

Docker Compose

これまで見てきた内容は覚えるのが大変かもしれません。特に、複数のパソコンで同じ設定を繰り返さなければならない場合にはなおさらです。そこで活躍するのが **Docker Compose** です。

これは、**Docker** プロジェクトを構築するために必要な操作を記述し、1 つのコマンドでプロジェクト全体を構築することを可能にします。

compose.yaml ファイルの作成

これを実現するために、プロジェクトのルートに "**compose.yaml**" という名前のファイルを作成します。

Apache の設定

以前作成した **Dockerfile** を使って、イメージを追加できます。例えば、次のように記述します：

```
services:
  web:
    build: .
    container_name: serverApache2
    restart: always
    depends_on:
      - mysql
    ports:
      - "8080:80"
    volumes:
      - ../var/www/html
    environment:
      - SUPER_ENVIRONEMENT_VARIABLE=Mon Super Site
```

YAML ではインデント（字下げ）が非常に重要です。インデントが 1 つ少ない、または多いだけでもエラーになります。

compose.yaml の各項目の説明：

- **services**：プロジェクトで使用するサービス（イメージ）をリストアップするためのプロパティです。
- **web**：各サービスは名前付きのプロパティを持つ必要があります。ここでは **Apache** サーバーに "web" という名前を付けました。
- **build**：使用する **Dockerfile** のある場所を指定します（ここでは "." が **compose.yaml** と同じ場所を意味します）。
- **container_name**：任意で指定可能。コンテナに一意の名前を付けます。
- **restart**：コンテナが停止したときに自動的に再起動するかどうかを指定します。
- **depends_on**：サービスの起動順を指定します（ここでは "mysql" サービスの後に "web" を起動するように指定しています）。
- **ports**：仮想マシンとコンテナで開くポートを指定します。
- **volumes**：コンテナとホストとのファイル共有（ボリュームやバインドマウント）を定義します。
- **environment**：環境変数をコンテナまたはアプリケーションに渡します（この例は意味のないダミー変数です）。

MySQL の設定

すでに存在するイメージを使って、サービスを追加することも可能です。先ほどの **services** セクションに次を追加します：

```
mysql:
  image: mariadb
  container_name: serverMySQL2
  restart: always
  environment:
    - MARIADB_ROOT_PASSWORD=root
  ports:
    - "3308:3306"
  volumes:
    - ./mysql.sql:/docker-entrypoint-initdb.d/mysql.sql
```

すでに説明したプロパティもありますが、ここでは追加の要点を紹介します：

- **mysql**：これは任意に付けた名前です。
- **image**：使用するイメージの名前とタグ（バージョン）です。
- **environment**：この環境変数には実際の意味があります。MySQL の初回インストール時にも使用した内容です。
- **/docker-entrypoint-initdb.d/**：このディレクトリには特別な意味があり、ここに置いた SQL ファイルは、コンテナ作成時に自動的にデータベースへインポートされます。

PHPMyAdmin の設定

インストールを補完するために、PHPMyAdmin 用のイメージを新たに追加します。こちらも **services** に次のように追記します：

```
phpmyadmin:
  image: phpmyadmin
  container_name: serverPHPMyAdmin2
  restart: always
  depends_on:
    - mysql
  environment:
    PMA_ARBITRARY: 1
    PMA_HOST: mysql
  ports:
    - "8081:80"
```

ここでは新しい内容はあまりありませんが、以下の環境変数について補足します：

- **PMA_ARBITRARY**：true にすると、MySQL サーバーを選択するフィールドが表示されます。
- **PMA_HOST**：接続するデータベース名を指定します。

もちろん、他にも多くの構成オプションがありますが、それについては各自で調査してみてください。

Docker Compose のコマンド

ファイルが作成できたら、プロジェクトのルート（[compose.yaml](#)がある場所）で次のコマンドを実行して、インストールを開始します：

```
docker compose up
```

このコマンドは、3つのイメージを含むネットワークを作成します。これは最低限必要なコマンドですが、オプションを追加したり、ターミナルをログから解放したいこともあるでしょう。

その場合、まず次のコマンドでコンテナを停止・削除します：

```
docker compose stop
```

```
docker compose rm
```

その後、次のように再起動します：

```
docker compose -p cours-docker up -d
```

- **-p** : Compose に任意の名前を付けます（指定しない場合はカレントディレクトリ名が使用されます）。

- **-d** : Compose のログ出力を非表示にして、ターミナルを解放します。

Compose に名前を付けた場合、その名前を使って停止や削除を行う必要があります。

Compose を停止・削除するには、次のコマンド1つで済みます：

```
docker compose down
```