

Familiarização com Docker

TP1- Parte 2

2º Semestre 2017/2018

Gestão e Virtualização de Redes

4º ano MIEI

Fábio Gonçalves

Computer Communications and Networks

Departamento de Informática, Universidade do Minho

A segunda fase deste trabalho tem como objetivo a obtenção dos conhecimentos necessários para a criação de volumes Docker que poderão ser utilizados para a partilha de dados entre os diversos containers. Esta fase termina com a elaboração de um trabalho usando as ferramentas estudadas.

1. Contexto e experimentação

Volumes Docker

O Docker permite a criação de volumes para poder gerir e persistir os conteúdos de cada container associado. Na primeira parte do trabalho foi estudado o mapeamento diretamente no host. Nesta fase, o armazenamento e persistência de dados será feito com a ajuda dos volumes Docker.

Alguns comandos básicos para usar volumes Docker estão listados de seguida:

- **docker volume create <my-vol-name>**: criar um volume com o respetivo nome;
- **docker volume ls**: listar os volumes existentes;
- **docker volume inspect <my-vol-name>**: ver informações sobre um determinado volume;
- **docker volume rm <my-vol-name>**: remover um volume.

Criar um container com um volume associado

- Criar um volume docker:
 - **docker volume create db-vol**
- Criar o container com o volume associado:
 - **docker run -d -p 5432:5432 --name db --mount source=db-vol \ target=/var/lib/postgresql/data postgres:8.4**
- Experimentar criar uma nova base de dados (por defeito a palavra passe e o nome de utilizador é postgresql), apagar o container e voltar a criar, de forma a verificar se os dados persistiram;
- Usar **docker volume inspect** para ver informações do volume criado;

Criar um container com um volume associado usando

Para este passo basta descarregar o docker-compose.yml fornecido, conclui-lo e executar o comando **docker-compose up -d**.

Deverão executar os mesmos passos feitos no exercício anterior (parar o container, remover o container e iniciar novamente), para verificar a persistência dos dados e inspecionar as informações dos volumes.

2. Exercício Prático

O presente exercício prático consiste na implementação de um serviço de autenticação e um serviço de email. O serviço de autenticação deverá funcionar de forma semelhante ao OAuth (de uma forma muito simplificada), onde o utilizador se deverá autenticar (utilizando, por exemplo, nome de utilizador e palavra passe), de forma a obter um token. Este pode ser utilizado para, posteriormente, aceder ao serviço de email. No mínimo, o serviço de autenticação deverá guardar os tokens gerados na base de dados.

O serviço de envio de mails, utilizará o token recebido para, através da rede interna (Network 2), confirmar com o servidor de autenticação se este é válido.

Para este exercício, deverão ser criados diversos containers docker, os quais deverão conseguir comunicar entre si. A persistência dos dados deverá ser feita usando volumes docker. A forma como os serviços, armazenamento e redes deverão ser configurados, está representada na Figura 1.

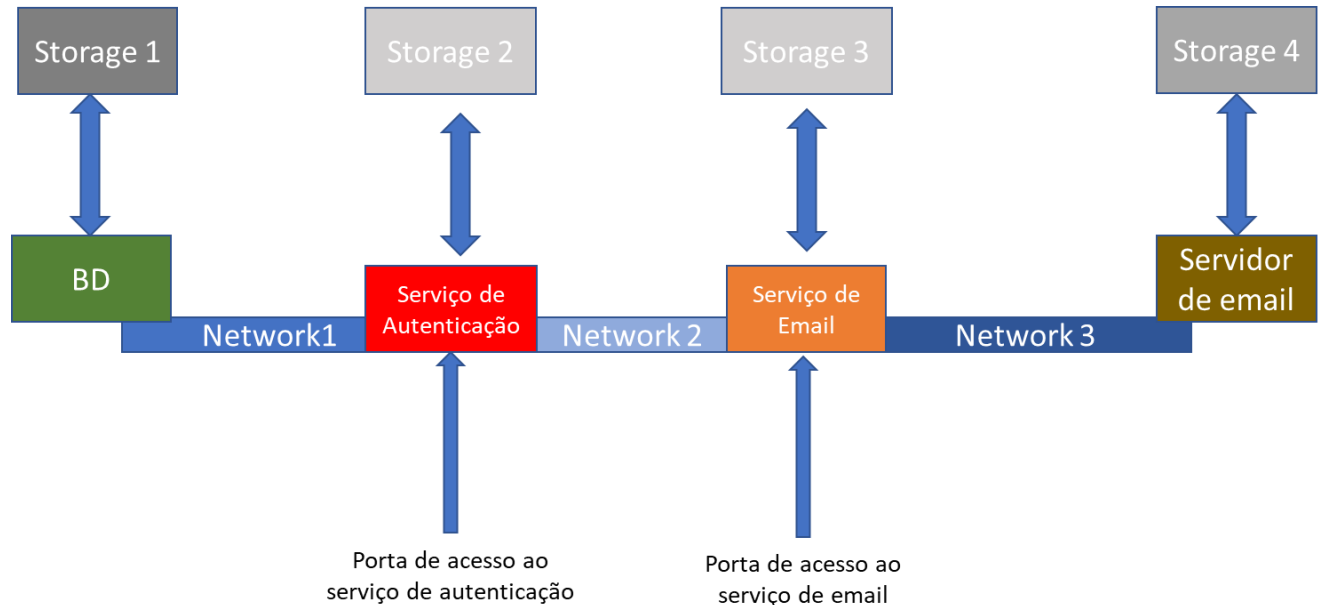


Figura 1 – Esquema

- O serviço de autorização deverá ser capaz de comunicar com a base de dados para inserir e obter dados;
- O serviço de email deverá ser capaz de comunicar com o servidor de email de forma a efetuar o envio de emails;
- Os serviços anteriormente descritos poderão ser implementados na linguagem de programação à escolha dos alunos. No entanto, sugere-se a utilização de php (pode ser utilizada uma configuração semelhante à estudada na parte 1)

- A BD e o servidor de email são à escolha dos alunos e poderão ser utilizadas imagens pré construídas existentes no docker-hub. Sugere-se a utilização da imagem oficial de postgres existente no docker hub para a base de dados e a imagem namshi/smtp;
- O serviço de mail deverá ser capaz de comunicar com o serviço de autorização de forma a verificar a veracidade dos tokens;
- O serviço de mail deverá ser configurado de forma a enviar emails com o domínio vr-<Nome_do_Grupo>.gcom.di.uminho.pt;
- O acesso exterior pode ser feito apenas pelos serviços de autenticação e de email;
- A persistência deverá ser feita através do uso de volumes docker;
- Os serviços deverão ser implementados usando um docker-compose.yml

Além do exercício proposto, podem ainda ser implementados, para valorização extra do TP, os seguintes pontos:

- Mapear um volume docker numa diretoria à escolha;
- Criar um novo container com um serviço (haproxy, nginx, etc) que permita aceder ao serviço 1 e serviço 2 na mesma porta mas com diferentes caminhos:
 - Por exemplo <http://host:port/servico1> e <http://host:port/servico2>
 - Este serviço apenas poderá comunicar com o serviço 1 e serviço 2
- Usar um serviço para gerar certificados (letsencrypt) que poderão ser usados para poder usar https para aceder ao host;

3. Avaliação

A avaliação terá duas componentes, um relatório e uma demo. Cada grupo deve realizar o relatório do TP e submeter um ficheiro comprimido com os diversos ficheiros de configuração, código elaborado e docker-compose.yml via plataforma de elearning. O relatório deverá ser entregue até dia 9 de Março.

A demo deverá ser feita no inicio da aula do dia 8 de Março.