# Fedora 32: Simple Local File-Sharing with Samba

Posted by da2ce7 on June 12, 2020


Sharing files with Fedora 32 using Samba is cross-platform, convenient, reliable, and performant.

## What is 'Samba'?

Samba is a high-quality implementation of Server Message Block protocol (SMB). Originally developed by Microsoft for connecting windows computers together via local-area-networks, it is now extensively used for internal network communications.

Apple used to maintain it's own independent file sharing called "Apple Filing Protocol (**AFP**)", however in recent times, it also has also switched to SMB.

**In this guide we provide the minimal instructions to enable:**

- Public Folder Sharing (Both Read Only and Read Write)
- User Home Folder Access

```
Note about this guide: The convention '~]$' for a local user command prompt, and '~]#' for a
super user prompt will be used.
```

## Public Sharing Folder

Having a shared public place where authenticated users on an internal network can access files, or even modify and change files if they are given permission, can be very convenient. This part of the guide walks through the process of

setting up a shared folder, ready for sharing with Samba.

```
Please Note: This guide assumes the public sharing folder is on a Modern Linux Filesystem; other
filesystems such as NTFS or FAT32 will not work. Samba uses POSIX Access Control Lists (ACLs).

For those who wish to learn more about Access Control Lists, please consider reading the
documentation: "Red Hat Enterprise Linux 7: System Administrator's Guide: Chapter 5. Access
Control Lists", as it likewise applies to Fedora 32.

In General, this is only an issue for anyone who wishes to share a drive or filesystem that was
created outside of the normal Fedora Installation process. (such as a external hard drive).

It is possible for Samba to share filesystem paths that do not support POSIX ACLs, however this
is out of the scope of this guide.
```

## Create Folder

For this guide the ***/srv/public/*** folder for sharing will be used.

> *The /srv/ directory contains site-specific data served by a Red Hat Enterprise Linux system. This directory gives users the location of data files for a particular service, such as FTP, WWW, or CVS. Data that only pertains to a specific user should go in the /home/ directory.*
>
> — **RED HAT ENTERPRISE LINUX 7, STORAGE ADMINISTRATION GUIDE: CHAPTER 2. FILE SYSTEM STRUCTURE AND MAINTENANCE: 2.1.1.8. THE /SRV/ DIRECTORY**

```
Make the Folder (will provide an error if the folder already exists).
~]# mkdir --verbose /srv/public

Verify folder exists:
~]$ ls --directory /srv/public

Expected Output:
/srv/public
```

## Set Filesystem Security Context

To have *read and write* access to the public folder the *public_content_rw_t* security context will be used for this guide. Those wanting *read only* may use: *public_content_t*.

> *Label files and directories that have been created with the public_content_rw_t type to share them with read and write permissions through vsftpd. Other services, such as Apache HTTP Server, Samba, and NFS, also have access to files labeled with this type. Remember that booleans for each service must be enabled before they can write to files labeled with this type.*
>
> — **RED HAT ENTERPRISE LINUX 7, SELINUX USER'S AND ADMINISTRATOR'S GUIDE: CHAPTER 16. FILE TRANSFER PROTOCOL: 16.1. TYPES: PUBLIC_CONTENT_RW_T**

Add */srv/public* as *“public_content_rw_t”* in the system’s local filesystem security context customization registry:

```
Add new security filesystem security context:
~]# semanage fcontext --add --type public_content_rw_t "/srv/public(/.*)?"

Verifiy new security filesystem security context:
~]# semanage fcontext --locallist --list

Expected Output: (should include)
/srv/public(/.*)? all files system_u:object_r:public_content_rw_t:s0
```

Now that the folder has been added to the local system’s filesystem security context registry; The **restorecon** command can be used to ‘restore’ the context to the folder:

```
Restore security context to the /srv/public folder:
$~]# restorecon -Rv /srv/public

Verify security context was correctly applied:
~]$ ls --directory --context /srv/public/

Expected Output:
unconfined_u:object_r:public_content_rw_t:s0 /srv/public/
```

# User Permissions

## Creating the Sharing Groups

To allow a user to either have *read only*, or *read and write* accesses to the public share folder create two new groups that govern these privileges: *public_readonly* and *public_readwrite*.

User accounts can be granted access to *read only*, or *read and write* by adding their account to the respective group (and allow login via Samba creating a smb password). This process is demonstrated in the section: “Test Public Sharing (localhost)”.

```
Create the public_readonly and public_readwrite groups:
~]# groupadd public_readonly
~]# groupadd public_readwrite

Verify successful creation of groups:
~]$ getent group public_readonly public_readwrite

Expected Output: (Note: x:1...: number will probability differ on your System)
public_readonly:x:1009:
public_readwrite:x:1010:
```

## Set Permissions

Now set the appropriate user permissions to the public shared folder:

```
Set User and Group Permissions for Folder:
~]# chmod --verbose 2700 /srv/public
~]# setfacl -m group:public_readonly:r-x /srv/public
~]# setfacl -m default:group:public_readonly:r-x /srv/public
~]# setfacl -m group:public_readwrite:rwx /srv/public
~]# setfacl -m default:group:public_readwrite:rwx /srv/public

Verify user permissions have been correctly applied:
~]$ getfacl --absolute-names /srv/public

Expected Output:
file: /srv/public
owner: root
group: root
flags: -s-
user::rwx
group::---
group:public_readonly:r-x
group:public_readwrite:rwx
mask::rwx
other::---
default:user::rwx
default:group::---
default:group:public_readonly:r-x
default:group:public_readwrite:rwx
default:mask::rwx
default:other::---
```

# Samba

## Installation

```
~]# dnf install samba
```

## Hostname (systemwide)

Samba will use the name of the computer when sharing files; it is good to set a hostname so that the computer can be found easily on the local network.

```
View Your Current Hostname:
~]$ hostnamectl status
```

If you wish to change your hostname to something more descriptive, use the command:

```
Modify your system's hostname (example):
~]# hostnamectl set-hostname "simple-samba-server"
```

```
For a more complete overview of the hostnamectl command, please read the previous Fedora
Magazine Article: "How to set the hostname on Fedora".
```

# Firewall

Configuring your firewall is a complex and involved task. This guide will just have the most minimal manipulation of the firewall to enable Samba to pass through.

```
For those who are interested in learning more about configuring firewalls; please consider
reading the documentation: "Red Hat Enterprise Linux 8: Securing networks: Chapter 5. Using and
configuring firewall", as it generally applies to Fedora 32 as well.
```

```
Allow Samba access through the firewall:
~]# firewall-cmd --add-service=samba --permanent
~]# firewall-cmd --reload

Verify Samba is included in your active firewall:
~]$ firewall-cmd --list-services

Output (should include):
samba
```

# Configuration

## Remove Default Configuration

The stock configuration that is included with Fedora 32 is not required for this simple guide. In particular it includes support for sharing printers with Samba.

For this guide make a backup of the default configuration and create a new configuration file from scratch.

```
Create a backup copy of the existing Samba Configuration:
~]# cp --verbose --no-clobber /etc/samba/smb.conf /etc/samba/smb.conf.fedora0

Empty the configuration file:
~]# > /etc/samba/smb.conf
```

## Samba Configuration

```
Please Note: This configuration file does not contain any global definitions; the defaults
provided by Samba are good for purposes of this guide.
```

```
Edit the Samba Configuration File with Vim:
~]# vim /etc/samba/smb.conf
```

Add the following to */etc/samba/smb.conf* file:

```
# smb.conf - Samba Configuration File

# The name of the share is in square brackets [],
```

```
#   this will be shared as //hostname/sharename

# There are a three exceptions:
#   the [global] section;
#   the [homes] section, that is dynamically set to the username;
#   the [printers] section, same as [homes], but for printers.

# path: the physical filesystem path (or device)
# comment: a label on the share, seen on the network.
# read only: disable writing, defaults to true.

# For a full list of configuration options,
#   please read the manual: "man smb.conf".

[global]

[public]
path = /srv/public
comment = Public Folder
read only = No
```

## Write Permission

By default Samba is not granted permission to modify any file of the system. Modify system's security configuration to allow Samba to modify any filesystem path that has the security context of *public_content_rw_t*.

For convenience, Fedora has a built-in SELinux Boolean for this purpose called: *smbd_anon_write*, setting this to *true* will enable Samba to write in any filesystem path that has been set to the security context of *public_content_rw_t*.

For those who are wishing Samba only have a read-only access to their public sharing folder, they may choose skip this step and not set this boolean.

```
There are many more SELinux boolean that are available for Samba. For those who are interested,
please read the documentation: "Red Hat Enterprise Linux 7: SELinux User's and Administrator's
Guide: 15.3. Samba Booleans", it also apply to Fedora 32 without any adaptation.
```

```
Set SELinux Boolean allowing Samba to write to filesystem paths set with the security context
public_content_rw_t:
~]# setsebool -P smbd_anon_write=1

Verify bool has been correctly set:
$ getsebool smbd_anon_write

Expected Output:
smbd_anon_write --> on
```

# Samba Services

The Samba service is divided into two parts that we need to start.

## Samba 'smb' Service

The Samba “Server Message Block” (SMB) services is for sharing files and printers over the local network.

Manual: “smbd – server to provide SMB/CIFS services to clients“

# Enable and Start Services

```
For those who are interested in learning more about configuring, enabling, disabling, and
managing services, please consider studying the documentation: "Red Hat Enterprise Linux 7:
System Administrator's Guide: 10.2. Managing System Services".
```

```
Enable and start smb and nmb services:
~]# systemctl enable smb.service
~]# systemctl start smb.service

Verify smb service:
~]# systemctl status smb.service
```

# Test Public Sharing (localhost)

To demonstrate allowing and removing access to the public shared folder, create a new user called *samba_test_user*, this user will be granted permissions first to read the public folder, and then access to read and write the public folder.

The same process demonstrated here can be used to grant access to your public shared folder to other users of your computer.

The *samba_test_user* will be created as a locked user account, disallowing normal login to the computer.

```
Create 'samba_test_user', and lock the account.
~]# useradd samba_test_user
~]# passwd --lock samba_test_user

Set a Samba Password for this Test User (such as 'test'):
~]# smbpasswd -a samba_test_user
```

## Test Read Only access to the Public Share:

```
Add samba_test_user to the public_readonly group:
~]# gpasswd --add samba_test_user public_readonly

Login to the local Samba Service (public folder):
~]$ smbclient --user=samba_test_user //localhost/public

First, the ls command should succeed,
Second, the mkdir command should not work,
and finally, exit:
smb: \> ls
smb: \> mkdir error
smb: \> exit
```

```
Remove samba_test_user from the public_readonly group:
gpasswd --delete samba_test_user public_readonly
```

### Test Read and Write access to the Public Share:

```
Add samba_test_user to the public_readwrite group:
~]# gpasswd --add samba_test_user public_readwrite

Login to the local Samba Service (public folder):
~]$ smbclient --user=samba_test_user //localhost/public

First, the ls command should succeed,
Second, the mkdir command should work,
Third, the rmdir command should work,
and finally, exit:
smb: \> ls
smb: \> mkdir success
smb: \> rmdir success
smb: \> exit

Remove samba_test_user from the public_readwrite group:
~]# gpasswd --delete samba_test_user public_readwrite
```

After testing is completed, for security, disable the **samba_test_user**'s ability to login in via samba.

```
Disable samba_test_user login via samba:
~]# smbpasswd -d samba_test_user
```

# Home Folder Sharing

In this last section of the guide; Samba will be configured to share a user home folder.

For example: If the user bob has been registered with *smbpasswd*, bob's home directory */home/bob*, would become the share *//server-name/bob*.

This share will only be available for bob, and no other users.

```
This is a very convenient way of accessing your own local files; however naturally it carries at
a security risk.
```

## Setup Home Folder Sharing

### Give Samba Permission for Public Folder Sharing

```
Set SELinux Boolean allowing Samba to read and write to home folders:
~]# setsebool -P samba_enable_home_dirs=1

Verify bool has been correctly set:
```

```
$ getsebool samba_enable_home_dirs

Expected Output:
samba_enable_home_dirs --> on
```

## Add Home Sharing to the Samba Configuration

**Append the following to the systems smb.conf file:**

```
# The home folder dynamically links to the user home.

# If 'bob' user uses Samba:
# The homes section is used as the template for a new virtual share:

# [homes]
# ...   (various options)

# A virtual section for 'bob' is made:
# Share is modified: [homes] -> [bob]
# Path is added: path = /home/bob
# Any option within the [homes] section is appended.

# [bob]
#       path = /home/bob
# ...   (copy of various options)


# here is our share,
# same as is included in the Fedora default configuration.

[homes]
        comment = Home Directories
        valid users = %S, %D%w%S
        browseable = No
        read only = No
        inherit acls = Yes
```

## Reload Samba Configuration

```
Tell Samba to reload it's configuration:
~]# smbcontrol all reload-config
```

# Test Home Directory Sharing

```
Switch to samba_test_user and create a folder in it's home directory:
~]# su samba_test_user
samba_test_user:~]$ cd ~
samba_test_user:~]$ mkdir --verbose test_folder
samba_test_user:~]$ exit

Enable samba_test_user to login via Samba:
~]# smbpasswd -e samba_test_user

Login to the local Samba Service (samba_test_user home folder):
```

```
$ smbclient --user=samba_test_user //localhost/samba_test_user

Test (all commands should complete without error):
smb: \> ls
smb: \> ls test_folder
smb: \> rmdir test_folder
smb: \> mkdir home_success
smb: \> rmdir home_success
smb: \> exit

Disable samba_test_user from login in via Samba:
~]# smbpasswd -d samba_test_user
```

FAQS AND GUIDES | FOR SYSTEM ADMINISTRATORS | FIREWALL | SAMBA | SELINUX

**da2ce7**

## 31 COMMENTS

### RMPR

Instead of
Make the Folder (will provide an error if the folder already exists).
~]# mkdir –verbose /srv/public

Verify folder exists:
~]$ ls –directory /srv/public

Expected Output:
/srv/public

I would suggest
~]# mkdir -p /srv/public

JUNE 12, 2020

### da2ce7

Hello RMPR, I considered '–recursive' option for creating the path for /srv/public, however, I felt it was important that a user of the guide be notified that on this path might have been used previously. (instead of silently continuing).

JUNE 14, 2020

### Osqui

Why use Samba instead of NFS? From a technically perspective. Thanks!

JUNE 12, 2020

### Alberto Patino

If you have your Windows 10 machine then you can use File Explorer and point to \hostname\srv\public and voila, your Windows machine will open a folder to read/write files in a Linux/Unix Server.

JUNE 12, 2020

### Gregory Bartholomew

Hi Osqui:

As with many things, I think it depends on what you are trying to do. Generally, NFS tends to be used when you want to centralize a system-wide directory like /opt or even /. Samba tends to be used for “smaller” file shares that will only be accessed by a few users.

One of the differences between them that used to be a big problem for sharing home directories over samba is the allowed characters. I haven’t verified this with more recent versions, but it used to be that, among others, the colon character “:” was not allowed in file names on samba shares. This was sometimes a problem for X11 users because X11 needed to create a file with a colon in its name in the user’s home directory in order to function properly.

Authentication is handled differently as well. NFS (v4+) uses kerberos tickets. Samba can be configured to prompt for a username and password combination. In the past, samba has had some security problems with its authentication systems (see for example pass-the-hash). But I think modern versions are much better (as long as you keep the legacy authentication methods disabled).

JUNE 12, 2020

### Ralph

The first half of this article seems to be devoted entirely to SElinux issues.
What, if I don’t care about SElinux and have my RHEL/CentOS/Fedora based hosts run in SElinux permissive mode at best?
Furthermore my RasPi device with attached disk storage, which seems to lend itself to host a Samba server for my wireless home network (admittedly throughput-wise not the best choice) , doesn’t even have any notion about SElinux since it runs on Raspbian (i.e. some Debian derivative), and I have never heard of even the most Debian die-hards that they would go into the fuss to turn their OS into an SElinux hardened one.
Though I have nothing against SElinux, especially on a RHEL based OS, I feel it kind of obfuscates a little the Samba tutorial for the average linux aficionado who just wants to setup quickly a low-maintenance file, print and scanner server that can also be used by the rest of the family members, who could not be convinced to recant their strange obsession for Windows and Android devices.
Apropos, it would be highly appreciated if you could provide a continuation that would show how to extend your Samba server to also offer print and scanner services.

P.S. could Fedora Magazine issue an article about Fedora or CentOS on ARM respectively on SBCs like RasPis?
Is the Pidora project still alive, and how does it compare to Raspbian or Arch on ARM?

JUNE 12, 2020

### Gregory Bartholomew

As a magazine editor, I'm approving this comment, but in anticipation of the response, I'm asking that people not "jump on" Ralph. I will likely delete negative responses to Ralph, don't expect them to get through. He has a valid viewpoint and SELinux isn't without its issues. In particular, SELinux has a UI issue that is well explained here: https://dev.to/goober99/selinux-has-a-ui-problem-4knj.

SElinux is FAR more complex than those wonderful 9 bits that we can all see with ls and set with chmod, but in this new world of connectivity, complex program-to-program interactions, and endless data breaches, it is necessary.

Ralph, admittedly, this article probably should have been titled "Local File-Sharing with Samba and SELinux". If you want to do it without SELinux, you can just skip all the SELinux related stuff and instead set

write list = @public_readwrite
on the share to get the same behavior. If your server supports it, you will also have to google how to disable SELinux, but because it is something that we don't want to encourage people to do, as a policy, we do not tell people how to do that on this magazine.

Thanks for your understanding. Hopefully the UI for SELinux will improve with time.

Also, this is a magazine for Fedora users, not Debian.

JUNE 12, 2020

**Geeker**

There's a UI for SELinux?

JUNE 23, 2020

**Stephan Schutter**

this is nonsense. Why not do the more straightforward solution and install a plug-in 2 Nautilus using the package manager UI? Then right click on the folder you want to share and select share.

JUNE 13, 2020

**Tomasz Gąsior**

Because that plugin does not exist in Fedora repos. That's because it's deprecated and abandoned.

JUNE 18, 2020

**Marcus**

This guide is awsome! Thank you! 🙂 I even learnd alot about selinux.

JUNE 13, 2020

**Ixo**

With all due respect, I would feel hesitant to call this 'simple'. Sometimes I really miss old good approach 'everything is a file'.

JUNE 13, 2020

**Bill Gates**

Nobody is going to switch from Windows to Fedora, that's for sure. SELinux sux!

JUNE 13, 2020

**Gregory Bartholomew**

Hmm, Bill Gates' comment above sounds a bit defensive … I wonder if this Linux thing is starting to worry him … :-),

I have no idea if that is actually him BTW. I kind of doubt it, but I wouldn't reply to it since the email address is set to an @microsoft.com domain.

JUNE 13, 2020

**Jaap**

on $ sudo setfacl -m group:public_readonly:r-x /srv/public

error: setfacl: Option -m: Invalid argument near character 7
cannot find a solution on that problem, do you know?

JUNE 14, 2020

**Gregory Bartholomew**

Hi Jaap:

Did you (successfully) run the previous command:

```
# groupadd public_readonly
```

I think character 7 in this case is the start of the group name ("group:" being the 6 characters before it).

JUNE 14, 2020

**Jaap**

i did all of the editing over again and now it is working fine. Just finished editing. Your answer is faster than I expected. Thanks.

JUNE 14, 2020

**Jeroen**

Thanks for the detailed and methodical walk through including verification checks and explanation around the interaction with filesystem security context.

JUNE 14, 2020

**Marcus**

Hello! I had some problem with the files I created witin the publicfolder did get the root group as default.

I fixed it with changing the default group of the folder to “public_readwrite”.

chgrp public_readwrite /srv/public

JUNE 14, 2020

**Marcus**

I also had to add this to the smb.conf to get everyting to work.

Im using Fedora 31 so I dont know if that is the problem.

[global]

vfs objects = acl_xattr
map acl inherit = yes

JUNE 14, 2020

**Nathan Smith**

How does this compare with simply turning on File sharing under Settings > Sharing? Seems to work just fine with Selinux enabled and hasn’t been a problem for any of my use cases.

I’m uncertain when I would go through all the extra effort outlined in sharing /srv/public.

JUNE 14, 2020

**Tomasz Gąsior**

This does not use SMB. This GNOME feature used webdav implemented through apache user instance. It won’t be visible in “network neighborhood” in Windows systems. It will be visible only in other GNOMEs (and maybe KDEs) and in Mac OSes.

JUNE 18, 2020

**Gregory Bartholomew**

For what it’s worth, it looks like webdav (aka “web folders”) is supported by Windows. It apparently doesn’t work as transparently as SMB shares, but it looks like it *should* work with a bit of effort:

https://answers.microsoft.com/en-us/windows/forum/windows_7-networking/how-to-open-web-folder-in-windows-7/0552a422-9e7d-41e6-8971-200d2bd14621

JUNE 18, 2020

**Tomasz Gąsior**

Yep. That’s what I wanted to say. Windows supports webdav but does not support automatic service discovery for webdav — you can’t just find GNOME shared directory in Windows “network” window. I wrote some script which prints path which can be used in Windows:

```
#!/bin/bash

port=$(pgrep -a -u $USER httpd | tail -n 1 | grep -Eo '[0-9
```

```
if [ -z "$port" ]; then
    zenity --error --text "gnome-user-share-webdav is not
    exit
fi

ip_addr=$(nmcli | grep -m1 inet4 | awk '{print $2}')
ip_addr=${ip_addr%???}

message=$(cat &lt;&lt;-END
&lt;span font="Cantarell 22"&gt;
URLs of GNOME user share for user "$USER"
```

JULY 11, 2020

**Yves**

Hi,
congrats to the author ! When someone is searching howto to share a folder with samba, the first results concern heavy configurations with connection with an AD server…
This article provides a clear and simple configuration path of sharing stuff sufficient most of people and it also addresses selinux and firewall's eventual problems.
So in short : I very much like this article and I will use it as a reference !

JUNE 14, 2020

**Steve**

@Bill Gates I did, and will never look back

JUNE 15, 2020

**Jaap**

Thanks for the very clear and thorough explanation. Very good lesson for me.
Question.
I did the smb install on one computer. Now I want to exchange files from computer A (with SMB) to computer B (should I install SMB in computer B)./ Both are in one local network.
Do I have to add something more except install SMB?

JUNE 15, 2020

**Jaap**

Thanks, I did find a solution.

JUNE 15, 2020

**froqstar**

Me:
– Reads "Simple Local File-Sharing" headline.
– Looks at 100+ lines of arcane command line/config file incantations.
– Looks at first line of Fedora mission statement: "We envision a world where free and open source software is accessible and usable."
– Cries.

Don’t get me wrong, thanks a lot for the detailed write-up. I just don’t think the “simple” word in the headline is appropriate.
More like “An in-depth tutorial on how to properly configure a SMB share”.

JUNE 23, 2020

**da2ce7**

@froqstar, Believe-it-or-not, this is the ‘simple’ guide. Samba and file sharing can lots more complicated that this!

I agree, Fedora, and the GUI really should do much better than what is required in this guide.

JUNE 25, 2020

**vs**

I am have reach the part to add user and give them readwrite access but get the following error:
[vs@localhost ~]$ sudo gpasswd –add vS public_readwrite
Adding user varoonS to group public_readwrite
[vs@localhost ~]$ sudo smbclient –user=vS //localhost/public
Enter WORKGROUP\vS’s password:
Try “help” to get a list of possible commands.
smb: > mkdir test
NT_STATUS_ACCESS_DENIED making remote directory \test
smb: > ls
. D 0 Tue Jun 23 22:01:19 2020
.. D 0 Tue Jun 23 22:01:19 2020

35038800 blocks of size 1024. 26926868 blocks

NT_STATUS_ACCESS_DENIED making remote directory \test
smb: > ls
. D 0 Tue Jun 23 22:01:19 2020
.. D 0 Tue Jun 23 22:01:19 2020

35038800 blocks of size 1024. 26926868 blocks

JUNE 26, 2020

***Comments are Closed***

Search Fedora Magazine

Fedora Linux 38 is available now. Read the release announcement for all the details.

### SUBSCRIBE TO FEDORA MAGAZINE VIA EMAIL

Email Address

Subscribe

Join 9,571 other subscribers

### CONTRIBUTE TO THE MAGAZINE

Fedora Magazine is looking for contributors!

Propose a new article

Become a writer

Become an editor