



Começo

Resumo: Este documento é um exercício relacionado à Administração do Sistema.

Versão: 1.3

Conteúdo

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

Capítulo I

Preâmbulo



Capítulo II

Introdução

Este projeto tem como objetivo ampliar seu conhecimento em administração de sistemas utilizando Docker. Você virtualizará diversas imagens Docker, criando-as em sua nova máquina virtual pessoal.

Capítulo III

Diretrizes gerais

- Este projeto precisa ser feito 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 construir as imagens Docker usando `docker-compose.yml`).
- Esta disciplina exige a colocação em prática de conceitos que, dependendo da sua formação, você talvez ainda não tenha aprendido. Portanto, aconselhamos que você não hesite em ler muita documentação relacionada ao uso do Docker, bem como qualquer outra coisa que você considere útil para concluir esta tarefa.

Capítulo IV

Parte obrigatória

Este projeto consiste em montar uma pequena infraestrutura composta por diversos serviços sob regras específicas. Todo o projeto deve ser feito em uma máquina virtual. Você tem que usar o docker compose.

Cada imagem 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 docker-compose.yml pelo seu Makefile.

Isso significa que você mesmo deve construir as imagens Docker do seu projeto. É então proibido extrair imagens Docker prontas, bem como usar serviços como DockerHub (sendo Alpine/Debian excluído desta regra).

Você então deve configurar:

- Um contêiner Docker que contém NGINX somente com TLSv1.2 ou TLSv1.3.
- Um contêiner Docker que contém WordPress + php-fpm (deve ser instalado e configurado) apenas sem nginx.
- Um contêiner Docker que contém MariaDB apenas sem nginx.
- Um volume que contém seu banco de dados 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 Docker não é uma máquina virtual. Assim, não é recomendado 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.

Começo



Claro, usar `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 executando um loop infinito. Portanto, isso também se aplica a qualquer comando usado como ponto de entrada ou em scripts de ponto de entrada. A seguir estão alguns patches de hacky proibidos: `tail -f`, `bash`, `sleep infinity`, `while true`.



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

- Em seu banco de dados 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`, `administrador`, `Administrador`, `admin-123` e assim por diante).



Seus volumes estarão disponíveis na pasta `/home/login/data` da máquina host usando Docker. Claro, você deve substituir o `login` pelo seu.

Para simplificar as coisas, você deve configurar seu nome de domínio para que aponte para seu endereço IP local.

Este nome de domínio deve ser `login.42.fr`. Novamente, você deve usar seu próprio login.

Por exemplo, se o seu login for `wil`, `wil.42.fr` irá redirecionar para o endereço IP que aponta para o site do `wil`.



A tag mais recente é proibida.

Nenhuma senha deve estar presente em seus Dockerfiles.

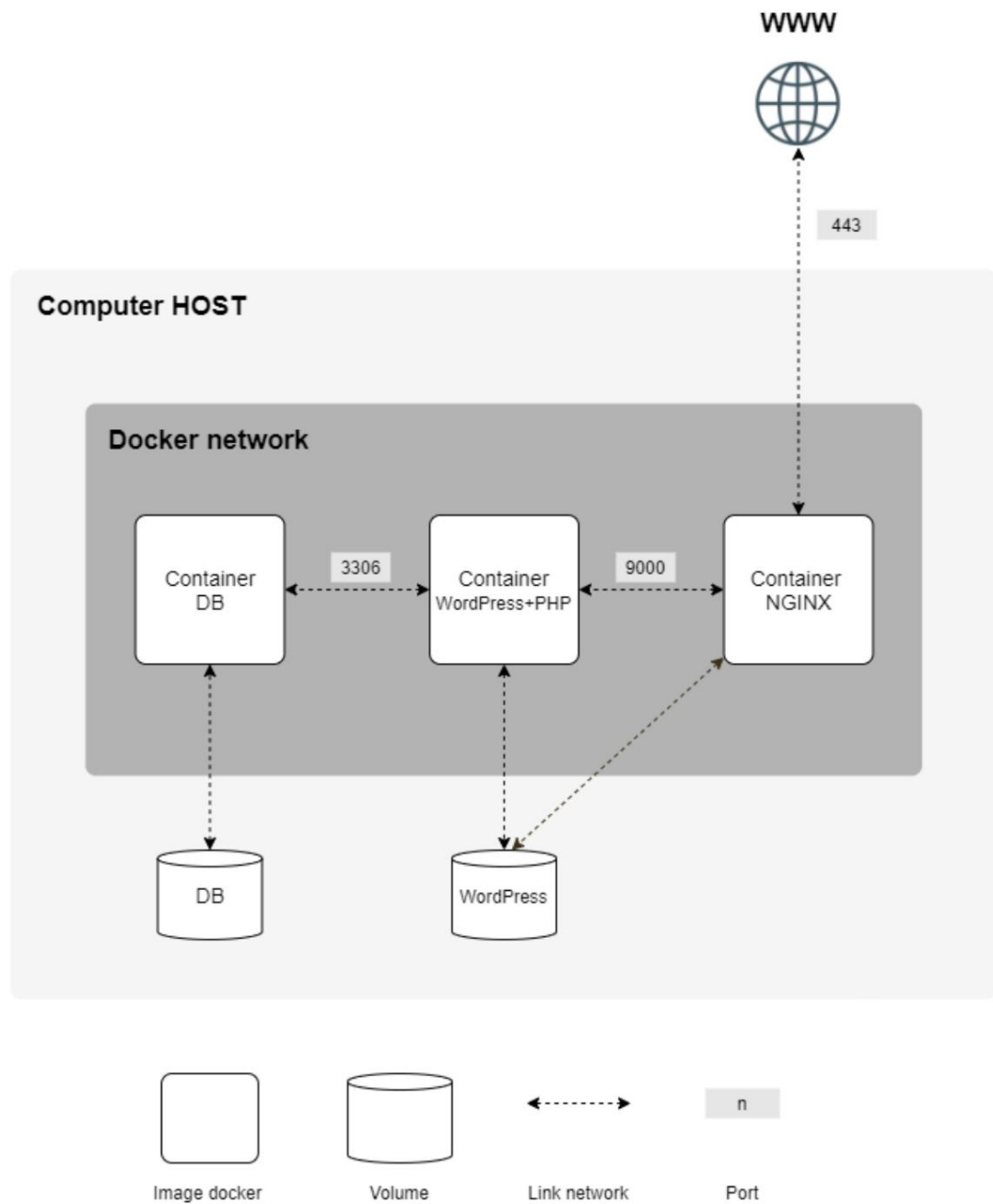
É obrigatório o uso de variáveis de ambiente.

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 através da porta 443, usando o protocolo TLSv1.2 ou TLSv1.3.

Começo

Aqui está um diagrama de exemplo do resultado esperado:



Começo

Abaixo está um exemplo da estrutura de diretórios esperada:

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

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

./srcs/requisitos:
total XX
drwxrwxr-x 5 wil wil 4096 42 de abril 20:42 . drwxrwxr-x 3
wil wil 4096 42 de abril 20:42 ..
drwxrwxr-x 4 wil wil 4096 42 de abril 20:42 bonus drwxrwxr-x 4
wil wil 4096 42 de abril 20:42 mariadb
drwxrwxr-x 4 wil wil 4096 42 de abril 20:42 nginx drwxrwxr-x 4 wil
wil 4096 42 de abril 20:42 ferramentas
drwxrwxr-x 4 wil wil 4096 42 de abril 20:42 wordpress

./srcs/requirements/mariadb: total XX

drwxrwxr-x 4 wil wil 4096 42 de abril 20:45 . drwxrwxr-x 5
wil wil 4096 42 de abril 20:42 .. drwxrwxr-x 2 wil wil 4096 42 de abril
42 20:42 conf -rw-rw-r-- 1 wil wil XXXX 42 de abril 20:42 Dockerfile
-rw-rw-r-- 1 wil wil XXXX 42 de abril 20:42 .dockerignore drwxrwxr-x 2
wil wil 4096 42 de abril 20:42 ferramentas

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

$> cat srcs/.env
DOMAIN_NAME=wil.42.fr #
certificados
CERTS_=/XXXXXXXXXXXXX
# MYSQL SETUP
MYSQL_ROOT_PASSWORD=XXXXXXXXXXXXX
MYSQL_USER=XXXXXXXXXXXXX
MYSQL_PASSWORD=XXXXXXXXXXXXX
[...] $>
```



Por razões óbvias de segurança, quaisquer credenciais, chaves de API, variáveis de ambiente, etc... devem ser salvas localmente em um arquivo .env e ignoradas pelo git. Credenciais armazenadas publicamente levarão você diretamente ao fracasso do projeto.

Capítulo V

Parte bônus

Para este projeto, a parte bônus pretende ser simples.

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

Lista de bônus:

- Configure o cache redis para o seu site WordPress para 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, PHP está excluído!). Por exemplo, um site de demonstração ou um site para apresentar seu currículo.
- Configure o administrador.
- Configure um serviço de sua preferência que você considere útil. Durante a defesa, você terá que justificar sua escolha.



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



A parte bônus só será avaliada se a parte obrigatória for PERFEITA. Perfeito significa que a parte obrigatória foi feita integralmente e funciona sem mau funcionamento. Se você não passou em TODOS os requisitos obrigatórios, sua parte do bônus não será avaliada de forma alguma.

Capítulo VI

Envio e avaliação por pares

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