

ウェブ・アプリのDocker コンテナ化にトライ

ご購入はこちら

館野 昌一



図1 シンプルなログイン・システムを作る…他のウェブ・アプリでも使い回せる

本章では、Pythonでウェブ・アプリケーションを作成できるライブラリStreamlitを使いアプリケーションをデプロイする手法を示します。Dockerを用いることで、アプリケーションのスケラビリティと移植性が向上し、開発者はより多くの時間を機能の改善に費やすことができます。

● 作るもの

現代のウェブ・アプリケーションでは、ユーザごとの個別のニーズに応えるために、個人アカウントの管理機能を持つことが一般的となっています。本稿で紹介するウェブ・アプリケーションは、ユーザ登録、ログイン、パスワード変更を行えるシンプルなログイン・システムを備えています。

本システムではユーザの役割を、

- アドミン (admin)
- メンバ (member)
- 候補者 (candidate)

の3種類に分け、最初に登録したユーザを自動的にアドミンとして設定することで、管理の手間を軽減しています。ログイン機能によってユーザの認証と認可を適切に行うことで、アプリケーションのセキュリティを確保し、ユーザの役割に応じたアクセス制御を行います(図1)。

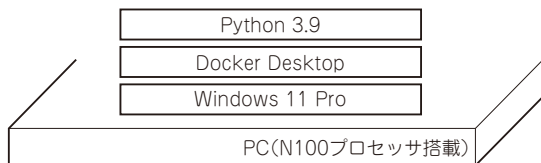


図2 利用したのはN100プロセッサ搭載のWindows 11マシン

● 使用した機材

Windows 11 Proを搭載したミニPCを使用しました。このPCのスペックは、CPUがN100(インテル)、メモリが16Gバイト、SSDが500Gバイトで、価格は2万円前半です。このPCにDocker Desktopをインストールして使用しました(図2)。Docker Desktopは、公式サイト(<https://www.docker.com/ja-jp/products/docker-desktop/>)からダウンロードしてインストールできます。

ステップ①…ウェブ・アプリの作成

ここで作るウェブ・アプリケーションの全体像と各ファイルの役割について説明します。アプリケーションは主にPythonスクリプトと設定ファイルで構成されており、ユーザの登録、ログイン、パスワード変更といった基本的な機能を提供します。

● ファイルのディレクトリ構成

今回使用するファイルは全部で7つあります(図3)。Dockerに関するファイルが3つと、プログラム本体が4つです。

以下にプログラム本体について説明します。

● app/app.py…ユーザ・インターフェースを構築する

app/app.pyは、このウェブ・アプリケーションのエントリ・ポイントであり、ユーザ・インターフェースを構築する役割を担っています(リスト1)。このスクリプトは、Streamlitを使って、ユーザが登