**Setup NGINX Web Server with Wagtail CMS using uWSGI**

**Create a new MySQL database and user for Wagtail CMS. The MySQL user should be restricted to the Wagtail CMS database only.**

Assuming mysql database server is installed:

create database wagtail;

create user 'wagcms'@'localhost' identified by 'wagpassword';

grant all on wagtail.\* to 'wagcms'@'localhost' ;

show grants for 'wagcms'@'localhost';

**Create a new Linux user and deploy the application within the user's home directory.**

adduser wagtail

su – wagtail

mkdir wagtail-cms

**Use Python virtual environment when installing Wagtail CMS.**

apt-get install build-essential python-dev

apt-get install python3-pip

apt install virtualenv

cd wagtail-cms

pip install wagtail

wagtail start mydemosite (folder will be created with name mydemosite)

cd mydemosite

virtualenv --python=python3 env

source env/bin/activate

pip install -r requirements.txt

./manage.py migrate (By default sqllite3 database will be used)

./manage.py createsuperuser

enter email

enter user

enter password

./manage.py runserver ip\_address:port

wagtail project now can be accessed on browser via http://ip\_address:port

**Configure Wagtail CMS to use the MySQL database.**

Make sure that following library is installed with root:  
apt-get install libmysqlclient-dev

We have to update the db settings to use the mysql database:

In project folder -> settings -> base.py:

add the syntax below:

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'mydatabase',

'USER': 'mydatabaseuser',

'PASSWORD': 'mypassword',

'HOST': '127.0.0.1',

'PORT': '6033',

}

}

Then activate the virtual environment and run below commands:

pip3 install mysqlclient

./manage.py migrate

Now run the server:

./manage.py runserver ipaddress:port

**Use uWSGI to run Wagtail. Make sure to configure uWSGI run the Wagtail as the Linux user you have created.**

Installed via command below:  
apt-get install uwsgi

Created uwsgi file at path *etc*uwsgi/apps-enabled/mydemosite.ini

File Content:

[uwsgi]

chdir = /home/wagtail/wagtail-cms/mydemosite

env = DJANGO\_SETTINGS\_MODULE = mydemosite.settings

wsgi-file = /home/wagtail/wagtail-cms/mydemosite/mydemosite/wsgi.py

workers = 1

max-requests = 500

plugins = python3

home = /home/wagtail/wagtail-cms/mydemosite/env

pyhtonpath = /home/wagtail/wagtail-cms/mydemosite/env/lib/python3.8/site-packages

processes = 3

threads = 1

master = true

die-on-term = true

uid = wagtail

gid = wagtail

socket = /tmp/mywagtail\_project.sock

chmod-socket = 664

vacuum = true

Save the file and restart the service:  
service uwsgi restart

Now to check if the uwsgi process is running with same linux user with we have created above:

ps aux | grep wagtail (it will show output like this)

wagtail 878 0.0 2.5 103028 51012 ? S 06:03 0:00 /usr/bin/uwsgi --ini /usr/share/uwsgi/conf/default.ini --ini /etc/uwsgi/apps-enabled/mydemosite.ini --daemonize /var/log/uwsgi/app/mydemosite.log

**Make sure that NGINX version, OS version and PHP version in HTTP headers is not publicly visible when browsing via curl command e.g. curl http://localhost/ or http://IP\_ADDRESS\_OF\_YOUR\_VM.**

First we check the headers for such info via command below:

curl -v <http://localhost/> OR curl -IsL <http://localhost/> | grep -i server

Found OS and Nginx version in header info. So we open the file and edit below:

vi /etc/nginx/nginx.conf

un-comment line or add if not there

server\_tokens off;

save config file and restart nginx service

In addition to remove Server block add below line in nginx.conf file and restart service:

more\_clear\_headers Server;

**Useful Links:**

[https://docs.djangoproject.com/en/4.0/ref/settings/#databases](https://docs.djangoproject.com/en/4.0/ref/settings/" \l "databases)

<https://prog.world/lazy-deploy-django-project-uwsgi-nginx-ubuntu-20-04/>