



## ネットワーク

Apache と PHP だけではなく、MySQL も必要になります。さらに PHPMyAdmin があると便利です。

複数のコンテナを相互接続するには「ネットワーク（network）」を作成する必要があります。デフォルトでは、外部からアクセス可能にしても、コンテナ同士は仮想マシン内で隔離されています。同じネットワークを共有することで、コンテナ同士が通信できるようになります。

ネットワークの作成には以下のコマンドを使用します：

```
docker network create firstNetwork
```

ここで "firstNetwork" はネットワーク名です。その他の便利なコマンド：

```
# すべてのネットワークを表示
docker network ls

# ネットワークを削除
docker network rm ネットワーク名
```

コンテナをネットワークに接続するには：

```
# コンテナ起動時に指定：
--network ネットワーク名

# 既存のコンテナをネットワークに接続：
docker network connect ネットワーク名 コンテナ名
```

---

## MySQL のインストール

MySQL をインストールするには、以下のようにいくつかのパラメータを加えたコマンドを使います：

```
docker run --name firstMySQL --network firstNetwork -p 3307:3306 -e
MARIADB_ROOT_PASSWORD=root -d mariadb
```

ここでの新しい要素：

- `-e`：環境変数を指定します。
- `MARIADB_ROOT_PASSWORD=`：デフォルトユーザー（root）のパスワードを設定します。

これで MySQL データベースが動作します。

---

## データベースへの接続

コマンドラインからデータベースに接続するには、コンテナに入るだけでOKです。

仮想マシン外部のツール（VS Code、MySQL Workbenchなど）を使って接続する場合は、ホスト名に "localhost" または "127.0.0.1" を使用し、ポートにはVMのポート（ここでは3307）を使用します。

同じネットワーク上の他のコンテナから接続するには、コンテナのポート（ここでは3306）を使います。接続先としては、コンテナ名またはIPアドレスを使用できます。

IPアドレスを確認するには：

```
docker inspect コンテナ名
```

"NetworkSettings/Networks/ネットワーク名/IPAddress" の項目にIPアドレスがあります。

または、次のようにして直接IPだけを取得することも可能です：

```
docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' コンテナ名
```

---

## ファイルのインポート

データベースにインポートを行うには、インポート対象のファイルをコンテナ内に配置する必要があります。

1つのファイルを移動するだけであれば、以下のコピーコマンドが便利です：

```
# ローカルからコンテナへコピー：
docker cp ファイルパス コンテナ名:保存先パス

# コンテナからローカルへコピー：
docker cp コンテナ名:ファイルパス 保存先パス
```

---

## PHPMyAdmin の導入

コマンドラインは便利ですが、GUIがあればもっと快適です。

以下のコマンドで PHPMyAdmin を実行できます：

```
docker run --name phpmyadmin --network ネットワーク名 -e PMA_HOST=IP -d -p 8087:80 phpmyadmin
```

- `PMA_HOST=` の後には、MySQL コンテナの IP またはコンテナ名を指定します。

この設定により、先ほど設定した "root" ユーザーとパスワード（ここでは "root"）で接続できるようになります。