



インセプション

概要: このドキュメントは、システム管理に関する演習です。

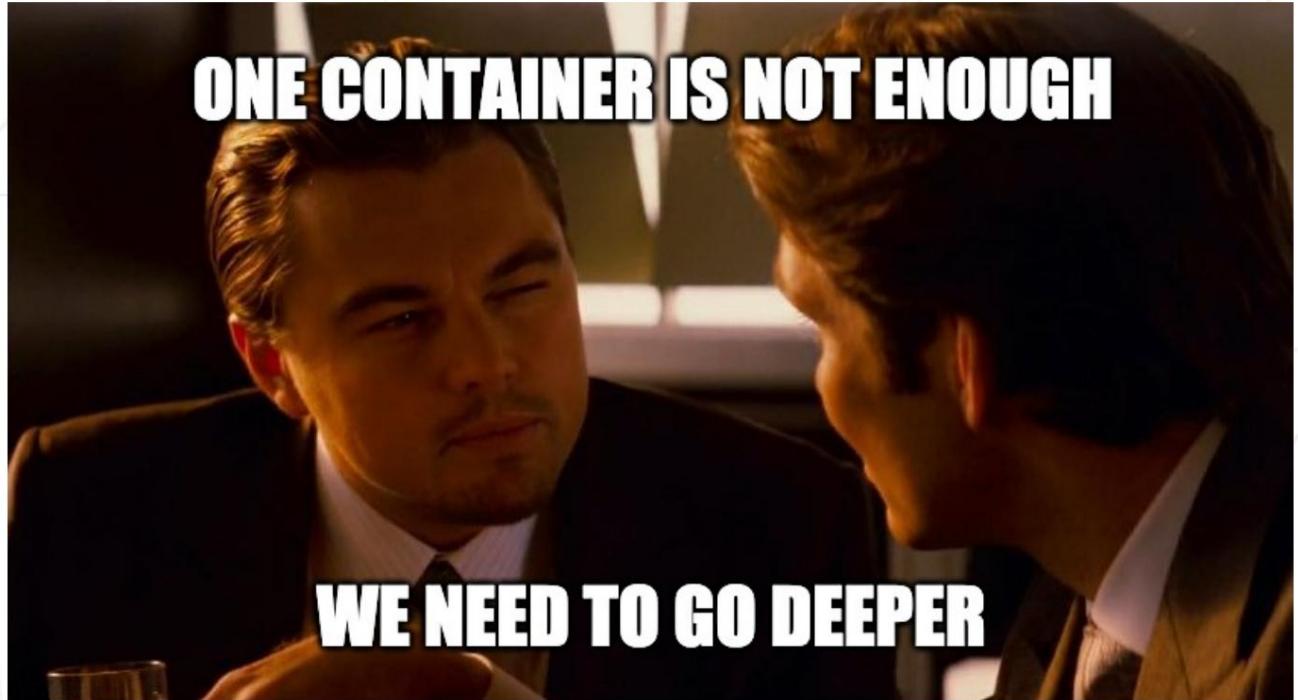
バージョン: 4.0

コンテンツ

私	前文	2
II	導入	3
3	一般的なガイドライン	4
IV	A指示	5
V	必須部分	7
VI	ボーナスパート	12
VII	提出とピア評価	13

第1章

前文



第2章

導入

このプロジェクトは、Dockerテクノロジーの活用を通してシステム管理に関する知識を深めることを目的としています。新しい個人用仮想マシンに複数のDockerイメージを作成し、仮想化します。

第3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完了する必要があります。
- プロジェクトの構成に必要なすべてのファイルは、srcsに配置する必要があります。
フォルダ。
- Makefile も必須で、ディレクトリのルートに配置する必要があります。Makefile はアプリケーション全体の設定を行う必要があります（つまり、docker-compose.yml を使用して Docker イメージをビルドする必要があります）。
- この科目では、これまでの経験に応じて、まだ習得していない概念を実践する必要があります。そのため、Docker の使用に関する詳細なドキュメントや、この課題を完了するために役立つと思われる他のリソースを読むことをお勧めします。

第4章

AI指示

•文脈

学習の過程で、AIは様々なタスクを支援してくれます。AIツールの様々な機能と、それらがどのようにあなたの仕事に役立つかをじっくりと探求してください。しかし、常に慎重にアプローチし、結果を批判的に評価してください。コード、ドキュメント、アイデア、技術的な説明など、どんなものであっても、質問が適切に構成されているか、生成されたコンテンツが正確であるかを完全に確信することはできません。同僚は、間違いや盲点を回避するための貴重なリソースです。

•主なメッセージ

AIを使用して、反復的な作業や面倒な作業を削減します。

コーディングと非コーディングの両方で、あなたのビジネスに役立つプロンプトスキルを身につけましょう。
将来のキャリア。

AIシステムが一般的なリスク、バイアス、倫理的問題をより適切に予測して回避する仕組みを学びます。

同僚と協力して、技術スキルとパワースキルの両方を継続的に構築します。

AI生成コンテンツは、完全に理解し、責任を負えるもののみ使用してください

のために。

•学習者のルール:

- AIツールを倫理的に使用し、潜在的なバイアスを減らすことができるよう、時間をかけてAIツールを調べ、その仕組みを理解する必要があります。
- 質問する前に、自分の問題についてよく考えてください。こうすることで、より明確に、正確な語彙を使用した、より詳細で関連性の高いプロンプト。
- AIによって生成されたものはすべて体系的にチェック、レビュー、質問、テストする習慣を身につける必要があります。
- 常にピアレビューを求める必要があります。自分自身の検証だけに頼らないでください。

インセプション

・フェーズの成果:

- 汎用的なプロンプトスキルとドメイン固有のプロンプトスキルの両方を開発します。
- AIツールを効果的に活用して生産性を向上します。
- 計算思考、問題解決能力、適応力、そしてコラボレーション。

・コメントと例:

- 試験や評価など、真の理解を示すことが求められる状況に頻繁に遭遇するでしょう。技術スキルと対人スキルの両方を磨き続け、万全の準備をしてください。
- 自分の考えを説明し、仲間と議論することで、自分の理解のギャップが明らかになることが多い。理解を深める。ピアラーニングを優先しましょう。
- AIツールは、ユーザー固有のコンテキストを欠き、一般的な回答を提供する傾向があります。同じ環境にいる同僚は、より関連性の高い正確な洞察を提供できます。
- AIが最も可能性の高い回答を生成する傾向がある場合でも、同僚は別の視点や貴重なニュアンスを提供できます。品質チェックポイントとして頼りにしましょう。

✓ 良い実践例:

AIに「ソート関数をテストするにはどうすればいいですか？」と尋ねると、いくつかのアイデアが浮かびました。それを試してみて、同僚と結果を確認します。そして、一緒にアプローチを改良していきます。

悪い習慣:

AIに関数全体を書いてもらい、それをプロジェクトにコピー＆ペーストしました。しかし、ピア評価の際に、それが何をするのか、なぜそうするのかを説明できませんでした。信頼を失い、プロジェクトは失敗に終わりました。

✓ 良い実践例:

AIを活用してパーサーの設計を支援します。その後、同僚と一緒にロジックを検証します。2つのバグを見つけ、一緒に書き直します。より良く、よりクリーンで、完全に理解できるものになります。

悪い習慣:

プロジェクトの重要な部分のコードをCopilotで生成しました。コンパイルは通るもの、パイプ処理の仕組みが理解できません。評価時に説明がつかず、プロジェクトは失敗に終わりました。

第5章

必須部分

このプロジェクトでは、特定のルールに基づいて、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ます。プロジェクト全体は仮想マシン内で実行する必要があります。Docker Composeを使用する必要があります。

各 Docker イメージの名前は、対応するサービスと同じである必要があります。
各サービスは専用のコンテナ内で実行する必要があります。

パフォーマンス上の理由から、コンテナは Alpine または Debian の最新安定版からビルドする必要があります。どちらをお選びいただくかは自由です。

また、サービスごとに独自の Dockerfile を作成する必要があります。これらの Dockerfile は、Makefile から docker-compose.yml 内で呼び出される必要があります。

つまり、プロジェクト用の Docker イメージは必ず自分でビルドする必要があります。そのため、既製の Docker イメージをプルしたり、DockerHubなどのサービスを利用したりすることは禁止されます（Alpine/Debian はこのルールの対象外です）。

次に、以下を設定する必要があります。

- TLSv1.2 または TLSv1.3 のみの NGINX を含む Docker コンテナ。
- nginx なしで、php-fpm (インストールおよび構成が必要) のみを含む WordPress を含む Docker コンテナー。
- nginx なしで MariaDB のみを含む Docker コンテナ。
- WordPress データベースが含まれるボリューム。
- WordPress ウェブサイトのファイルが含まれる 2 番目のボリューム。
- コンテナ間の接続を確立する docker-network。

クラッシュが発生した場合、コンテナは自動的に再起動する必要があります。



Docker コンテナは仮想マシンではありません。したがって、
実行する際に、「tail -f」などの方法に基づいたハックパッチを使用することをお勧めします。デーモンの動作原理と、デーモンを使用するのが適切かどうかについては、こちらの記事をご覧ください。

インセプション



もちろん、network: host や -link や links: の使用は禁止されています。
docker-compose.yml ファイルにネットワーク ラインが存在している必要があります。
コンテナは、無限ループを実行するコマンドで起動してはいけません。これは、エントリポイントとして使用されるコマン
ドや、エントリポイントスクリプトで使用されるコマンドにも適用されます。以下のハックバッヂは禁止されています :tail
-f、bash、sleep infinity、while

真実。



PID 1 と Dockerfile の作成に関するベスト プラクティスについて説明します。

- WordPressデータベースには2人のユーザーがいる必要があります。そのうちの1人は管理者です。管理者のユーザー名には「admin」、「Admin」、「admin-istrator」、「Administrator」を含めることはできません（例 :admin、administrator、Administrator、admin-123など）。



ボリュームは、Docker を使用しているホストマシンの /home/login/data フォルダで利用できるようになります。もちろん、ログイン情報はご自身のものに置き換える必要があります。

プロセスを簡素化するには、ドメイン名をローカル 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 である必要があります。繰り返しますが、ご自身のログイン名を使用する必要があります。
たとえば、ログインが「wil」の場合、wil.42.fr は Wil の Web サイトを指す IP アドレスにリダイレクトされます。

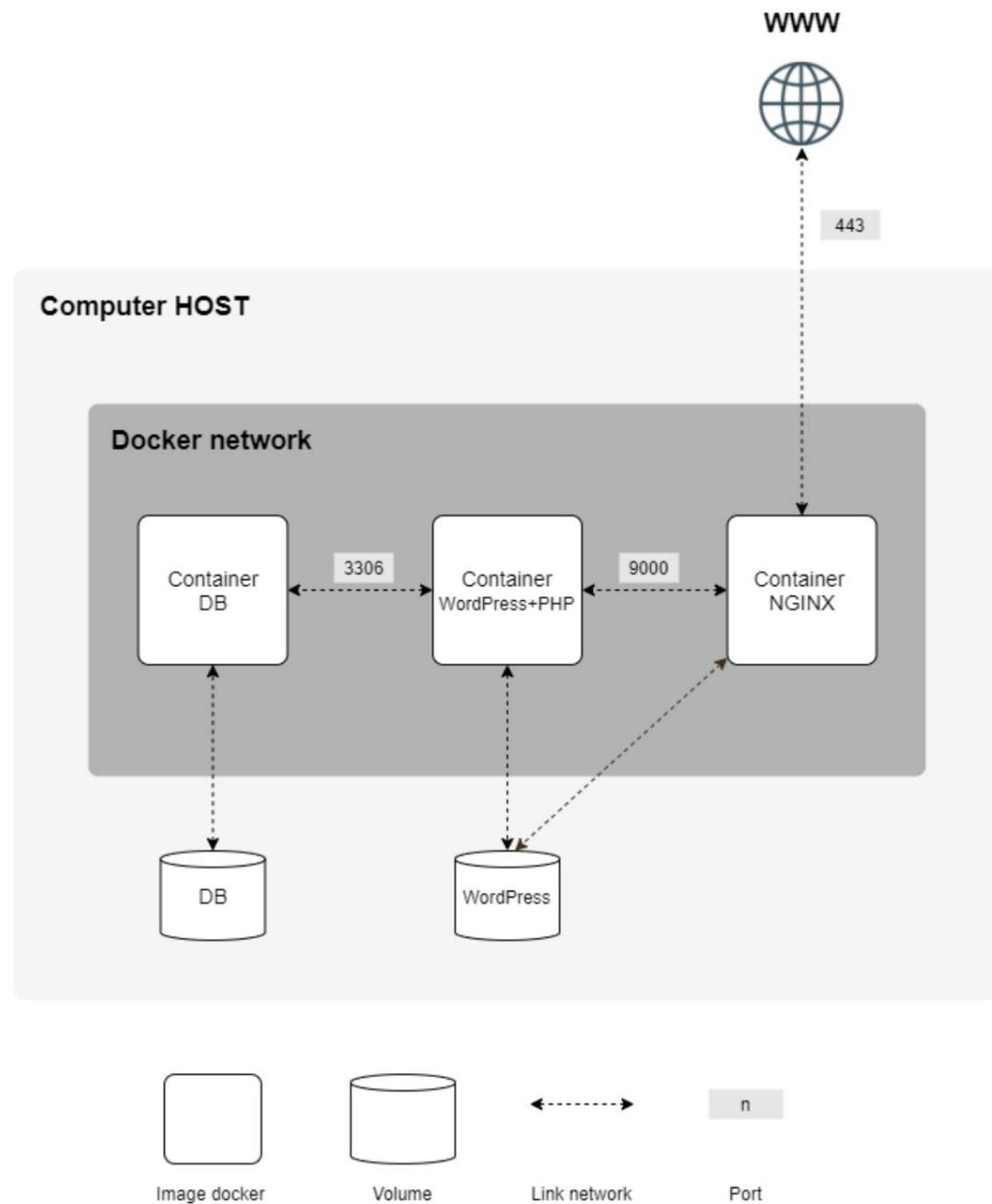


最新タグは禁止です。
Dockerfile にパスワードが存在してはいけません。
環境変数の使用は必須です。
また、環境変数を保存するには .env ファイルを使用し、機密情報を保存するには Docker シークレットを使用することを強くお勧めします。

NGINX コンテナは、インフラストラクチャへの唯一のエントリ ポイントである必要があり、
TLSv1.2 または TLSv1.3 プロトコルを使用してポート 443 経由でのみアクセス可能でなければ
なりません。

インセプション

期待される結果の例の図を以下に示します。



インセプション

以下は、予想されるディレクトリ構造の例です。

```
$> ls -alR
合計XX
drwxrwxr-x 3 ウィル ウィル 4096 42 20:42。 drwxrwxrwt 17 ウィル ウィル 4096 42
20:42 ..
-rw-r--r-- 1 wil wil XXXX 4月42日 20:42 Makefile drwxrwxr-x 3 wil wil 4096 4月42日 20:42 秘密
drwxrwxr-x 3 ウィル ウィル 4096 アヴリル 42 20:42 srcs

./秘密:
合計XX
drwxrwxr-x 2 ウィル ウィル 4096 42 20:42。 drwxrwxr-x 6 ウィル ウィル 4096 42
20:42 ..
-rw-r--r-- 1 は XXXX になります 4月 42 20:42 credentials.txt -rw-r--r-- 1 は XXXX になります 4月 42 20:42
db_password.txt -rw-r--r-- 1 は XXXX になります 4月 42 20:42 db_root_password.txt

./srcs:
合計 XX
drwxrwxr-x 3 wil wil 4096 avril 42 20:42。
drwxrwxr-x 3 は 4096 年 4 月 42 日 20:42 に実行されます ... -rw-r--r-- 1 は XXXX になります
す 42 年 4 月 42 日 20:42 に実行されます docker-compose.yml -rw-r--r-- 1 は XXXX になります 42 年 42 日 20:42 に実行さ
れます .env drwxrwxr-x 5 は 4096 年 4 月 42 日 20:42 に実行されます 要件

./srcs/requirements: total XX drwxrwxr-
x 5 wil wil 4096
4月 42 20:42 .
drwxrwxr-x 3 ウィル ウィル 4096 アヴリル 42 20:42 .. drwxrwxr-x 4 ウィル ウィル 4096 アヴリ
ル 42 20:42 ボーナス drwxrwxr-x 4 ウィル ウィル 4096 アヴリル 42 20:42 mariadb drwxrwxr-x 4 ウ
ィル ウィル 4096 アヴリル 42 20:42 nginx drwxrwxr-x 4 ウィル ウィル 4096 アヴリル 42 20:42 ツール

drwxrwxr-x 4 wil wil 4096 4月42日 20:42 wordpress

./srcs/要件/mariadb:
合計XX
drwxrwxr-x 4 ウィル ウィル 4096 42 20:45。 drwxrwxr-x 5 ウィル ウィル 4096 42
20:42 ..
drwxrwxr-x 2 は 4096 4月 42 20:42 に実行されます conf -rw-r--r-- 1 は XXXX 4月 42 20:42
に実行されます Dockerfile
-rw-r--r-- 1 wil wil XXXX 4月 42 20:42 .dockermanage drwxrwxr-x 2 wil wil 4096 4月 42 20:42 ツール

[...]
./
srcs/requirements/nginx:
合計XX
drwxrwxr-x 4 ウィル ウィル 4096 42 20:42。 drwxrwxr-x 5 ウィル ウィル 4096 42
20:42 ..
drwxrwxr-x 2 は 4096 4月 42 20:42 に実行されます conf -rw-r--r-- 1 は XXXX 4月 42 20:42
に実行されます Dockerfile
-rw-r--r-- 1 wil wil XXXX 4月 42 20:42 .dockermanage drwxrwxr-x 2 wil wil 4096 4月 42 20:42 ツール

[...]

$> cat srcs/.env
DOMAIN_NAME=wil.42.fr # MYSQL SETUP
MYSQL_USER=XXXXXXXXXXXX
[...] $>
```



セキュリティ上の理由から、認証情報、APIキー、パスワードなどは、様々な方法やファイルでロー
カルに保存し、Gitによって無視される必要があります。認証情報を公開保存すると、プロジェク
トの失敗に直結します。

インセプション



変数を（ドメイン名として）環境に保存することができます
.envのような変数ファイル

第6章

ボーナス部分

このプロジェクトでは、ボーナス部分はシンプルなものになる予定です。

追加サービスごとにDockerfileを作成する必要があります。これにより、各サービスは独自のコンテナ内に保存され、必要に応じて専用のボリュームが確保されます。

ボーナスリスト:

- WordPressウェブサイトにRedisキャッシュを設定し、適切に管理します。
キャッシュ。
- WordPress ウェブサイトのボリュームを指す FTP サーバー コンテナーを設定します。
- PHP以外の任意の言語で、シンプルな静的ウェブサイトを作成してください（PHPは対象外です）。例えば、ショーケースサイトや履歴書のプレゼンテーションサイトなどです。
- Adminer を設定します。
- 役に立つと思うサービスを自由に設定してください。審査中は、あなたの選択を正当化する必要があります。



ボーナス部分を完了するには、追加のサービスを設定することができます。この場合、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。



ボーナス部分は、必須部分が完璧に完了した場合にのみ評価されます。完璧とは、必須部分が完全に完了し、不具合なく機能することを意味します。必須要件をすべて満たしていない場合、ボーナス部分は評価されません。

まったく評価されません。

第7章

提出とピア評価

課題は通常通りGitリポジトリに提出してください。審査ではリポジトリ内の作業のみが評価されます。フォルダとファイルの名前が正しいか、必ず再確認してください。

評価中に、プロジェクトの簡単な変更が求められる場合があります。これには、軽微な動作変更、数行のコードの追加または書き換え、あるいは簡単に追加できる機能などが含まれる場合があります。

この手順はすべてのプロジェクトに当てはまるとは限りませんが、評価ガイドラインに記載されている場合は準備しておく必要があります。

このステップは、プロジェクトの特定の部分についての実際の理解を確認することを目的としています。変更は、選択した任意の開発環境（通常のセットアップなど）で実行でき、評価の一部として特定の時間枠が定義されていない限り、数分以内に実行できるはずです。

たとえば、関数やスクリプトに小さな更新を加えたり、表示を変更したり、新しい情報を保存するためにデータ構造を調整したりすることが求められる場合があります。

詳細（範囲、対象など）は評価ガイドラインに明記され、同じプロジェクトでも評価ごとに異なる場合があります。



16D85ACC441674FBA2DF65190663EC3C3C258FEA065D090A715F1B62F5A57F0B75403
61668BD6823E2F873124B7E59B5CE94BB7ABD71CD01F65B959E14A3838E414F1E871
F7D91730B