

Instituto Superior de Engenharia de Lisboa  
BEng in Computer Science and Engineering  
System Virtualization Techniques, Autumn/Winter 2023/2024  
Fourth coursework assignment

---

For this assignment, we will use the same web application as in the third coursework assignment to achieve the same general goals, but this time with a solution based on Docker containers, orchestrated via Docker Compose.

In the end, we expect a solution with the following components:

- One container with a single-node *elasticsearch* (8.11.1)
- One or more containers running the *tvapp* web application on Node.js
  - The number of containers running *tvapp* replicas shall be adjustable in runtime
- One container running a load balancer (*nginx*), distributing incoming requests among *tvapp* replicas.

## Exercises

1. Create a Dockerfile under `cw4/tvapp` to produce a Docker image with the following characteristics:
  - Base image: `node:alpine`
  - Working directory: `/home/node/app`
  - Web application location: `/home/node/app` (`package.json`, `tvapp.js`, `files/*`)
  - `npm install` executed for `/home/node/app/package.json`
  - All files and directories under `/home/node` must belong to `node:node`
  - Environment variable `NODE_PORT` defined to `(4000 + your group number)` and that port *exposed*.
  - Define `USER` as `node` and set `CMD` to run `node tvapp.js`
  - Reduce the number of layers but allow for reuse of the build cache when `tvapp.js` is modified.

Produce the `tvapp` image from this Dockerfile and use `docker run` to get four containers in execution, making the application available in ports 4001, 4002, 4003, and 4004 of the host system.

Be ready to explain the values observed for **HOST** and **PORT** in the four instances.

Tag this exercise on the GitHub repository with: **CW4-1**

2. Create a `docker-compose.yml` file in `cw4` with the following characteristics:
  - Three services: **entry**, **webapp**, and **datastore**
  - Service **entry** uses an `nginx:alpine` image with a modified configuration file (a template is available at the end of this document), located at `/etc/nginx/conf.d/default.conf`
  - Service **webapp** is built directly from the Dockerfile developed in exercise 1.
  - Service **datastore** is an `elasticsearch:8.11.1`, with the these environment variables defined:
    - `discovery.type=single-node`
    - `xpack.security.enabled=false`
  - Use a **volume** to ensure that `elasticsearch` persistent data is not lost between executions. The relevant directory is `/usr/share/elasticsearch/data`
  - Ensure that **entry** **can** access **webapp** and **webapp** **can** access **datastore**, but **entry** **cannot** access **datastore**
  - The solution will use a single **port** in the host system for all incoming requests: **2023**

Tag this exercise on the GitHub repository with: **CW4-2**

In the end, you must be able to **run the solution** with:

```
docker compose up -d
```

and you must also be able to **adjust the number of *webapp* replicas**, at any time, using the `--scale` option of `docker compose up` without interfering with the rest of the solution, that must keep working during the reconfiguration.

```
docker compose up -d --scale webapp=4 --no-recreate
```

Confirm that the solution is **using all the replicas** to process incoming requests. You may use the following command for that:

```
seq 32 | xargs -I{} curl -s http://localhost:2023/ | grep "HOST" |  
sed "s/<\|\/\|[a-z]\+>\/g" | sed "s/^\[:space:]*\/" | sort | uniq -c
```

You are **also expected** to:

- Check service logs for the solution
- Run a shell in any container in the solution to get its IP address and observe the running processes
- Demonstrate proper connectivity and unreachability between containers in the solution
- Explain and show why <http://webapp:PORT> is enough on `nginx` to reach all the replicas
- Explain the purpose of `resolver 127.0.0.11 valid=5s` in the `nginx` configuration

**Do not submit binaries and other unneeded files to the repository.**

For the absolute final version, use the tag **CW4-DONE** on the GitHub repository.

ISEL, December 6<sup>th</sup>, 2023

Submission last date: December 20<sup>th</sup>, 2023

---

Template for the `nginx` configuration file:

```
server {  
    listen 80;  
  
    location / {  
        set $TVSSVC http://webapp:4000;  
        proxy_pass $TVSSVC;  
        resolver 127.0.0.11 valid=5s;  
    }  
}
```