

Инструкция

Куркурин Никита

Содержание

1. Подготовка сервера	2
1.1. Работа с пользователем	2
1.2. Фаервол	2
1.3. Установка необходимого софта	2
1.4. Установка Python	2
1.5. Установка MySQL, Nginx	2
2. Настройка MySQL	3
2.1. Первоначальная настройка	3
2.2. Работа с пользователем	3
2.2.1 Настройка рута	3
2.3. Создаем нового пользователя	3
3. Django	5
3.1. Запуск на без сервера	5
3.2. Gunicorn	5
4. Nginx	7
5. Генерация SSL сертификата	8
6. Supervisor	9
6.1. Настройка	9
6.2. Доп. параметры	9
6.3. Подключение сайта	10

1. Подготовка сервера

1.1. Работа с пользователем

```
adduser sammy && usermod -aG sudo sammy
```

создание пользователя и наделение его правами администратора

1.2. Фаервол

```
ufw app list && ufw allow OpenSSH && ufw enable && ufw status
```

Установка фаерлоула и разрешение порта ssh, также, проверка его работоспособности

Перезагрузить сервер и зайти под sammy

1.3. Установка необходимого софта

```
sudo apt update && sudo apt upgrade && sudo apt install vim tmux
```

-y

Установка vim'а и tmux'а;

1.4. Установка Python

```
sudo apt install python3-pip python3-dev libpq-dev python3-virtualenv
```

-y

1.5. Установка MySQL, Nginx

```
sudo apt install nginx curl mysql-server -y
```

2. Настройка MySQL

2.1. Первоначальная настройка

`sudo mysql_secure_installation`

Далее такая последовательность ответов:

- *If system request about VALIDATE PASSWORD* - N
- *Password* - root
- *Remove anon* - Y
- *Dissallow* - N !!!
- *Remove test db* - Y
- *Reload priv.* - Y

2.2. Работа с пользователем

2.2.1 Настройка рута

- `sudo mysql`
- `ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'ПАРОЛЬ';`
- `FLUSH PRIVILEGES;`
- `exit;`

2.3. Создаем нового пользователя

- `mysql -u root -p`
- пароль который введен был в настройке
- `CREATE USER 'sammy'@'localhost' IDENTIFIED BY 'ПАРОЛЬ ДЛЯ SAMMY';`

- **GRANT ALL PRIVILEGES ON *.* TO 'sammy'@'localhost' WITH GRANT OPTION;**
- **exit;**
- **systemctl status mysql.service**

3. Django

3.1. Запуск на без сервера

- `Allow port (if not root != 8000)`
- `Write in ALLOWED_HOST ip-server.`
- `sudo ufw allow 8001`
- `start – python manage.py runserver 0.0.0.0:8001`
- `start with gunicorn – gunicorn –bind 0.0.0.0:8001 web_kazan.wsgi`

3.2. Gunicorn

- `sudo vim /etc/systemd/system/gunicorn.socket`

В этом файле пишем:

```
[Unit]
Description=gunicorn socket

[Socket]
ListenStream=/run/gunicorn.sock

[Install]
WantedBy=sockets.target
```

- `sudo vim /etc/systemd/system/gunicorn.service`

В этом файле пишем:

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

[Service]
User=sammy Group=www-data
WorkingDirectory=/home/sammy/myprojectdir
ExecStart=/home/sammy/myprojectdir/myprojectenv/bin/gunicorn \
-access-logfile - \
-workers 3 \
-bind unix:/run/gunicorn.sock \
myproject.wsgi:application
```

```
[Install]
```

```
WantedBy=multi-user.target
```

- **sudo journalctl -u gunicorn.socket**

4. Nginx

- `sudo vim /etc/nginx/sites-available/myproject`

```
server {  
    listen 8001;  
    server_name SERVER_IP_ADDRESS;  
  
    location /static/ {  
        root /home/sammy/web_kazan;  
    }  
  
    location / {  
        include proxy_params;  
        proxy_pass http://unix:/run/gunicorn.sock;  
    }  
}
```

- `sudo ln -s /etc/nginx/sites-available/myproject /etc/nginx/sites-enabled`
- `sudo nginx -t`

5. Генерация SSL сертификата

- `openssl genrsa -out webhook_pkey.pem 2048`
- `openssl req -new -x509 -days 3650 -key webhook_pkey.pem -out webhook_cert.pem`
- RU
- Moscow
- Moscow
- .
- .
- IP_ADDRESS
- .

6. Supervisor

6.1. Настройка

- `sudo apt-get install supervisor`
- `sudo supervisord`
- `sudo service supervisor status`
- `sudo service supervisor start`
- `sudo vim /etc/supervisor/conf.d/APPLICATION_TITLE.conf`

```
[program:APPLICATION_TITLE]
command=/home/sammy/parser_details/venv/bin/python -u main.py
directory=/home/sammy/parser_details
stdout_logfile=/home/sammy/parser_details/logs.txt
redirect_stderr=true
user=sammy
autostart=true
autorestart=true
```

- `sudo supervisorctl` - заходим в супервизор
- - `supervisor > reread` - обновление конфигов
 - `supervisor > add APPLICATION_TITLE` - добавление конфига в супервизор
 - `supervisor > status` - просмотр статуса всех активных конфигов

6.2. Доп. параметры

- `reread` : Reloads conf files
- `add <program>` : Adds a newly created conf file and starts the process
- `status` : Checks all status of programs managed by supervisor
- `start <program>` : Starts the program
- `restart <program>` : Restart the program
- `tail -f <program>` : Watch log file
- `exit` : Exits supervisorctl
- `help` : List commands

6.3. Подключение сайта

- `sudo vim /etc/supervisor/supervisord.conf`
[inet_http_server]
port=0.0.0.0:9001
username=1
password=1
- `sudo ufw allow 9001`
- `sudo service supervisor restart`