Установка и настройка Samba на Ubuntu и Debian

Рассмотрим процесс установки ПО Samba на ОС Ubuntu 20.04. Для работы будем использовать облачные ресурсы. Также эта инструкция подойдет в том числе для установки Samba на Debian и Astra Linux. А начнем с краткого описания этого программного обеспечения.

Что такое Samba

Samba представляет собой программный пакет, разработанный для обеспечения совместимости и взаимодействия UNIX-подобных систем с Windows. ПО распространяется по свободной лицензии уже на протяжении 30 лет. Samba обеспечивает беспроблемную интеграцию серверов и ПК, работающих под управлением UNIX, в систему AD (Active Directory). Это ПО может использоваться как контроллер и в качестве стандартной составляющей домена. Таким образом, у пользователей появляются возможности по гибкой настройке облачных файловых хранилищ. Samba обеспечивает широкий функционал по управлению правами доступа к файлам и базам данных с назначением определенных групп пользователей.

Добавление пользователя

Здесь всё просто, введите команду:

```
sudo useradd -p new_server_pass new_server_user
```

Вместо `new_server_pass` и `new_server_user` можно использовать любой пароль и любое имя пользователя. Указанные здесь имя и пароль приведены просто для примера. Обратите внимание, что мы сразу ввели пароль, что стало возможным благодаря команде `-p`. Теперь используем команду `update` для обновления хранилища, и можно переходить к дополнительным настройкам.

Дополнительные настройки

Для подготовки к установке Linux Samba потребуется выполнить еще пару действий:

- Синхронизировать время. Это необходимо сделать для того, чтобы впоследствии корректно отображались даты. Это поможет в дальнейшем облегчить поиск данных, с поиском которых при некорректно отображаемом времени могли бы возникнуть проблемы.
- Настроить работу брандмауэра. Хотя в большинстве систем на основе Linux все порты открыты, некоторые нужные для работы Samba (чаще всего это 445 и порты 137-139) могут быть закрыты. Поэтому проверьте их доступность и при необходимости откройте их.

Для синхронизации времени используйте специальную утилиту `chrony` — добавленная в автозагрузку, она будет поддерживать правильные временные значения. Это добавление осуществляется с помощью простой команды:

```
sudo systemctl enable chrony
```

Теперь запускаем программу, используя команду `start`, и выставляем нужный часовой пояс, а корректность установки смотрим при помощи команды `date`. Вывод должен быть примерно таким:

```
Fri Oct 28 11:25:34 UTC 2022
```

Вот и всё, сервер готов к установке и настройке Samba на Ubuntu.

Установка Samba на Ubuntu 20.04

Для удобства разбили процесс установки на отдельные этапы.

Шаг 1. Подготовка

Для старта процесса установки используйте следующую команду:

```
sudo apt install samba -y
```

Теперь нужно запомнить системное имя сервиса. В большинстве случаев это `smbd`. Поэтому, если хотите вызвать сервис, то вводите именно это значение.

- И для начала настроим автозапуск, что делается командой `enable smbd`.
- Теперь запускаем, используя уже знакомую команду `start`.
- Затем проверяем статус системы при помощи `status`.

- Для остановки Samba используется `stop`.
- Чтобы перезапустить сервис, введите команду `restart`.
- Если хотите, чтобы Samba больше не запускалась автоматически, используйте инструкцию `disable`.
- Команда `reload` нужна для обновления конфигурации.

Следующая команда позволит принудительно открыть порт 445, а также 137-139 (помните про настройки брандмауэра?). Чтобы разрешить их в брандмауэре *ufw*, используйте:

```
sudo ufw allow Samba
```

Шаг 2. Настройка анонимного доступа

Допустим, у нас есть какой-то удаленный сервер, который находится за пределами нашего облака. Правила сетевой безопасности требуют ни в коем случае не открывать прямой доступ к нему через IP. Сделать это можно только через VPN-туннель, который к тому же уже настроен. Обычно серверы, к которым предоставлен VPN-доступ, имеют адрес 10.8.0.1, и именно с этим адресом в дальнейшем нам и предстоит работать.

Чтобы расшарить данные и предоставить к ним анонимный доступ, сначала откройте конфигурационный файл. Он находится здесь: `/etc/samba/smb.conf`. Рекомендуем сделать бэкап чистого файла — это поможет затем быстро восстановить исходное состояние программы без необходимости ее переустановки. Теперь удалите все комментарии, оставив только код, и введите команду `testparm`, чтобы убедиться, что программа работает нормально. В настройках общей папки введите следующие параметры:

```
[share]
    comment = шара
    path = /data/public_share
    public = yes
    writable = yes
    read only = no
    guest ok = yes
```

Также следите за тем, чтобы в следующих четырех полях (`mask` и `mode`) совпадали числовые значения (например, 0777). Что касается конкретных строк, то здесь:

- `[share]` — имя расшаренной папки, которое будет видно всем подключившимся к вашему серверу;
- `comment` — комментарий, который может быть каким угодно;
- `path` — путь к папке для хранения данных;
- `public` — дает разрешение на общий доступ: если не хотите, чтобы пользователи могли просматривать содержимое папки, установите значение `no`;
- `writable` — определяет, можно ли записывать данные в папке;
- `read only` — указывает, что папка только для чтения: чтобы пользователи могли создавать новые, поставьте значение `no`;
- `guest ok` — определяет, могут ли гости получать доступ к папке.

Таким образом, название папки и путь могут отличаться в зависимости от того, какие значения вы указали для общей папки. Комментарий тоже может быть любым, а для нижних четырех параметров значения устанавливаются в виде `yes` или `no`. Теперь перезапускаем программу и на всякий случай проверяем, можно ли подключиться к серверу из-под Windows.

Шаг 3. Настройка доступа по учетным данным пользователя

Для создания доступа по логину и паролю нужно в первую очередь создать новую директорию и настроить права. В файле конфигурации установите все параметры на `no` (см. выше), кроме `writable`: в этой строке должно стоять значение `yes`, то есть запись в папке должна быть включена.

Командой `mkdir` создаем новую директорию, далее с помощью `useradd someone` (вместо *someone* может стоять любое имя пользователя) создаем пользователя и задаем ему пароль командой `passwd`. Например, так:

```
passwd something
```

Теперь командой `sudo smbpasswd -a someone` добавляем нового пользователя и пытаемся выполнить вход: если всё настроено корректно, то нам будет открыт доступ к папке.

Шаг 4. Настройка группового доступа

Это понадобится для создания ограниченного доступа отдельных групп пользователей. В `smb.conf` после строки `guest ok` дополнительно пропишите следующие строки (все имена пользователей были сгенерированы просто для примера):

```
valid users = admin, vlasova_ulyana, usova_elena, sokolova_eva, spiridonova_kseniya
write list = admin, spiridonova_kseniya
```

В строке `valid users` указываются пользователи, которым предоставлен доступ к директории. А в строке `write list` указаны те, кто может изменять данные в папке. Кроме того, после строки `force directory mode` добавьте еще одну строку со следующим значением:

```
inherit owner = yes
```

Она необходима для того, чтобы можно было включать наследование создаваемых объектов. Теперь записываем настройки и перезапускаем сервис, после чего новые настройки должны вступить в силу.

Шаг 5. Подключение к ресурсу из-под Windows и Linux

- Для быстрого подключения к Самбе из-под Windows нажмите *Ctrl+E* и введите путь. Только учтите, что нужно использовать \\ для указания пути к ресурсу в сети. А чтобы не соединяться с сервером постоянно, можно выбрать опцию подключения ресурса как диска, если это позволяет ваша политика безопасности. В новом окне задайте букву для наименования диска и заполните необходимые данные.
- Для подключения к Самбе из-под Linux используются утилиты `cifs`, которые устанавливаются с помощью команды `sudo apt install cifs-utils -y`. Далее ресурс монтируется и выполняется подключение. Это делается с помощью `mount.cifs //10.8.0.1/our_share /share` (путь и название ресурса могут быть любыми). Можно выполнять монтирование и автоматическим способом, для чего используется файл конфигурации `fstab` со своими настройками.

Шаг 6. Настройка сетевой корзины

Эта операция понадобится, чтобы избежать случайного безвозвратного удаления файлов. Для этого создаем такую директорию:

```
[Recycle]
    comment = Корзина для временного хранения файлов
    path = /directory/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
```

Теперь рассмотрим построчно, что означают эти параметры (кроме первых шести строк, с ними понятно):

- `vfs objects = recycle` — указание на использование соответствующей подсистемы;
- `repository` — путь для хранения удаленных данных;
- `keeptree` — сохранять ли дерево каталогов после удаления;
- `touch` — нужно ли изменять временные метки файлов при их помещении в корзину;
- `versions` — нужно ли указывать номер версии, если удаляются файлы с одинаковыми именами;
- `maxsize` — максимальный размер помещаемого в корзину файла. Значение 0 отключает лимиты;
- `exclude` — какие типы файлов нужно исключить;

- `exclude_dir` — какие каталоги нужно исключить.

Заключение

На этом всё, теперь вы знаете, что нужно делать перед установкой Самбы, как ее ставить на облачный сервер и настраивать под собственные нужды.

Revision #1
Created 2023-11-14 18:24:39 UTC by odiljonov
Updated 2023-11-14 18:25:31 UTC by odiljonov