

Docker コンテナの利用に慣れるための例題集

ご購入はこちら

館野 昌一

Docker コンテナの利用に慣れるための例題集です。前章のログイン機能付きウェブ・アプリケーションを

ベースにしています。手元のPCで試せるので、ぜひ、体験してみてください。

1 複数のサービスやプロセスを同時に起動、管理する

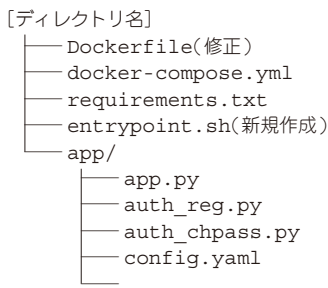


図1-1 ディレクトリ構成

複数のサービスやプロセスを同時に起動、管理する場合、`entrypoint.sh`を使用すると便利です。`entrypoint.sh`は、Dockerコンテナが起動する際に最初に実行されるスクリプトです。ここでは`entrypoint.sh`を作成し、Dockerfileと一緒に利用する方法を解説します。ディレクトリ構成を図1-1に示します。

Dockerfileの修正

`entrypoint.sh`を利用するため、Dockerfileをリスト1-1のように修正します。日本語フォントのインストールも追加しました。

リスト1-1 Dockerfileの修正

```

# ベースイメージの指定
FROM python:3.9-slim

# 作業ディレクトリの設定
WORKDIR /app

# 必要なファイルをコピー
COPY requirements.txt /app/

# 日本語フォントのインストール
RUN apt-get update && apt-get install -y \
    fonts-ipafont-gothic

# パッケージのインストール
RUN pip install --no-cache-dir -r requirements.txt

# エントリーポイントスクリプトのコピーと実行権限の付与
COPY entrypoint.sh /
RUN chmod +x /entrypoint.sh

# ポートを公開(必要に応じて他のポートも追加)
EXPOSE 8501

# エントリーポイントの設定
ENTRYPOINT ["/entrypoint.sh"]

```

entrypoint.shの作成

`entrypoint.sh` ファイルを新規作成し、リスト1-2の内容を記述します。ここでは、2番目のプログラム `app2.py` を起動する場合の書き方を示しています。

リスト1-2 entrypoint.sh ファイルを新規作成

```

#!/bin/bash
# このスクリプトがBashシェルで実行されることを指定

# Streamlitをバックグラウンドで起動
streamlit run app.py --server.port 8501 &

# 必要に応じて他のサービスをバックグラウンドで起動可能
# 例: 別のstreamlitアプリ(app2.py)を起動する場合
streamlit run app2.py --server.port 8502 &

# フォアグラウンドで実行するプロセスが必要なため待機
wait -n

```