



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 : 1.2

内容

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	提出と相互評価	10

第一章 前文



第二章 はじめに

このプロジェクトは、**Docker**を使用してシステム管理に関する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、新しい個人用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`ディレクトリに配置する必要があります。
フォルダーに格納されます。
- `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。このファイル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` を使って `Docker` イメージをビルドする必要があります)。
- この課題は、あなたのバックグラウンドによっては、まだ学んでいない概念を实践する必要があります。そのため、`Docker`の使い方に関連するドキュメントや、その他この課題を達成するために役立つと思われるものは、遠慮なくたくさん読むことをお勧めします。

第4章 必須項目

このプロジェクトは、特定のルールのもとで様々なサービスからなる小さなインフラストラクチャを構築してもらおうというものです。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**を使用しなければなりません。

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実行する必要があります。パフォーマンス上の問題から、コンテナはAlpineまたはDebianの最新の安定版からビルドする必要があります。選択するのはあなたです。また、1つのサービスにつき1つのDockerfilesを記述する必要があります。Dockerfilesはdocker-compose.ymlの中でMakefileから呼び出される必要があります。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ります。そのため、DockerHubなどの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推奨されています（Alpine/Debianはこのルールから除外されています）。

その後、セットアップする必要があります。

- TLSv1.2またはTLSv1.3のみのNGINXを格納したDockerコンテナです。
- WordPress + php-fpm (インストールと設定が必要) を含む Docker コンテナで、nginx を含まないものです。
- nginxを使わずにMariaDBだけを入れたDockerコンテナ。
- WordPressのデータベースを格納するボリュームです。
- WordPressのウェブサイトファイルが格納されている第2巻。
- コンテナ間の接続を確立するDocker-Network。クラッシュした場合、コンテナを再起動する必要があります。



Dockerコンテナは、仮想のマシンではありません。したがって実行しようとするときに **'tail -f'** などに基づいたハッキングパッチを使用することはお勧めできません。デーモンがどのように機能するのか、そしてそれ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。



もちろん、`network`を使用します。 `host`や`-link`、`links:`の使用は禁止されています。`docker-compose.yml`ファイルに`network`の行があることが必要です。コンテナは無限ループするようなコマンドで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、いくつかの禁止されているハックパッチです。 `tail -f, bash, sleep infinity, while true.`



PID 1とDockerfilesの書き方の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 人がアドミニストレーターである必要があります。管理者のユーザー名には `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）。



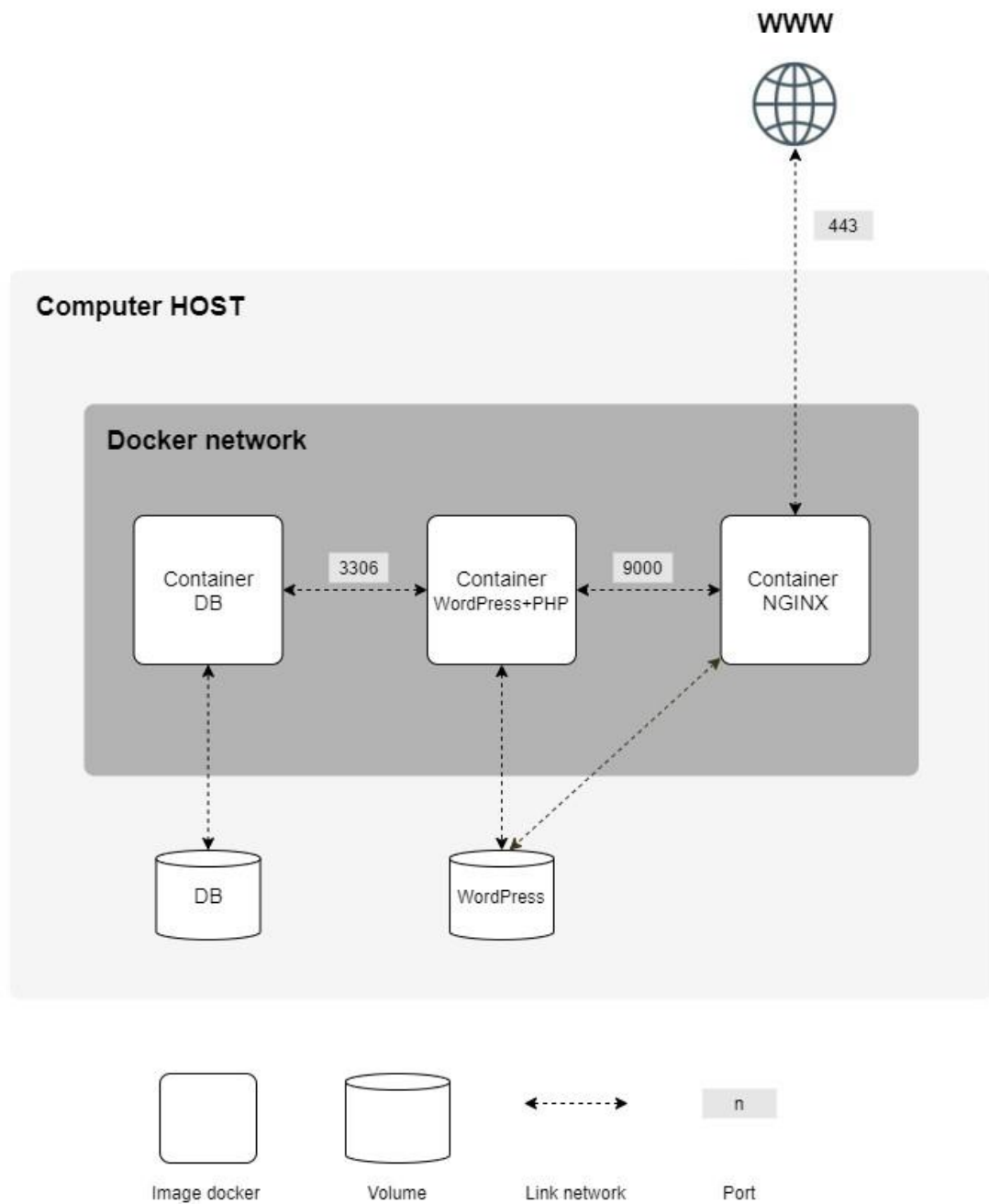
Dockerを使用しているホストマシンの`/home/login/data`フォルダにあなたのボリュームが用意されます。もちろん、ログインは自分のものに置き換えてください。

より簡単にするために、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、あなた自身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のみを介してインフラストラクチャへの唯一のエントリーポイントにする必要があります。

以下は、予想される結果の例図です。



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:
合計 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:
合計 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 ナインテ
                               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:
合計 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:
合計 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
# 証明書
CERTS=./XXXXXXXXXX#
MYSQL SETUP
mysql_root_password=xxxxxxxxx
mysql_user=xxxxxxxxx
mysql_password=xxxxxxxxx [...]...
$>
```


第五章 ボーナスパート

今回の企画では、特典部分をシンプルにすることを目指しました。

Dockerfileは、それぞれの追加サービスに対して記述する必要があります。このように、各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

- キャッシュを適切に管理するために、WordPressのウェブサイトに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は除きます！）。例えば、ショーケースサイトや履歴書を発表するためのサイトなど。
- Adminerをセットアップする。
- 自分が便利だと思う好きなサービスを設定する。答弁では、選んだサービスを正当化する必要があります。



ボーナス・パートを完成させるために、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放することができます。



ボーナスパーツは、必須パーツがPERFECTである場合にのみ査定されます。パーフェクトとは、必須パーツが統合的に行われ、誤動作することなく動作することを意味します。必須項目をすべてクリアしていない場合、ボーナスパーツの評価は一切行われません。

第六章

提出と相互評価

通常通り、Gitリポジトリに課題を提出する。防衛戦では、あなたのリポジトリ内の作品だけが評価されます。フォルダ名やファイル名に間違いがないか、遠慮なく再確認してください。