

Perfect! Let's go **step by step with a full, detailed guide to deploy a Django project for free on Render**, which is one of the easiest and most reliable free hosts right now. I'll include **all commands, settings, and explanations**, so you can literally deploy in under 15 minutes.

---

## Step 1: Prepare Your Django Project for Deployment

### 1.1 Install Required Packages

Open your terminal in your project folder and run:

```
pip install gunicorn dj-database-url whitenoise python-decouple psycopg2-binary
```

```
(env) D:\crud_example>pip install gunicorn dj-database-url whitenoise python-decouple psycopg2-binary
Collecting gunicorn
  Downloading gunicorn-25.1.0-py3-none-any.whl.metadata (5.5 kB)
Collecting dj-database-url
  Downloading dj_database_url-3.1.2-py3-none-any.whl.metadata (13 kB)
Collecting whitenoise
  Downloading whitenoise-6.12.0-py3-none-any.whl.metadata (3.7 kB)
```

What these do:

- **gunicorn** → Production server (instead of `manage.py runserver`)
- **dj-database-url** → Easily read database URL (like PostgreSQL)
- **whitenoise** → Serve static files (CSS/JS) in production
- **python-decouple** → Store sensitive data (`SECRET_KEY`) in `.env`
- **psycopg2-binary** → PostgreSQL driver (Render gives free Postgres)

---

### 1.2 Create a `.env` File

In your project root:

```
SECRET_KEY='your-secret-key'
DEBUG=False
```

Make sure you **never commit** `.env` to GitHub. Add it to `.gitignore`.

---

### 1.3 Update `settings.py`

Add imports at the top

```
import os
from decouple import config
import dj_database_url
```

## Set secret key and debug

```
SECRET_KEY = config('SECRET_KEY')
DEBUG = config('DEBUG', default=False, cast=bool)
```

## Set allowed hosts

```
ALLOWED_HOSTS = ['your-app-name.onrender.com'] # Replace with your Render
URL
```

## Database configuration

At the bottom of `settings.py`, replace default SQLite for production:

```
DATABASES = {
    'default': dj_database_url.config(
        default='sqlite:///db.sqlite3', # fallback to SQLite
        conn_max_age=600
    )
}
```

---

## 1.4 Configure Static Files

### Add to `settings.py`:

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

# Whitenoise for serving static files
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'whitenoise.middleware.WhiteNoiseMiddleware', # Add here
    # ... rest of middleware
]
STATICFILES_STORAGE =
'whitenoise.storage.CompressedManifestStaticFilesStorage'
```

```
D:\crud_example\crud_example\settings.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
settings.py models.py views.py urls.py urls.py record_list.html record_data

100
101
102 # Static files (CSS, JavaScript, Images)
103 # https://docs.djangoproject.com/en/5.2/howto/static-files/
104 import os
105
106 STATIC_URL = '/static/'
107 STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')
108
109 # Whitenoise for serving static files
110 MIDDLEWARE = [
111     'django.middleware.security.SecurityMiddleware',
112     'whitenoise.middleware.WhiteNoiseMiddleware', # Add here
113     # ... rest of middleware
114 ]
115 STATICFILES_STORAGE = 'whitenoise.storage.CompressedManifestStaticFilesS
116
117 # Default primary key field type
118 # https://docs.djangoproject.com/en/5.2/ref/settings/#default-auto-field
119
120 DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
121
122
123
124
125
126
127
```

Then, collect static files locally to check everything works:

```
python manage.py collectstatic
```

```
(env) D:\crud_example>python manage.py collectstatic
```

---

## 1.5 Create requirements.txt

Render uses this to install dependencies:

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

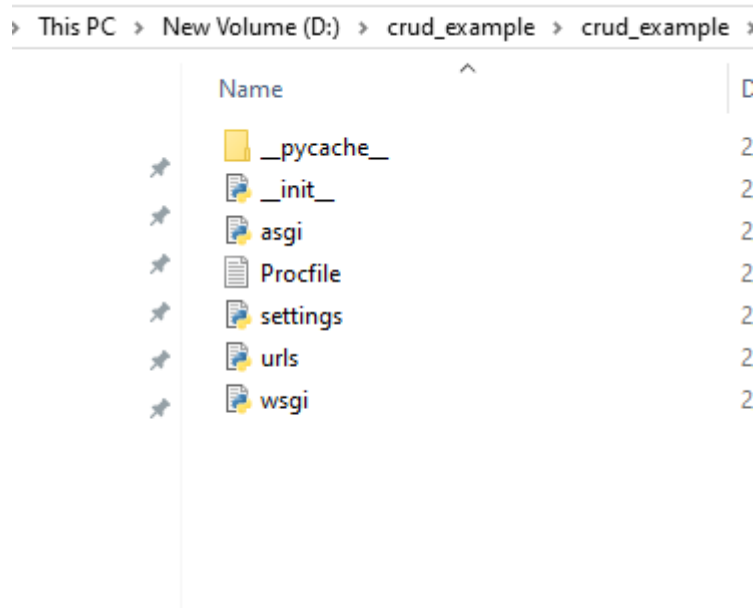
---

## 1.6 Create Procfile

In your project root, create a file named Procfile (no extension) with:

```
web: gunicorn myproject.wsgi
```

Replace myproject with your Django project folder name (where wsgi.py is located).



Note:-

### Embed migrations in your code (hacky, not recommended)

You could technically run migrations in wsgi.py:

```
import os
from django.core.wsgi import get_wsgi_application
from django.core.management import call_command

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "crud_example.settings")

# Run migrations automatically (dangerous on each start!)
call_command('migrate', interactive=False)
application = get_wsgi_application()
```

- Pros: works on free Render
-

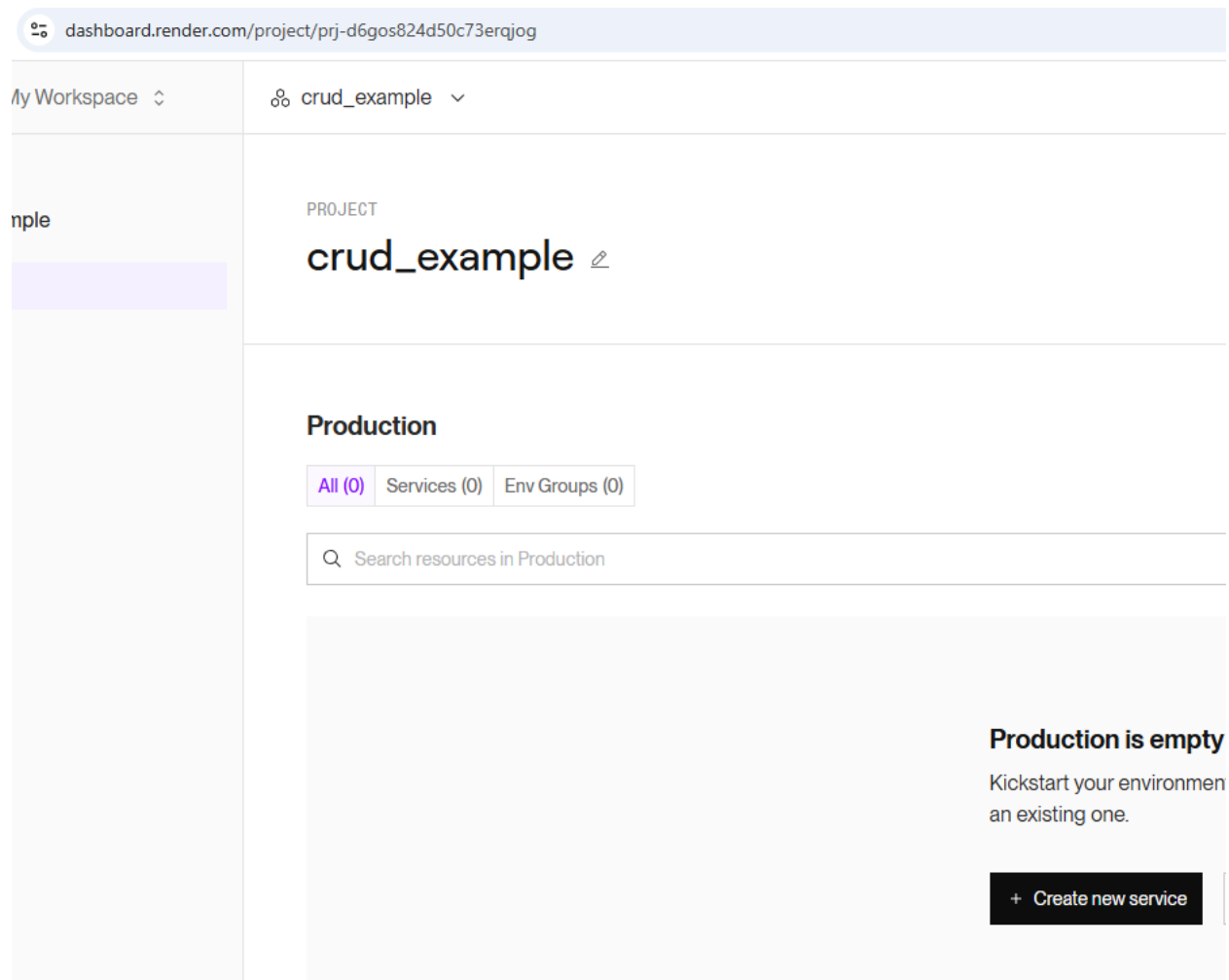
## Step 2: Push to GitHub

Render connects to GitHub for deployment.

```
git init
git add .
git commit -m "Ready for Render deployment"
git branch -M main
git remote add origin https://github.com/username/repo.git
git push -u origin main
```

## Step 3: Create a Free Render Account

1. Go to [Render.com](https://render.com) → Sign up.
2. Connect your **GitHub account**.



The screenshot shows the Render dashboard for a project named 'crud\_example'. The breadcrumb navigation at the top reads 'dashboard.render.com/project/prj-d6gos824d50c73erqjog'. Below the navigation, the workspace is identified as 'my Workspace' and the project as 'crud\_example'. The main content area shows the project name 'crud\_example' under the 'PROJECT' heading. Below this, the 'Production' environment is displayed, with tabs for 'All (0)', 'Services (0)', and 'Env Groups (0)'. A search bar is present with the text 'Search resources in Production'. At the bottom right, a message states 'Production is empty' and 'Kickstart your environment with an existing one.', accompanied by a '+ Create new service' button.

And click on new web services:-

The screenshot shows the Render dashboard interface. At the top, the browser address bar displays the URL: `dashboard.render.com/project/prj-d6gos824d50c73erqjog/environment/evm-d6gos824d50c73erqjp0/new`. Below the address bar, the navigation bar includes a profile icon, a dropdown menu labeled 'My Workspace', and a breadcrumb trail showing 'crud\_example / Production'. The main content area is titled 'Create a new Service' and includes the instruction 'Select the type of service you'd like to create'. There are four service cards displayed in a 2x2 grid:

- Static Sites**: Static content served over a global CDN. Ideal for frontend, blogs, and content sites. Includes a link 'New Static Site →'.
- Web Services**: Dynamic web app. Ideal for full-stack apps, API servers, and mobile backends. Includes a link 'New Web Service →'. This card is highlighted with a blue border.
- Cron Jobs**: Short-lived tasks that run on a periodic schedule. Includes a link 'New Cron Job →'.
- Postgres**: Relational data storage. Supports point-in-time recovery, read replicas, and high availability. Includes a link 'New Postgres →'.

On the right side of the dashboard, there is a vertical sidebar with a search icon and a list of items, including 'W fr' and 'Ni'.

And then connect your github account with all repositories

Browser tabs: Login, ChatGPT, New Web Service - Render, digitalommaurya/crud-ex, my-django-project/projec, my-django-project

Address bar: dashboard.render.com/project/prj-d6gos824d50c73erqjog/environment/evm-d6gos824d50c73erqjp0/web/new

Workspace: My Workspace

Breadcrumbs: crud\_example / Production / New Web Service

## New Web Service

Source Code

Git Provider | Public Git Repository | Existing Image

Search

- digitalommaurya / crud-example 37m ago
- digitalommaurya / my-django-project 2h ago
- digitalommaurya / php Nov 15, 2025
- digitalommaurya / mem-stack-projects Oct 25, 2025
- digitalommaurya / omsir Sep 15, 2025

And then

Browser tabs: Login, ChatGPT, New Web Service · Rende, digitalommaurya/crud-ex, my-django-pr

Address bar: dashboard.render.com/project/prj-d6gos824d50c73erqjog/environment/evm-d6gos824d50c73erqjp0/web/new

Navigation: My Workspace, crud\_example / Production / New Web Service

## New Web Service

It looks like you're using **Django**, so we've autofilled some fields accordingly.

### Source Code

digitalommaurya / crud-example · 38m ago

### Name

A unique name for your web service.

crud-example1

### Project Optional

Add this web service to a **project** once it's created.

crud\_example

### Language

Choose the **runtime environment** for this service.

Python 3

### Branch

The Git branch to build and deploy.

main

### Region

Your services in the same **region** can communicate over a **private network**. You

Oregon (US West)

And you will see



