



インセ プショ ン

概要 本書は、システム管理に関する演習です。

バージョン: 1.3

目次

I	前文	2
II	はじめに	3
III	一般的なガイドライン	4
IV	必須項目	5
V	ボーナスパート	9
VI	提出物および相互評価	10

第1章 前文



第II章 はじめに

このプロジェクトは、Dockerを使用することによって、システム管理の知識を広げることを目的としています。いくつかのDockerイメージを仮想化し、新しい個人の仮想マシンに作成します。

第III章 一般的な

ガイドライン

- このプロジェクトは、仮想マシン上で行う必要があります。
- プロジェクトの設定に必要なファイルは、すべてsrcsに置く必要があります。フォルダーを使用します。
- Makefileも必要で、ディレクトリのルートに配置する必要があります。これはアプリケーション全体をセットアップする必要があります（つまり、docker-compose.ymlを使用してDockerイメージを構築する必要があります）。
- この課題では、あなたのバックグラウンドによっては、まだ学んだことのない概念を実践する必要があります。そのため、この課題をクリアするために、Dockerの使い方に関連するドキュメントや、その他役に立つと思われるものは、躊躇せずにたくさん読むことをお勧めします。

第IV章 必須パー

ト

このプロジェクトは、特定のルールのもと、さまざまなサービスで構成される小規模なインフラストラクチャを構築していただくものです。プロジェクト全体は、仮想マシン上で行わなければなりません。docker composeを使用する必要があります。

各Dockerイメージは、対応するサービスと同じ名前である必要があります。各サービスは専用のコンテナで動作させる必要があります。性能面を考慮し、コンテナはAlpineまたはDebianの最終安定版からビルドする必要があります。選択はあなた次第です。

また、1つのサービスにつき1つ、独自のDockerfilesを書く必要があります。Dockerfileはdocker-compose.ymlの中でMakefileから呼び出す必要があります。

つまり、プロジェクトのDockerイメージは自分で構築する必要があるのです。そして、DockerHubのようなサービスを利用したり、既製のDockerイメージを引っ張ってくることが禁じられています（Alpine/Debianはこのルールから除外されています）。

その後、セットアップが必要です：

- TLSv1.2またはTLSv1.3のみのNGINXを含むDockerコンテナです。
- nginxを使わずにWordPress+php-fpm（インストールと設定が必要）だけを入れたDockerコンテナです。

- nginxを使わずにMariaDBだけを入れたDockerコンテナです。
- WordPressのデータベースが格納されているボリュームです。
- WordPress ウェブサイトのファイルを格納する第 2 ボリュームです。
- コンテナ間の接続を確立するdocker-networkです。

コンテナはクラッシュした場合、再起動しなければなりません。



Dockerコンテナは、仮想マシンではありません。したがって、それを実行しようとするときに、「tail -f」などに基づくハッキリとしたパッチを使用することはお勧めしません。デーモンがどのように機能するのか、デーモンを使うのが良いのかどうかについては、こちらをお読みください。

インセプション



もちろん、`network: host`や`-link`、`links:`を使うことは禁止されています。
`docker-compose.yml`ファイルには必ず`network`の行を記述してください。無限ループを実行するようなコマンドでコンテナを起動してはいけません。したがって、これはエントリーポイントとして使用されるコマンドや、エントリーポイントのスクリプトで使用されるコマンドにも適用されます。以下は、禁止されているハッキーパッチです: `tail -f`, `bash`, `sleep infinity`, `while true`.



PID 1とDockerfilesの書き方に関するベストプラクティスについてお読みください。

- WordPressのデータベースには、2人のユーザーが存在し、そのうちの1人がad-ministratorである必要があります。管理者のユーザー名にadmin/Admin や admin- istrator/Administrator を含めることはできません（例: admin, administrator, Administrator, admin-123, など）。



あなたのボリュームは、Dockerを使ってホストマシンの`/home/login/data`フォルダーで利用できるようになります。もちろん、ログインはあなたのものに置き換えてください。

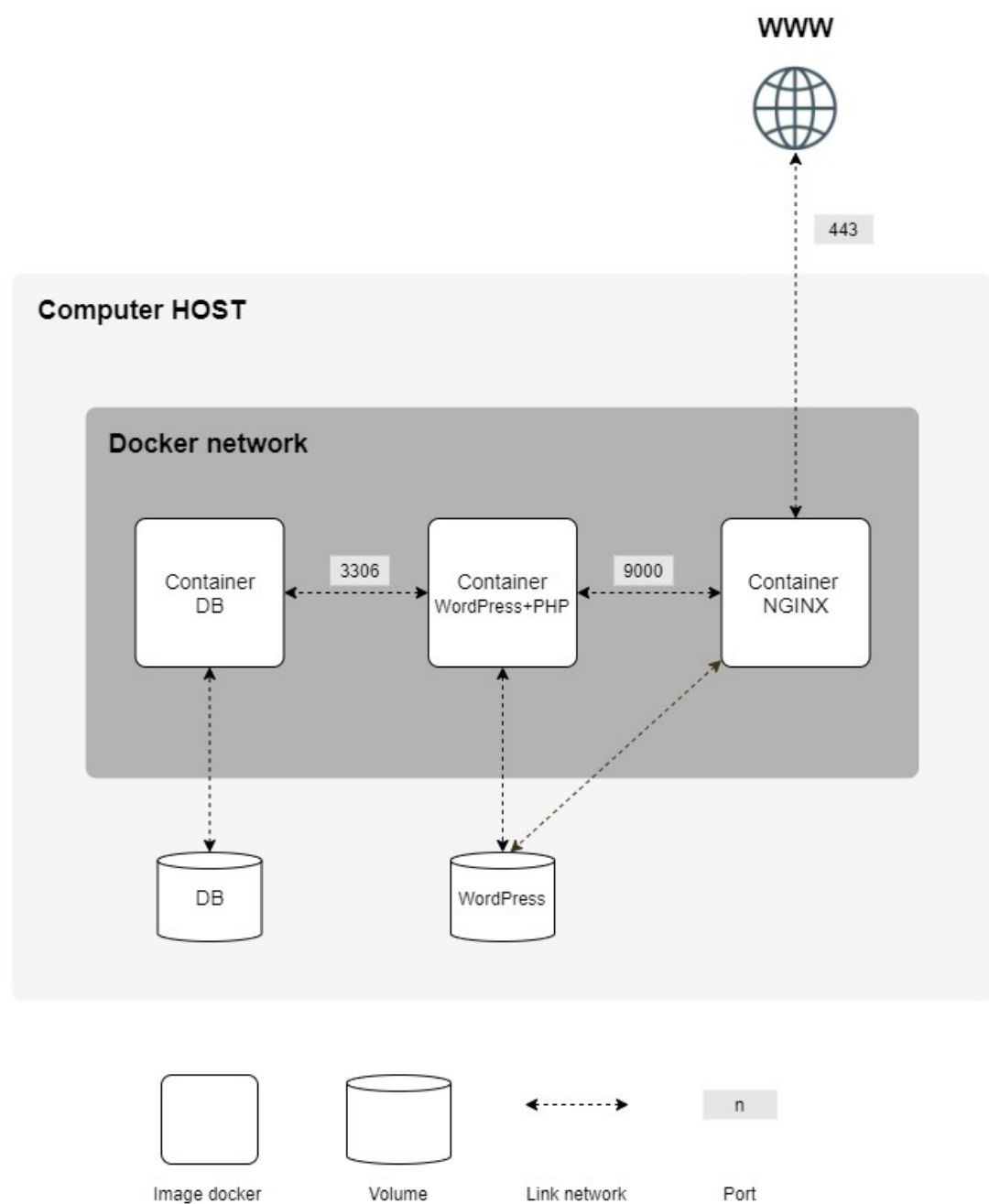
物事を簡単にするために、ドメイン名があなたのローカルIPアドレスを指すように設定する必要があります。
このドメイン名は`login.42.fr`でなければなりません。ここでも、自分のログインを使用する必要があります。
例えば、ログインが`wil`の場合、`wil.42.fr`は`wil`のウェブサイトを目指すIPアドレスにリダイレクトされます。



最新のタグは禁止されています。
Dockerfilesにパスワードが存在してはいけません。環境変数の使用は必須です。
また、環境変数を保存するために、`.env`ファイルを使用することを強くお勧めします。`.env`ファイルは、`srcs`ディレクトリのルートに配置する必要があります。
NGINXコンテナは、`TLSv1.2`または`TLSv1.3`プロトコルを使用して、ポート443のみを経由するインフラへの唯一のエントリポイントである必要があります。

インセプション

ここでは、予想される結果の例図を示します：



インセプション

以下は、想定されるディレクトリ構造の例です:

```
$> 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:
トータル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:
トータルXX
drwxrwxr-x 5 wil wil 4096 アヴ 42 20:42 .
                        リル
drwxrwxr-x 3 wil wil 4096 アヴ 42 20:42 ..
                        リル
drwxrwxr-x 4 wil wil 4096 アヴ 42 20:42 ボーナス
                        リル
drwxrwxr-x 4 wil wil 4096 アヴ 42 20:42 マリアドビ
                        リル
drwxrwxr-x 4 wil wil 4096 アヴ 42 20:42 nginx
                        リル
drwxrwxr-x 4 wil wil 4096 アヴ 42 20:42 用具
                        リル
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 wordpress

./srcs/requirements/mariadb:
トータル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
[...]です。

./srcs/requirements/nginx:
トータル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
[...]です。

$> cat srcs/.env
DOMAIN_NAME=wil.42.fr
# 証明書
CERTS_=/XXXXXXXXXXXXX
# MYSQL セットアップ
mysql_root_password=xxxxxxxxxxxx
mysql_user=xxxxxxxxxxx
mysql_password=xxxxxxxxxxxx
[...]...
$>
```



セキュリティ上の理由から、認証情報、APIキー、env変数などは.envファイルにローカルに保存し、gitでは無視する必要があります。公開されている認証情報は、プロジェクトの失敗に直接つながります。

第V章 ボーナス

パート

今回の企画では、特典部分はシンプルなものを目指しています。

Dockerfileは、追加サービスごとに記述する必要があります。このように、それぞれのサービスは独自のコンテナ内で実行され、必要に応じて専用のボリュームを持つことになります。

ボーナスリストです：

- キャッシュを適切に管理するために、WordPressのWebサイトにredisキャッシュを設定します。
- WordPressウェブサイトのボリュームを指すFTPサーバーコンテナを設定します。
- PHPを除くお好きな言語で、簡単な静的ウェブサイトを作成してください（はい、PHPは除外します！）。例えば、ショーケースサイトや履歴書を紹介するためのサイトなど。
- Adminerを設定する。
- 便利だと思う自分の好きなサービスを設定する。ディフェンスでは、自分の選んだサービスを正当化する必要があります。



ボーナスパートを完了するために、あなたは追加のサービスを設定する可能性があります。この場合、お客様のニーズに合わせてより多くのポートを開くことができます。



ボーナスパートは、必須パートがPERFECTである場合にのみ評価されます。完璧とは、必須パートが統合的に行われ、誤動作することなく動作することを意味します。必須条件をすべてクリアしていない場合、ボーナスパートはまったく評価されません。

第六章

提出物および相互評価

通常通り、Gitリポジトリに課題を提出してください。あなたのリポジトリ内の作品だけが、ディフェンス時に評価されます。フォルダ名やファイル名が正しいかどうか、遠慮なく再確認してください。

