



OFFICIAL REPOSITORY

telegraf (/r/ /telegraf/) ☆

Last pushed: 9 days ago

Repo Info (/ /telegraf/)

Short Description

Telegraf is an agent for collecting metrics and writing them to InfluxDB or other outputs.

Full Description

Supported tags and respective Dockerfile links

- 1.6 , 1.6.4 (telegraf/1.6/Dockerfile) (<https://github.com/influxdata/influxdata-docker/blob/e90ed7bbdcd69305c615fe9be7903ef2e93ad978/telegraf/1.6/Dockerfile>)
- 1.6-alpine , 1.6.4-alpine (telegraf/1.6/alpine/Dockerfile) (<https://github.com/influxdata/influxdata-docker/blob/e90ed7bbdcd69305c615fe9be7903ef2e93ad978/telegraf/1.6/alpine/Dockerfile>)
- 1.7 , 1.7.4 (telegraf/1.7/Dockerfile) (<https://github.com/influxdata/influxdata-docker/blob/e90ed7bbdcd69305c615fe9be7903ef2e93ad978/telegraf/1.7/Dockerfile>)
- 1.7-alpine , 1.7.4-alpine (telegraf/1.7/alpine/Dockerfile) (<https://github.com/influxdata/influxdata-docker/blob/e90ed7bbdcd69305c615fe9be7903ef2e93ad978/telegraf/1.7/alpine/Dockerfile>)
- 1.8 , 1.8.3 , latest (telegraf/1.8/Dockerfile) (<https://github.com/influxdata/influxdata-docker/blob/e90ed7bbdcd69305c615fe9be7903ef2e93ad978/telegraf/1.8/Dockerfile>)
- 1.8-alpine , 1.8.3-alpine , alpine (telegraf/1.8/alpine/Dockerfile) (<https://github.com/influxdata/influxdata-docker/blob/e90ed7bbdcd69305c615fe9be7903ef2e93ad978/telegraf/1.8/alpine/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-](https://blog.docker.com/2016/11/introducing-docker-community-directory-docker-slack)

[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/influxdata/influxdata-docker/issues>
(<https://github.com/influxdata/influxdata-docker/issues>)

- **Maintained by:**

[InfluxData \(https://github.com/influxdata/influxdata-docker\)](https://github.com/influxdata/influxdata-docker)

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

[amd64 \(https://hub.docker.com/r/amd64/telegraf/\)](https://hub.docker.com/r/amd64/telegraf/), [arm32v7 \(https://hub.docker.com/r/arm32v7/telegraf/\)](https://hub.docker.com/r/arm32v7/telegraf/), [arm64v8 \(https://hub.docker.com/r/arm64v8/telegraf/\)](https://hub.docker.com/r/arm64v8/telegraf/)

- **Published image artifact details:**

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

- **Image updates:**

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

- **Source of this description:**

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

- **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)

Telegraf

Telegraf is an open source agent written in Go for collecting metrics and data on the system it's running on or from other services. Telegraf writes data it collects to InfluxDB in the correct format.



Using this image

Exposed Ports

- 8125 StatsD
- 8092 UDP
- 8094 TCP

Using the default configuration

The default configuration requires a running InfluxDB instance as an output plugin. Ensure that InfluxDB is running on port 8086 before starting the Telegraf container.

Minimal example to start an InfluxDB container:

```
$ docker run -d --name influxdb -p 8083:8083 -p 8086:8086 influxdb
```

Starting Telegraf using the default config, which connects to InfluxDB at `http://localhost:8086/`:

```
$ docker run --net=container:influxdb telegraf
```

Using a custom config file

First, generate a sample configuration and save it as `telegraf.conf` on the host:

```
$ docker run --rm telegraf telegraf config > telegraf.conf
```

Once you've customized `telegraf.conf`, you can run the Telegraf container with it mounted in the expected location:

```
$ docker run -v $PWD/telegraf.conf:/etc/telegraf/telegraf.conf:ro t
```

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

Read more about the Telegraf configuration [here](#)

[\(https://docs.influxdata.com/telegraf/latest/administration/configuration/\)](https://docs.influxdata.com/telegraf/latest/administration/configuration/).

Using the container with input plugins

These examples assume you are using a custom configuration file that takes advantage of Docker's built-in service discovery capability. In order to do so, we'll first create a new network:

```
$ docker network create influxdb
```

Next, we'll start our InfluxDB container named `influxdb` :

```
$ docker run -d --name=influxdb \  
  --net=influxdb \  
  influxdb
```

The `telegraf.conf` configuration can now resolve the `influxdb` container by name:

```
[[outputs.influxdb]]  
  urls = ["http://influxdb:8086"]
```

Finally, we start our Telegraf container and verify functionality:

```
$ docker run -d --name=telegraf \  
  --net=influxdb \  
  -v $PWD/telegraf.conf:/etc/telegraf/telegraf.conf:ro \  
  telegraf  
$ docker logs -f telegraf
```

Aerospike

Start an instance of aerospike:

```
$ docker run -d --name aerospike \  
  --net=influxdb \  
  -p 3000-3003:3000-3003 \  
  aerospike
```

Edit your Telegraf config file and set the correct connection parameter for Aerospike:

```
[[inputs.aerospike]]  
  servers = ["aerospike:3000"]
```

Restart your `telegraf` container to pick up the changes:

```
$ docker restart telegraf
```

Nginx

Create an `nginx_status.conf` configuration file to expose metric data:

```
server {  
    listen 8090;  
    location /nginx_status {  
        stub_status;  
        access_log off;  
    }  
}
```

Start an Nginx container utilizing it:

```
$ docker run -d --name=nginx \
  --net=influxdb \
  -p 8090:8090 -p 8080:80 \
  -v $PWD/nginx_status.conf:/etc/nginx/conf.d/nginx_status.conf
nginx
```

Verify the status page: http://localhost:8090/nginx_status
(http://localhost:8090/nginx_status).

Configure the nginx input plugin in your Telegraf configuration file:

```
[[inputs.nginx]]
  urls = ["http://nginx:8090/nginx_status"]
```

Restart your telegraf container to pick up the changes:

```
$ docker restart telegraf
```

StatsD

Telegraf has a StatsD plugin, allowing Telegraf to run as a StatsD server that metrics can be sent to. In order for this to work, you must first configure the [StatsD plugin](https://github.com/influxdata/telegraf/tree/master/plugins/inputs/statsd) (<https://github.com/influxdata/telegraf/tree/master/plugins/inputs/statsd>) in your config file.

Run Telegraf with the UDP port 8125 exposed:

```
$ docker run -d --name=telegraf \
  --net=influxdb \
  -p 8125:8125/udp \
  -v $PWD/telegraf.conf:/etc/telegraf/telegraf.conf:ro \
  telegraf
```

Send Mock StatsD data:

```
$ for i in {1..50}; do echo $i;echo "foo:1|c" | nc -u -w0 127.0.0.1
```

Check that the measurement `foo` is added in the DB.

Supported Plugins Reference

- [Input Plugins](https://docs.influxdata.com/telegraf/latest/plugins/inputs/) (<https://docs.influxdata.com/telegraf/latest/plugins/inputs/>)
- [Output Plugins](https://docs.influxdata.com/telegraf/latest/plugins/outputs/) (<https://docs.influxdata.com/telegraf/latest/plugins/outputs/>)

Monitoring the host filesystem

One of the more common use cases for Telegraf is running it in a container to monitor the host filesystem using the inputs that take information from the `/proc` filesystem. This section only applies to monitoring a Linux host.

To do this, you can mount the host's `/proc` filesystem inside of the container and set the location of `/proc` to an alternate location by using the `HOST_PROC` environment variable to change the location of where `/proc` is located. As an example:

```
$ docker run -d --name=telegraf \  
  --net=influxdb \  
  -e HOST_PROC=/host/proc \  
  -v /proc:/host/proc:ro \  
  -v $PWD/telegraf.conf:/etc/telegraf/telegraf.conf:ro \  
  telegraf
```

Monitoring docker containers

To monitor other docker containers, you can use the docker plugin and mount the docker socket into the container. An example configuration is below:

```
[[inputs.docker]]  
  endpoint = "unix:///var/run/docker.sock"
```

Then you can start the telegraf container.

```
$ docker run -d --name=telegraf \  
  --net=influxdb \  
  -v /var/run/docker.sock:/var/run/docker.sock \  
  -v $PWD/telegraf.conf:/etc/telegraf/telegraf.conf:ro \  
  telegraf
```

Refer to the docker [plugin documentation](#)

(<https://github.com/influxdata/telegraf/blob/master/plugins/inputs/docker/README.md>) for more information.

Image Variants

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

`telegraf:<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.

`telegraf:<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 \(https://github.com/influxdata/telegraf/blob/master/LICENSE\)](https://github.com/influxdata/telegraf/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 `telegraf/` directory \(https://github.com/docker-library/repo-info/tree/master/repos/telegraf\)](https://github.com/docker-library/repo-info/tree/master/repos/telegraf).

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 telegraf
```