inceptionプチ説明会資料

inceptionプチ説明会資料

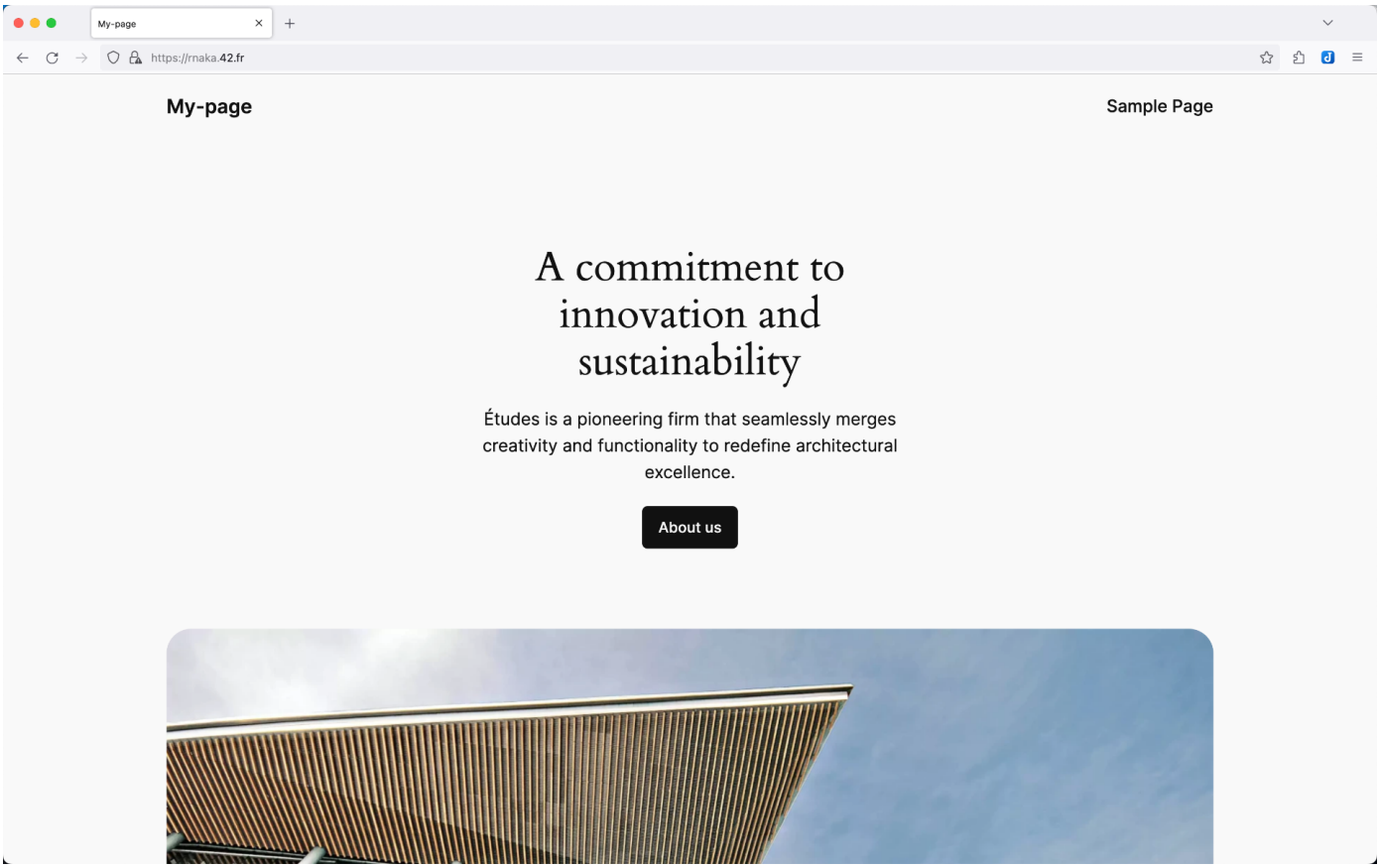
目次

inceptionとは

wordpressを自動的に作成してくれるdockercompose.ymlやdockerファイルを作成する。

目標

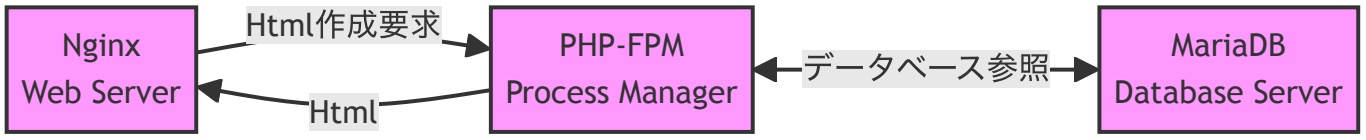
makeし、ブラウザでhttps://login.42.fr(login は自分のintra名) を検索した時に写真のようなページが表示されれば完成。



構造

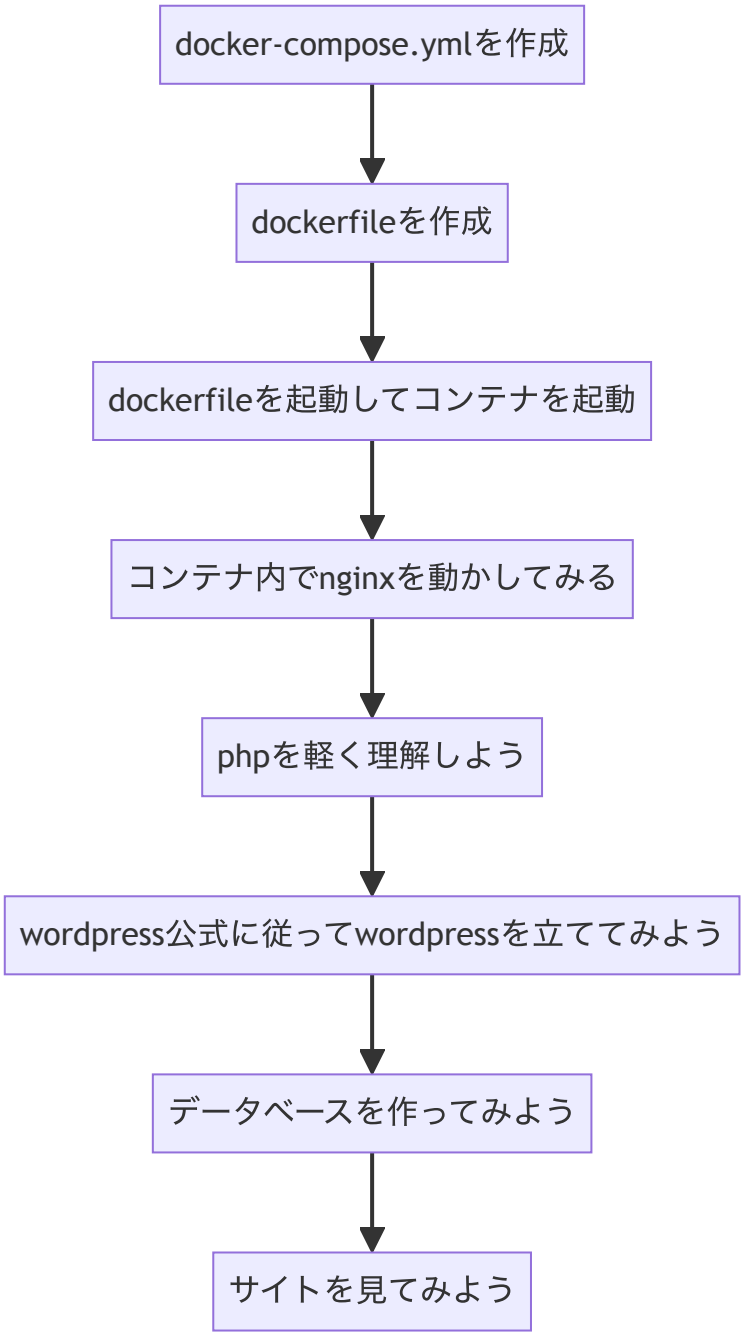
inceptionでは三つのサービスをお互いに繋げてサイトを作成する。ただし、三つのサービスをそれぞれ独立したコンテナに分けなければいけない。

1. nginx
   webサーバー。要求されたサイトページを送信する。
2. php-fpm
   nginxから要求されたphpファイルからhtmlファイルを作成し、nginxに返す。
3. mariadb
   データベースを管理するシステム。今回の場合は、wordpressのユーザーやコメント情報を管理する。



WordPressを一つのコンテナで作成！

流れを理解



Step 1 docker-compose.ymlとdockerfileを書く

1. 以下の構造通りになるようディレクトリーとファイルを作る。

```
inception
├── all-in-one
│   └── Dockerfile
└── docekr-compose.yml

2 directories, 2 files
```

2. docker-compose.ymlファイルに以下を貼り付け。
   tabの数に注意！！ここ大事！

```
version: '3'  #dockercomposeのバージョン

services:     #コンテナ定義セクション
  incep:      #名前
    build: ./all-in-one   #Dockerfileの場所
    ports:    #繋げたいポートの指定。左がホスト、右がコンテナのポート番号
      - "80:80"
```

3.Dockerfileに以下を貼り付け

```
FROM debian:latest

RUN apt update
RUN apt install -y vim wget unzip
RUN apt install -y nginx
RUN apt install -y php7.4-fpm php7.4-mysqli php7.4-gd php7.4-curl
php7.4-xml php7.4-mbstring php7.4-zip
RUN apt install mariadb-server
```

Step 2 コンテナ内に入って、nginxの設定を変える

1. `docker compose exec コンテナ名 bash`
   注意！！docker-compose.ymlファイルがあるディレクトリーでしかcompose系のコマンドは動かない
2. `service --status-all`で各サービスの状態を確認
3. `service nginx start` でNginxを起動
4. ブラウザで`localhost:80/`と検索
5. `vim /etc/nginx/sites-available/default`で設定項目をいじってみる。

```
server {
listen 80 default_server; #リッスンポートを指定している
listen [::]:80 default_server;#Ipv6用にも設定している

root /var/www/html;#80番ポートから接続してきたら、/var/www/html配下のファイルしか見せないよ

# Add index.php to the list if you are using PHP
index index.php;#アクセスしてきたらデフォルトで返すファイルを指定するよ

server_name _;#よくわかんない

location / {#元から設定されている、いじらない
    try_files $uri $uri/ =404;
}
```

```
    location ~ \.php$ {#元はコメントアウトされている、外して使う。
        include snippets/fastcgi-php.conf;

        fastcgi_pass unix:/run/php/php8.2-fpm.sock;#二つあるけど、どちらか片方を使用する。元はphp7.4-fpmになっているから気をつけてください。
        #fastcgi_pass 127.0.0.1:9000;#こっちはtcpというプロトコルを使用して通信する。こっちの方が遅いが、今後はこっちを使用する。
    }
}
```

Step 3 Wordpressをダウンロード

1. wordpress公式ダウンロードページにアクセス
2. コンテナ内で`wget https://wordpress.org/latest.zip`
3. `unzip latest.zip`
4. wordpressファイル群を移動。`mv wordpress/* /var/www/html/`
5. wordpress公式インストールガイドを見ながら進める。

Step 4 Mariadbでデータベースを構築

1. wordpressのデータベース構築を見ながら、cliでデータベースを構築する。

```
$ mysql -u root -p
Enter password:

mysql> CREATE DATABASE wordpress_db;#データベース名は何でも良い(ここではwordpress_db)

mysql> CREATE USER "user"@"%" IDENTIFIED BY "password";#ここではユーザーを作成している。"user"@"localhost"でlocalhostoからしかアクセスできないuserを作成している。%は外部からでもアクセスできるようになる。将来的には%を使用する。

mysql> GRANT ALL PRIVILEGES ON wordpress_db.* TO "user"@"%";

mysql> FLUSH PRIVILEGES;

mysql> EXIT
```

2. localhost:80/にアクセスして、wordpress用のユーザーをポチポチしながら作る。
   今後はwp-cliコマンドを使用して作る。

Step 5 作成したサイトを見てみよう！

1. ログアウトして、もう一度localhost:80/にアクセスしてみよう！

役立つコマンド一覧

compose系コマンド

`docker compose up --build -d` イメージとコンテナを作成
`docker compose down -v` イメージとコンテナを削除(ボリュームやnetworkも)
`docker compose exec コンテナ名 bash` コンテナ内にbashで入る
`docker compose ps` 起動中のコンテナを表示

serviceコマンド(コンテナ内で各サービスを起動、停止させる)

`service サービス名 start` サービスの起動
`service サービス名 stop` サービスの停止
`service サービス名 restart` サービスの再起動
`service --status-all` 全サービスの状態確認