

# Deploying Django App on Heroku with Postgres as Backend



Harman Deep Singh  
Dec 9, 2018 · 3 min read



Photo by rawpixel on Unsplash

---

*Deploying a webapp on heroku is usually a difficult task for a beginner especially when you want to use postgres as backend . This post is step by step guide to help you deploy easily.*

---

*You can checkout my **deployed app** and the code on **github**. If you want a sample app you can use **this app**.*

**Step 1 :** Create a Django app( if you haven't already created).

**Step 2 :** Download & Install Heroku Command Line Interface.

**Step 3 :** Open in terminal your main project folder. Create and activate virtual environment.

- `cd main_project_folder`
- `virtualenv venv`
- `source venv/bin/activate`

**Step 4 :** Install the dependencies and these packages.

- `pip install django gunicorn whitenoise dj-database-url psycopg2`

**Step 5 :** Create a Procfile `Procfile` and add the following line.

```
web: gunicorn <nameOfProject>.wsgi --log-file -
```

Here `<nameOfProject>` is name of folder which includes `settings.py` .

**Step 6 :** Create requirements file `requirements.txt` by running `pip freeze > requirements.txt` .

**Step 7 :** Create runtime file `runtime.txt` and add the following `python-3.6.7` or any other runtime supported by heroku.

**Step 8 :** Initialise git repository (if not already done) and create `.gitignore` file. Also add and save changes in git.

- `git init`
- `touch .gitignore` and add the following lines to the file

```
*.log  
*.pot  
*.pyc  
__pycache__/  
local_settings.py
```

```
db.sqlite3  
venv/
```

Add all the changes to git by `git add .` and commit them by `git commit -m "message" .`



**Step 9 :** Login to Heroku terminal by `heroku login` and create heroku app by `heroku create` OR `heroku create nameofapp` for custom name.

**Step 10 :** Now let's modify the `settings.py` file.

- Modify allowed hosts by adding `nameofapp.herokuapp.com`

```
ALLOWED_HOSTS = ['0.0.0.0', 'localhost', '127.0.0.1',  
'nameofapp.herokuapp.com']
```

- Set environment variables (`SECRET_KEY` and others)

```
SECRET_KEY = os.environ.get('SECRET_KEY')  
EMAIL_HOST_USER = os.environ.get('EMAIL_HOST_USER')  
EMAIL_HOST_PASSWORD = os.environ.get('EMAIL_HOST_PASSWORD')
```

and running in the terminal `heroku config:set SECRET_KEY=secretkey`

- Set `DEBUG = False` and you could use `heroku logs` for debugging.
- Modify `INSTALLED_APPS` by adding `whitenoise.runserver_nostatic` , also `MIDDLEWARE` by adding `'whitenoise.middleware.WhiteNoiseMiddleware'` , and add

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

- Add `import dj_database_url` at the top. After `DATABASES` add

```
db_from_env = dj_database_url.config(conn_max_age=600)
DATABASES['default'].update(db_from_env)
```

- Also ensure you file has following variables `STATIC_URL` , `STATIC_ROOT` , `STATICFILES_DIRS` accordingly. Just for example my file has

```
STATIC_URL = '/static/'

#location where django collect all static files
STATIC_ROOT = os.path.join(BASE_DIR, 'static')

# location where you will store your static files
STATICFILES_DIRS = [os.path.join(BASE_DIR, 'project_name/static')]
]
```

- Also ensure the media file variables.

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
MEDIA_URL = '/media/'
```

- Commit and save changes in git by `git add .` and `git commit -m "change settings"`

## Step 11: Adding and configuring Postgres

- The following commands create postgresql database on heroku for you app and fetch its url.

```
heroku addons:create heroku-postgresql:hobby-dev  
heroku config -s | grep DATABASE_URL
```

You can also run `heroku pg:info` to get details of your database on heroku. Add-on: will give you `nameOfHerokuDB` .

- Now lets push local database to herokuDB

```
push local database:PGUSER=postgres PGPASSWORD=password heroku  
pg:push postgres://name_of_host/name_of_local_database  
nameOfHerokuDB
```

For example

```
PGUSER=postgres PGPASSWORD=possword heroku pg:push  
postgres://localhost/myDB postgresql-convex-23876
```

**Step 12 :** Disable Collectstatic and push the files to heroku.

```
heroku config:set DISABLE_COLLECTSTATIC=1  
  
git push heroku master
```

Open the deployed app by `heroku open` .

You can also `heroku bash` to connect with server's terminal and run django commands directly.

You can checkout my **deployed app here** and the code on **github**.

***THAT'S ALL!!***

Congrats!!! You have deployed you Django app to heroku. I hope you found this post helpful while deploying your app. If you have any suggestions to improve this post, please share with me in the comments or add an issue in github repo.

[Heroku](#)[Django](#)[Postgresql](#)[Python](#)[Web Development](#)[About](#) [Help](#) [Legal](#)

Get the Medium app

