Установка и настройка файлового сервера Samba

Оглавление

[Подготовка сервера 2](#_Toc164121591)

[1. Время 2](#_Toc164121592)

[2. Брандмауэр 2](#_Toc164121593)

[Установка и запуск Samba 3](#_Toc164121594)

[Создание первого общего ресурса и предоставление к нему гостевого доступа (анонимного) 3](#_Toc164121595)

[Доступ к папке по логину и паролю 5](#_Toc164121596)

[Доступ к папке определенным пользователям и группам 6](#_Toc164121597)

[Сетевая корзина 8](#_Toc164121598)

[Подключение к общему ресурсу из Linux 10](#_Toc164121599)

[Монтирование 10](#_Toc164121600)

[SMB Browser 10](#_Toc164121601)

[Некоторые опции Samba 11](#_Toc164121602)

[hosts allow 11](#_Toc164121603)

[hosts deny 11](#_Toc164121604)

[interfaces 12](#_Toc164121605)

[Force User/Group 12](#_Toc164121606)

[Include 12](#_Toc164121607)

Samba позволяет настроить файловое хранилище различных масштабов — от малых офисов для крупных организаций. В данной инструкции мы рассмотрим процесс настройки файлового сервера. Сначала мы выполним установку и базовую настройку с предоставлением гостевого доступа. После будет приведены примеры разграничения доступа по пользователям, группам и пользователям Active Directory.

# Подготовка сервера

Прежде чем перейти к установке файлового сервера, выполним подготовку.

## 1. Время

Для корректного отображения дат, необходимо позаботиться о синхронизации времени. Для этого будем использовать демон chrony. Установим его:

apt update

apt install chrony

Разрешим автозапуск сервиса:

systemctl enable chrony

## 2. Брандмауэр

По умолчанию, в системах на базе Debian брандмауэр разрешает все соединения. Но если в нашем случае мы используем брандмауэр, необходимо открыть порты:

iptables -I INPUT -p tcp --dport 445 -j ACCEPT

iptables -I INPUT -p udp --dport 137:138 -j ACCEPT

iptables -I INPUT -p tcp --dport 139 -j ACCEPT

\* где порт **445** используется для samba, а порты **137**, **138** и **139** — для работы NetBIOS (использование имени компьютера для доступа).

Применяем настройки:

apt install iptables-persistent

netfilter-persistent save

## Установка и запуск Samba

Установка выполняется из репозитория одной командой:

apt update

apt install samba

Разрешаем автостарт сервиса:

systemctl enable smbd

И проверим, что сервис запустился:

systemctl status smbd

Проверяем, что сервер самба позволяет к себе подключиться. Для этого можно с компьютера обратиться к серверу по SMB, например, на компьютере с Windows это можно сделать из проводника, прописав путь к серверу с двух слэшей (см рисунок 1):

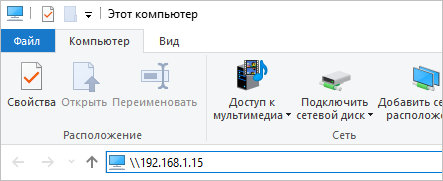


Рисунок 1. Пример подключения к SMB шаре с компьютера на Windows

\* в данном примере мы подключаемся к серверу Samba с IP-адресом **192.168.1.15**.

Если мы настроили сервер правильно, система откроет пустую папку. На данном этапе проверка закончена, и мы можем переходить к созданию первого общего ресурса (или, в простонародье, *шары*).

# Создание первого общего ресурса и предоставление к нему гостевого доступа (анонимного)

Разберем самый простой пример предоставления доступа к папке — анонимный доступ всем пользователям без запроса пароля.

Открываем на редактирование конфигурационный файл samba:

nano /etc/samba/smb.conf

И добавляем настройку для общей папки:

[Общая папка]

comment = Public Folder

path = /data/public

public = yes

writable = yes

read only = no

guest ok = yes

create mask = 0777

directory mask = 0777

force create mode = 0777

force directory mode = 0777

\* где:

**[Общая папка]** — имя общей папки, которое увидят пользователи, подключившись к серверу.

**comment** — свой комментарий для удобства.

**path** — путь на сервере, где будут храниться данные.

**public** — для общего доступа. Установите в yes, если хотите, чтобы все могли работать с ресурсом.

**writable** — разрешает запись в сетевую папку.

**read only** — только для чтения. Установите no, если у пользователей должна быть возможность создавать папки и файлы.

**guest ok** — разрешает доступ к папке гостевой учетной записи.

**create mask, directory mask, force create mode, force directory mode** — при создании новой папки или файла назначаются указанные права. В нашем примере права будут полные.

Создаем каталог на сервере и назначим права:

mkdir -p /data/public

chmod 777 /data/public

Применяем настройки samba, перезагрузив сервис:

systemctl restart smbd

Пробуем подключиться к папке. Мы должны зайти в нее без необходимости ввода логина и пароля.

## Доступ к папке по логину и паролю

Теперь создадим каталог, в который вход будет разрешен только авторизованным пользователям.

Открываем конфигурационный файл samba:

nano /etc/samba/smb.conf

Добавляем настройку для новой папки:

[Папка сотрудников]

comment = Staff Folder

path = /data/staff

public = no

writable = yes

read only = no

guest ok = no

create mask = 0777

directory mask = 0777

force create mode = 0777

force directory mode = 0777

\* эти настройки, во многом, похожи на те, что использовались в примере выше. Вот основные различия:

**path = /data/staff** — используем новый путь до папки.

**public = no** — запрещаем публичный доступ.

**guest ok = no** — не разрешаем гостевое подключение.

Создаем каталог для новой папки:

mkdir /data/staff

Задаем права на созданный каталог:

chmod 777 /data/staff

Создаем пользователя в системе Linux:

useradd staff1

\* где **staff1** — имя пользователя.

Задаем пароль для пользователя:

passwd staff1

Теперь создадим пользователя в samba:

smbpasswd -a staff1

Перезапускаем samba:

systemctl restart smbd

Пробуем зайти на сервер — общую папку мы должны открыть без авторизации, а при попытке открыть папку сотрудников должно появиться окно ввода логина и пароля.

**Если мы авторизованы на компьютере, с которого пытаемся подключиться к серверу, под той же учетной записью, что создали для доступа к папке, samba может и не потребовать аутентификации.**

## Доступ к папке определенным пользователям и группам

Теперь создадим папку, доступ к которой будут иметь ограниченное количество пользователей.

Открываем конфигурационный файл samba:

nano /etc/samba/smb.conf

Добавляем настройку для новой папки:

[Приватная папка]

comment = Private Folder

path = /data/private

public = no

writable = no

read only = yes

guest ok = no

valid users = admin, staff2, staff3, @privateusers

write list = admin, staff2

create mask = 0777

directory mask = 0777

force create mode = 0777

force directory mode = 0777

inherit owner = yes

\* стоит обратить внимание на следующие настройки:

**path = /data/private** — используем новый путь до папки.

**writable = no** и **read only = yes** — в данном примере мы разрешим запись в каталог только некоторым пользователям. Поэтому общие настройки, разрешающие запись в папку, должны быть запрещены.

**valid users** — список пользователей, которым разрешено подключаться к каталогу. В данном примере разрешения работают для пользователей **admin**, **staff2** и **staff3**, а также для всех, кто входим в группу **privateusers**.

**write list** — список пользователей, которые имеют доступ к папке на чтение и запись. В данном примере мы разрешаем это только для пользователей **admin** и **staff2**.

**inherit owner** — опция позволяем включить наследование владельца при создании папок и файлов.

\* если мы хотим, чтобы доступ к каталогу был полный у определенных пользователей (без разделения на тех, кто может только читать и тех, кто может также писать в папку), то опцию **write list** можно не указывать, а опции **writable** и **read only** оставить как в примерах выше.

Создаем каталог для новой папки:

mkdir /data/private

Задаем права на созданный каталог:

chmod 777 /data/private

Для применения настроек перезапускаем samba:

systemctl restart smbd

Создать группу пользователей можно командой:

groupadd privateusers

Добавить ранее созданного пользователя в эту группу:

usermod -a -G privateusers staff1

Проверяем возможность работы с новым каталогом.

# Сетевая корзина

При удалении файлов из общей папки, данные удаляются навсегда. Но мы можем настроить сетевую корзину — скрытый каталог, в который будут перемещаться удаляемые с самбы объекты.

Открываем конфигурационный файл:

nano /etc/samba/smb.conf

Создаем общий ресурс:

[Recycle]

comment = Snap Directories

path = /data/recycle

public = yes

browseable = yes

writable = yes

vfs objects = recycle

recycle:repository = .recycle/%U

recycle:keeptree = Yes

recycle:touch = Yes

recycle:versions = Yes

recycle:maxsize = 0

recycle:exclude = \*.tmp, ~$\*

recycle:exclude\_dir = /tmp

\* где:

**vfs objects = recycle** — использовать подсистему recycle.

**recycle:repository** — где хранить удаленные объекты. В данном примере удаленные файлы попадут в скрытый каталог **.recycle** к котором создастся каталог с именем пользователя, удалившего файл или папку.

**recycle:keeptree** — удалять объекты с сохранение дерева каталогов.

**recycle:touch** — изменить ли дату изменения файла при его перемещении в корзину.

**recycle:versions** — при удалении файлов с совпадающими именами, добавлять номер версии.

**recycle:maxsize** — не помещать в корзину файлы, размер которых больше заданного параметра (в байтах). В данном примере, помещать файлы любого размера.

**recycle:exclude** — исключить файлы.

**recycle:exclude\_dir** — исключить каталог.

Создаем каталог и задаем права:

mkdir /data/recycle

chmod 777 /data/recycle

Перезапускаем самбу:

systemctl restart smbd

Пробуем зайти в сетевой каталог Recycle и создать, а после удалить файл. Он должен оказаться к скрытой папке .recycle.

Для автоматической чистки сетевой корзины можно создать скрипт:

mkdir /scripts

nano /scripts/cleanrecycle.sh

Внести следующие данные:

#!/bin/bash

recyclePath="/data/recycle/.recycle"

maxStoreDays="30"

/usr/bin/find $recyclePath -name "\*" -ctime +$maxStoreDays -exec rm {} \;

\* в данном скрипте мы ищем все файлы в каталоге /data/recycle/.recycle, которые старше 30 дней и удаляем их.

Разрешаем запуск скрипта:

chmod +x /scripts/cleanrecycle.sh

Создаем задание в планировщике:

crontab -e

0 5 \* \* \* /scripts/cleanrecycle.sh

\* в данном примере мы будем запускать скрипт по очистке сетевой корзины каждый день в 05:00.

# Подключение к общему ресурсу из Linux

## Монтирование

В Linux мы можем монтировать удаленный каталог с помощью команды mount, например:

mount -t cifs "//192.168.1.15/ad" /mnt -o user=user

\* где **192.168.1.15** — IP-адрес сервера; **mnt** — каталог, куда монтируем сетевую шару; **user** — пользователь, под которым выполняем подключение к сетевому каталогу.

\*\* в систему должен быть установлен пакет **cifs-utils**.

## SMB Browser

Также мы можем увидеть содержимое удаленных папок на samba при помощи клиента smb. Для начала установим данного клиента:

1. на Red Hat / CentOS / Fedora:

yum install samba-client

1. на Debian / Ubuntu / Mint:

apt install samba-client

После вводим команду:

smbclient -L 192.168.1.15 -U staff@user.local

\* где **192.168.1.15** — сервер samba, к которому мы пытаемся подключиться**; staff@user.local** — учетная запись, под которой выполняется подключение.

... мы получим список каталогов, к которым предоставлен общий доступ на сервере.

Также мы можем подключиться к конкретной папке, например:

smbclient \\\\192.168.1.15\\ad -U staff@user.local

Мы подключимся клиентом samba — можно выполнить запрос на показ содержимого:

smb: \> ls

Или полный список возможных команд:

smb: \> help

# Некоторые опции Samba

Рассмотрим некоторые полезные опции, которые могут пригодится при настройке Samba.

## hosts allow

Параметр задает список разрешенных хостов или сетей, с которых можно подключаться к серверу. Если его прописать в секцию с настройкой шары, то он будет действовать только для нее, если прописать в [global], то для всех общих папок.

Пример использования параметра.

hosts allow = comp1, 192.168.1., 192.168.160.0/255.255.252.0

\* в нашем примере мы разрешим доступ только для компьютера **comp1**, компьютеров из сетей **192.168.1.0/24** и **192.168.160.0/22**.

## hosts deny

Параметр аналогичный hosts allow, только он наоборот — запрещает доступ для хостов и сетей. Например:

hosts deny = comp2, 192.168.2., 192.168.164.0/255.255.252.0

hosts deny = ALL EXCEPT 192.168.3.

## interfaces

По умолчанию samba использует все сетевые интерфейсы, кроме локальной петли (127.0.0.1). Параметр interfaces позволит указать, на каком сетевом адаптере файловый сервер должен принимать запросы. Пример:

interfaces = ens32 192.168.1.15/24 192.168.2.15/255.255.255.0

## Force User/Group

Опции force user и force group позволяют задать пользователя и группу, от которых будет работать подключение пользователя к samba. Параметр может оказаться полезным для создания шары из каталога с уже назначенным владельцем, которого мы не хотим менять.

Прописывается для общего ресурса:

force user = apache

force group = apache

\* данная настройка позволит подключаться к общему ресурсу под пользователем apache.

## Include

Позволяет подключить дополнительный файл с конфигурацией.

include = /etc/samba/conf.d/shares.conf

\* данная опция будет дополнительно подключать конфигурацию из файла **/etc/samba/conf.d/shares.conf** (прописывается полный путь).