

deploy django app to heroku

Deploy Django app to heroku for free

Prerequisite:

- [Django project](#)
- an account on [heroku](#)
- Note the database should be postgresql

Postgres database in django

install postgres

```
sudo apt install postgresql postgresql-contrib
```

A user with username postgres will be created automatically, so login as postgres user by:

```
sudo su - postgres
```

When inside the PostgreSQL shell session, execute the following command to enter the console:

```
psql
```

Create database and user

```
CREATE DATABASE heroku_demo_db;  
CREATE USER rahul WITH PASSWORD '123456';  
ALTER ROLE rahul SET client_encoding TO 'utf-8';  
GRANT ALL PRIVILEGES ON DATABASE heroku_demo_db TO rahul;
```

install packages for postgresql for django

```
pip install psycopg2
```

if this shows some error:

```
sudo apt-get install libpq-dev python3-dev
```

```
# settings.py
```

```
DATABASES = {
```

```
'default': {
    'ENGINE': 'django.db.backends.postgresql_psycopg2',
    'NAME': 'heroku_demo_db',
    'USER': 'rahul',
    'PASSWORD': '123456',
    'HOST': '127.0.0.1',
    'PORT': '5432',
}
```

Then lets migrate the datas

```
python manage.py makemigrations
python manage.py migrate
```

To see the databases:

```
sudo su - postgres
psql

<!-- To list all tables -->
\l

<!-- To use heroku_demo_db -->
\c heroku_demo_db

<!-- To list all databases in this table -->
\dt
```

Install heroku cli

```
sudo snap install --classic heroku
```

Authenticate heroku

```
heroku login
```

Configure django app for heroku

```
touch Procfile
```

Open the Procfile and add this:

```
web: gunicorn projectname.wsgi --log-file -
```

install some packages

```
pip install gunicorn dj-database-url whitenoise
```

```
pip install psycopg2
```

if this shows some error:

```
sudo apt-get install libpq-dev python3-dev
```

add these to the requirements.txt

```
pip freeze > requirements.txt
```

```
# settings.py
```

```
MIDDLEWARE = [  
    'whitenoise.middleware.WhiteNoiseMiddleware',  
    ...  
]
```

```
# Add configuration for static files storage using whitenoise
```

```
STATICFILES_STORAGE =  
'whitenoise.storage.CompressedManifestStaticFilesStorage'
```

```
# Database settings
```

```
import dj_database_url  
prod_db = dj_database_url.config(conn_max_age=500)  
DATABASES['default'].update(prod_db)
```

Connecting to heroku

Lets create an app in heroku, the name should be unique

```
heroku create heroku-deploy-django
```

Make sure that the allowed host is set

```
git init
heroku git:remote -a heroku-deploy-django
git add --all
git commit -m "initial commit"
git push heroku master
```

If you get an error message with collectstatic, simply disable it by instructing Heroku to ignore running the `manage.py collectstatic` command during the deployment process.

```
heroku config:set DISABLE_COLLECTSTATIC=1
```

Otherwise you have to set static url and static root correctly

```
heroku run python manage.py migrate
heroku run python manage.py createsuperuser
```