

コンテナ利用システムを作る②

…開発環境

土屋 健

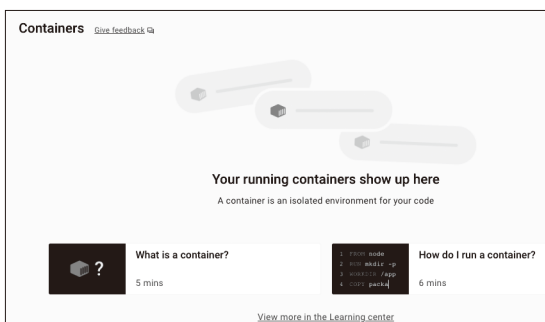


図1 Docker Desktopの画面

開発環境として コンテナを利用するメリット

ここからは、開発環境としてコンテナを利用し、システム開発を体験します。

開発環境としてコンテナを利用すると、以下のよう
なメリットがあります。

- 開発環境（コンパイラ、ライブラリ、フレームワーク、DBなどのミドルウェア）を統一できる
- 試験環境と運用環境を全く同じ環境にできる
- 開発環境、試験環境をどこにでもすぐにくつでも用意できる

開発と運用で環境が違っていたためにプログラムが動かないというのは意外と多いので、このように開発から試験/運用まで一貫性のある環境を用意できるというのは非常に大きなメリットです。

以下のステップで環境構築からプログラム開発まで実施します。

ステップ1：開発端末にコンテナ環境をセットアップ

ステップ2：開発用コンテナ・イメージの作成

ステップ3：開発用コンテナを利用したシステム開発

ステップ1：開発端末に コンテナ環境をセットアップ

macOSでDockerコンテナを利用できるように環境準備を行います。Windows環境で試す場合はコラム

リスト1 nginxコンテナの起動

```
(base) tsuchiya@macbook ~ % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES

(base) tsuchiya@macbook ~ % docker images
REPOSITORY TAG          IMAGE ID   CREATED   SIZE
(base) tsuchiya@macbook ~ % docker run nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
8a1e25ce7c4f: Pull complete
省略
2024/04/01 07:46:57 [notice] 1#1: start worker
process 32
```

を参考にしてください。

今回は、Docker Desktopを利用して環境構築します。なお、Docker Desktopがサポートしている環境では、インストール方法などの違いはありますが、Docker Desktopの操作やdockerコマンドの操作は同じようにできます。

● Docker Desktopのインストール

以下のサイトからDocker Desktopをダウンロードします。

<https://www.docker.com/ja-jp/products/docker-desktop/>

ダウンロードしたファイルを開き、DockerをApplicationsにコピーすればインストールは完了です。

インストールしたDocker Desktopを起動してみます。図1の画面が表示されれば正常にインストールできています。

● コンテナの起動確認

まずは、dockerコマンドでコンテナを起動してみます。ここでは、nginxのコンテナ・イメージ⁽²⁾を使用します。

リスト1のように、nginxコンテナを起動します。dockerコマンドでのコンテナ起動をスキップして、Docker Desktopでコンテナ起動することも可能ですが、現在のDocker DesktopではGUI操作でコンテナ・