Asynchronous Web Chat Project

Guidelines for production server

September 13, 2022

Those sections details how to setup up the VM for the production environment. **You must first reproduce the same steps than for the development environment and then move one with the following steps.**

1 WSGI server

1. Install the application requirements out of the virtual environment
2. Install Gunicorn using the pip tool from Python 3.8.
3. Try launching the application with gunicorn :

   ```
   $ : gunicorn —bind 0.0.0.0:8000 ChatServerPlayground.wsgi
   ```

   If it does not work, then there is something wrong with your setup. **You must fix it before moving onto next steps.**

1.1 Gunicorn setup

1. Create Gunicorn socket and service files :

   ```
   $ : cd /etc/systemd/system
   $ : sudo cp ∼/web/files/gunicorn.socket gunicorn.socket
   $ : sudo cp ∼/web/files/gunicorn.service gunicorn.service
   ```

2. Enable Gunicorn to start whenever the VM starts :

   ```
   $ : sudo systemctl start gunicorn.socket
   $ : sudo systemctl enable gunicorn.socket
   ```

1.2 Nginx setup

1. Create the Nginx configuration :

```
$ : cd /etc/nginx/sites-available
$ : sudo cp ∼/WebChatApp/ProjectFiles/ChatServerPlayground ./
```

2. Update Nginx config file at ”/etc/nginx/nginx.conf” with the following :

```
http{
client_max_body_size 10M;
}
```

3. Set up the firewall :

```
$ : sudo ln -s ChatServerPlayground ../nginx/sites-enabled
$ : sudo nginx -t
$ : sudo systemctl restart nginx
$ : sudo ufw allow ’Nginx Full’
$ : sudo systemctl restart gunicorn
```

2 ASGI server

2.1 Redis setup

1. Allow redis to be managed by systemd :

   Open ”/etc/redis/redis.conf”
   change ’supervised no’ to ’supervised systemd’

```
$ : sudo systemctl restart redis.service
$ : sudo systemctl status redis
```

2. Confirm Redis is running at 127.0.0.1. Port should be 6379 by default :

```
$ : sudo apt install net-tools
$ : sudo netstat -lnp — grep redis
$ : sudo systemctl restart redis.service
```

2.2 Channels and Daphne

1. Install the python-decouple library :

```
$ : pip install python-decouple
```

2. At the top of ”WebChatApp/ChatServerPlayground/settings.py”¨, add :

   from pathlib import Path
   from decouple import config

1. Create the django ASGI application

```
$ : cd WebChatApp/ChatServerPlayground
$ : mv ../ProjectFiles/asgi.py ./
```

2. Creating daphne service :

```
$ : sudo apt install daphne
$ : cd /etc/sustemd/system
$ : sudo cp ∼/WebChatApp/ProjectFiles/daphne.service ./
$ : systemctl daemon-reload
$ : systemctl start daphne.service
$ : systemctl status daphne.service
```

3. Starting daphne automatically :

```
$ : cp WebChatApp/ProjectFiles/boot.sh ∼/
$ : chmod u+x ∼/boot.sh
$ : cd /etc/systemd/system
$ : sudo cp ∼/WebChatApp/on_boot.service ./
$ : systemctl daemon-reload
$ : sudo systemctl start on_boot
$ : sudo systemctl enable on_boot
```

4. Allow daphne service through firewall :

```
$ : ufw allow 8001
$ : sudo shutdown -r now
```

5. Check the status of on_boot.service :

```
$ : systemctl status on_boot.service
```

1. Correct the ws_path in WebSockets connections in HTML templates :

   window.location.host =¿ window.location.hostname rajouter le port 8001 à chaque fois

2. Set up port forwarding : 8000 to 80 and 8001 to 8001

3 Systemctl services tips

- **Must be executed if you change the gunicorn.service file.**
  sudo systemctl daemon-reload
- **If you change code on your server you must execute this to see the changes take place.**
  sudo systemctl restart gunicorn
- **To check the status of a service**
  sudo systemctl status gunicorn
- **Where all the logs are consolidated to.**
  sudo journalctl
- **View the last entries in the error log**
  sudo tail -F /var/log/nginx/error.log
- **Nginx process logs**
  sudo journalctl -u nginx
- **Nginx access logs**
  sudo less /var/log/nginx/access.log
- **Nginx error logs**
  sudo less /var/log/nginx/error.log
- **gunicorn application logs**
  sudo journalctl -u gunicorn
- **check gunicorn socket logs**
  sudo journalctl -u gunicorn.socket