

Hands-on L3: Containers with Docker

Git hub url: <https://github.com/yaminibuyya27/hands-on-l3-docker>.

This hands-on assignment demonstrates installation of Docker Desktop and deployment of a multi-container microservice application using Flask and Redis. Docker Compose was used to manage container communication and execution.

Docker desktop app showing the containers

The screenshot shows the Docker Desktop application interface. The top bar is blue with the Docker logo, 'docker desktop PERSONAL', a search bar, and various icons including a bell with a red notification badge, a mail icon, a settings gear, and a 'Sign in' button. Below the top bar is a dark blue sidebar with navigation options: Containers, Images, Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area displays the 'docker-assignment-handson' project, which is located at '/Users/sahasrayamini/docker-assignment-handson'. It shows two containers: 'web' (based on 'docker-assignment-handson:8000-5000') and 'redis' (based on 'redis:alpine'). To the right of the container list is a log viewer showing the output of the containers. The logs for 'redis' show Redis initialization and configuration. The logs for 'web' show the Flask application starting and listening on port 5000. At the bottom of the interface, a status bar indicates 'Engine running', system resources (RAM 2.05 GB, CPU 0.25%, Disk 2.42 GB used), and a terminal icon with the version 'v4.40.0'.

```
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.455 * <ReJSON> Exported RedisJSON_V3 API
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.455 * <ReJSON> Exported RedisJSON_V4 API
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.455 * <ReJSON> Exported RedisJSON_V5 API
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.455 * <ReJSON> Exported RedisJSON_V6 API
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.455 * <ReJSON> Enabled diskless replication
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.455 * <ReJSON> Initialized shared string cache
, thread safe: true.
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.455 * Module 'ReJSON' loaded from /usr/local/l
ib/redis/modules/rejson.so
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.455 * <search> Acquired RedisJSON_V6 API
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.459 * Server initialized
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.459 * <search> Loading event starts
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.459 * <search> Changing workers threadpool siz
e from 0 to 4
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.459 * <search> Enabled workers threadpool of s
ize 4
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.459 * Loading RDB produced by version 8.4.0
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.459 * RDB age 3302 seconds
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.459 * RDB memory usage when created 1.12 Mb
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.459 * Done loading RDB, keys loaded: 1, keys e
xplred: 0.
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.459 * <search> Changing workers threadpool siz
e from 4 to 0
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.459 * <search> Disabled workers threadpool of
size 4
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.459 * <search> Loading event ends
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.459 * DB loaded from disk: 0.000 seconds
2026-02-03 17:07:15 redis-1 | 1:M 03 Feb 2026 22:07:15.459 * Ready to accept connections tcp
2026-02-03 17:07:15 web-1 | * Serving Flask app 'app.py'
2026-02-03 17:07:15 web-1 | * Debug mode: off
2026-02-03 17:07:15 web-1 | WARNING: This is a development server. Do not use it in a production de
ployment. Use a production WSGI server instead.
2026-02-03 17:07:15 web-1 | * Running on all addresses (0.0.0.0)
2026-02-03 17:07:15 web-1 | * Running on http://127.0.0.1:5000
2026-02-03 17:07:15 web-1 | * Running on http://172.18.0.3:5000
2026-02-03 17:07:15 web-1 | Press CTRL+C to quit
```

Output obtained after the handson:

