


Wagtail

Wagtail 



Django 

CMS 


Wagtail 


1.





- Python 3.7+
- PostgreSQL/MySQL/SQLite ( PostgreSQL)
- Node.js ()





```
# 
python -m venv venv
source venv/bin/activate # Linux/Mac
# venv\Scripts\activate # Windows

#  Wagtail
pip install wagtail

# 
wagtail start mysite
cd mysite

# 
pip install -r requirements.txt

# 
python manage.py migrate

# 
python manage.py createsuperuser
```

```
# 000000
```

```
python manage.py runserver
```

```
00 http://localhost:8000 00000 http://localhost:8000/admin 00000
```

2. 0000000 (Docker 00)

docker-compose.yml 00

```
version: '3.8'

services:
  db:
    image: postgres:13
    environment:
      POSTGRES_DB: wagtail
      POSTGRES_USER: wagtail
      POSTGRES_PASSWORD: wagtail
    volumes:
      - postgres_data:/var/lib/postgresql/data

  web:
    build: .
    command: gunicorn mysite.wsgi:application --bind 0.0.0.0:8000
    volumes:
      - ./code
    ports:
      - "8000:8000"
    depends_on:
      - db
    environment:
      DATABASE_URL: postgres://wagtail:wagtail@db:5432/wagtail
      SECRET_KEY: your-secret-key-here

volumes:
  postgres_data:
```

Dockerfile

```
FROM python:3.9-slim

WORKDIR /code

ENV PYTHONDONTWRITEBYTECODE 1
ENV PYTHONUNBUFFERED 1


RUN apt-get update && apt-get install -y \
    libpq-dev \
    gcc \
    && rm -rf /var/lib/apt/lists/*

COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

EXPOSE 8000
```



```
# 
docker-compose up -d --build

# 
docker-compose exec web python manage.py migrate

# 
docker-compose exec web python manage.py createsuperuser

# 
docker-compose exec web python manage.py collectstatic --no-input
```

3.  ()

☐☐ Gunicorn + Nginx

1. ☐☐ Gunicorn:

```
pip install gunicorn
```

2. ☐☐ Gunicorn ☐☐☐ (/etc/systemd/system/gunicorn.service):

```
[Unit]
Description=gunicorn daemon
After=network.target

[Service]
User=youruser
Group=www-data
WorkingDirectory=/path/to/your/project
ExecStart=/path/to/venv/bin/gunicorn --access-logfile - --workers 3 --bind unix:/run/gunicorn.sock
mysite.wsgi:application

[Install]
WantedBy=multi-user.target
```

3. Nginx ☐☐☐ (/etc/nginx/sites-available/yourdomain):

```
server {
    listen 80;
    server_name yourdomain.com;

    location = /favicon.ico { access_log off; log_not_found off; }

    location /static/ {
        root /path/to/your/project;
    }

    location /media/ {
        root /path/to/your/project;
    }

    location / {
        include proxy_params;
    }
}
```

```
    proxy_pass http://unix:/run/gunicorn.sock;
  }
}
```

4. `sudo ln -s /etc/nginx/sites-available/yourdomain /etc/nginx/sites-enabled`

```
sudo ln -s /etc/nginx/sites-available/yourdomain /etc/nginx/sites-enabled
sudo systemctl restart nginx
```

4. `requirements.txt`

Heroku `Procfile`

1. `Procfile`:

```
web: gunicorn mysite.wsgi:application --bind 0.0.0.0:$PORT
```

2. `runtime.txt` `Python` :

```
python-3.9.7
```

3. `heroku` :

```
heroku create
heroku addons:create heroku-postgresql:hobby-dev
heroku config:set SECRET_KEY=your-secret-key
heroku config:set DISABLE_COLLECTSTATIC=1
git push heroku main
heroku run python manage.py migrate
heroku run python manage.py createsuperuser
```

AWS Elastic Beanstalk `requirements.txt`

1. `requirements.txt` `requirements.txt`

2. `.ebextensions/django.config` :

```
option_settings:
  aws:elasticbeanstalk:container:python:
```

```
WSGIPath: mysite/wsgi.py
```

3. `eb init` :

```
eb init -p python-3.9 wagtail-app
eb create wagtail-env
```

5. `eb init`

1. `settings.py` :

```
# settings.py
DEBUG = False
ALLOWED_HOSTS = ['yourdomain.com', 'localhost']
SECURE_SSL_REDIRECT = True
SESSION_COOKIE_SECURE = True
CSRF_COOKIE_SECURE = True
```

2. `requirements.txt` :

- `django`
- `django-cors-headers`
- `django-heroku` CDN

3. `manage.py` :

```
# backup
docker-compose exec db pg_dump -U wagtail wagtail > backup.sql

# restore
tar -czvf media_backup.tar.gz /path/to/media
```

6. `eb init`

1. `eb init` **404** :

- `collectstatic`
- `nginx` Nginx/Apache `nginx`

2. `eb init` :

- `django`
- `DATABASE_URL` `settings.py` `django`

3. `eb init` :

- 1 Gunicorn worker (2-4 * CPU)
- 1 (Redis/Memcached)

1

Wagtail CMS 1

Revision #1

Created 20 May 2025 01:08:57 by Admin

Updated 20 May 2025 01:10:58 by Admin