Docker Microservices

TP2

2º Semestre 2019/2020

Gestão e Virtualização de Redes

4º Ano MIEI

Fábio Gonçalves

Computer Communications and Networks

Departamento de Informática, Universidade do Minho

The present work consists in the implementation of several microservices in docker containers. These should be able to follow a set of rules that define how they can interact with each other. The main services to be implemented are an authentication service and two file servers. The authentication service should work similarly to Oauth, a simplified version. This service should, after correct user authentication, provide a token. This token can be then used to access the file servers and download or upload files. The authorization server should keep the generated tokens in the database.

Upon receiving the user token, the file server should, though the internal network, verify if it is valid.

The docker data should be persisted using docker volumes.

All the applications installed in the containers should be created using docker files. These should be turned into automated builds and put in docker hub. The final goal is to have a docker-compose.yml that can be executed in any computer and reproduce the architecture.

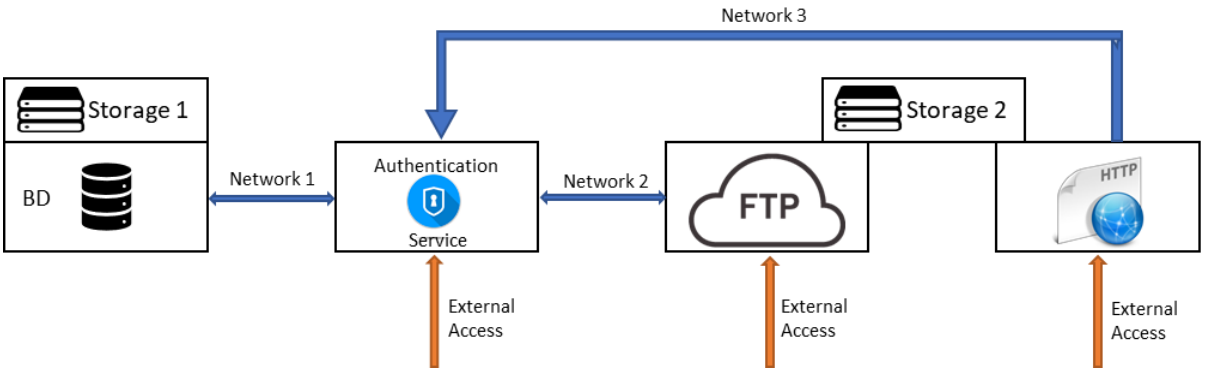

Figure 1 – Architecture

Phase 1

In the first phase, the students should implement:

- The authentication service should be able to communicate with BD;
- The FTP and HTTP services should be able to communicate with the authentication service;
- The images used can be chosen by the students with, these can be pre-built images from docker-hub;
- The external access can only be done though the authentication service, FPT server or HTTP file server;
- The BD service has its own persistence;
- The FTP and HTTP file server share their file system;

Phase 2

- Change the FTP server to a modified one that:
    - Instead of username and password authentication receives an authentication token;
    - Verifies the token with the authentication service;
- Change the HTTP file server to a modified one that:
    - Instead of having any kind of authentication, redirects the user to the authentication service to be authenticated (Oauth like);
    - Verifies the token;

Phase 3

- Use a driver to map a docker volume into a host path;
- Create a new container with a reverse proxy, that allows all services to be accessed in the same port but, with different paths;
- Use some service to generate certificates (letsencrypt, for example) and enable https;
- Add bind mounts for log files to all the containers;

Phase 4

- The authentication server allows to define roles;
- Define permissions according to roles;

Evaluation

The evaluation of the work will have two components, a written report and a demo. Each group should submit the report with all the configuration files, docker-compose and code in a zip file via elearning. The report should be submitted until 26 April.