

Instituto Superior de Engenharia de Lisboa
BEng in Computer Science and Engineering
System Virtualization Techniques, Autumn/Winter 2024/2025
Fourth coursework assignment

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

The solution will include the following components:

- A container running a single-node *Elasticsearch* instance (8.16.1)
- One or more containers running the *tv sapp* web application on Node.js
 - The number of containers running *tv sapp* replicas is adjustable at runtime
- A container running a load balancer (*nginx*), distributing incoming requests among the *tv sapp* replicas.

Exercises

1. Create a Dockerfile under `cw4/tv sapp` 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`, `tv sapp.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 `(8000 + your group number)` and that port *exposed*.
 - Define `USER` as `node` and configure the image to always run `node tv sapp.js`
 - Reduce the number of layers but allow for reuse of the build cache when `tv sapp.js` is modified.

Produce the *tv sapp* image from this Dockerfile and use `docker run` to execute four containers, making the application available in ports 12001, 12002, 12003, and 12004 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** [Reference date: December 4th, 2024]

2. Create a `docker-compose.yml` file in `cw4` with the following characteristics:
 - Three services: **entry**, **webapp**, and **storage**
 - 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 **storage** is an `elasticsearch:8.16.1`, 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 **storage**, while **entry** **cannot** access **storage**
 - The solution will use a single **port** in the host system for all incoming requests: **2024**

Tag this exercise on the GitHub repository with: **CW4-2** [Reference date: December 11th, 2024]

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

```
docker compose up -d
```

You must also be able to **adjust the number of *webapp* replicas** at any time, using the `--scale` option with `docker compose up` without affecting the rest of the solution, which is expected to continue running 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 can use the following command for that:

```
seq 32 | xargs -I{} curl -s http://localhost:2024/ | grep "HOST" |  
sed "s/<\/\?[a-z]\+>\/g" | sed "s/^[[:space:]]*\/" | sort | uniq -c
```

Additionally, you are **expected** to be able 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, November 29th, 2024

Submission last date: December 14th, 2024

Template for the nginx configuration file:

```
server {  
    listen 80;  
  
    location / {  
        set $TVSSVC http://webapp:8000;  
        proxy_pass $TVSSVC;  
        resolver 127.0.0.11 valid=5s;  
    }  
}
```