

Using Samba in your File Manager

Warning: (30 June 2018) This page needs updating; search the forums for more current info

The following tutorial will guide you through setting up usersharing with Samba so that you can use your file manager to share folders. Ubuntu style.

Contents

[\[hide\]](#)

- [1_Installation](#)
 - [1.1_GTK+](#)
 - [1.2_KDE](#)
- [2_Configuration](#)
 - [2.1_Automatic](#)
 - [2.2_Manual](#)
- [3_Post-install \(Users and Groups\)](#)
- [4_Sharing In The Home Directory](#)
- [5_See Also](#)

Installation

First we need to install a few packages. The base package `samba` is required in addition to a few more depending on your desktop or file manager.

```
sudo pacman -S samba libwbclient nemo-share smb4k thunar-shares-plugin smbclient  
smbnetfs gvfs-smb libsmbios manjaro-settings-samba
```

GTK+

Install `samba`, `gvfs-smb`, and the share package for your file manager.

Cinnamon / Nemo

```
sudo pacman -S samba gvfs-smb nemo-share
```

Gnome / Nautilus

```
sudo pacman -S samba gvfs-smb nautilus-share
```

Mate / Caja

```
sudo pacman -S samba gvfs-smb caja-share
```

XFCE / Thunar

```
sudo pacman -S samba gvfs-smb thunar-shares-plugin
```

KDE

KDE has the ability to browse shares built-in, but still requires `samba` to share folders and `kdenetwork-filesharing` for settings and easier use.

```
sudo pacman -S samba kdenetwork-filesharing
```

Configuration

Automatic

```
sudo pacman -S manjaro-settings-samba
```

Manual

Now we are ready to set up Manjaro for usershares. Usershares allows a non-root user to add, modify, and delete their own samba shares.

First we're going to create the usershare path. This is where samba stores the share configuration (so it's not going in `/etc/samba/smb.conf`) In the terminal, enter:

```
mkdir -p /var/lib/samba/usershare
```

We have now added the usershares directory in `/var/lib/samba`.

Next we need to create the sambashare group. In the terminal, enter:

```
groupadd sambashare
```

We need to make user `root` owner of both the usershares directory and the sambashare group.

In the terminal, enter:

```
chown root:sambashare /var/lib/samba/usershare
```

Because `/var/lib/samba/usershare` is now owned by `root`, we need to make the usershare directory accessible for non-root users. In the terminal, enter:

```
chmod 1770 /var/lib/samba/usershare
```

This `chmod` command sets the sticky bit (makes the permissions fixed for non-root users), as signified by the preceding 1 in the 1770 string. The 7+7 signifies that users and groups can read, write and execute. The 0 means that "others" have no rights to the directory.

Now we need to create a new `smb.conf` from the template configuration file. In the terminal, enter:

```
cp /etc/samba/smb.conf.default /etc/samba/smb.conf
```

Open the newly created `smb.conf` in a text editor. In the terminal, enter:

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

Replace `nano` with the name of your preferred text editor.

To make usershares possible we need to add the following parameters under section `[global]`:

```
usershare path = /var/lib/samba/usershare
usershare max shares = 100
usershare allow guests = yes
usershare owner only = yes
```

Approximately halfway in the `[global]` section is the parameter `security = user`. Find this line and add the following immediately after:

```
map to guest = bad user
```

This line makes it possible for users without a "proper username" to still connect to a share.

Save the `smb.conf` file with `CTRL+O` and close nano with `CTRL+X`.

Post-install (Users and Groups)

Now add your user to the `sambashare` group. Replace `<username>` with your real username. In the terminal, enter:

```
usermod -a -G sambashare <username>
```

We still need to enable the samba service. In the terminal, enter:

```
systemctl enable smb nmb  
systemctl start smb nmb
```

Log out and log back in. It should now be possible to configure samba shares using the GUI. For instance, in Gnome Files you can right click on any directory and share it on the network.

Sharing In The Home Directory

To be able to share directories in your home (`/home/<username>`) you also need to add new permissions to your home (`/home/<username>`). Replace `<username>` with your own username. In the terminal, enter:

```
chmod 701 /home/<username>
```

The 701 gives read, write and execute permissions to the user, zero rights to groups and execute rights to "other". The execute rights for "other" seems to be required for samba to be able to access the lower directories under `/home/<username>`. Other users can't enter your home directory with only the execute bit set, but it might lessen security, as others now do have permission to execute stuff under your home. There needs to be executable stuff in there beforehand, though, and others need to know the path to the executable by heart, before they can run it. It doesn't seem to have much room for mischief, but caveat emptor.