Google Cloud Logging driver

- Usage
- gcplogs options

The Google Cloud Logging driver sends container logs to Google Cloud Logging Logging.

Usage

To use the `gcplogs` driver as the default logging driver, set the `log-driver` and `log-opt` keys to appropriate values in the `daemon.json` file. For more about configuring Docker using `daemon.json`, see daemon.json.

{{% include "daemon-cfg-desktop.md" %}}

The following example sets the log driver to `gcplogs` and sets the `gcp-meta-name` option.

```
{
  "log-driver": "gcplogs",
  "log-opts": {
    "gcp-meta-name": "example-instance-12345"
  }
}
```

Restart Docker for the changes to take effect.

You can set the logging driver for a specific container by using the `--log-driver` option to `docker run`:

```
$ docker run --log-driver=gcplogs ...
```

If Docker detects that it's running in a Google Cloud Project, it discovers configuration from the instance metadata service. Otherwise, the user must specify which project to log to using the `--gcp-project` log option and Docker attempts to obtain credentials from the Google Application

Default Credential. The `--gcp-project` flag takes precedence over information discovered from the metadata server, so a Docker daemon running in a Google Cloud project can be overridden to log to a different project using `--gcp-project`.

Docker fetches the values for zone, instance name and instance ID from Google Cloud metadata server. Those values can be provided via options if metadata server isn't available. They don't override the values from metadata server.

gcplogs options

You can use the `--log-opt NAME=VALUE` flag to specify these additional Google Cloud Logging driver options:

Option	Required	Description
`gcp-project`	optional	Which Google Cloud project to log to. Defaults to discovering this value from the Google Cloud metadata server.
`gcp-log-cmd`	optional	Whether to log the command that the container was started with. Defaults to false.
`labels`	optional	Comma-separated list of keys of labels, which should be included in message, if these labels are specified for the container.
`labels-regex`	optional	Similar to and compatible with `labels`. A regular expression to match logging-related labels. Used for advanced log tag options.
`env`	optional	Comma-separated list of keys of environment variables, which should be included in message, if these variables are specified for the container.
`env-regex`	optional	Similar to and compatible with `env`. A regular expression to match logging-related environment variables. Used for advanced log tag options.
`gcp-meta-zone`	optional	Zone name for the instance.
`gcp-meta-name`	optional	Instance name.
`gcp-meta-id`	optional	Instance ID.

If there is collision between `label` and `env` keys, the value of the `env` takes precedence. Both options add additional fields to the attributes of a logging message.

The following is an example of the logging options required to log to the default logging destination which is discovered by querying the Google Cloud metadata server.

```
$ docker run \
    --log-driver=gcplogs \
    --log-opt labels=location \
    --log-opt env=TEST \
    --log-opt gcp-log-cmd=true \
    --env "TEST=false" \
    --label location=west \
    your/application
```

This configuration also directs the driver to include in the payload the label `location`, the environment variable `ENV`, and the command used to start the container.

The following example shows logging options for running outside of Google Cloud. The `GOOGLE_APPLICATION_CREDENTIALS` environment variable must be set for the daemon, for example via systemd:

```
[Service]
Environment="GOOGLE_APPLICATION_CREDENTIALS=uQWVCPkMTI34bpssr1HI"
```

```
$ docker run \
    --log-driver=gcplogs \
    --log-opt gcp-project=test-project \
    --log-opt gcp-meta-zone=west1 \
    --log-opt gcp-meta-name=`hostname` \
    your/application
```

□□□ #2
□ □□□ □□□ 2026-10-31 19:24:36 UTC
□ □□□ □□□ 2026-10-31 19:24:38 UTC