENTRY_REDIS_HOST`, `SENTRY_REDIS_PORT`, `SENTRY_REDIS_DB`

Connection information for your Redis server. These values aren't needed if a linked `redis` container exists.

`SENTRY_MEMCACHED_HOST`, `SENTRY_MEMCACHED_PORT`

Connection information for a Memcache server. These values aren't needed if a linked `memcached` container exists.

`SENTRY_FILESTORE_DIR`

Directory where uploaded files will be stored. This defaults to `/var/lib/sentry/files` and is a `VOLUME` for persistent data.

`SENTRY_SERVER_EMAIL`

The email address used for `From:` in outbound emails. Default: `root@localhost`

`SENTRY_EMAIL_HOST`, `SENTRY_EMAIL_PORT`, `SENTRY_EMAIL_USER`, `SENTRY_EMAIL_PASSWORD`, `SENTRY_EMAIL_USE_TLS`

Connection information for an outbound smtp server. These values aren't needed if a linked `smtp` container exists.

`SENTRY_MAILGUN_API_KEY`

If you're using Mailgun for inbound mail, set your API key and configure a route to forward to `/api/hooks/mailgun/inbound/`.

Image Variants

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

`sentry:<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.

`sentry:onbuild`

This image makes it easy to custom build your own Sentry instance by copying in a custom `config.yml` and/or `sentry.conf.py` file and installing plugins from `requirements.txt`.

It's also possible to develop custom extensions within your `onbuild` package. If the build directory contains a `setup.py` file, this will also get installed.

See the official Sentry documentation (https://docs.getsentry.com/on-premise/server/installation/) for more information.

To create your custom `sentry:onbuild` package, simply do the following:

1. Create a Dockerfile containing `FROM sentry:onbuild`
2. In the same directory, add your custom configuration files.
3. You can get copies of those files to use as templates from the docker-sentry GitHub repo (https://github.com/getsentry/docker-sentry/).
4. Build your image: `docker build -t mysentry .`
5. Run your custom image using `mysentry` instead of `sentry`.

License

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

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 sentry
```