Configure logging drivers

Estimated reading time: 6 minutes

Docker includes multiple logging mechanisms to help you get information from running containers and services (https://docs.docker.com/engine/admin/logging/view_container_logs/). These mechanisms are called logging drivers.

Each Docker daemon has a default logging driver, which each container uses unless you configure it to use a different logging driver.

In addition to using the logging drivers included with Docker, you can also implement and use logging driver plugins (https://docs.docker.com/engine/admin/logging/plugins/).

Configure the default logging driver

To configure the Docker daemon to default to a specific logging driver, set the value of `log-driver` to the name of the logging driver in the `daemon.json` file, which is located in `/etc/docker/` on Linux hosts or `C:\ProgramData\docker\config\` on Windows server hosts. The default logging driver is `json-file`. The following example explicitly sets the default logging driver to `syslog`:

```
{
  "log-driver": "syslog"
}
```

If the logging driver has configurable options, you can set them in the `daemon.json` file as a JSON array with the key `log-opts`. The following example sets two configurable options on the `json-file` logging driver:

```
{
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "10m",
    "max-file": "3",
    "labels": "production_status",
    "env": "os,customer"
  }
}
```

> **Note**: `log-opt` configuration options in the `daemon.json` configuration file must be provided as strings. Boolean and numeric values (such as the value for `max-file` in the example above) must therefore be enclosed in quotes (`"`).

If you do not specify a logging driver, the default is `json-file` . Thus, the default output for commands such as `docker inspect <CONTAINER>` is JSON.

To find the current default logging driver for the Docker daemon, run `docker info` and search for `Logging Driver` . You can use the following command:

```
$ docker info --format '{{.LoggingDriver}}'

json-file
```

Configure the logging driver for a container

When you start a container, you can configure it to use a different logging driver than the Docker daemon's default, using the `--log-driver` flag. If the logging driver has configurable options, you can set them using one or more instances of the `--log-opt <NAME>=<VALUE>` flag. Even if the container uses the default logging driver, it can use different configurable options.

The following example starts an Alpine container with the `none` logging driver.

```
$ docker run -it --log-driver none alpine ash
```

To find the current logging driver for a running container, if the daemon is using the `json-file` logging driver, run the following `docker inspect` command, substituting the container name or ID for `<CONTAINER>` :

```
$ docker inspect -f '{{.HostConfig.LogConfig.Type}}' <CONTAINER>

json-file
```

Configure the delivery mode of log messages from container to log driver

Docker provides two modes for delivering messages from the container to the log driver:

- (default) direct, blocking delivery from container to driver
- non-blocking delivery that stores log messages in an intermediate per-container ring buffer for consumption by driver

The `non-blocking` message delivery mode prevents applications from blocking due to logging back pressure. Applications are likely to fail in unexpected ways when STDERR or STDOUT streams block.

> **WARNING**: When the buffer is full and a new message is enqueued, the oldest message in memory is dropped. Dropping messages is often preferred to blocking the log-writing process of an application.

The `mode` log option controls whether to use the `blocking` (default) or `non-blocking` message delivery.

The `max-buffer-size` log option controls the size of the ring buffer used for intermediate message storage when `mode` is set to `non-blocking`. `max-buffer-size` defaults to 1 megabyte.

The following example starts an Alpine container with log output in non-blocking mode and a 4 megabyte buffer:

```
$ docker run -it --log-opt mode=non-blocking --log-opt max-buffer-size=4m
 alpine ping 127.0.0.1
```

Use environment variables or labels with logging drivers

Some logging drivers add the value of a container's `--env|-e` or `--label` flags to the container's logs. This example starts a container using the Docker daemon's default logging driver (let's assume `json-file`) but sets the environment variable `os=ubuntu`.

```
$ docker run -dit --label production_status=testing -e os=ubuntu alpine s
h
```

If the logging driver supports it, this adds additional fields to the logging output. The following output is generated by the `json-file` logging driver:

```
"attrs":{"production_status":"testing","os":"ubuntu"}
```

Supported logging drivers

The following logging drivers are supported. See the link to each driver's documentation for its configurable options, if applicable. If you are using logging driver plugins (https://docs.docker.com/engine/admin/logging/plugins/), you may see more options.

Driver	Description
`none`	No logs are available for the container and `docker logs` does not return any output.
`json-file` (https://docs.docker.com/config/containers/logging/json-file/)	The logs are formatted as JSON. The default logging driver for Docker.
`local` (https://docs.docker.com/config/containers/logging/local/)	Writes logs messages to local filesystem in binary files using Protobuf.
`syslog` (https://docs.docker.com/config/containers/logging/syslog/)	Writes logging messages to the `syslog` facility. The `syslog` daemon must be running on the host machine.

Driver	Description
`journald` (https://docs.docker.com/config/containers/logging/journald/)	Writes log messages to `journald`. The `journald` daemon must be running on the host machine.
`gelf` (https://docs.docker.com/config/containers/logging/gelf/)	Writes log messages to a Graylog Extended Log Format (GELF) endpoint such as Graylog or Logstash.
`fluentd` (https://docs.docker.com/config/containers/logging/fluentd/)	Writes log messages to `fluentd` (forward input). The `fluentd` daemon must be running on the host machine.
`awslogs` (https://docs.docker.com/config/containers/logging/awslogs/)	Writes log messages to Amazon CloudWatch Logs.
`splunk` (https://docs.docker.com/config/containers/logging/splunk/)	Writes log messages to `splunk` using the HTTP Event Collector.
`etwlogs` (https://docs.docker.com/config/containers/logging/etwlogs/)	Writes log messages as Event Tracing for Windows (ETW) events. Only available on Windows platforms.

Driver	Description
`gcplogs` (https://docs.docker.com/config/containers/logging/gcplogs/)	Writes log messages to Google Cloud Platform (GCP) Logging.
`logentries` (https://docs.docker.com/config/containers/logging/logentries/)	Writes log messages to Rapid7 Logentries.

Limitations of logging drivers

The `docker logs` command is not available for drivers other than `json-file` and `journald`.

docker (https://docs.docker.com/glossary/?term=docker), logging (https://docs.docker.com/glossary/?term=logging), driver (https://docs.docker.com/glossary/?term=driver)