

## deployの手順

今回ご紹介する手順は、基本的には以下urlに記載されている内容に沿ったものです。

<https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-gunicorn-on-ubuntu-18-04>

上記urlの内容は英語ですので、より簡潔にした上で順番に手順をお伝えしていきます。

なお、赤字部分は、Djangoのプロジェクトで付けた任意の名前を入れていきます。今回は、todolistで作成したフォルダ名に基づいて作成していきます。

また、linux(ubuntu)で作成したユーザーの名前は「ryota」を前提としています。

### 1. 初期設定

まずは初期設定を進めていきます。

#### 1-1. パッケージマネージャーの更新

ubuntuのパッケージマネージャーの更新をします。

```
sudo apt update
```

#### 1-2. pip、PostgreSQL、Nginxのインストール

デプロイに必要なソフトをまとめてインストールします。

```
sudo apt install python3-pip python3-dev libpq-dev postgresql postgresql-contrib nginx  
curl
```

### 2. PostgreSQLの設定

次に、データベースの設定をしていきます。

#### 2-1. データベースへのログイン

以下のコマンドで、PostgreSQLにログインします。

```
sudo -u postgres psql
```

#### 2-2. データベーステーブルの作成

データベーステーブルを作成します。

```
CREATE DATABASE myproject;
```

### 2-3. ユーザーの作成

データベース内でユーザーを作成します。

```
CREATE USER myprojectuser WITH PASSWORD 'password';
```

### 2-4. データベースの初期設定

データベースの初期設定を進めていきます。

```
ALTER ROLE myprojectuser SET client_encoding TO 'utf8';
```

```
ALTER ROLE myprojectuser SET default_transaction_isolation TO 'read committed';
```

```
ALTER ROLE myprojectuser SET timezone TO 'UTC';
```

### 2-5. データベースへの権限の付与

データベースに権限を与えていきます。

```
GRANT ALL PRIVILEGES ON DATABASE myproject TO myprojectuser;
```

### 2-6. データベースから抜ける

これでデータベースの設定は完了ですので、以下のコマンドでデータベースから抜けます。

```
\q
```

## 3. Gitを使ってプロジェクトをコピーしていく

ローカルで開発したtodoプロジェクトを、VPSにコピーしていきましょう。

### 3-1. Gitのインストール

まずはgitをインストールします。

```
sudo apt-get install git
```

### 3-2. リポジトリのclone

Github上にあるリポジトリをクローンします。

```
git clone https://github.com/ryotax3/todoproject.git
```

## 3. 仮想環境の構築

本番環境では、仮想環境を作っていきます。

### 3-1. 仮想環境のインストール

まずは仮想環境を実行するために必要なファイルをインストールします。

```
sudo -H pip3 install --upgrade pip  
sudo -H pip3 install virtualenv
```

### 3-2. 開発環境の作成と起動

仮想環境の作成と起動を進めていきます。

```
cd todoproject  
virtualenv todoenv
```

```
source todoenv/bin/activate
```

### 3-3. Djangoとgunicornのインストール

仮想環境内でdjangoとgunicornをインストールしていきます。

```
pip install django gunicorn psycopg2-binary
```

## 4. Djangoプロジェクトの設定

Djangoプロジェクトの設定を進めていきます。

### 4-1. settings.pyファイルの設定

settings.pyファイルの設定を進めていきます。

```
vim ~/todoproject/todoproject/settings.py
```

以下、settings.pyファイル内の設定

+++ここから

```
ALLOWED_HOSTS = ['VPSのIP', 'localhost']
```

```
DATABASES = {
```

```
'default': {  
    'ENGINE': 'django.db.backends.postgresql_psycopg2',  
    'NAME': 'myproject',  
    'USER': 'myprojectuser',  
    'PASSWORD': 'password',  
    'HOST': 'localhost',  
    'PORT': '',  
}  
}
```

```
STATIC_ROOT = os.path.join(BASE_DIR, 'static/')
```

+++ここまで

#### 4-2. Migration等のコマンドの実行

Deployに先立ち、以下のコマンドを実行していきます。

```
manage.py makemigrations
```

```
manage.py migrate
```

```
manage.py collectstatic
```

```
manage.py createsuperuser
```

#### 4-3. gunicornが起動するか確かめてみる

gunicornがサーバーとして機能しているか確認します。

```
sudo ufw allow 8000
```

```
gunicorn --bind 0.0.0.0:8000 todoproject.wsgi
```

#### 4-4. 仮想環境を抜ける

以下のコマンドで、仮想環境を抜けます。

```
deactivate
```

#### 5. gunicornの設定

デーモンプロセスでアクセスを受けることができるように、gunicornの設定を進めていきます。

### 5-1. gunicorn socketの作成

Nginxからアクセスを受けるソケットの作成をしていきます。

```
sudo vim /etc/systemd/system/gunicorn.socket
```

+++ここから

```
[Unit]
```

```
Description=gunicorn socket
```

```
[Socket]
```

```
ListenStream=/run/gunicorn.sock
```

```
[Install]
```

```
WantedBy=sockets.target
```

+++ここまで

### 5-2. gunicorn serviceの作成

socketからアクセスを受けた時に実行するファイルの設定をしていきます。

```
sudo vim /etc/systemd/system/gunicorn.service
```

+++ここから

```
[Unit]
```

```
Description=gunicorn daemon
```

```
Requires=gunicorn.socket
```

```
After=network.target
```

```
[Service]
```

```
User=ryota
```

```
Group=www-data
```

```
WorkingDirectory=/home/ryota/todoproject
```

```
ExecStart=/home/ryota/todoproject/todoenv/bin/gunicorn \
```

```
--access-logfile - \
```

```
--workers 3 \
```

```
--bind unix:/run/gunicorn.sock \
```

```
todoproject.wsgi:application
```

[Install]

WantedBy=multi-user.target

+++ここまで

### 5-3. gunicorn socketの実行

gunicorn.socketファイルを実行していきます。

```
sudo systemctl start gunicorn.socket
```

```
sudo systemctl enable gunicorn.socket
```

## 6. Nginxの設定

最後にNginxの設定を進めていきましょう。

### 6-1. Nginxの繋ぎこみの設定

ブラウザから受けたリクエストをgunicornに繋げるための設定を行っていきます。

```
sudo vim /etc/nginx/sites-available/todoproject
```

+++ここから

```
server {  
    listen 80;  
    server_name VPSのIP;  
  
    location = /favicon.ico { access_log off; log_not_found off; }  
    location /static/ {  
        root /home/ryota/todoproject;  
    }  
  
    location / {  
        include proxy_params;  
        proxy_pass http://unix:/run/gunicorn.sock;  
    }  
}
```

### 6-2. 設定の反映

シンボリックリンクの設定をすることで、上記の内容を反映させていきます。

```
sudo ln -s /etc/nginx/sites-available/todoproject /etc/nginx/sites-enabled
```

### 6-3. Nginxの再起動

設定を反映させるために再起動します。

```
sudo systemctl restart nginx
```

### 6-4. Firewallの設定

ブラウザからのアクセスを許可するための設定をします。

```
sudo ufw allow 'Nginx Full'
```

これで設定は完了です。IPアドレスをブラウザに入力すると、todolistが表示されます。