



# Настройка Nginx на Arch Linux с поддержкой HTTPS, перенаправлением, alias и виртуальными хостами

Обо мне

Меня зовут **Иван Фёдоров** aka Fara Limansk  
Мой GitHub для доступа к этому и другим проектам: [FaraLimansk](#)

В этом гайде мы рассмотрим настройку Nginx на примере проекта, с такими требованиями:

- Должен работать по https с сертификатом
- Настроить принудительное перенаправление HTTP-запросов (порт 80) на HTTPS (порт 443) для обеспечения безопасного соединения.

- Использовать `alias` для создания псевдонимов путей к файлам или каталогам на сервере.
- Настроить виртуальные хосты для обслуживания нескольких доменных имен на одном сервере.

## Ход работы

### Установка nginx

Для начала нам необходимо установить nginx

`sudo pacman -S nginx` - установка nginx

`sudo systemctl start nginx` - запуск nginx

`sudo systemctl enable nginx` - запуск nginx при запуске ос

```
[fara@archlinux ~]$ sudo pacman -S nginx
warning: nginx-1.26.2-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (1) nginx-1.26.2-1

Total Installed Size: 1.77 MiB
Net Upgrade Size: 0.00 MiB

:: Proceed with installation? [Y/n] y
(1/1) checking keys in keyring [#####] 100%
(1/1) checking package integrity [#####] 100%
(1/1) loading package files [#####] 100%
(1/1) checking for file conflicts [#####] 100%
(1/1) checking available disk space [#####] 100%
:: Processing package changes...
(1/1) reinstalling nginx [#####] 100%
:: Running post-transaction hooks...
(1/2) Reloading system manager configuration...
(2/2) Arming ConditionNeedsUpdate...
[fara@archlinux ~]$ sudo systemctl start nginx
sudo systemctl enable nginx
```

`sudo systemctl status nginx` - это нужно для проверки статуса сервера(в частности работает ли твой сервер или нет)

```
[fara@archlinux ~]$ sudo systemctl status nginx
nginx.service - nginx web server
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
  Active: active (running) since Sun 2024-10-20 17:26:25 UTC; 1h 11min ago
  Invocation: b1ce0b6de467454a84f14b214bcfe632
  Main PID: 9837 (nginx)
  Tasks: 2 (limit: 13105)
  Memory: 2.5M (peak: 3M)
  CPU: 46ms
  CGroup: /system.slice/nginx.service
          ┌─────────┐9837 "nginx: master process /usr/bin/nginx"
          └─────────┘9838 "nginx: worker process"
```

Теперь ты можешь убедиться сам что nginx работает :)

Если сервер работает то после ввода локального адреса (localhost) Тебя будет ожидать вот такое приветствие :

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working.  
Further configuration is required.

For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*

## Настройка HTTPS с сертификатом

Создайте директории для проекта и SSL сертификатов:

```
sudo mkdir -p /etc/nginx/ssl
```

Генерация самоподписанных сертификатов:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
/etc/nginx/ssl/project.local.key -out /etc/nginx/ssl/project.local.crt
```

```
[fara@archlinux ~]$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/project.local.key -out /etc/nginx/ssl/project.local.crt
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:RU
State or Province Name (full name) [Some-State]:SP
Locality Name (eg, city) []:SP
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Limansk
Organizational Unit Name (eg, section) []:Fara
Common Name (e.g. server FQDN or YOUR name) []:Fara
Email Address []:fara@limansk7777@mail.ru
[fara@archlinux ~]$
```

Настройте конфигурацию для HTTPS: Создайте директорию для проекта:

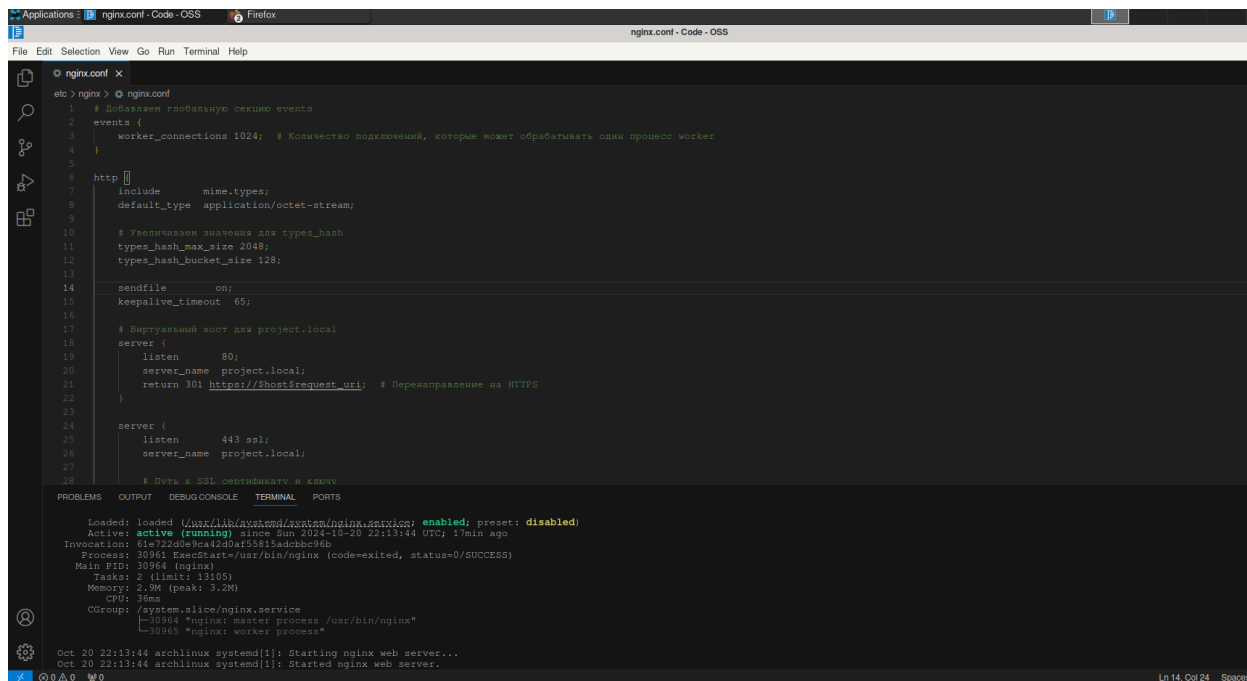
`sudo mkdir -p /var/www/project.local/html` - можете использовать этот путь для теста или создать свой уникальный проект с другим названием .

`sudo nano /etc/nginx/nginx.conf` - создаем конфиг файл или используем уже существующий при установке nginx

-----

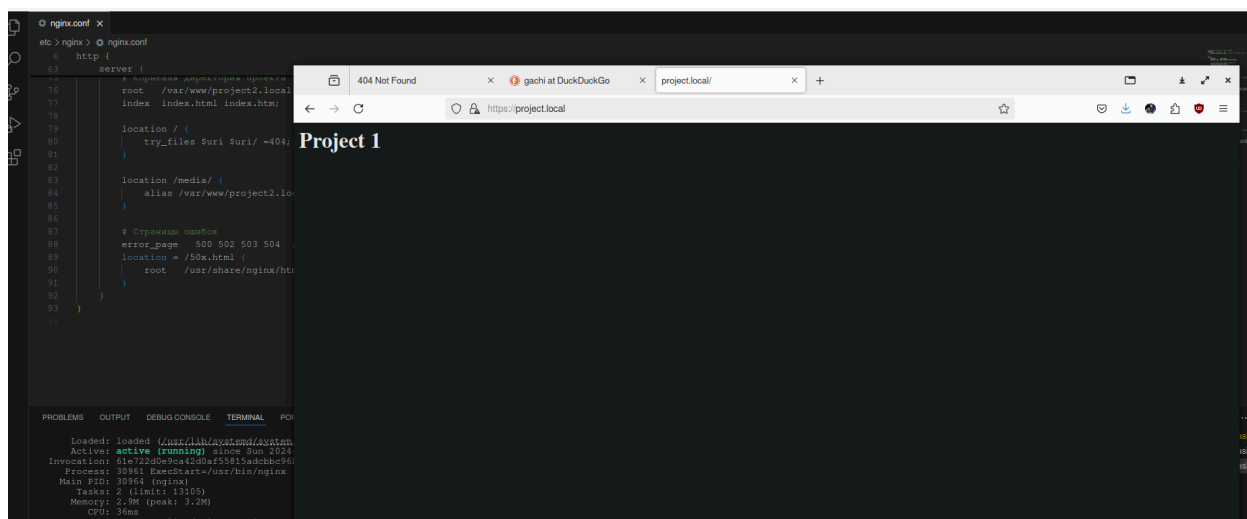
-----

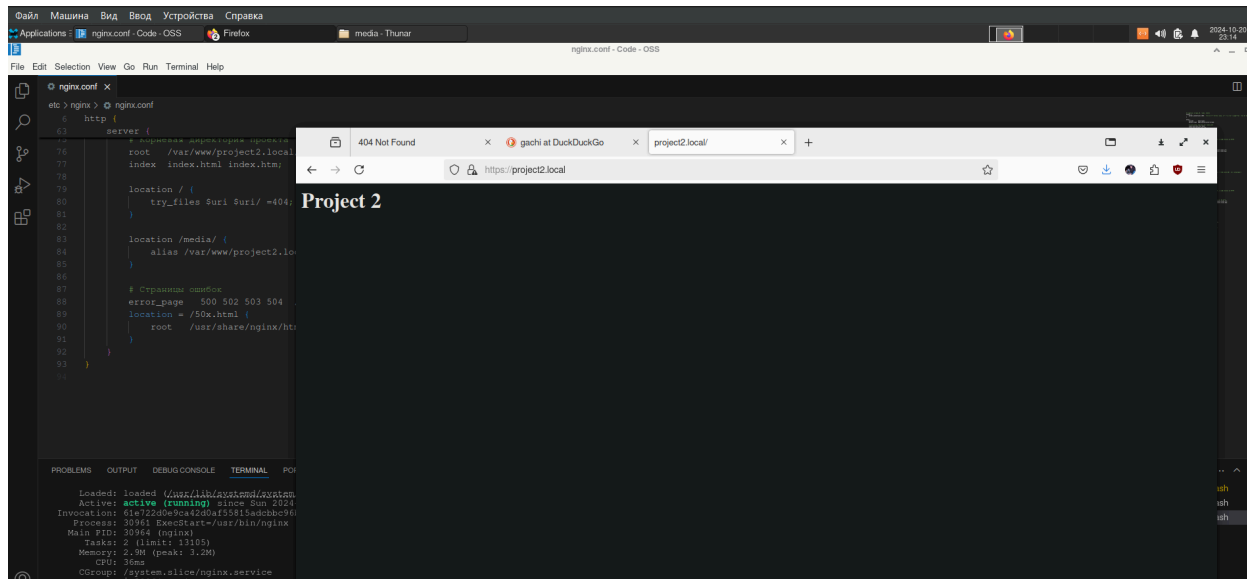
Я лично работая на arch linux использую OSS , он очень удобен в этом плане .



```
etc > nginx > nginx.conf
1 # Добавляем глобальную секцию events
2 events {
3     worker_connections 1024; # Количество подключений, которые может обрабатывать один процесс worker
4 }
5
6 http {
7     include mime.types;
8     default_type application/octet-stream;
9
10    # Увеличиваем значения для types_hash
11    types_hash_max_size 2048;
12    types_hash_bucket_size 128;
13
14    sendfile on;
15    keepalive_timeout 65;
16
17    # Виртуальный хост для project.local
18    server {
19        listen 80;
20        server_name project.local;
21        return 301 https://$host$request_uri; # Перенаправление на HTTPS
22    }
23
24    server {
25        listen 443 ssl;
26        server_name project.local;
27
28        # Путь к SSL сертификату и ключу
29    }
30 }
31
32 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
33
34 Loaded: loaded (/usr/lib/ systemd/systemd/nginx.service; enabled; preset: disabled)
35 Active: active (running) since Sun 2024-10-20 22:13:44 UTC; 17min ago
36 Invocation: 61e722d0e9ca42d0af55815adcbbe96b
37 Process: 30961 ExecStart=/usr/bin/nginx (code=exited, status=0/SUCCESS)
38 Main PID: 30964 (nginx)
39 Tasks: 2 (limit: 13105)
40 Memory: 2.9M (peak: 3.2M)
41 CPU: 36ms
42 CGroup: /system.slice/nginx.service
43         └─30964 "nginx: master process /usr/bin/nginx"
44             └─30965 "nginx: worker process"
45
46 Oct 20 22:13:44 archlinux systemd[1]: Starting nginx web server...
47 Oct 20 22:13:44 archlinux systemd[1]: Started nginx web server.
```

Если вы сделали все верно то у вас должно работать перенаправление с http на https , вот пример того что у меня все работает без тормозов :





Вот мой config для общего пользования :

# Добавляем глобальную секцию events

```
events {
```

```
    worker_connections 1024; # Количество подключений, которые может  
    обрабатывать один процесс worker
```

```
}
```

```
http {
```

```
    include    mime.types;
```

```
    default_type  application/octet-stream;
```

```
# Увеличиваем значения для types_hash
```

```
types_hash_max_size 2048;
```

```
types_hash_bucket_size 128;
```

```
sendfile          on;

keepalive_timeout 65;


# Виртуальный хост для project.local

server {

listen          80;

server_name     project.local;

return 301 https://$host$request_uri; # Перенаправление на HTTPS

}


server {

listen          443 ssl;

server_name     project.local;


# Путь к SSL сертификату и ключу

ssl_certificate  /etc/nginx/ssl/project.local.crt;

ssl_certificate_key /etc/nginx/ssl/project.local.key;


# Включить безопасные SSL параметры

ssl_protocols TLSv1.2 TLSv1.3;

ssl_ciphers HIGH:!aNULL:!MD5;
```

```
# Корневая директория проекта

root    /var/www/project1.local/html;

index   index.html index.htm;


location / {

    try_files $uri $uri/ =404;

}


# Использование alias для медиа-файлов

location /media/ {

    alias /var/www/project1.local/media/; # Исправлен путь для
медиа-файлов

}


# Страницы ошибок

error_page   500 502 503 504   /50x.html;

location = /50x.html {

    root    /usr/share/nginx/html;

}

}


# Виртуальный хост для project2.local

server {
```



```
listen      80;

server_name  project2.local;

return 301 https://$host$request_uri; # Перенаправление на HTTPS
}
```

```
server {

listen      443 ssl;

server_name  project2.local;


# Путь к SSL сертификату и ключу для второго проекта

ssl_certificate      /etc/nginx/ssl/project2.local.crt;
ssl_certificate_key  /etc/nginx/ssl/project2.local.key;


# Включить безопасные SSL параметры

ssl_protocols TLSv1.2 TLSv1.3;

ssl_ciphers HIGH:!aNULL:!MD5;


# Корневая директория проекта

root    /var/www/project2.local/html;

index  index.html index.htm;


location / {

    try_files $uri $uri/ =404;
```

Давайте пробежимся по важным аспектам которые выполняют указания тз :

Это псевдонимы путей или же `alias` .

**alias** в **Nginx** используется для создания псевдонимов путей к файлам или папкам на сервере. Он позволяет указать директорию, отличную от

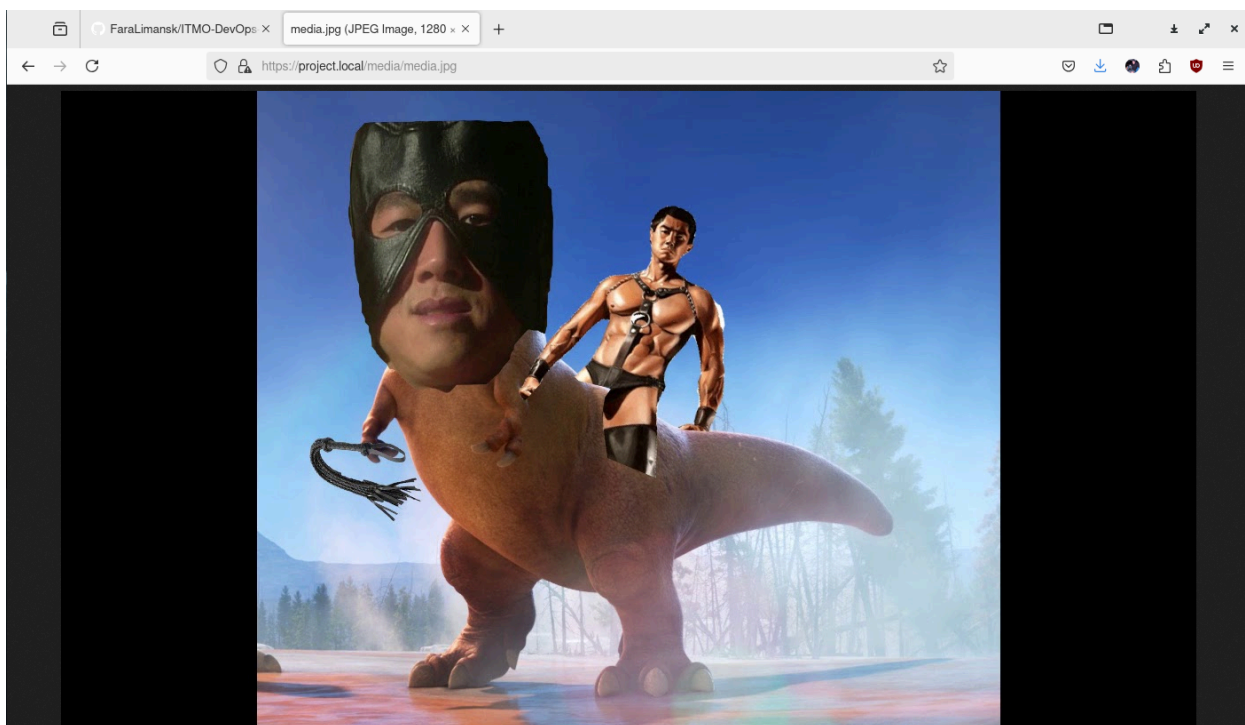
корневой (document root), чтобы, например, обслуживать статические файлы (изображения, документы) из другой директории.

**\*Важно!** Проверяйте файлы на их доступ и правильность прописывания путей , а также не забывайте про то что папка или файл не будут открываться если они находятся вне корневой папки сервера.

На скрине можно заметить пример рабочего alias для петов №1 и №2 :

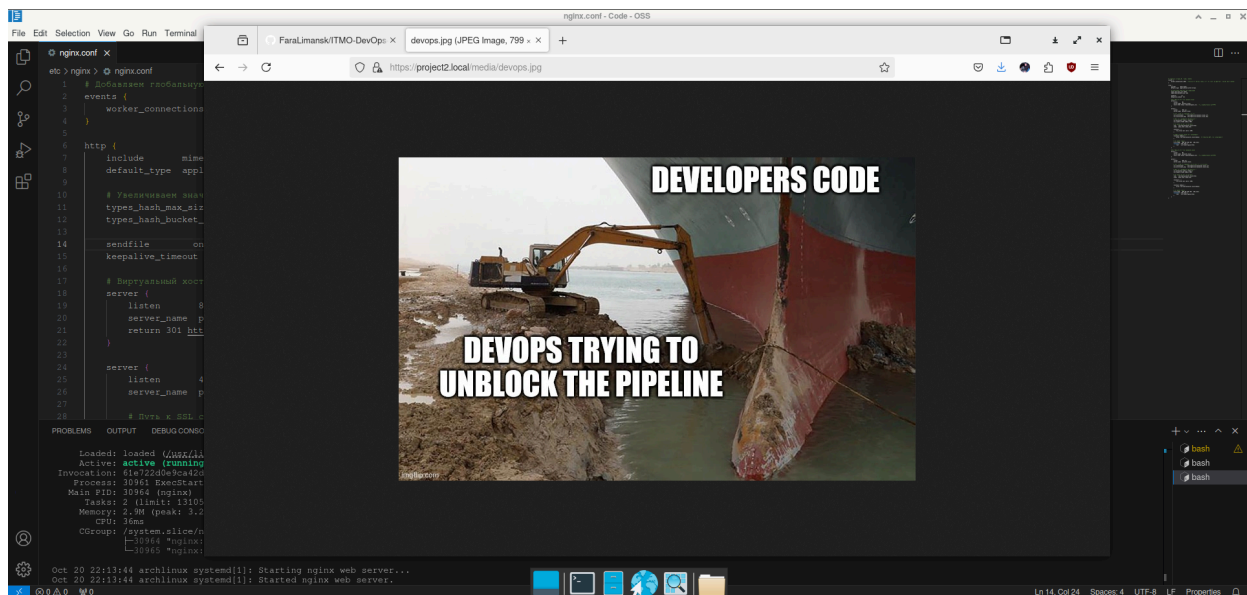
Настройка для `alias` в `config` под пет №1:

```
location /media/ {
    alias /var/www/project1.local/media/;
}
```



Настройка для alias в config под пет №2:

```
location /media/ {
    alias /var/www/project2.local/media/;
}
```



**Вот и все ! Ничего сложного в этом нет ,  
теперь вы знаете как работает nginx !**

**NGINX**

