Lab Log Files

Neha Singh
67
500069028

When you start a container, Docker will track the Standard Out and Standard Error outputs from the process and make them available via the client. Example In the background, there is an instance of Redis running with the name redis-server. Using the Docker client, we can access the standard out and standard error outputs using

```
$ docker run -d --name redis-server redis
b1a537729d5d22d449b30553e9c7b9cd4e579555d509a5ec7e28d435e1bfb30e
$
$
$ docker logs redis-server
1:C 03 May 10:03:47.587 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
1:C 03 May 10:03:47.588 # Redis version=4.0.11, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 03 May 10:03:47.588 # Warning: no config file specified, using the default config. In order to specify a
config file use redis-server /path/to/redis.conf
1:M 03 May 10:03:47.590 * Running mode=standalone, port=6379.
1:M 03 May 10:03:47.590 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/co
re/somaxconn is set to the lower value of 128.
1:M 03 May 10:03:47.591 # Server initialized
1:M 03 May 10:03:47.591 # WARNING overcommit_memory is set to 0! Background save may fail under low memory co
ndition. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the comm
and 'sysctl vm.overcommit_memory=1' for this to take effect.
1:M 03 May 10:03:47.591 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This
will create latency and memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/
kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain the sett
ing after a reboot. Redis must be restarted after THP is disabled.
1:M 03 May 10:03:47.591 * Ready to accept connections
$ docker run -d --name redis-syslog --log-driver=syslog redis
adc9a2fd9db05282852ba35a3a979d92a3b7ef07dcff748afee100f94c9246de
```

Syslog The Syslog log driver will write all the container logs to the central syslog on the host. "syslog is a widely used standard for message logging. It permits separation of the software that generates messages, the system that stores them, and the software that reports and analyses them." Wikipedia This log-driver is designed to be used when syslog is being collected and aggregated by an external system. Example The command below will redirect the redis logs to syslog.

```
$ docker run -d --name redis-syslog --log-driver=syslog redis
adc9a2fd9db05282852ba35a3a979d92a3b7ef07dcff748afee100f94c9246de
```

Step 3 - Disable Logging The third option is to disable logging on the container. This is particularly useful for containers which are very verbose in their logging. Example When the container is launched simply set the log-driver to none. No output will be logged.

```
$ docker run -d --name redis-syslog --log-driver=syslog redis
adc9a2fd9db05282852ba35a3a979d92a3b7ef07dcff748afee100f94c9246de
```

Which Config? The inspect command allows you to identify the logging configuration for a particular container. The command below will output the LogConfig section for each of the containers. Server created in step 1

```
$ docker run -d --name redis-none --log-driver=none redis
63b14ea90eb5e936b0644bd5862c610a0aea7daa726995bbd2649eda65004eeb
```

Server created in step 2

```
$ docker inspect --format '{{ .HostConfig.LogConfig }}' redis-syslog
{syslog map[]}
```

Server created in this step

```
$ docker inspect --format '{{ .HostConfig.LogConfig }}' redis-none
{none map[]}
```