

【Docker Compose】 Nginx - Django - MySQL の検証環境を作成

Docker,Django, Nginx, MySQL

Docker Composeで動作するDjango環境を作成したので、備忘のため記事にしました。

作成したDocker Composeは、 Nginx - Django - MySQL のコンテナが動作します。

以下がそのコードです。

https://github.com/ruruyuki/django_server_container-

ダウンロードしたファイルを解凍すれば、すぐにコンテナが使用できるようになっています。

また、このコードは次のサイトを参考に作成しています。

<https://qiita.com/kenkono/items/6221ad12670d1ae8b1dd>

コンテナ使用手順

前提

- ・ DockerとDocker Composeがインストール済みであること。
- ・ コマンドラインで作業すること。

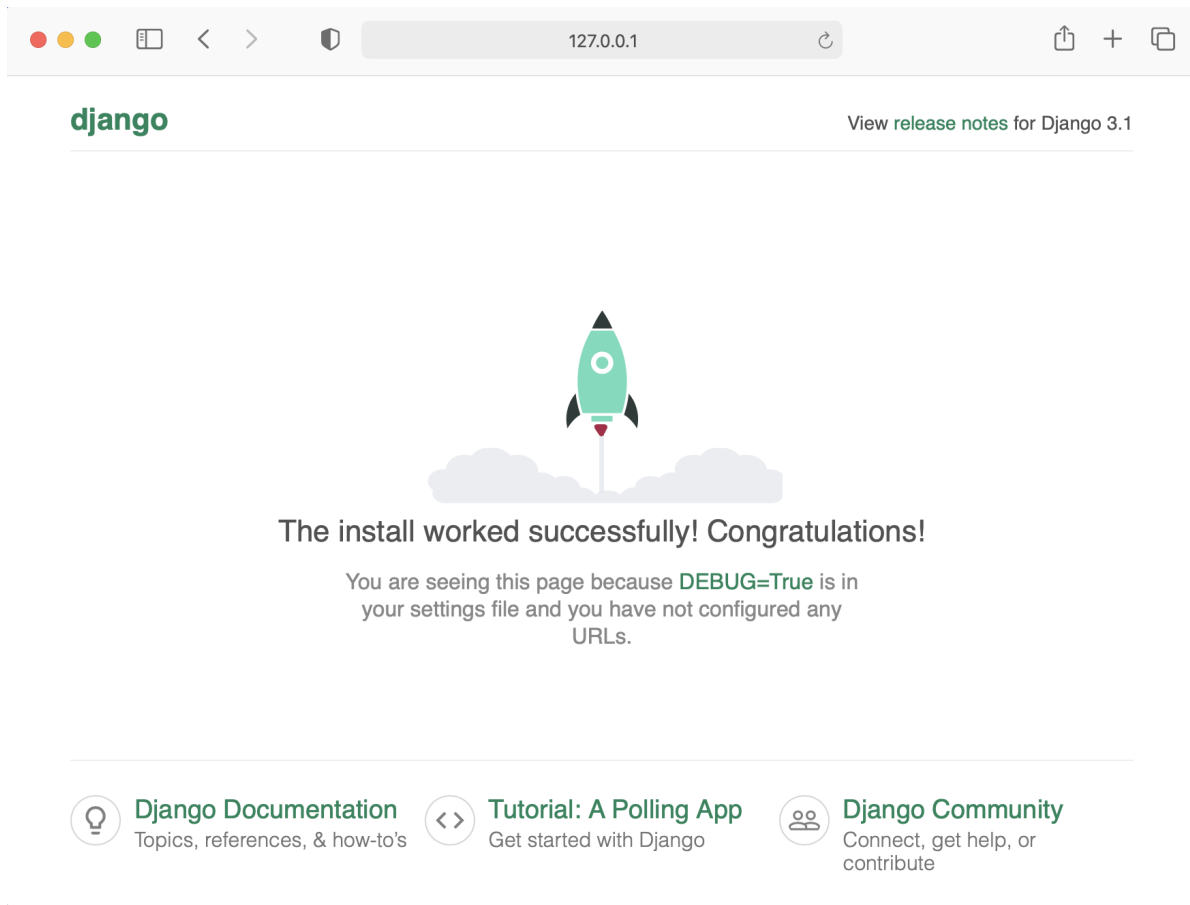
作業手順

- ・ 上記URL（Git）からダウンロードしたソースを解凍する。
- ・ 解凍したら、コマンドラインでディレクトリの中に移動する。
- ・ 以下コマンドを実行する。

```
docker-compose up -d --build
```

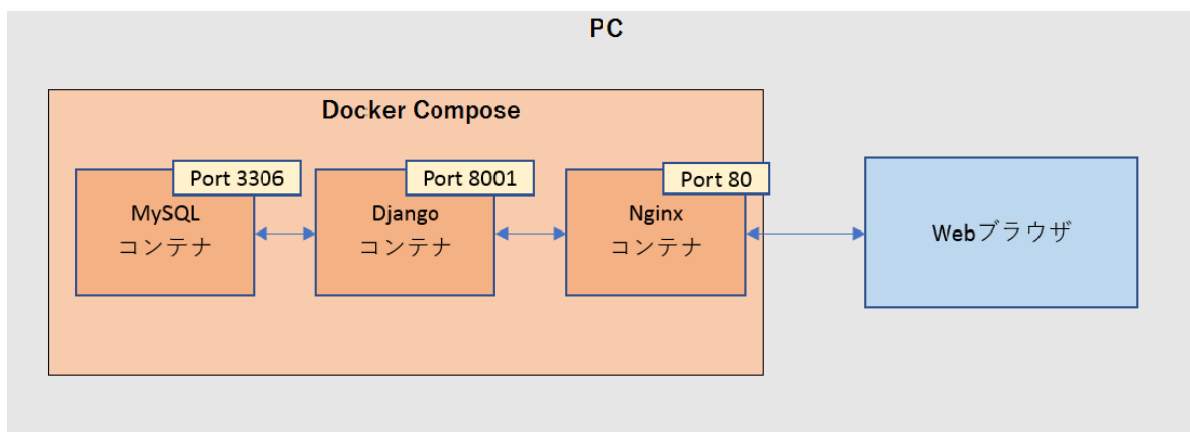
- Webブラウザで「http://127.0.0.1:80」にアクセスする。

すると、Djangoのアクセス成功画面が表示されます。



コンテナの構成と注意点

ざっくりとした構成は次のようになっています。



また、docker-compose.yml の構成は下記の通りです。

```
version: '3'

services:
  WebServer-django:
    image: nginx
    container_name: 'WebServer-django'
    volumes:
      - ./django_server/nginx/nginx.conf:/etc/nginx/nginx.conf
      - ./django_server/nginx/uwsgi_params:/etc/nginx/uwsgi_params
    ports:
      - 80:80
    depends_on:
      - python

  python:
    build: ./django_server/django
    container_name: 'WebAPServer-django'
    command: uwsgi --socket :8001 --module mysite.wsgi --py-autoreload 1 --logto
/tmp/uwsgi.log
    volumes:
      - ./django_server/django/src:/code
      - ./django_server/django/log:/tmp
    expose:
      - "8001"
    depends_on:
      - DBServer-django

  DBServer-django:
    image: mysql:5.7
    container_name: 'DBServer-django'
    command: mysqld --character-set-server=utf8mb4 --collation-
server=utf8mb4_unicode_ci
    ports:
      - "3306:3306"
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: djangodb
      MYSQL_USER: user
      MYSQL_PASSWORD: password
      TZ: 'Asia/Tokyo'
    volumes:
      - ./django_server/mysql/lib:/var/lib/mysql
      - ./django_server/mysql/etc:/etc/mysql
      - ./django_server/mysql_ini:/docker-entrypoint-initdb.d
```

コンテナをカスタムして使用する場合は、DBServer-djangoの「environment:」内のパラメータは下記構成ファイル「settings.py」と内容を合わせないとMySQLとの接続が行えなくなるので注意。

docker-compose.yml で呼び出しているDockerfile や設定ファイルなどの配置は次のようになっています。

```
django_server
├── docker-compose.yml
├── django_server
│   ├── django
│   │   ├── Dockerfile
│   │   ├── requirements.txt
│   │   ├── src
│   │   │   ├── manage.py
│   │   │   └── project
│   │   │       └── settings.py
│   │   └── uwsgi.ini
│   ├── mysql
│   ├── mysql_ini
│   │   └── init.sql
│   ├── nginx
│   │   ├── nginx.conf
│   │   └── uwsgi_params
└── Makefile
```

作成済のDjangoプロジェクトを使用する場合

以下の内容は作成済のDjangoプロジェクトが既に取り、それをDockerに移行したい方向けのものです。

まず、既にあるDjangoプロジェクトを用意します。

次に、本サイトからDL、解凍したzipファイルの中のdjango_server/django_server/django/src 配下のファイルを削除します。

そして、manage.pyなどが置いてある階層に既存プロジェクトを配置してください。

それが完了したら、以下を参考に settings.py の設定を見直します。

settings.py

```
import pymysql # 追加
pymysql.install_as_MySQLdb() # 追加

#~~~ 中略 ~~~

# ブラウザからdjangoに接続するために必要な設定。
# ローカル以外からDjangoに接続する場合はALLOWED_HOSTSにIPを追加または修正する。
ALLOWED_HOSTS = ['127.0.0.1']

#~~~ 中略 ~~~

# mysqlに接続するための設定。docker-compose.yml の設定と合わせる必要がある。
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'djangodb',
        'USER': 'user',
        'PASSWORD': 'password',
        'HOST': 'DBServer-django',
        'PORT': '3306',
    }
}
```

DBの設定をいじったら、manage.py のあるフォルダで次のコマンドを実行します。

```
docker-compose run python ./manage.py makemigrations
docker-compose run python ./manage.py migrate
```

もしくは、Makefileのあるディレクトリで下記コマンドを実行してもOKです。

こちらを実行すると上記と同じコマンドが実行されます。

```
make mm
make mg
```

マイグレートが完了したらdocker-composeを一旦停止します。

```
docker-compose down --volumes
```

その後起動すれば上手く動いているはずです。

```
docker-compose up -d --build
```

ちなみに、docker-composeコマンドについても同じくmakeコマンドを用意しています。

upは起動、downは停止です。

```
make up
make down
```

Linux用ですが、docker-composeの起動・停止状況をログに残したい場合はこちらが参考になるか
と思います。<[bashでシンプルなログ出力を行う](#)