

Разработка Веб-Фронтенда

На нашем будущем сервере будет постоянно запущен веб-сервер Nginx. Он предоставит пользователю возможность управлять системой через веб-приложение в окне браузера по временному порту 7000. Если пользователь захочет использовать собственный веб-сервер, в приложении ему будет доступен веб-сервер Apache2 с поддержкой PHP. Службы OpenSSH и Samba, автозапуск которых настраивается для создания веб-приложения, также будут по умолчанию отключены с возможностью их постоянного включения через панель управления.

!!!!!!!!!!!!настройка nginx !!!!!!!!!!!!!

Настройка Nginx на обработку PHP

Файл nginx.conf

Мы перепишем конфигурационный файл nginx.conf, добавив правильный блок location ~ \.php\$, работающий через UNIX-сокеты.

#bash

```
cat << 'EOF' | sudo tee /etc/nginx/nginx.conf > /dev/null
worker_processes 1;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;

    server {
        listen 7000;
        server_name localhost;
        root /usr/share/nginx/html;
        index index.html index.htm index.php;

        location / {
            try_files $uri $uri/ =404;
        }

        location ~ \.php$ {
            include fastcgi.conf;
```

```
        fastcgi_pass    unix:/run/php-fpm/php-fpm.sock;
        fastcgi_index   index.php;
    }
}
EOF
```

```
[eva@tom1 ~]$ cat << 'EOF' | sudo tee /etc/nginx/nginx.conf > /dev/null
> worker_processes 1;
>
> events {
>     worker_connections 1024;
> }
>
> http {
>     include       mime.types;
>     default_type  application/octet-stream;
>     sendfile      on;
>     keepalive_timeout 65;
>
>     server {
>         listen      7000;
>         server_name localhost;
>         root        /usr/share/nginx/html;
>         index       index.html index.htm index.php;
>
>         location / {
>             try_files $uri $uri/ =404;
>         }
>
>         location ~ \.php$ {
>             include   fastcgi.conf;
>             fastcgi_pass unix:/run/php-fpm/php-fpm.sock;
>             fastcgi_index index.php;
>         }
>     }
> }
> EOF
```

Тестирование синтаксиса nginx.conf

Нам нужно протестировать синтаксис Nginx, чтобы убедиться, что все скобки закрыты и сокет прописан без ошибок.

#bash

```
sudo nginx -t
```

```
[eva@tom1 ~]$ sudo nginx -t
2026/05/24 05:20:19 [warn] 1462#1462: could not build optimal types_hash, you should increase either types_hash_max_size: 1024 or types_hash_bucket_size: 64; ignoring types_hash_bucket_size
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[eva@tom1 ~]$
```

(Тест пройден успешно (*syntax is ok, test is successful*). Предупреждение о *types_hash* — это стандартный информационный ворнинг, на работу он не влияет.)

Проверка содержимого файла nginx.conf

#bash

```
cat -n /etc/nginx/nginx.conf
```

```
[eva@tom1 ~]$ cat -n /etc/nginx/nginx.conf
1 worker_processes 1;
2
3 events {
4     worker_connections 1024;
5 }
6
7 http {
8     include mime.types;
9     default_type application/octet-stream;
10    sendfile on;
11    keepalive_timeout 65;
12
13    server {
14        listen 7000;
15        server_name localhost;
16        root /usr/share/nginx/html;
17        index index.html index.htm index.php;
18
19        location / {
20            try_files $uri $uri/ =404;
21        }
22
23        location ~ \.php$ {
24            include fastcgi.conf;
25            fastcgi_pass unix:/run/php-fpm/php-fpm.sock;
26            fastcgi_index index.php;
27        }
28    }
29 }
```

Перезапуск веб-сервера

#bash

```
sudo systemctl restart nginx
```

```
[eva@tom1 ~]$ sudo systemctl restart nginx
[eva@tom1 ~]$
```

Проверка статуса службы

#bash

```
sudo systemctl status nginx
```

```
[eva@tom1 ~]$ 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 2026-05-24 05:31:24 UTC; 2min 13s ago
  Invocation: c70deed23faa4bcca7e4807d5183a936
    Process: 1520 ExecStart=/usr/bin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 1521 (nginx)
      Tasks: 2 (limit: 11644)
    Memory: 4.8M (peak: 5.3M)
       CPU: 16ms
    CGroup: /system.slice/nginx.service
            └─1521 "nginx: master process /usr/bin/nginx"
              └─1522 "nginx: worker process"

May 24 05:31:24 tom1 systemd[1]: Stopping nginx web server...
May 24 05:31:24 tom1 systemd[1]: nginx.service: Deactivated successfully.
May 24 05:31:24 tom1 systemd[1]: Stopped nginx web server.
May 24 05:31:24 tom1 systemd[1]: Starting nginx web server...
May 24 05:31:24 tom1 nginx[1520]: 2026/05/24 05:31:24 [warn] 1520#1520: could not build optimal types_hash, you should increase either types_hash_max_size: 1024 or types_hash_bucket_size: 64; ignoring types_hash_bucket_size
May 24 05:31:24 tom1 systemd[1]: Started nginx web server.
[eva@tom1 ~]$
```

Проверим директорию

Чтобы не запутаться в структуре бэкенда и фронтенда, давайте сначала проверим, что сейчас вообще находится внутри корневой папки Nginx на tom_1.

#bash

```
ls -la /usr/share/nginx/html/
```

```
[eva@tom1 ~]$ ls -la /usr/share/nginx/html/
total 8
drwxrwxrwx 1 root root 36 May 23 09:21 .
drwxr-xr-x 1 root root 8 May 21 16:06 ..
-rwxrwxrwx 1 root root 497 May 22 16:16 50x.html
-rwxrwxrwx 1 root root 896 May 22 16:16 index.html
```

Проверим запуск и работу веб-сервера на порту 7000 в браузере на странице <http://192.168.1.72:7000/>



Проверяем автозапуск службы php-fpm

Убедимся, что служба PHP-FPM активирована в автозапуске, чтобы при старте флешки в ОЗУ она запустилась сама вместе с Nginx.

#bash

```
systemctl is-enabled php-fpm
```

```
[eva@tom1 ~]$ systemctl is-enabled php-fpm
disabled
[eva@tom1 ~]$
```

(Примечание: статус disabled - выключена)

Включаем службу php-fpm в автозапуск

Включим службу PHP-FPM в автозапуске, чтобы при старте флешки в ОЗУ она запустилась сама вместе с Nginx.

#bash

```
sudo systemctl enable php-fpm
```

```
[eva@tom1 ~]$ sudo systemctl enable php-fpm
[sudo] password for eva:
created symlink '/etc/systemd/system/multi-user.target.wants/php-fpm.service' -> '/usr/lib/systemd/system/php-fpm.service'
[eva@tom1 ~]$
```

(Примечание: созданы системные симлинки)

Проверяем статус автозапуска службы php-fpm

Убедимся, что служба PHP-FPM активирована в автозапуске, чтобы при старте флешки в ОЗУ она запустилась сама вместе с Nginx. Убедимся, что система теперь рапортует правильный статус.

#bash

```
systemctl is-enabled php-fpm
```

```
[eva@tom1 ~]$ systemctl is-enabled php-fpm
enabled
[eva@tom1 ~]$
```

(Примечание: статус *enabled* - включена)

Снятие системной изоляции с PHP

В Arch Linux служба php-fpm по умолчанию заперта (ProtectSystem=full). Из-за этого PHP видит системные файлы пользователей как Read-Only. Снимем это ограничение.

Откройте переопределение настроек службы:

#bash

```
sudo systemctl edit php-fpm
```

Вставьте свои строки строго между первой и второй строками комментариев (обычно там есть явная подсказка `### Lines below this comment will be discarded` или аналогичная):

ini

```
[Service]
ProtectSystem=false
ProtectHome=false
```

```
GNU nano 9.0 /etc/systemd/system/php-fpm.service.d/.#override.conf08e8a472e69155c5 Modified
# Takes away the ability to create or manage any kind of namespace
# RestrictNamespaces=true
#
# [Install]
# WantedBy=multi-user.target
[Service]
ProtectSystem=false
ProtectHome=false
^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute  ^C Location ^U Undo
^X Exit      ^R Read File ^N Replace   ^J Paste     ^D Justify  ^V Go To Line ^E Redo
```

(Ctrl+O для сохранения, затем Enter и Ctrl+X для выхода)

```
[eva@tom1 ~]$ sudo systemctl edit php-fpm
Successfully installed edited file '/etc/systemd/system/php-fpm.service.d/override.conf'.
[eva@tom1 ~]$
```

(Вывод означает правильно отредактированную конфигурацию, и systemd успешно создал конфигурационный файл (drop-in файл) с вашими настройками по пути /etc/systemd/system/php-fpm.service.d/override.conf)

Перезапустите службу:

```
sudo systemctl restart php-fpm
```

```
[eva@tom1 ~]$ sudo systemctl restart php-fpm
[eva@tom1 ~]$
```

!!!!!!!!!!!!!! настройка Samba-сервера !!!!!!!!!!!!!!!

Проверяем состояние Samba-сервера на tom_1

Нам нужно узнать, запущен ли демон Samba (smb), чтобы понять, сможем ли мы сразу подключить сетевой диск в Windows.

Выполните в терминале команду:

#bash

```
sudo systemctl status smb
```

```
[eva@tom1 ~]$ sudo systemctl status smb
[sudo] password for eva:
Unit smb.service could not be found.
[eva@tom1 ~]$
```

Служба Samba (smb.service) отсутствует в системе tom_1 (пакет не установлен).

Установка пакета Samba

Установим сам сервер Samba.

Выполните в терминале команду:

#bash

```
sudo pacman -S samba --noconfirm
```

```
[eva@tom1 ~]$ sudo pacman -S samba --noconfirm
[sudo] password for eva:
resolving dependencies...
looking for conflicting packages...

Packages (15) avahi-1:0.9rc4-1 cifs-utils-7.5-1 ldb-2:4.24.2-1 libbsd-0.12.2-2 libcups-2:2.4.19-1 libmd-1.2.0-1
               liburing-2.14-1 libwbclient-2:4.24.2-1 mpdecimal-4.0.1-3 python-3.14.5-1 smbclient-2:4.24.2-1
               talloc-2.4.4-1 tdb-1.4.15-1 tevent-1:0.17.1-2 samba-2:4.24.2-1

Total Download Size: 31.57 MiB
Total Installed Size: 173.03 MiB

:: Proceed with installation? [y/n]
:: Retrieving packages...
avahi-1:0.9rc4-1-x86_64           444.2 KiB   705 KiB/s  00:01 [#####] 100%
ldb-2:4.24.2-1-x86_64           488.0 KiB   775 KiB/s  00:01 [#####] 100%
libcups-2:2.4.19-1-x86_64       273.4 KiB   598 KiB/s  00:00 [#####] 100%
liburing-2.14-1-x86_64          253.9 KiB   577 KiB/s  00:00 [#####] 100%
libbsd-0.12.2-2-x86_64         163.5 KiB   790 KiB/s  00:00 [#####] 100%
mpdecimal-4.0.1-3-x86_64        103.2 KiB   577 KiB/s  00:00 [#####] 100%
cifs-utils-7.5-1-x86_64         98.7 KiB   584 KiB/s  00:00 [#####] 100%
tdb-1.4.15-1-x86_64             72.0 KiB   497 KiB/s  00:00 [#####] 100%
libmd-1.2.0-1-x86_64           59.8 KiB   398 KiB/s  00:00 [#####] 100%
tevent-1:0.17.1-2-x86_64       58.2 KiB   416 KiB/s  00:00 [#####] 100%
```

Проверка установки

Нам нужно убедиться, что бинарные файлы утилиты встали в систему без ошибок.

Выполните в терминале команду проверки установленной версии:

#bash

```
smbd --version
```

```
[eva@tom1 ~]$ smbd --version
Version 4.24.2
[eva@tom1 ~]$
```

Создание конфигурационного файла smb.conf

Создадим конфигурационный файл в редакторе nano.

Выполните в терминале команду:

#bash

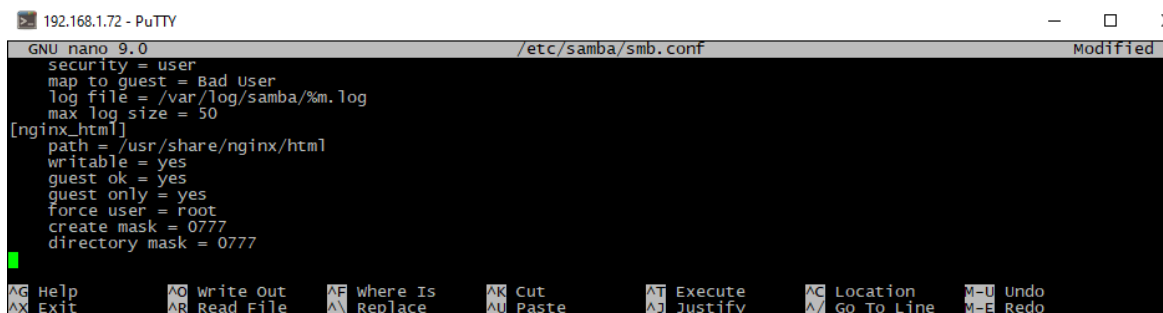
```
sudo nano /etc/samba/smb.conf
```

Файл пустой и готов к заполнению. Вставьте в него следующий минимальный рабочий конфиг, чтобы открыть доступ к директории Nginx (/usr/share/nginx/html) с правами на запись для всех:

ini

```
[global]
workgroup = WORKGROUP
server string = Arch Linux Tom1
security = user
map to guest = Bad User
log file = /var/log/samba/%m.log
max log size = 50

[nginx_html]
path = /usr/share/nginx/html
writable = yes
guest ok = yes
guest only = yes
force user = root
create mask = 0777
directory mask = 0777
```



Файл изменен. Нажмите последовательно: CTRL + O, затем клавишу Enter (для записи файла). CTRL + X (для выхода из редактора nano)

Проверка синтаксиса

выполните встроенную команду Samba для проверки синтаксиса файла конфигурации:

#bash

```
testparm -s
```

```

server role: ROLE_STANDALONE

# global parameters
[global]
    log file = /var/log/samba/%m.log
    map to guest = Bad User
    max log size = 50
    security = USER
    server string = Arch Linux Tom1
    idmap config * : backend = tdb

[nginx_html]
    create mask = 0777
    directory mask = 0777
    force user = root
    guest ok = Yes
    guest only = Yes
    path = /usr/share/nginx/html
    read only = No
[eva@tom1 ~]$

```

Тест синтаксиса пройден успешно (Loaded services file OK). Ошибок в файле smb.conf нет.

Сетевая папка nginx_html определена верно.

Следующий шаг — запуск и добавление службы Samba в автозагрузку.

#bash

```
sudo systemctl enable --now smb
```

```

[eva@tom1 ~]$ sudo systemctl enable --now smb
Created symlink '/etc/systemd/system/multi-user.target.wants/smb.service' - '/usr/lib/systemd/system/smb.service'.
[eva@tom1 ~]$

```

(Симлинк успешно создан, служба добавлена в автозапуск.)

Следующий шаг — обязательная проверка статуса запущенного демона Samba.

Проверка статуса

#bash

```
sudo systemctl status smb
```

```

[eva@tom1 ~]$ sudo systemctl status smb
● smb.service - Samba SMB Daemon
   Loaded: loaded (/usr/lib/systemd/system/smb.service; enabled; preset: disabled)
   Active: active (running) since Sat 2026-05-23 18:58:14 UTC; 2min 46s ago
     Invocation: a207e7d6c62342f2bd16dd411ea5bb34
       Docs: man:smbd(8)
             man:samba(7)
             man:smb.conf(5)
    Main PID: 2373 (smbd)
      Status: "smbd: ready to serve connections..."
        Tasks: 3 (Limit: 11644)
       Memory: 11.3M (peak: 11.4M)
          CPU: 42ms
      CGroup: /system.slice/smb.service
              └─2373 /usr/bin/smbd --foreground --no-process-group
                └─2379 "smbd: notifyd"
                  └─2380 "smbd: cleanupd"

May 23 18:58:13 tom1 systemd[1]: Starting Samba SMB Daemon...
May 23 18:58:14 tom1 systemd[1]: Started Samba SMB Daemon.
[eva@tom1 ~]$

```

(Служба smb работает в режиме active (running) и полностью готова принимать сетевые подключения.)

Службы настроены. Теперь нам нужно выдать права Linux на целевую папку, чтобы гостевой пользователь Samba (nobody / root) и веб-сервер Nginx (http) могли без ограничений читать,

создавать и изменять файлы внутри.

Выполните в терминале команду:

#bash

```
sudo chmod -R 777 /usr/share/nginx/html
```

```
[eva@tom1 ~]$ sudo chmod -R 777 /usr/share/nginx/html
[eva@tom1 ~]$
```

(Права 777 назначены успешно.)

Следующий обязательный шаг по нашему плану — проверка того, как система применила эти права к содержимому каталога.

#bash

```
ls -la /usr/share/nginx/html
```

```
[eva@tom1 ~]$ ls -la /usr/share/nginx/html
total 8
drwxrwxrwx 1 root root 36 May 23 09:21 .
drwxr-xr-x 1 root root 8 May 21 16:06 ..
-rwxrwxrwx 1 root root 497 May 22 16:16 50x.html
-rwxrwxrwx 1 root root 896 May 22 16:16 index.html
[eva@tom1 ~]$
```

(Проверка прав прошла успешно. Строки `drwxrwxrwx` для текущей папки (`.`) и `-rwxrwxrwx` для файлов `index.html` и `50x.html` подтверждают, что доступ полностью открыт на чтение, запись и исполнение для всех пользователей системы.)

Подключение сетевой папки в Windows

Получим точный текущий IP-адрес сервера `tom_1`

Выполните в терминале команду:

#bash

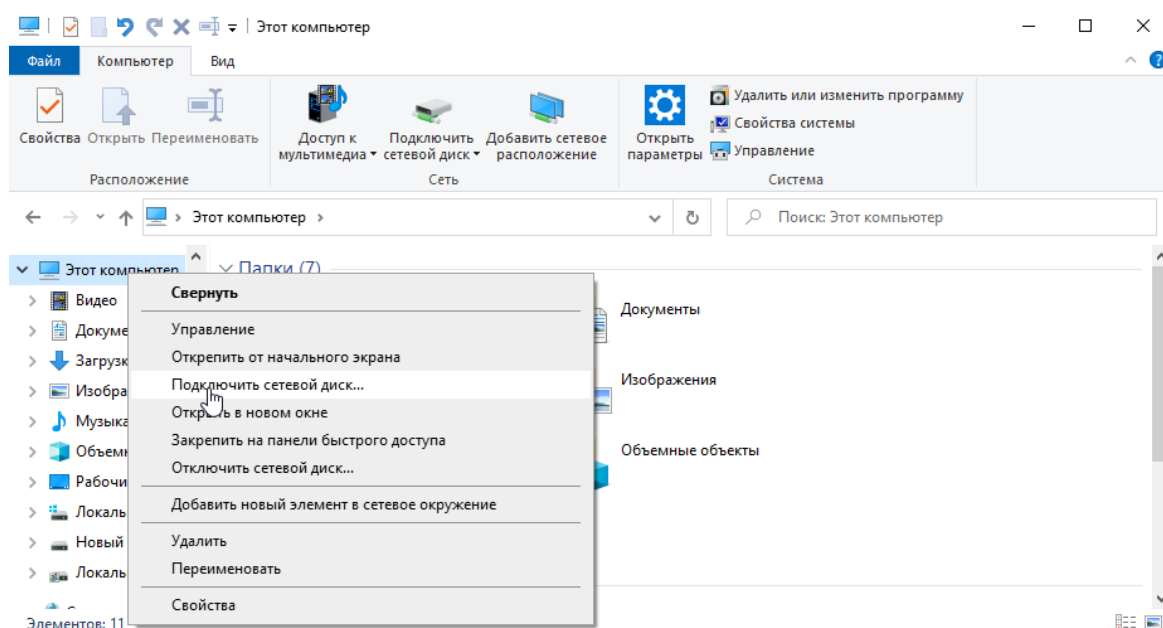
```
ip -br address show scope global | awk '{print $3}' | cut -d/ -f1
```

```
[eva@tom1 ~]$ ip -br address show scope global | awk '{print $3}' | cut -d/ -f1
192.168.1.72
[eva@tom1 ~]$
```

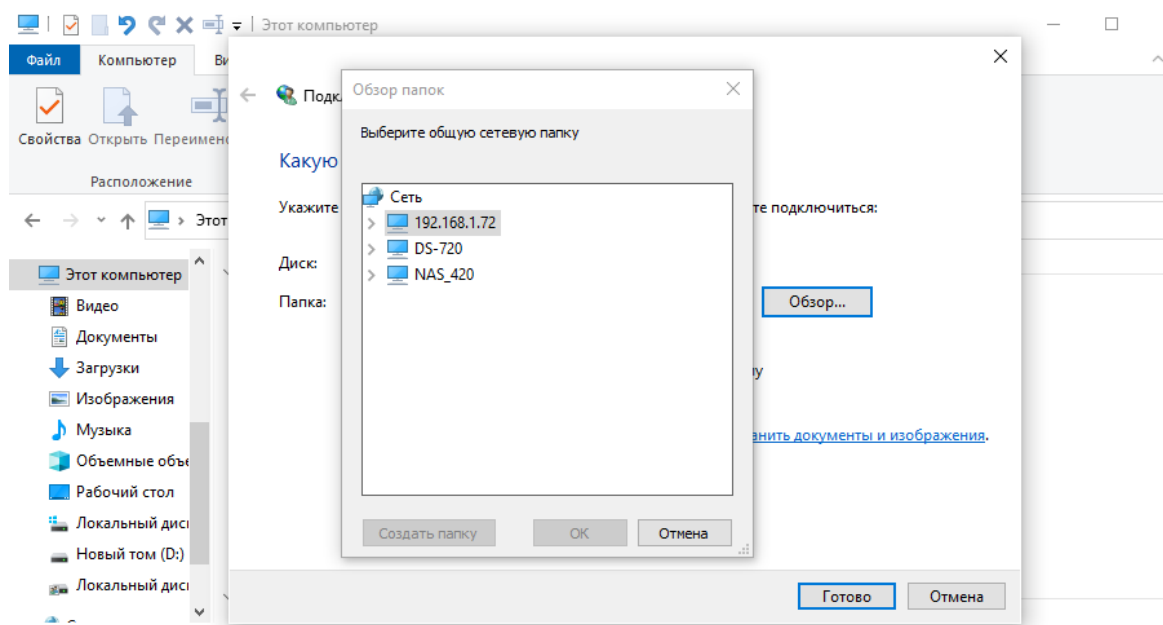
Теперь папка готова к подключению в качестве сетевого диска в среде Windows, чтобы вы могли открыть её через Notepad++.

1. Откройте Проводник (Этот компьютер) на вашей Windows-машине.

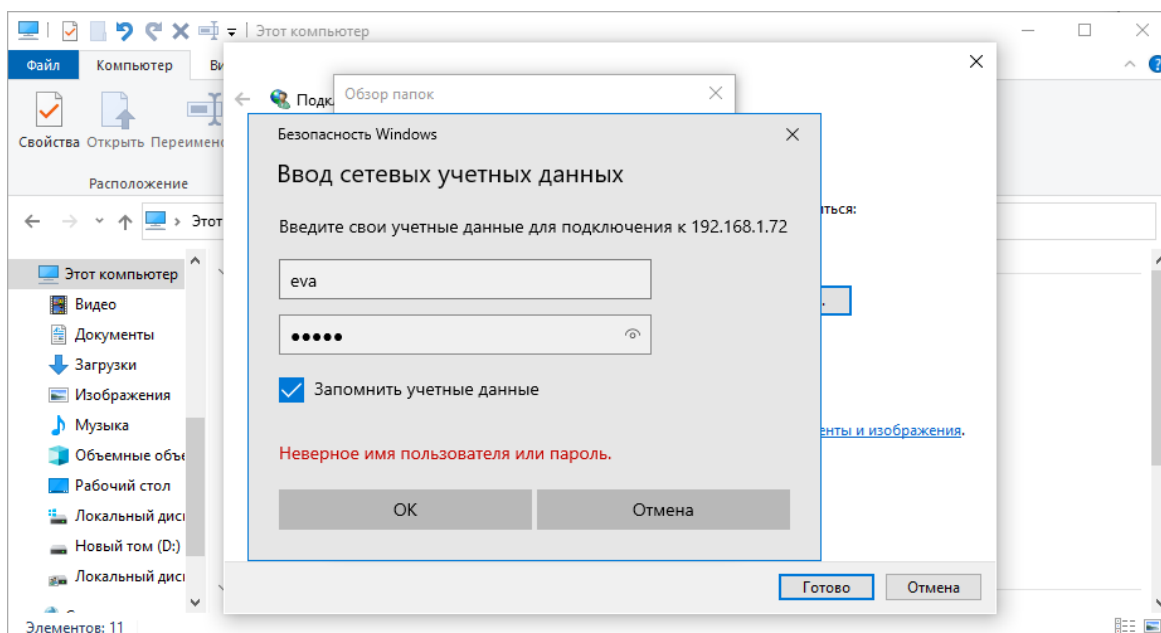
- В верхнем меню нажмите кнопку «Подключить сетевой диск» (или нажмите правой кнопкой мыши по «Этот компьютер» → «Подключить сетевой диск»).



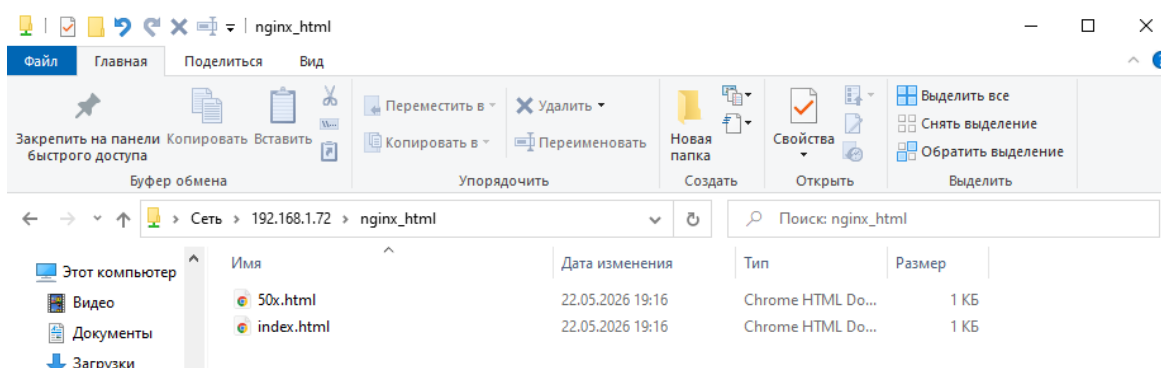
- В поле «Папка» введите сетевой путь, используя IP-адрес вашего сервера tom_1 (из логов PuTTY: 192.168.1.72 или через кнопку обзор)



Введите сетевые учетные данные



Зайдите через проводник



!!!!!!!!!!!!Что это и к чему!!!!!!!!!!!!!!!!!!!!

(В корне /usr/share/nginx/html/ только файлы 50x.html и index.html. Полные права 777)

Разворачиваем структуру каталогов для нашего веб-интерфейса. Создадим стандартные папки для стилей, скриптов и серверной логики.

!!!!!!!!!!!!настройка http пользователя !!!!!!!!!!!!

Системный пользователь http

Пользователь http в Arch Linux — это встроенный системный пользователь, от имени которого по умолчанию работают веб-серверы (например, Apache или Nginx) и сопутствующие им службы.

Он создается автоматически при установке этих программ для изоляции процессов и обеспечения безопасности.

Назначение

- **Безопасность:** Веб-серверы не должны работать под правами суперпользователя (root). Если злоумышленник найдет уязвимость в вашем сайте, он получит доступ только к файлам с правами пользователя http, что убережет остальную систему от взлома.
- **Права на файлы:** Этот пользователь владеет файлами и папками, к которым сервер имеет доступ (обычно они расположены в директории /srv/http/).
- **Группа http:** Для удобства существует одноименная группа http, в которую могут входить другие пользователи, чтобы иметь возможность редактировать файлы сайта без изменения прав доступа к ним через sudo

Применение

- **Размещение сайтов:** При настройке Nginx, Apache или PHP-FPM часто требуется указывать, что процесс должен запускаться от имени http.
- **Настройка разрешений (Permissions):** Если сайт выдает ошибку доступа (например, 403 Forbidden), обычно это означает, что у пользователя http нет прав на чтение нужных файлов или папок.
- **Безопасность каталогов:** Если скриптам на сайте нужно загружать файлы на сервер, папке загрузки необходимо выдать права (владельца) для пользователя http

Вывод строк пользователей системы

Выводим строки трех нами известных пользователей из базы данных.

#bash

```
sudo grep -E '^(root|eva|http):' /etc/shadow
```

```
[eva@tom1 ~]$ sudo grep -E '^(root|eva|http):' /etc/shadow
[sudo] password for eva:
root:$y$j9T$.Uty5qx7EJbR8eqoPnmR.$aAD2CoNq.XSFF1Js9h68cKmE5CRahxxDjAyk.NPnuU9:20594:0:0:
http:!*:20594:0:0:1:
eva:$y$j9T$XqsIjp.dE297cMeuUpNks0$g5P02vn/07UzTbFb/yr9CFmPVqVP9KbaQMkRNOY1SA:20594:0:99999:7:
[eva@tom1 ~]$
```

root:\$y\$j9T\$...:20594:0:0:

- Второе поле — длинный хэш \$y\$. Это зашифрованный пароль суперпользователя. Число 20594 — дата последнего изменения пароля (в днях от 1970 года). В конце строки — пустые поля. Это значит, что для root нет никаких ограничений.

eva:\$y\$j9T\$...:20594:0:99999:7:0:

- Также видим хэш личного пароля.
- Параметры 0:99999:7 означают стандартные правила пользователя: пароль можно

менять сразу (0), он действует 99999 дней, а за 7 дней до истечения система начнет предупреждать. Восьмое поле пустое — аккаунт не блокируется.

http:!*:20594::::::1:

- Второе поле содержит !*. Это маркер того, что пароль заблокирован (вход по паролю невозможен, учетка техническая). Вниманию в самый конец строки: :::::1:
- На предпоследней позиции (8-е поле) стоит цифра 1. В системе Linux это означает, что учетная запись принудительно заблокирована подсистемой безопасности PAM через 1 день после начала эпохи Unix (то есть 2 января 1970 года).

Изменяем параметры пользователя http

Уберем эту единицу из конца строки, чтобы сделать учетную запись бессрочной.

#bash

```
sudo chage -E -1 http
```

```
[eva@tom1 ~]$ sudo chage -E -1 http
[sudo] password for eva:
[eva@tom1 ~]$
```

Проверка изменений в файле /etc/shadow

#bash

```
sudo grep '^http:' /etc/shadow
```

```
[eva@tom1 ~]$ sudo grep '^http:' /etc/shadow
http:!*:20594::::::
[eva@tom1 ~]$
```

(Строка завершается чистыми двоеточиями (:::), что означает: блокировка PAM полностью снята, аккаунт http стал бессрочным)

!!!!!!!!!!!!!! Настройка ssh !!!!!!!!!!!!!!!

Схема веб-панели управления в основной системе (tom_1)

Папка веб-сервера nginx_html находится по пути /usr/share/nginx/html/ и имеет следующую структуру файлов бэкенда (PHP) и фронтенда (JS/CSS):

```
/usr/share/nginx/html/
├─ index.html
таблицы, модальные окна)
├─ css/
│   └─ style.css
управления
├─ js/
│   └─ app.js
запросы к API, фильтры)
└─ api/
    └─ users.php
пользователей (/etc/passwd)
    └─ groups.php
(/etc/group)
```

Корневая директория веб-сервера Nginx
Главный интерфейс панели (вкладки,
Стили оформления интерфейса панели
Клиентская логика (асинхронные Fetch-
Серверный обработчик для системных
Серверный обработчик для системных групп

!!!!!!!!!!!!!! Разработка Веб-Фронтенда !!!!!!!!!!!!!!

!!!!!! Убрать перед сборкой iso !!!!!!! Проверить версии linux Драйверы сетевой карты
реального сервера Изменить временно на 192.168.1.150 Стереть sfid

запись iso Изменить постоянно на 192.168.1.72 Вернуть sfid

!!!!!!!!!!!!!! переходим к iso !!!!!!!!!!!!!!!

From:
<https://wwoss.ru/> - worldwide open-source software

Permanent link:
https://wwoss.ru/doku.php?id=tmp_26.05.26_frontend&rev=1779807939

Last update: **2026/05/26 18:05**

