Nginx

Создать директорию nginx_config.conf и внутри файл python_microservices

```
server {
  listen 8080;

  location /api/firstendpoint {
    proxy_pass http://firstendpoint:8000/api/firstendpoint;
  }

  location /api/secondendpoint {
    proxy_pass http://secondendpoint:8000/api/secondendpoint;
  }
}
```

Compose:

```
version: '3.7'

services:
  nginx:
    image: nginx:latest
    ports:
      - "8080:8080"
    volumes:
      - ./nginx_config.conf:/etc/nginx/conf.d/default.conf
```

Источник

NGINX reverse proxy с https терминацией

Обслуживает несколько доменов, для одного из доменов путь /auth ведет на отдельный сервер

Структура проекта:

```
>certs
.env
```

```
docker-compose.yaml
nginx.conf.template
```

Размещение статических файлов

```
location /static {
    alias /app/static;
    # Опционально: настройки для статики
    expires 30d;
    add_header Cache-Control "public, immutable";
}
```

Директория certs: сертификаты в формате <domain name>.crt и <domain name>.key

.env

```
# Backend сервисы
WOOD_BACKEND_IP=192.168.1.194
WOOD_BACKEND_PORT=8000
HUB_BACKEND_IP=192.168.1.194
HUB_BACKEND_PORT=8021
HUB_BACKEND_WEB_PORT=8020

WOOD_AUTH_IP=192.168.1.194
WOOD_AUTH_PORT=8001

# Домены
HUB_DOMAIN=hub.bobrobotirk.ru
HUB_WEB_DOMAIN=hubui.bobrobotirk.ru
WOOD_DOMAIN=wood.bobrobotirk.ru
```

nginx.conf.template

```
worker_processes auto;

events {
    worker_connections 1024;
}

http {
    include mime.types;
```

```
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;

    # SSL-настройки
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers on;
    ssl_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-
GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384';
    ssl_session_timeout 1d;
    ssl_session_cache shared:SSL:50m;

    # Upstreams
    upstream hub_backend {
        server ${HUB_BACKEND_IP}:${HUB_BACKEND_PORT};
    }

    upstream hub_backend_web {
        server ${HUB_BACKEND_IP}:${HUB_BACKEND_WEB_PORT};
    }

    upstream wood_backend {
        server ${WOOD_BACKEND_IP}:${WOOD_BACKEND_PORT};
    }

    upstream wood_auth_backend {
        server ${WOOD_AUTH_IP}:${WOOD_AUTH_PORT};
    }

    # HTTP → HTTPS редирект
    server {
        listen 80;
        server_name ${HUB_DOMAIN} ${WOOD_DOMAIN};
        return 301 https://$host$request_uri;
    }

    # Конфиг для hub.bobrobotirk.ru
    server {
        listen 443 ssl;
        server_name ${HUB_DOMAIN};
```

```
    ssl_certificate /etc/nginx/certs/${HUB_DOMAIN}.crt;
    ssl_certificate_key /etc/nginx/certs/${HUB_DOMAIN}.key;

    location / {
        proxy_pass http://hub_backend;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

server {
    listen 443 ssl;
    server_name ${HUB_WEB_DOMAIN};

    ssl_certificate /etc/nginx/certs/${HUB_WEB_DOMAIN}.crt;
    ssl_certificate_key /etc/nginx/certs/${HUB_WEB_DOMAIN}.key;

    location / {
        proxy_pass http://hub_backend_web;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        }
    }

# Конфиг для wood.bobrobotirk.ru
server {
    listen 443 ssl;
    server_name ${WOOD_DOMAIN};

    ssl_certificate /etc/nginx/certs/${WOOD_DOMAIN}.crt;
    ssl_certificate_key /etc/nginx/certs/${WOOD_DOMAIN}.key;

    location /auth {
        proxy_pass http://wood_auth_backend;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
```

```
        location / {
            proxy_pass http://wood_backend;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
        }
    }
}
```

Алгоритм добавления еще одного endpoint:

Файл .env Константы, связанные с endpoint

```
ELSE_BACKEND_IP=192.168.1.198
ELSE_BACKEND_PORT=8000
```

Файл docker-compose .yaml

Добавляем эти константы в envsubst для машины nginx

```
envsubst '$${ELSE_BACKEND_IP} $${ELSE_BACKEND_PORT} ...'
```

И добавляем еще один сервис

```
  else-backend:
    image: else-backend:0.1
    container_name: else-backend
    #restart: unless-stopped  # Автоперезапуск при ошибках
    ports:
      - "${ELSE_BACKEND_PORT}:8000"  # host:container
    environment:
      - APP_ENV=production
      - LOG_LEVEL=info
      - WORKERS_PER_CORE=1  # Оптимально для CPU-bound приложений
      - MAX_WORKERS=4      # Лимит на случай автоскейлинга
      - TIMEOUT=120        # Макс. время выполнения запроса (сек)
      - KEEP_ALIVE=5       # Поддержка keep-alive соединений
    command: sh -c "uvicorn main:app --host=0.0.0.0 --port=8000 --reload"
    volumes:
      - ./backend/app:/app
```

В файле nginx.conf.template добавляем два раздела

```
    ...
    # Upstreams
    upstream else-backend {
        server ${ELSE_BACKEND_IP}:${ELSE_BACKEND_PORT};
    }
    ...
    location /else {
            proxy_pass http://else-backend;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
        }
```

docker-compose.yaml

```
services:
  nginx-proxy:
    image: nginx:latest
    container_name: nginx-proxy
    hostname: nginx-proxy
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./nginx.conf.template:/etc/nginx/templates/nginx.conf.template
      - ./certs:/etc/nginx/certs
    env_file:
      - .env  # Подключаем переменные из файла
    command: >
      /bin/sh -c "
      envsubst '$${HUB_BACKEND_IP} $${HUB_BACKEND_PORT} $${HUB_BACKEND_WEB_PORT}
$${WOOD_BACKEND_IP}
      $${WOOD_BACKEND_PORT} $${WOOD_AUTH_IP} $${WOOD_AUTH_PORT}
      $${HUB_DOMAIN} $${HUB_WEB_DOMAIN} $${WOOD_DOMAIN}'
      < /etc/nginx/templates/nginx.conf.template
      > /etc/nginx/nginx.conf
      && nginx -g 'daemon off;'
      "
    restart: no
```

Revision #6
Created 28 July 2024 15:58:55 by Admin
Updated 26 March 2026 16:13:00 by Admin