Docker Dockerizing Node.js application

Step 1: 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.

Using the Docker client, we can access the standard out and standard error outputs using docker logs redis-server



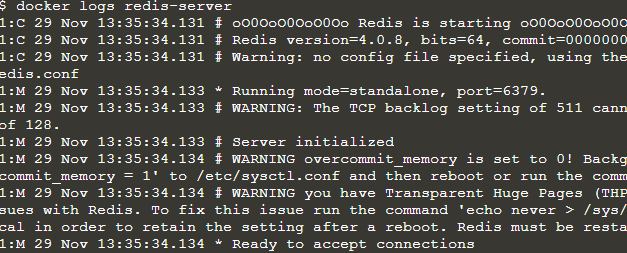
Step 2: Syslog

By default, the Docker logs are outputted using the *json-file* logger meaning the output is 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.

Example:

The command below will redirect the redis logs to syslog.

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



Step 3: Disable 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



### **ring Node.js application**