Working with SAMBA

By: Fabian Cooper & Aesia Cohen

# **Introduction:**

SAMBA is an open-source software suite capable of file sharing and printing services. Utilizing SMB and CIFS clients, SAMBA is used on various systems, however it is most often used for Linux-to-Windows and vice versa sharing.

**NOTE:** The following guide is for SAMBA v.4.4.4 and connecting between two Linux VirtualBox Virtual Machines (VMs) with the CentOS7 system and Gnome GUI installed. All names used in the examples are arbitrary.

**Contents**

[Introduction: 1](#_Toc481790638)

[Prerequisites: 2](#_Toc481790639)

[Creating the Share Folder: 3](#_Toc481790640)

[Using an Existing Folder: 3](#_Toc481790641)

[Creating a New Folder: 4](#_Toc481790642)

[SAMBA Configuration: 4](#_Toc481790643)

[SAMBA Share Mounting Setup: 6](#_Toc481790644)

[Starting the SAMBA Server: 7](#_Toc481790645)

[Connecting to the SAMBA Server: 8](#_Toc481790646)

[Connecting Through the GUI: 8](#_Toc481790647)

[Connecting Through the Terminal: 8](#_Toc481790648)

# **Prerequisites:**

The Bridged Adapter attachment uses the default network device that allocates IP addresses on the physical network to change the inet address of the VM. As CentOS7 gives a default inet address for VMs, the Bridged Adapter attachment makes it so that two VMs will have different inet addresses.

> **Settings 🡪 Network 🡪 Attached To: Bridged Adapter**

>**Start the VM that will host the server**

It’s very important to have the VM that will serve as the host for the server to be up-to-date. Especially the smbclient packages.

> **sudo yum update**

As SAMBA and some of the other commands that will be used later require root access, it is appropriate to switch to the root user account.

**> su root or sudo -s**

Now to install SAMBA and nmap. The Network Mapper (nmap) performs a security scan that discovers the hosts and services of the current working network. In a sense, it shows a “map” of the working network. nmap will be used in this sense to get the NetBIOS and Workgroup name.

> **yum install -y samba**

> **yum install -y nmap**

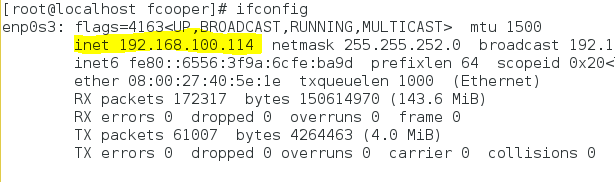
The Network Basic Input/Output System (NetBIOS) is a program that allows for communication on a LAN (Local Area Network) connection. It provides a message location and the name of the destination in the form of a NCB (Network Control Block). NetBIOS is needed when creating the SAMBA server as SAMBA relies on namespace messages.

Some of the commands in Linux are secured by File Locking files. File Locking is a protocol that only allows one defined process to access the attributed file. Ex: SAMBA uses SMB which has a smbpasswd command, like that of the passwd command. With the passwd.lock file existing, smbpasswd will return an error as it doesn’t match the passwd command. Therefore, the following files must be removed for many of the SAMBA commands to work.

> **rm -f** **/etc/passwd.lock /etc/group.lock /etc/gshadow.lock /etc/shadow.lock**

ifconfig is a Linux utility that shows the network interface configuration for the system. The system inet address can be found at the top of the configuration file. If the inet address is 10.0.2.15, then shut down the VM, reapply the Bridged Adapter setting, and restart the VM. Otherwise, make note of the inet address for later implementation.

**> ifconfig**



Then make note of the NetBIOS and Workgroup names from the nmap display.

**>nmap <inet address>**

At this point it is necessary to discern which way is best to create the share folder. There is a fast and unsecure way, and then there is a longer and secure way. Both will be outlined and explained below.

## 

## **Creating the Share Folder:**

Using an existing folder is faster to implement, however it requires that the predefined permissions be changed to all access so that it can be mounted and manipulated by the client. While this is an easy fix, it should not be used for everyday use.

### **Using an Existing Folder:**

**>>chmod 777 /home/fcooper/Public**

**>> touch file1 file2 file3 /home/fcooper/Public**

**>>Use path to the existing folder in the configuration file**

Creating a new folder is more secure, but it requires more time and steps. A group must be created to own the folder and the settings must be all access for the group and user. However, it is the best way to go when using SAMBA for everyday use.

### **Creating a New Folder:**

**>> mkdir / home/fcooper/sharefolder**

**>> chmod 770 /home/fcooper/sharefolder**

**>> touch file1 file2 file3 /home/fcooper/sharefolder**

**>>groupadd smbusers**

**>> chown :smbusers /home/fcooper/sharefolder**

# **SAMBA Configuration:**

The first step to creating the server is to create a SAMBA user. Depending on the security level of the share, the SAMBA user credentials can be crucial.

> **smbpasswd –a fcooper**

smbpasswd -a automatically adds the username and password created to the SAMBA database. However, it is more effective to create a file in the */etc/samba/* that will house this data.Should the share folder have been made from scratch, also perform:

**>usermod -G smbusers fcooper**

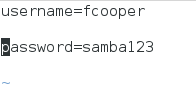
If using existing folder for the share folder, skip the previous command.

**>touch /etc/samba/user.cred**

The username and password of the previously made samba user must be added into the user.cred file.

**> vim /etc/samba/user.cred**

**> Add the previously defined user credentials in this format.**



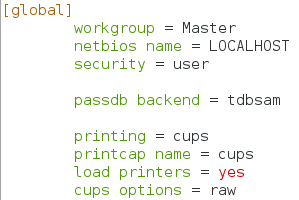
Now it’s time to edit the SMB configuration file such that the SAMBA server will function as a simple Stand-Alone Server.

> **vim /etc/samba/smb.conf**

The *[global]* section of the SMB configuration file sets the server-wide settings and the default share definitions. The *security* parameter is set to user so that the server is only available to certain users with the correct password.

**NOTE:** The parameter *netbios name = LOCALHOST* was added below the *workgroup* parameter. *LOCALHOST* and *Master* should be replaced with the appropriate Workgroup and NetBIOS of the current system.

**> Change the global directory to the following:**

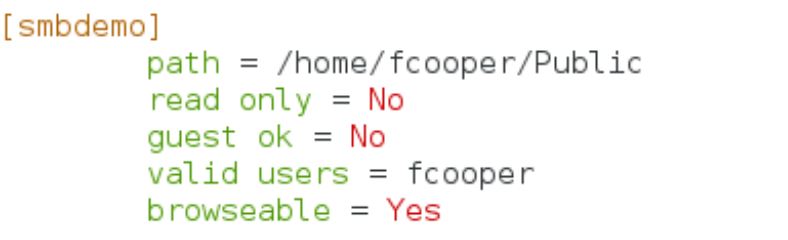


The *[homes]* section is used only when there are unresolved share names in the *[global]* section. User and password creation can be defined here as needed. The *[printers]* section is checked if both previous sections still return unresolved share names. This section reads all capabilities of the printer/s as defined in the */etc/printcap* file.

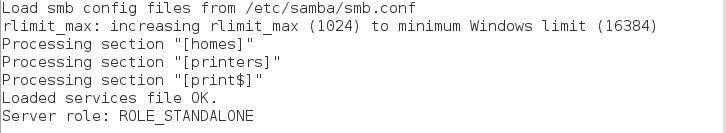
For this guide, there will be no need to change the *[home]* and *[printer]* section. If preferred, they can be commented out. Below these sections will go the settings and definitions for the share folder.

**NOTE:** The name of the section will be the name the share folder will be recognized as in further steps and when connection to the Server.

**> Add the following section to the bottom of the smb.conf file. Where *path* is defined to the path of your share folder. The parameter *create mask* defines the permissions of any files created in the folder; this is optional. *valid users* should be defined by the name of the previously created SAMBA user.**



After the configuration is finished, save the changes and use *testparm* to make sure all parameters are defined correctly.



Once the configuration is complete, it’s time to mount the share folder.

# **SAMBA Share Mounting Setup:**

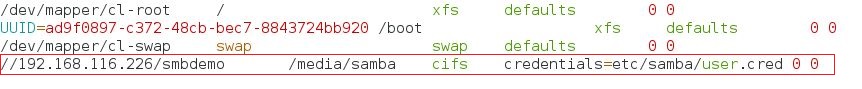
A core understanding of Linux is that everything is a file. This allows for a much wider range of manipulation. That is why the share folder is mounted using CIFS. Once the client connects to the share folder, they are essentially connecting to a new file system.

To start the mounting process, the share folder and its credentials need to be added to the *fstab* file. The *fstab* file defines all the file systems on the VM.

**Note:** The spaces in between the options are tabbed.

> **vim /etc/fstab**

***>* Insert the following directly below the last file system.**



At this point, the permissions can be changed for the user.cred file

> **chmod 600 /etc/samba/user.cred**

Return to the home directory and create the /media/samba directories. This is where the share folder will be mounted. Change permissions to give full access.

**> mkdir /media/samba**

**> chmod 777 /media/samba**

Now to begin the server and mount the folder.

# **Starting the SAMBA Server:**

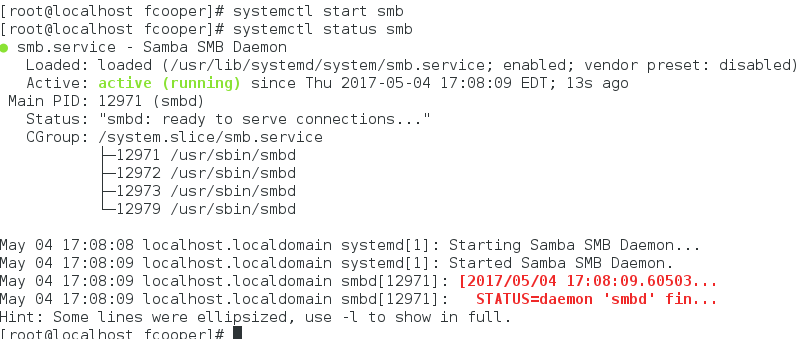
Enable the server first so that when the system reboots, it will save all the changes done for Samba (user and group creation and configuration file) and start the SMB service automatically.

**>systemctl enable smb**

Then start the server and check that is running. The output should look similar to that of the image below

**>systemctl start smb**

**>systemctl status smb**



Next the share needs to be mounted.

**Note:** The share will not mount unless SMB is running. The initial mount command will ask for the credentials of the user. Provide the correct password and the mount will be successful. The second mount command makes the mount automated.

**>** **mount –t cifs //192.168.116.226/smbdemo -o username=fcooper /media/samba**

> **mount -a**

# **Connecting to the SAMBA Server:**

There are two ways to connect to the SAMBA server—through the GUI or through the terminal.

## **Connecting Through the GUI:**

**>Applications 🡪 Files 🡪 Connect to Server**

A window will pop up and the server address should be inputted**.**

**>Server Address: smb:// 192.168.116.226/smbdemo**

The next window should be filled out as follows:

**>Username = fcooper**

**>Domain = SAMBA**

**>Password = samba123**

The only way to manipulate these files is through a text editor. This would be fine for quick appendages to the files, but it is best to connect through the terminal as it will offer more options for editing files and directories.

## **Connecting Through the Terminal:**

smbclient is a ftp-like interface that offers various ways to manipulate a SAMBA share. This guide provides definitions for the most basic commands. Use the following command in your client VM. A password authentication will pop up and then you will be in the server and able to access the shared folder.

**>smbclient //192.168.116.226/smbdemo -U fcooper**

Once inside the server, use the help command to view a full list of the commands available.



The basic commands for the smbclient are as follows:

**NOTE**: smbclient allows for scripting, printing, and other basic Linux capabilities.

