

Linux+Apache+Django環境を 爆速構築

(Django未経験者～初心者向け)

今日やること

- Django未経験者～初心者向けに、Linux上でApacheによりDjangoを動かすところまでを構築します。

使う環境

- CentOS7
- Apache2.4
- Django3.1
- 以上をDocker上で動かします。

CentOS7の準備

- Dockerコマンドで以下を入力します。

```
docker run -d -it -p 8080:80 --privileged --name centos7_web centos:7 /sbin/init
```

- 入力したらコンテナに入りましょう。

```
docker exec -it centos7_web /bin/bash
```

Yumの更新

- CentOSのライブラリインストールコマンドであるyumを更新します。

```
yum -y update
```

つまづきポイント：SELinuxの罨

- 今回はDockerでCentOS7を動かしているのでデフォルトではインストールされていませんが、
通常インストールの場合は初期状態でSELinuxが動いています。
これがあるとそのままではApacheが動きません。
Apacheを許可する方法もありますが、複雑&動かない可能性も高いので今回は無効化を進めます。

Apacheをインストール

- 以下のコマンドでApacheをインストールします。
開発版(devel)が必要です。

```
yum -y install httpd-devel
```

- インストールしたら、起動と永続化をしておきます。

```
systemctl start httpd
```

```
systemctl enable httpd
```

Apacheの起動確認

- Apacheがうまくインストール・起動できていたら、以下のアドレスにアクセスすると初期画面が表示されるはずです。

localhost:8080

(Dockerコンテナで8080番ポートを使っている場合です。
直にCentOSを動かしている場合は80番ポートになります)

Pythonのインストール準備

- Pythonをインストールする準備をします。
以下のライブラリをインストールします。

```
yum -y install wget gcc make openssl openssl-devel mod_ssl libffi-devel
```

Pythonのインストール準備

- 各ライブラリの説明
- wget: Pythonリポジトリをダウンロードするのに必要
- gcc: configure(後述)するのに必要
- make: make(後述)するのに必要
- openssl, openssl-devel, mod_ssl: pip(後述)を使うのに必要
- libffi-devel: makeする、mod_wsgi(後述)を使うのに必要

Python3のインストール

- CentOS7にはデフォルトでPython3が入っていないので、こちらでもインストールします。
まずは保存先のディレクトリに移動します。

```
cd /usr/local/src
```

- 以下のコマンドで圧縮ファイルを取得します。

```
wget https://www.python.org/ftp/python/3.9.1/Python-3.9.1.tgz
```

Python3のインストール

- 取得に成功したら解凍します。

```
tar xvfz Python-3.9.1.tgz
```

- 解凍してできたディレクトリに移動します。

```
cd Python-3.9.1
```

Python3のインストール

- 以下のコマンドを順に実行します。

```
./configure --prefix=/usr/local/python3 CFLAGS=-fPIC --enable-shared
```

```
make
```

```
make altinstall
```

Python3のインストール

- Python3.9コマンドを実行するためにパスを通します。

```
rm -rf /usr/bin/python
```

```
ln -s /usr/local/python3/bin/python3.9 /usr/bin/python
```

```
ln -s /usr/local/src/Python-3.9.1/libpython3.9.so.1.0 /lib64/libpython3.9.so.1.0
```

パスを通したらPython3が実行可能なことを確認します。

```
python --version
```

- バージョンが表示されればインストール成功です。

Djangoのインストール準備

- Apacheでは通常は/var/www以下に公開したいデータを置きますが、Djangoでは非推奨なので、ユーザーディレクトリ配下に置くためにユーザーを作り移動します。

```
adduser django
```

```
cd /home/django
```

- パーミッションも変更します。

```
chmod 701 /home/django
```

Djangoのインストール準備

- Pythonシステムを動かすための仮想環境(venv)を作ります。
仮想環境内にのみPythonライブラリをインストールできるため、
Python環境下でライブラリ等の干渉を防ぎます。

```
python -m venv pyenv
```

- 作った仮想環境のディレクトリに移動します。

```
cd pyenv
```


Djangoのインストール準備

- 仮想環境に入ります。

```
source bin/activate
```

- Pythonのライブラリインストールモジュールのpipを更新します。

```
pip install --upgrade pip
```

Djangoのインストール・開始

- ・ 更新したpipを使ってDjangoライブラリをインストールします。

```
pip install django
```

- ・ Djangoプロジェクトを作成します。

```
django-admin startproject testproject
```

- ・ ※Git等からプロジェクトを引っ張ってくる場合はインストールまででOK

Djangoの設定

- DB設定が原因でこのままでは動かないことがあるので、一旦DBの連携設定を止めます。
- `vi testproject/settings.py`
- DATABASESにある以下の2行をコメントアウトしてください。

`'ENGINE': 'django.db.backends.sqlite3',`
`'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),`
- ※実際の開発ではPostgreSQLやMySQLを使うことが大半なので、この項目はどちらにせよ開発時に変更することになります。

Djangoのテスト起動

- ここまで来たら、以下のコマンドでDjangoテストサーバーを起動します。

```
python manage.py runserver
```

- エラーが出なければ起動が成功しています。
ターミナルorコマンドプロンプトをもうひとつ開いて、

```
docker exec -it centos7_web /bin/bash
```

- 上記のコマンドを実行してCentOSコンテナに入り、

```
curl http://localhost:8000
```

- を実行してみてください。

DjangoとApacheの連携設定

- ApacheでDjangoを使うためには、WSGIモジュールを使います。
まずはmod_wsgiライブラリをインストールします。

```
pip install mod_wsgi
```

- インストールしたら、以下のコマンドでインストール先の場所を探します。

```
find /home/django/ -name 'mod_*.so'
```

- 見つかったファイルの場所をメモしておきましょう。今回は以下になるはず。

```
/home/django/pyenv/lib/python3.9/site-packages/mod_wsgi/server/mod_wsgi-  
py39.cpython-39-x86_64-linux-gnu.so
```

DjangoとApacheの連携

- ここで仮想環境から離脱します。

```
deactivate
```

- Apacheの設定ファイルがあるディレクトリに移動します。

```
cd /etc/httpd/conf
```

DjangoとApacheの連携

- httpd.confを編集します。以下の通り記述を追加します。

```
LoadModule wsgi_module /home/django/pyenv/lib/python3.9/site-packages/mod_wsgi/server/mod_wsgi-py39.cpython-39-x86_64-linux-gnu.so
```

```
WSGIPythonHome /home/django/pyenv
```

```
WSGIScriptAlias / /home/django/pyenv/testproject/testproject/wsgi.py
```

```
WSGIPythonPath /home/django/pyenv/mysite:/var/www/django/pyenv/lib/python3.9/site-packages
```

```
<Directory /home/django/pyenv/testproject>
```

```
<Files wsgi.py>
```

```
Require all granted
```

```
</Files>
```

```
</Directory>
```

DjangoとApacheの連携

- Apacheを再起動します。

```
systemctl restart httpd
```

- この状態でブラウザから以下にアクセスします。

localhost

- ロケットが飛んでいれば成功です！

動かないときは？

- 以下の可能性があります。
- ディレクトリのパーミッション設定が間違っている
- httpd.confの書き方が間違っている
-> `apachectl configtest` で間違いがないか確認してみましょう
- SELinuxが有効化されている
-> `setenforce 0` で無効化してみましょう