4.7 Самостоятельный хостинг

Самостоятельный хостинг

Размещайте приложение Flet на своем собственном сервере с помощью NGINX.

Есть бесплатные и недорогие облачные серверы, доступные на AWS, Oracle, Linode и других.

# Настройка среды Flet

**и**

**requirements.txt**

**virtualenv**

Создайте файл со списком зависимостей приложения.

requirements.txt

Минимально он должен содержать модуль flet:

flet>=0.2.4

Создайте виртуальное окружение и установите зависимости:

python -m venv .venv

source .venv/bin/activate

pip install -r requirements.txt

**Пример main.py приложения Flet**

import flet as ft

import os

# установите путь к Flet равным пустой строке для корневого URL

(например, https://lizards.ai/)

# или папку/путь для подгрузки под корневым URL (например,

https://lizards.ai/ui/path)

DEFAULT\_FLET\_PATH = '' # или 'ui/path'

DEFAULT\_FLET\_PORT = 8502

def main(page: ft.Page):

page.title = "Зацените моего чат-бота"

page.add(ft.Text("Сингулярность уже совсем близко"))

if name == "main":

flet\_path = os.getenv("FLET\_PATH", DEFAULT\_FLET\_PATH)

flet\_port = int(os.getenv("FLET\_PORT", DEFAULT\_FLET\_PORT))

ft.app(name=flet\_path, target=main, view=None, port=flet\_port)

# Автоматический запуск сервера Flet

**Создание файла блока systemd**

Автоматический запуск сервера Flet осуществляется с помощью системного файла flet.service .

Приведенная ниже установка предполагает, что сценарий приложения flet определен в файле $HOME/flet-app/main.py .

Замените User , Group , настройками.

WorkingDirectory

и т.д. в соответствии с вашими

[Unit]

Description=Flet Server

After=network.target

[Service]

User=ubuntu

Group=ubuntu

WorkingDirectory=/home/ubuntu/flet-app

Environment="PATH=/home/ubuntu/flet-app/.venv/bin"

ExecStart=/home/ubuntu/flet-app/.venv/bin/python /home/ubuntu/flet-

app/main.py

[Install]

WantedBy=multi-user.target

**Включение сервера Flet**

cd /etc/systemd/system

sudo ln -s /home/ubuntu/flet-app/flet.service

sudo systemctl start flet

sudo systemctl enable flet

sudo systemctl status flet

# Настройка прокси NGINX

NGINX будет проксировать приложение Flet и веб-сокет.

В вашем конфигурационном файле путь и порт при необходимости:

/etc/nginx/sites-available/\*

обновите

location / {

proxy\_pass

[http://127.0.0.1:8502/;](http://127.0.0.1:8502/%3B)

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection keep-alive;

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

location /ws {

proxy\_pass

[http://127.0.0.1:8502/ws;](http://127.0.0.1:8502/ws%3B)

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

Это часть конфигурации сервера NGINX, которая будет направлять запросы к вашему приложению Flet и веб-сокету. Обратите внимание, что есть два раздела location. Один для основных HTTP-запросов, а другой специально для веб-сокетов.

Настройка NGINX важна для обеспечения надежности и производительности вашего приложения. Эта конфигурация гарантирует, что запросы будут правильно маршрутизироваться к вашему приложению Flet и что веб-сокеты будут поддерживать постоянное соединение, что важно для функциональности реального времени Flet.