**Name:- Abhishek Singh**

Roll.No:- R171218120

SAP\_ID:- 500067726

Semester:-6

**EXPERIMENT 11**

**Managing Logs**

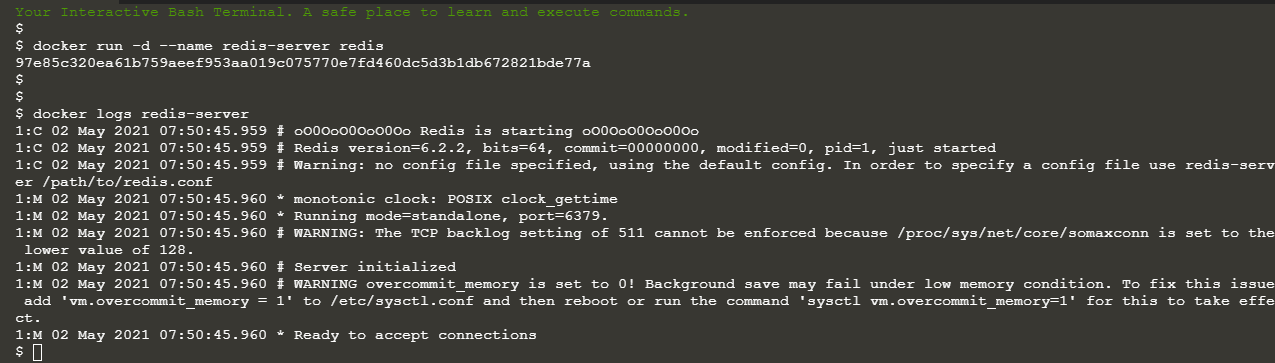
**Step 1 - Docker Logs**

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 logs redis-server



#### Step 2 - SysLog

By default, the Docker logs are outputting using the json-file logger meaning the output stored in a JSON file on the host. This can result in large files filling the disk. As a result, you can change the log driver to move to a different destination.

#### 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](https://en.wikipedia.org/wiki/Syslog)

This log-driver is designed to be used when syslog is being collected and aggregated by an external system.

#### Example

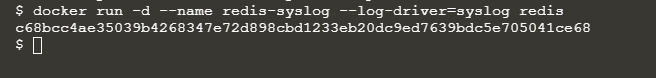
The command below will redirect the redis logs to syslog.

docker run -d --name redis-syslog --log-driver=syslog redis

#### Accessing Logs

If you attempted to view the logs using the client you'll recieve the error FATA[0000] "logs" command is supported only for "json-file" logging driver

Instead, you need to access them via the syslog stream.



#### 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-none --log-driver=none redis

#### 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 inspect --format '{{ .HostConfig.LogConfig }}' redis-server

Server created in step 2

docker inspect --format '{{ .HostConfig.LogConfig }}' redis-syslog

Server created in this step

docker inspect --format '{{ .HostConfig.LogConfig }}' redis-none

