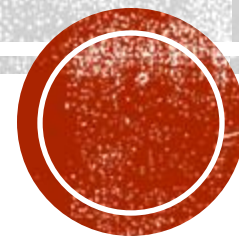


# DOCKERにおけるHTTP通信

#3 学生と社会人LT会 2024/03/01



# 自己紹介

- 名前: kitaaa
- 職業: 今年4月から修士一年
- 趣味: ギター、RPGゲーム
- ハッカソン3回、長期インターン
- 研究: 災害関連



異なる技術スタックの組み合わせが求められる場面

特定の言語に依存したパッケージの利用



# 発表のテーマ

Dockerを活用したコンテナ間のHTTP通信



# DOCKERとは

- 軽量な仮想サーバーのようなもの
- アプリケーションをコンテナ化する技術
- 環境を統一し、動作の再現性を高める

## メリット

- どこでも同じ環境で動作する
- 依存関係の解決が容易
- 仮想マシンよりも軽量で高速
- スケーラブルな構成が可能



# HTTPとは

- Web上でデータをやり取りするプロトコル
- クライアント（フロントエンド）とサーバー（バックエンド）を繋ぐ
- リクエストとレスポンスの仕組み



# DOCKERのネットワーク構成

## ブリッジネットワーク

- デフォルトのネットワーク設定
- 同じネットワーク内のコンテナは 名前（サービス名） で通信できる

## オーバーレイネットワーク

- 複数のサーバーをまたぐ通信が可能
- Docker SwarmやKubernetes環境で利用



```
version: '3'
services:
  app:
    build: .
    ports:
      - "8080:8080"
    networks:
      - app_network
  db:
    image: mysql:8.0
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: sample_db
    networks:
      - app_network
networks:
  app_network:
    driver: bridge
```

# ブリッジネットワークを使用した通信例





# ポートマッピング

```
python:
  build:
    context: ./
    dockerfile: Dockerfile
  ports:
    - '8080:8080'
```



# VOLUMES の役割

```
python:
  build:
    context: ./
    dockerfile: Dockerfile
  ports:
    - '8080:8080'
  volumes:
    - ./app/Python:/code/app/Python
    - ./storage/app/json:/code/app/json
```

ホストマシンとコンテナ間でデータを共有する仕組み



# PYTHONでのHTTP API実装

```
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
from typing import List

app = FastAPI()

class ISBNList(BaseModel):
    isbn_list: List[str]

@app.post("/get-book-info")
def run_get_book_info(isbn_data: ISBNList):
    return {"isbn_list": isbn_data.isbn_list, "message": "書籍情報を取得"}
```



# フロントエンドからAPIを呼び出す (PHP実装例)

```
use Illuminate\Support\Facades\Http;

class BookController {
    public function getBookInfo(Request $request) {
        $isbnList = $request->input('isbn_list');
        $response = Http::post('http://python:8080/get-book-info', ['isbn_list' => $isbnList]);
        return $response->json();
    }
}
```



# まとめ

- Dockerを使うと、異なる言語のアプリケーションをHTTPで簡単に連携可能



ご清聴ありがとうございました！

