



Início

Resumo: Este documento é um exercício relacionado à administração do sistema.

Versão: 2

Conteúdo

I	Preâmbulo	2
II	Introdução	3
III	Diretrizes gerais	4
IV	Parte obrigatória	5
V	Parte do bônus	9
VI	Envio e avaliação por pares	10

Capítulo I

Preâmbulo



Capítulo II

Introdução

Este projeto tem o objetivo de ampliar seu conhecimento sobre administração de sistemas usando o Docker. Você virtualizará várias imagens do Docker, criando-as em sua nova máquina virtual pessoal.

Capítulo III

Diretrizes gerais

- Esse projeto precisa ser realizado em uma máquina virtual.
- Todos os arquivos necessários para a configuração do seu projeto devem ser colocados em um arquivo `srcs` pasta.
- Um Makefile também é necessário e deve estar localizado na raiz do seu diretório. Ele deve configurar todo o seu aplicativo (ou seja, deve criar as imagens do Docker usando o `docker-compose.yml`).
- Esse assunto exige que você coloque em prática conceitos que, dependendo da sua formação, talvez ainda não tenha aprendido. Portanto, recomendamos que você não hesite em ler muita documentação relacionada ao uso do Docker, bem como qualquer outra coisa que considere útil para concluir esta tarefa.

Capítulo IV Parte

obrigatória

Este projeto consiste em fazer com que você configure uma pequena infraestrutura composta de diferentes serviços sob regras específicas. Todo o projeto deve ser feito em uma máquina virtual. Você deve usar o docker compose.

Cada imagem do Docker deve ter o mesmo nome do serviço correspondente. Cada serviço deve ser executado em um contêiner dedicado. Por questões de desempenho, os contêineres devem ser construídos a partir da penúltima versão estável do Alpine ou do Debian. A escolha é sua. Você também precisa escrever seus próprios Dockerfiles, um por serviço. Os Dockerfiles devem ser chamados em seu docker-compose.yml pelo seu Makefile. Isso significa que você mesmo deve criar as imagens do Docker do seu projeto. Em seguida, é proibido obter imagens prontas do Docker, bem como usar serviços como o DockerHub (o Alpine/Debian está excluído dessa regra).

Em seguida, é necessário fazer a configuração:

- Um contêiner do Docker que contém o NGINX apenas com TLSv1.2 ou TLSv1.3.
- Um contêiner do Docker que contém o WordPress + php-fpm (que deve ser instalado e configurado) apenas sem o nginx.
- Um contêiner do Docker que contém apenas o MariaDB sem o nginx.
- Um volume que contém o banco de dados do WordPress.
- Um segundo volume que contém os arquivos do seu site WordPress.
- Uma rede docker que estabelece a conexão entre seus contêineres.

Seus contêineres precisam ser reiniciados em caso de falha.



Um contêiner do Docker não é uma máquina virtual. Portanto, não é recomendável usar qualquer patch hacky baseado em "tail -f" e assim por diante ao tentar executá-lo. Leia sobre como os daemons funcionam e se é uma boa ideia usá-los ou não.



Obviamente, o uso de `network: host` ou `--link` ou `links:` é proibido. A linha de rede deve estar presente em seu arquivo `docker-compose.yml`. Seus contêineres não devem ser iniciados com um comando que execute um loop infinito. Portanto, isso também se aplica a qualquer comando usado como ponto de entrada ou usado em scripts de ponto de entrada. A seguir, algumas correções proibidas: `tail -f`, `bash`, `sleep infinity`, `while true`.



Leia sobre o PID 1 e as práticas recomendadas para escrever Dockerfiles.

- Em seu banco de dados do WordPress, deve haver dois usuários, sendo um deles o administrador. O nome de usuário do administrador não pode conter `admin/Admin` ou `admin-istrator/Administrator` (por exemplo, `admin`, `administrator`, `Administrator`, `admin-123` e assim por diante).



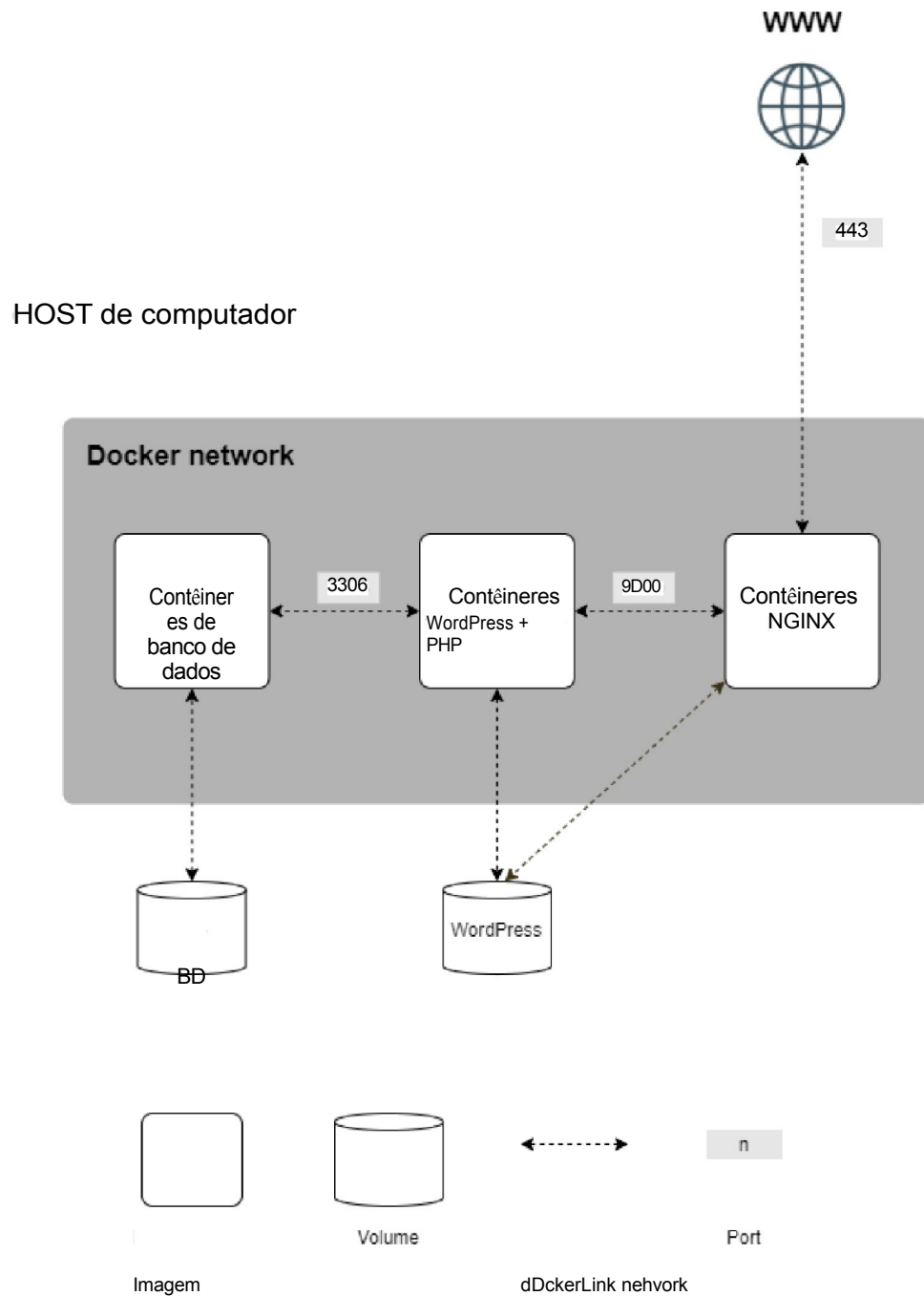
Seus volumes estarão disponíveis na pasta `/home/login/data` da máquina host usando o Docker. É claro que você precisa substituir o `login` pelo seu.

Para simplificar as coisas, você precisa configurar seu nome de domínio para que ele aponte para seu endereço IP local. Esse nome de domínio deve ser `login.42.fr`. Novamente, você precisa usar seu próprio `login`. Por exemplo, se seu `login` for `wil`, `wil.42.fr` será redirecionado para o endereço IP que aponta para o site da `wil`.



A última tag é proibida. Nenhuma senha deve estar presente em seus Dockerfiles. É obrigatório usar variáveis de ambiente. Além disso, é altamente recomendável usar um arquivo `.env` para armazenar variáveis de ambiente. O arquivo `.env` deve estar localizado na raiz do diretório `srcs`. Seu contêiner NGINX deve ser o único ponto de entrada em sua infraestrutura somente por meio da porta 443, usando o protocolo TLSv1.2 ou TLSv1.3.

Aqui está um exemplo de diagrama do resultado esperado:



Abaixo está um exemplo da estrutura de diretório esperada:

```
$> ls -alR
total XX
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 .
drwxrwxrwt 17 wil wil 4096 avril 42 20:42 ...
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 Makefile
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 srcs

./srcs:
total XX
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 .
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 ...
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 docker-compose.yml
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 .env
drwxrwxr-x 5 wil wil 4096 avril 42 20:42 requirements

./srcs/requirements:
total XX
drwxrwxr-x 5 wil wil 4096 avril 42 20:42 .
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 ..
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 bônus
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 mariadb
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 nginx
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 ferrament
as
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 wordpress

./srcs/requirements/mariadb:
total XX
drwxrwxr-x 4 wil wil 4096 avril 42 20:45 .
drwxrwxr-x 5 wil wil 4096 avril 42 20:42 ...
drwxrwxr-x 2 wil wil 4096 avril 42 20:42 conf
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 Dockerfile
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 .dockerignore
drwxrwxr-x 2 wil wil 4096 avril 42 20:42 tools
[...]

./srcs/requirements/nginx:
total XX
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 .
drwxrwxr-x 5 wil wil 4096 avril 42 20:42 ...
drwxrwxr-x 2 wil wil 4096 avril 42 20:42 conf
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 Dockerfile
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 .dockerignore
drwxrwxr-x 2 wil wil 4096 avril 42 20:42 tools
[...]

$> cat srcs/.env
DOMAIN_NAME=wil.42.fr
# Certificados
CERTS=./XXXXXXXXXXXX
XX # CONFIGURAÇÃO DO
MYSQL
MYSQL_ROOT_PASSWORD=XXXXXXXXXXXX
MYSQL_USER=XXXXXXXXXXXX
MYSQL_PASSWORD=XXXXXXXXXXXX
[...]
```



Por motivos óbvios de segurança, todas as credenciais, chaves de API, variáveis de ambiente etc. devem ser salvas localmente em um arquivo `.env` e ignoradas pelo git. As credenciais armazenadas publicamente o levarão diretamente a uma falha no projeto.

Capítulo V

Parte de bônus

Para este projeto, a parte do bônus deve ser simples.

Um Dockerfile deve ser gravado para cada serviço extra. Assim, cada um deles será executado dentro de seu próprio contêiner e terá, se necessário, seu volume dedicado.

Lista de bônus:

- Configure o cache do Redis para seu site WordPress a fim de gerenciar adequadamente o cache.
- Configure um contêiner de servidor FTP apontando para o volume do seu site WordPress.
- Crie um site estático simples na linguagem de sua escolha, exceto PHP (sim, o PHP está excluído!). Por exemplo, um site de demonstração ou um site para apresentar seu currículo.
- Configurar o Adminer.
- Configure um serviço de sua escolha que você considere útil. Durante a defesa, você terá de justificar sua escolha.



Para completar a parte do bônus, você tem a possibilidade de configurar serviços extras. Nesse caso, você pode abrir mais portas para atender às suas necessidades.



A parte do bônus só será avaliada se a parte obrigatória estiver PERFEITA. Perfeito significa que a parte obrigatória foi feita integralmente e funciona sem problemas. Se você não tiver sido aprovado em TODOS os requisitos obrigatórios, a parte do bônus não será avaliada.

Capítulo VI

Envio e avaliação por pares

Entregue sua tarefa em seu repositório Git como de costume. Somente o trabalho dentro de seu repositório será avaliado durante a defesa. Não hesite em verificar novamente os nomes de suas pastas e arquivos para garantir que estejam corretos.



16D85ACC441674FBA2DF65190663EC3C3C258FEA065D090A715F1B62F5A57F0B75403
61668BD6823E2F873124B7E59B5CE94BB7ABD71CD01F65B959E14A3838E414F1E871
F7D91730B