

****Устанавливаем Ubuntu 20.04****

****Обновляем****

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

****Устанавливаем веб сервер****

```
sudo apt install nginx
```

```
sudo apt install gunicorn
```

****Ngnix в автозагрузку****

```
sudo systemctl is-enabled nginx
```

```
sudo systemctl enable nginx
```

****Ставим фаервол****

```
sudo apt install ufw
```

****Добавляем правила****

```
sudo ufw app list
```

```
ls -la /etc/ufw/applications.d
```

****Создаем правило для фласк****

```
sudo nano /etc/ufw/applications.d/flask
```

```
[Flask]
```

```
title=Flask server
```

```
description=Flask development server, do not use it on production
```

```
ports=5000/tcp
```

****Смотрим что есть в правилах****

```
sudo ufw app list
```

****Добавляем правила для остальных служб что бы фаервол их пропускал****

```
sudo ufw allow OpenSSH
```

```
sudo ufw allow 'Nginx Full'
```

```
sudo ufw allow Flask
```

```
sudo ufw enable
```

****Ставим необходимое ПО****

```
sudo apt install python3
```

```
sudo apt install python3-pip
```

```
sudo apt install build-essential libssl-dev libffi-dev python3-dev python3-setuptools
```

****Ставим виртуальное окружение****

```
sudo apt install python3-venv
```

****Клонируем папку с гита (у меня будет techbase)****

```
cd techbase
```

****Файл с программой app.py файл для запуска wsgi.py****

```
-----
```

```
from app import app
```

```
if __name__ == "__main__":
```

```
    app.run()
```

```
-----
```

****Инициализируем виртуальное окружение****

```
python3 -m venv venv
```

****Активируем его****

```
source venv/bin/activate
```

****Ставим нужные нам библиотеки****

```
pip install flask mysql-connector-python wheel
```

****Тестируем****

```
gunicorn --bind 0.0.0.0:5000 wsgi:app
```

****Если все ок то отключаем тест CTRL+C****

****деактивируем venv****

```
deactivate
```

****Создаем службу проксирования gunicorn которая будет пробрасывать вебсервер на фласк****

```
sudo nano /etc/systemd/system/techbase.service
```

```
# Опишем что за сервис

[Unit]

# Описание

Description=Techbase gunicorn instance

# Здесь мы говорим systemd запускать наш сервер только после загрузки сетевых
служб.

After network.target

# Уже описывает наш юнит для системы.

[Service]

# От чьего имени запускать сервис

# Это я

User=lo

# Группа созданная для Nginx

Group=www-data

# Где его запускать

WorkingDirectory=/home/lo/techbase

# Путь к виртуальному окружению, в котором его запускать

Envirovement="PATH=/home/lo/techbase/venv/bin"

# Команда запуска сервиса с параметрами, о ней ниже

ExecStart=/home/lo/techbase/venv/bin/gunicorn --workers 2 --bind
unix:gunicorn.sock -m 007 wsgi:app

# Здесь опишем на каком уровне стартует наш сервис

[Install]

# Он будет соответствовать стандартному серверному (Runlevel 3)

# Многопользовательский режим с поддержкой сети, но без графического
интерфейса.

WantedBy=multi-user.target

-----

**теперь настройка Ngnix**

sudo nano /etc/nginx/nginx.conf

**раскомментировать**

server_names_hash_bucket_size 64;

**Удалить default в sitse enable**
```

Создаем файл конфигурации `sudo nano /etc/nginx/conf.d/techbase.conf`

```
-----  
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
    server_name 192.168.100.132;  
  
    location / {  
        include proxy_params;  
        proxy_pass http://unix:/home/lo/techbase/gunicorn.sock;  
    }  
}
```

****192.168.100.132 - адрес сервера****

****Тестируем конфигурацию Nginx****

`sudo nginx -t`

****Если ошибок не выдает****

****Мягкий перезапуск сервера****

`sudo service nginx reload`

****Удаляем правило пропуска 5000 порта****

`sudo ufw status`

`sudo ufw delete allow 5000`

****Пробуем http://192.168.100.132****