

???????????? ???? ???????

???????????? ??????????: ?? ??????????? ??
???????????? ?? ??????? ???????

После установки Nginx по умолчанию активен один сайт, который вы видите на 80 порту — это страница "Welcome to nginx!". Его конфигурация находится в файле `/etc/nginx/sites-enabled/default`. Чтобы разместить свой сайт, нужно отключить эту конфигурацию и создать свою.

??? 1: ??????????? ?????? ?? ???????????

Удалим символическую ссылку на конфигурацию по умолчанию. Это безопаснее, чем удалять сам файл, так как он останется в `/etc/nginx/sites-available/` в качестве примера.

```
sudo rm /etc/nginx/sites-enabled/default
```

??? 2: ?????????? ?????????????????? ??? ??????? ???????

Создадим новый файл конфигурации для нашего домена (например, `example.com`) в каталоге `sites-available`.

```
sudo nano /etc/nginx/sites-available/example.com
```

Вставьте в этот файл базовую конфигурацию для статического сайта. Не забудьте создать каталог `/var/www/example.com` и положить туда файлы вашего сайта (например, `index.html`).

```
server {  
    listen 80;  
    server_name example.com www.example.com;  
    root /var/www/example.com;  
    index index.html index.htm;  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

??? 3: ?????????? ?????? ??????

Теперь создадим символическую ссылку из `sites-available` в `sites-enabled`, чтобы Nginx "увидел" и загрузил нашу конфигурацию.

```
sudo ln -s /etc/nginx/sites-available/example.com /etc/nginx/sites-enabled/
```

??? 4: ?????????? ? ?????????????? ???????????????

Перед перезапуском всегда проверяйте конфигурацию на наличие синтаксических ошибок.

Проверить конфигурацию:

```
sudo nginx -t
```

Если ошибок нет (вывод: ``syntax is ok, test is successful``), применяем изменения командой ``reload``, которая делает это без разрыва текущих соединений.

```
sudo systemctl reload nginx
```

????????? ?????????????????? ??????????????????

1. Reverse Proxy ? ?????????????????? ?????????? ?? IP (??? ??????-???????)

Очень частый сценарий: защитить внутренний инструмент (например, Grafana, Prometheus, веб-интерфейс 1C), работающий на локальном порту (например, 3000), и разрешить доступ к нему только из офисной сети.

```
server {
    listen 80;
    server_name dashboard.example.com;
    # Блок ограничения доступа
    allow 88.77.66.55; # Ваш офисный/домашний IP
    allow 127.0.0.1;  # Доступ с самого сервера
    deny all;         # Запретить доступ всем остальным
    location / {
        proxy_pass http://localhost:3000; # Адрес вашего внутреннего приложения
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

```
}  
}
```

2. ?????????? ?????????? ??????????

Позволяет просматривать содержимое каталога, если в нем отсутствует индексный файл. Полезно для организации файловых архивов.

```
location /downloads/ {  
    root /var/www/files;  
    autoindex on; # Включить листинг  
    autoindex_exact_size off; # Показывать размеры файлов в удобном виде (KB, MB)  
    autoindex_localtime on; # Показывать локальное время файлов  
}
```

3. ??????? ?????????? ? ?????????? ???????

Блокирует доступ к файлам, начинающимся с точки, таким как `.htaccess` или `.git`.

```
location ~ /\. {  
    deny all;  
    return 404;  
}
```

????????????????? `htaccess` ? Nginx

Важное отличие Nginx от Apache: ****Nginx не поддерживает файлы `htaccess`****. Это сделано осознанно для повышения производительности. Nginx считывает конфигурацию один раз при старте или перезагрузке и держит ее в памяти, в то время как Apache проверяет наличие `htaccess` в каждом каталоге при каждом запросе.

Все правила, которые вы бы поместили в `htaccess`, в Nginx прописываются непосредственно в конфигурационном файле вашего сайта (в блоке `server` или `location`). Вот как эмулировать популярные правила:

????????????????? (Rewrite) ??? "???????????" URL

Типичная задача для фреймворков: все запросы, которые не ведут к существующим файлам, отправлять на главный `index.php`.

```
# Пример для Apache (.htaccess)  
# RewriteEngine On  
# RewriteCond %{REQUEST_FILENAME} !-f
```

```
# RewriteCond %{REQUEST_FILENAME} !-d
# RewriteRule . /index.php [L]
# Эквивалент для Nginx
location / {
    try_files $uri $uri/ /index.php?$query_string;
}
```

?????? ??????? ? ????????

```
# Пример для Apache (.htaccess в папке /secret)
# deny from all
# Эквивалент для Nginx
location /secret/ {
    deny all;
    return 404; # Опционально, можно вернуть 403 или 404
}
```

Revision #2

Created 5 October 2025 12:34:11 by Admin

Updated 5 October 2025 12:36:00 by Admin