

Inception プロジェクト評価シート（翻訳）

一般的な指示

- 評価プロセス全体を通じて、要件の確認方法や何かを検証する方法がわからない場合は、評価対象の学生があなたを手助けする必要があります。
- アプリケーションを設定するために必要なすべてのファイルが `srcs` フォルダー内に配置されていることを確認してください。 `srcs` フォルダーはリポジトリのルートに配置されている必要があります。
- Makefile がリポジトリのルートに配置されていることを確認してください。
- 評価を開始する前に、ターミナルで次のコマンドを実行してください：

```
docker stop $(docker ps -qa); docker rm $(docker ps -qa); docker rmi -f $(docker images -qa); docker volume rm $(docker volume ls -q); docker network rm $(docker network ls -q) 2>/dev/null
```
- `docker-compose.yml` ファイルを読んでください。その中に `network: host` や `links:` があってはいけません。ある場合は、評価はここで終了します。
- `docker-compose.yml` ファイルを読んでください。その中に `network(s)` が必要です。ない場合は、評価はここで終了します。
- Makefile と Docker が使用されているすべてのスクリプトを確認してください。その中に `--link` があってはいけません。ある場合は、評価はここで終了します。
- Dockerfile を確認してください。ENTRYPOINT セクションで `tail -f` やバックグラウンドで実行されるコマンドが見られる場合、評価はここで終了します。スクリプトを実行するためでない場合に `bash` や `sh` が使用されている場合も同様です（例： `nginx & bash` や `bash` ）。
- エントリーポイントがスクリプトの場合（例： `ENTRYPOINT ["sh", "my_entrypoint.sh"]`、`ENTRYPOINT ["bash", "my_entrypoint.sh"]` ）、バックグラウンドでプログラムを実行していないことを確認してください（例： `nginx & bash` ）。
- リポジトリ内のすべてのスクリプトを確認してください。無限ループを実行するものがないことを確認してください。

禁止されているコマンドの例：

`sleep infinity`、`tail -f /dev/null`、`tail -f /dev/random`

- Makefile を実行してください。

必須パート

このプロジェクトは、docker-compose を使用して異なるサービスで構成される小規模なインフラストラクチャのセットアップです。以下のすべてのポイントが正しいことを確認してください。

プロジェクト概要

評価対象者は、以下について簡単な言葉で説明する必要があります：

- Docker と docker-compose の仕組み
- docker-compose を使用した Docker イメージと使用しない場合の違い
- VM と比較した Docker の利点
- このプロジェクトに必要なディレクトリ構造の妥当性（サンプルは課題の PDF ファイルに記載されています）

シンプルなセットアップ

- NGINX がポート 443 のみでアクセス可能であることを確認してください。完了したら、ページを開いてください。
- SSL/TLS 証明書が使用されていることを確認してください。
- WordPress ウェブサイトが適切にインストールおよび設定されていることを確認してください（WordPress インストールページが表示されないはずです）。アクセスするには、ブラウザで `https://login.42.fr` を開いてください。login は評価対象の学生のログインです。 `http://login.42.fr` 経由でサイトにアクセスできないはずです。

期待通りに動作しない場合は、評価プロセスはここで終了します。

Docker の基本

- まず Dockerfile を確認してください。サービスごとに 1 つの Dockerfile が必要です。Dockerfile が空のファイルでないことを確認してください。該当しない場合、または Dockerfile が欠落している場合、評価プロセスはここで終了します。
- 評価対象の学生が独自の Dockerfile を書き、独自の Docker イメージを構築したことを確認してください。実際、既製のものを使用したり、DockerHub などのサービスを使用することは禁止されています。
- すべてのコンテナが Alpine Linux の最新から1つ前の安定バージョン、または Debian Buster から構築されていることを確認してください。Dockerfile が `FROM alpine`、

FROM `debian:buster`、またはその他のローカルイメージで始まっていない場合、評価プロセスはここで終了します。

- 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。そうでない場合、評価プロセスはここで終了します。
- Makefile が `docker-compose` を介してすべてのサービスをセットアップしていることを確認してください。つまり、コンテナは `docker-compose` を使用して構築されている必要があります、クラッシュが発生していないことが必要です。そうでない場合、評価プロセスは終了します。

Docker ネットワーク

- `docker-compose.yaml` ファイルを確認して、`docker-network` が使用されていることを確認してください。次に `docker network ls` コマンドを実行して、ネットワークが表示されることを確認してください。
- 評価対象の学生は、`docker-network` について簡単な説明をする必要があります。

上記のいずれかのポイントが正しくない場合、評価プロセスはここで終了します。

SSL/TLS を使用した NGINX

- Dockerfile があることを確認してください。
- `docker-compose ps` コマンドを使用して、コンテナが作成されたことを確認してください（必要に応じてフラグ `-p` の使用は許可されています）。
- `http`（ポート 80）経由でサービスにアクセスしてみて、接続できないことを確認してください。
- ブラウザで `https://login.42.fr/` を開いてください。login は評価対象の学生のログインです。表示されるページは設定された WordPress ウェブサイトである必要があります（WordPress インストールページが表示されないはずです）。
- TLS v1.2/v1.3 証明書の使用は必須であり、実証される必要があります。SSL/TLS 証明書は認識される必要はありません。自己署名証明書の警告が表示される場合があります。

上記のいずれかのポイントが明確に説明されておらず正しくない場合、評価プロセスはここで終了します。

php-fpm とそのボリュームを使用した WordPress

- Dockerfile があることを確認してください。
- Dockerfile に NGINX がないことを確認してください。

- `docker-compose ps` コマンドを使用して、コンテナが作成されたことを確認してください（必要に応じてフラグ `-p` の使用は許可されています）。
- ボリュームがあることを確認してください。そのために：
`docker volume ls` コマンドを実行し、次に `docker volume inspect <ボリューム名>` を実行してください。標準出力の結果に `/home/login/data/` というパスが含まれていることを確認してください。login は評価対象の学生のログインです。
- 利用可能な WordPress ユーザーを使用してコメントを追加できることを確認してください。
- 管理ダッシュボードにアクセスするために管理者アカウントでサインインしてください。管理者ユーザー名には `admin` や `Admin` を含めてはいけません（例：admin、administrator、Admin-login、admin-123 など）。
- 管理ダッシュボードから、ページを編集してください。ウェブサイトでページが更新されたことを確認してください。

上記のいずれかのポイントが正しくない場合、評価プロセスはここで終了します。

MariaDB とそのボリューム

- Dockerfile があることを確認してください。
- Dockerfile に NGINX がないことを確認してください。
- `docker-compose ps` コマンドを使用して、コンテナが作成されたことを確認してください（必要に応じてフラグ `-p` の使用は許可されています）。
- ボリュームがあることを確認してください。そのために：
`docker volume ls` コマンドを実行し、次に `docker volume inspect <ボリューム名>` を実行してください。標準出力の結果に `/home/login/data/` というパスが含まれていることを確認してください。login は評価対象の学生のログインです。
- 評価対象の学生は、データベースにログインする方法を説明する必要があります。パスワードなしで root として SQL データベースにログインしてみてください。ログインが成功した場合、評価プロセスはここで終了します。
- ユーザーアカウントとそのパスワードで SQL データベースにログインしてみてください。データベースが空でないことを確認してください。

上記のいずれかのポイントが正しくない場合、評価プロセスはここで終了します。

永続性!

- この部分は非常に簡単です。仮想マシンを再起動する必要があります。再起動後、`docker-compose` を再度起動してください。次に、すべてが機能していて、WordPress

と MariaDB の両方が設定されていることを確認してください。以前に WordPress ウェブサイトに加えた変更はまだ残っているはずです。

上記のいずれかのポイントが正しくない場合、評価プロセスはここで終了します。

ボーナス

必須パートが完全かつ完璧に完了し、エラー管理が予期しない使用や誤った使用を処理する場合にのみ、ボーナスパートを評価してください。防御中にすべての必須ポイントが合格しなかった場合、ボーナスポイントは完全に無視される必要があります。

ボーナス

- 課題で許可されているボーナスごとに 1 ポイントを追加してください。
- 各追加サービスの適切な機能と実装を確認およびテストしてください。
- 自由選択のサービスについては、評価対象の学生がその仕組みと、なぜそれが有用だと考えるかについて簡単な説明をする必要があります。

0（失敗）から 5（優秀）までで評価してください。

評価

防御に対応するフラグをチェックすることを忘れないでください

評価結果：

- ☐ Ok
- ☐ 空の作業 ☐ 不完全な作業 ☐ 不正行為 ☐ クラッシュ ☐ 不完全なグループ
- ☐ 懸念事項 ☐ 禁止された関数

結論

この評価にコメントを残してください：

（コメント欄）

