

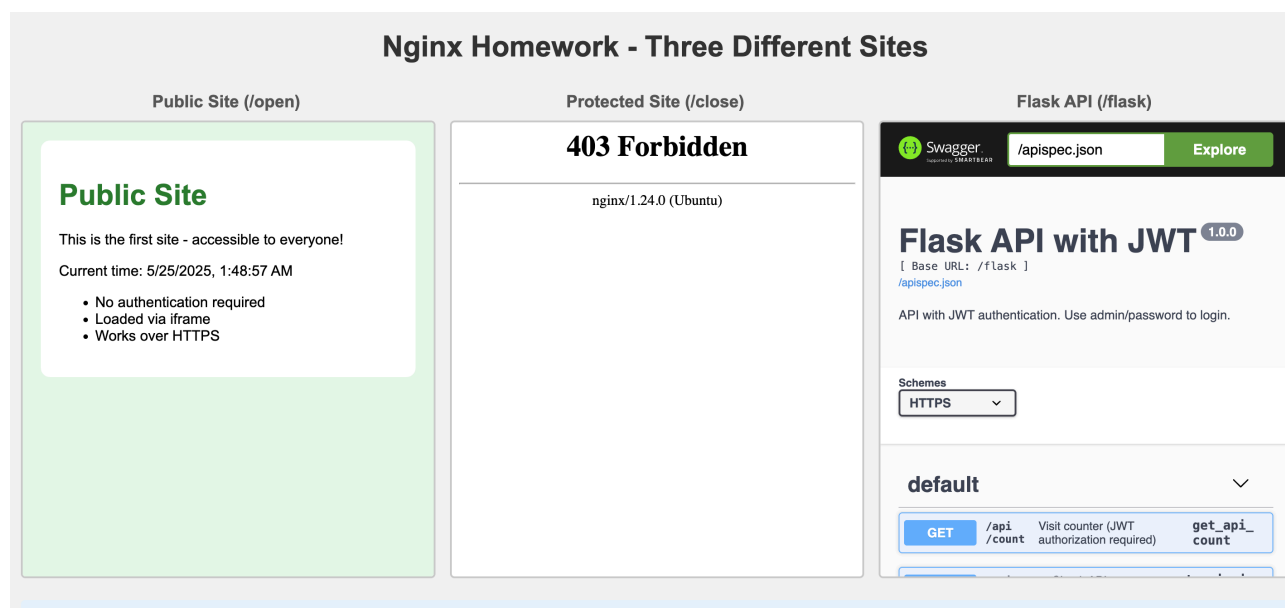
# Отчет по домашнему заданию к лекции 4 NGINX

**Выполнил:** Метельский Павел

**Дата выполнения:** 25 мая 2025

**Сервер:** Ubuntu 24.04.1 LTS (37.9.53.237)

- **Репозиторий с конфигами:** <https://github.com/PavelMetelsky/HW-lesta/tree/master/hw-NGINX#readme>
- **Исходный Flask проект:** <https://github.com/PavelMetelsky/Flask-app/tree/master#readme>



## 1. Создание виртуальной машины с Ubuntu и установка Nginx

### Характеристики VM:

- OS: Ubuntu 24.04.1 LTS
- CPU: 2 cores
- RAM: 4 GB
- Nginx: 1.24.0

## 2. HTML-страница с двумя сайтами через iframe

Главная страница (/var/www/main/index.html):

```
<details> <summary>Код HTML страницы</summary>
<!DOCTYPE html>
<html>
<head>
  <title>Main Page - Three Sites</title>
  <style>
```

```

    body {
        font-family: Arial;
        margin: 20px;
        background: #f0f0f0;
    }
    .container {
        display: flex;
        gap: 20px;
        margin-top: 20px;
    }
    iframe {
        width: 100%;
        height: 500px;
        border: 2px solid #ccc;
    }
</style>
</head>
<body>
    <h1>Nginx Homework - Three Different Sites</h1>

    <div class="container">
        <div class="site-frame">
            <h3>Public Site (/open)</h3>
            <iframe src="/open/" title="Public Site"></
iframe>
        </div>

        <div class="site-frame">
            <h3>Protected Site (/close)</h3>
            <iframe src="/close/" title="Protected Site"></
iframe>
        </div>

        <div class="site-frame">
            <h3>Flask API (/flask)</h3>
            <iframe src="/flask/swagger/" title="Flask
API"></iframe>
        </div>
    </div>
</body>
</html>

```

### 3. Настройка Nginx для доступа к обоим сайтам

Конфигурационный файл (/etc/nginx/sites-available/homework):

```

# Redirect HTTP to HTTPS
server {
    listen 80;
    server_name _;
    return 301 https://$host$request_uri;
}

# HTTPS server
server {
    listen 443 ssl;
    server_name _;

    # SSL certificates
    ssl_certificate /etc/nginx/ssl/homework.crt;
    ssl_certificate_key /etc/nginx/ssl/homework.key;

    # Main page
    location / {
        root /var/www/main;
        index index.html;
        access_log /var/log/nginx/main_access.log;
        error_log /var/log/nginx/main_error.log;
    }

    # Site 1 - Public (URL: /open)
    location /open {
        alias /var/www/site1;
        try_files $uri $uri/ /open/index.html;
        access_log /var/log/nginx/site1_access.log;
        error_log /var/log/nginx/site1_error.log;
    }

    # Site 2 - Protected (URL: /close)
    location /close {
        # IP restriction
        allow 127.0.0.1;
        allow 192.168.1.0/24;
        allow 10.0.0.5;
        deny all;

        # Basic authentication
        auth_basic "Restricted Area";
        auth_basic_user_file /etc/nginx/.htpasswd;

        alias /var/www/site2;
    }
}

```

```

        try_files $uri $uri/ /close/index.html;
        access_log /var/log/nginx/site2_access.log;
        error_log /var/log/nginx/site2_error.log;
    }

# Site 3 - Flask API (URL: /flask)
location /flask {
    rewrite ^/flask(.*) $1 break;
    proxy_pass http://unix:/run/gunicorn/flask.sock;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    access_log /var/log/nginx/flask_access.log;
    error_log /var/log/nginx/flask_error.log;
}
}

```

#### 4. Скриншоты конфигурационного файла и HTML-страницы

Структура файлов:

```

/etc/nginx/
├── sites-available/
│   └── homework                                # Nginx конфигурация
├── ssl/
│   ├── homework.crt                          # SSL сертификат
│   ├── homework.key                         # SSL приватный ключ
│   └── .htpasswd                             # Basic auth файл
└── /var/www/
    ├── main/                                # Главная страница с iframe
    │   └── index.html
    ├── site1/                               # Публичный сайт (/open)
    │   └── index.html
    └── site2/                               # Защищенный сайт (/close)
        └── index.html

```

#### 5. Настройка HTTPS с самоподписанным сертификатом

```

# Генерация сертификата
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -keyout /etc/nginx/ssl/homework.key \
    -out /etc/nginx/ssl/homework.crt \
    -subj "/C=BY/ST=Minsk/L=Minsk/O=Homework/
CN=homework.local"

```

Результат:

- HTTP автоматически перенаправляется на HTTPS

- Сертификат валиден 365 дней
- Браузер показывает предупреждение о самоподписанном сертификате

## 6. Настройка логирования запросов

Конфигурация логов:

```
access_log /var/log/nginx/main_access.log;      # Главная страница
access_log /var/log/nginx/site1_access.log;     # /open
access_log /var/log/nginx/site2_access.log;     # /close
access_log /var/log/nginx/flask_access.log;     # /flask API
```

Пример записей из лога:

```
192.168.1.100 - - [25/May/2025:14:23:45 +0000] "GET /flask/
auth/login HTTP/1.1" 200 521
192.168.1.100 - - [25/May/2025:14:23:50 +0000] "POST /flask/
auth/login HTTP/1.1" 200 234
```

## 7. Ограничение доступа к административной части по IP

Конфигурация IP-ограничений для **/close**:

```
location /close {
    # IP whitelist
    allow 127.0.0.1;          # localhost
    allow 192.168.1.0/24;    # локальная сеть
    allow 10.0.0.5;          # конкретный IP
    deny all;                # запретить всем остальным

    # Basic Auth
    auth_basic "Restricted Area";
    auth_basic_user_file /etc/nginx/.htpasswd;
}
```

- Доступ разрешен только с указанных IP
- Остальные получают 403 Forbidden
- Дополнительно требуется Basic Auth (admin/homework)

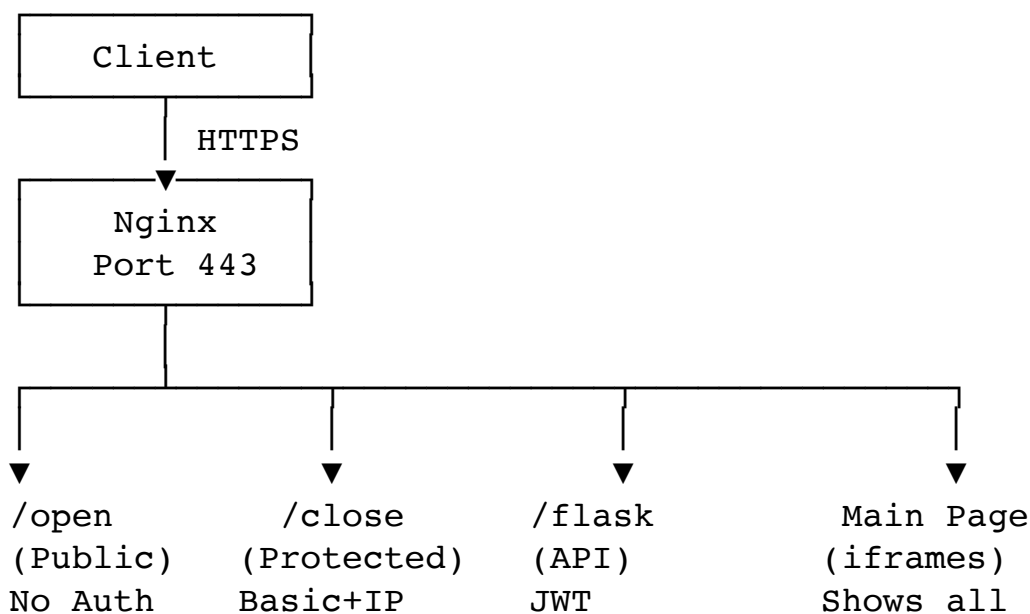
## Дополнительные улучшения

### Flask API с JWT аутентификацией

Помимо базовых требований, был развернут Flask API с:

- JWT токенами для аутентификации
- Swagger UI документацией
- Redis для хранения данных
- Systemd сервис для автозапуска

## Архитектура решения



## Тестирование

### Проверка HTTPS редиректа:

```
curl -I http://37.9.53.237
# HTTP/1.1 301 Moved Permanently
# Location: https://37.9.53.237/
```

### Проверка Basic Auth:

```
# Без авторизации
curl -k https://37.9.53.237/close/
# 401 Authorization Required
```

```
# С авторизацией
curl -k https://37.9.53.237/close/ -u admin:admin
# 200 OK (если IP разрешен)
```

### Проверка IP ограничений:

```
# С localhost (разрешен)
curl -k https://localhost/close/ -u admin:admin
# 200 OK
```

```
# С внешнего IP (запрещен)
curl -k https://37.9.53.237/close/ -u admin:admin
# 403 Forbidden
```