**Setup NGINX Web Server with Wagtail CMS using Gunicorn/ Supervisor**

**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 wagtail2;

create user 'wagcms2'@'localhost' identified by 'wagpassword2';

grant all on wagtail2.\* to 'wagcms2'@'localhost' ;

show grants for 'wagcms2'@'localhost';

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

adduser wagtail2 --shell /usr/sbin/nologin

id wagtail2

cat /etc/passwd | grep wagtail2

mkdir /home/wagtail2/wagtail\_gunicorn

su - wagtail2

This account is currently not available. (user has no shell)

## To Run commands as spesific user that has no shell ##

runuser -u username -- command

runuser -u wagtail2 -- wagtail start wagtail\_gunicorn

**Installing Wagtail CMS.**

apt install python3.8-venv

pip install wagtail

In user’s home directory run below commands:

runuser -u wagtail2 -- wagtail start wagtail\_gunicorn

runuser -u wagtail2 -- python3 -m venv env

source env/bin/activate

runuser -u wagtail2 -- pip install -r requirements.txt

deactivate

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

First install following package:

apt install libmysqlclient-dev

Now in virtual environment install mysql client using pip:

To Use MYSQL as database for this project we need to add this block in > project\_folder > settings > base.py

Connection String for mysql for python application:

DATABASES = {

'default': {

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

'NAME': 'wagtail2',

'USER': 'wagcms2',

'PASSWORD': 'wagpassword2',

'HOST': '127.0.0.1',

'PORT': '3306',

}

}

In user’s home run commands below:

source env/bin/activate

runuser -u wagtail2 -- pip3 install mysqlclient

runuser -u wagtail2 -- ./manage.py migrate (this migrates data into your mysqldb)

Now Create a admin user for wagtail admin portal:

runuser -u wagtail2 -- ./manage.py createsuperuser

Testing the application via below command:

runuser -u wagtail2 -- ./manage.py runserver 192.168.10.14:8000

OR

runuser -u wagtail2 -- pip3 install gunicorn

gunicorn --bind 0.0.0.0:8000 wagtail\_gunicorn.wsgi:application

Note that above two runserver commands run a process at provided ip and port but it will be terminated after you exit from this

deactivate

**Write a Gunicorn Python configuration file for your use case. Gunicorn must use this configuration file to run.**

Install below packages:  
apt-get install supervisor

apt-get install python-setuptools

##### Example File #####

[program:gunicorn]

command=/usr/local/bin/gunicorn main:application -c /path/to/project/gunicorn.conf.py

directory=/path/to/project

user=nobody

autorestart=true

redirect\_stderr=true

#################

Create a new file at below path:

vi /etc/supervisor/conf.d/wagtail\_gunicorn.conf

[program:wagtail\_gunicorn]

directory = /home/wagtail2/wagtail\_gunicorn

command = /home/wagtail2/wagtail\_gunicorn/env/bin/gunicorn --bind 127.0.0.1:8000 wagtail\_gunicorn.wsgi:application

workers = 3

autostart=true

autorestart=true

stderr\_logfile = /home/wagtail2/logs/gunicorn.out.log

stdout\_logfile = /home/wagtail2/logs/gunicorn.err.log

user = wagtail2

group = www-data

environment=LANG=en\_US.UTF-8,LC\_ALL=en\_US.UTF-8

Paste above content and restart the service of supervisor:

service supervisor restart

Check if the supercisor process is running.

supervisorctl status

It will show the process name that you have entered in gunicorn file as shown below:

wagtail\_gunicorn RUNNING 67124

Now we need to configure the nginx vhost for our application to accepts the requests on.

Create file at path below:

vi /etc/nginx/sites-enabled/gunicorn-wagtail.conf

server {

listen 80;

server\_name gunicornwag.local;

root /home/wagtail2/wagail\_gunicorn;

access\_log /var/log/nginx/guniwag-access.log;

error\_log /var/log/nginx/guniwag-error.log;

location / {

include proxy\_params;

proxy\_pass http://127.0.0.1:8000;

}

}

Paste above content and restart the service of nginx web server:

service nginx restart

Test the url given in vhost to check if the application is working fine:

http://gunicornwag.local

**Useful Links:**

<https://www.cyberciti.biz/open-source/command-line-hacks/linux-run-command-as-different-user/>

<https://cloudwafer.com/blog/how-to-install-and-configure-supervisor-on-ubuntu-16-04/>

<https://www.codingforentrepreneurs.com/blog/hello-linux-setup-gunicorn-and-supervisor/>