Install and Configure Samba - CentOS 7

What is Samba and why should I use it?

Samba is a service that allows Linux machines to access and share files, folders and printers with Windows machines. Specifically, it is an open source implementation of SMB/CIFS protocol. If you have a mixed networking environment with Windows and Linux machines, Samba is an essential tool to get all of your devices playing nicely.

Using Samba, we can also setup a domain controller on a Linux server, and integrate the Windows clients to the Domain controller.

This tutorial will describe you how to setup a basic Samba server in a CentOS 7 system. These steps will work on RHEL 7 and Scientific Linux 7 (and other Red Hat-based Linux distributions.)

Installation

The easiest part of this how to will be the actual installation. To install samba run the following command:

```
yum install samba samba-client samba-common
```

Configuration files are typically located in the `/etc` directory, and Samba is no exception - its main configuration file, `smb.conf`, is located in `/etc/samba`. It is generally considered to be a best practice to back up the original configuration file before modifying it, so it can be reverted to the original in case of a mistake or error. Let's back it up now.

```
mv /etc/samba/smb.conf /etc/samba/smb.conf.bak
```

OR

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

Finding your Windows Domain

At this point, we can proceed no further without a key piece of information: the Windows domain or workgroup we wish to place our machine on. To find this out, open a Command Prompt (`cmd.exe`) *on a Windows PC* and enter the following:

```
net config workstation
```

```
Software version        Windows 8.1 Pro
Workstation domain      WORKGROUP
```

Editing the configuration file

Open up the Samba configuration file in your favourite editor and fill it in with the appropriate information

```
EDITOR /etc/samba/smb.conf
```

```
[global]
workgroup = WORKGROUP
server string = Samba Server %v
netbios name = YOURHOSTNAME
security = user
map to guest = bad user
dns proxy = no
```

These global settings are required. Beneath them we can list our shares. To start, we will create a share accessible to all users – even unauthenticated ones. This share is akin to anonymous uploading with FTP.

```
[all]
path = /samba/all
browseable = yes
writeable = yes
guest ok = yes
read only = no
```

Adding shares with user access control is very similar, with an additional requirement that we need to create a group that is allowed to access the share, and add our user to it. In this example we will name our secured share 'restricted' and create a group with the same name on our host. Note that the group name does not have to match the share name, it is just convenient to do so in many cases.

```
[restricted]
path = /samba/restricted
valid users = @restricted
guest ok = no
writable = yes
browseable = yes
```

User permissions

Now we must create the directories we specified in our configuration file, namely */samba/all* and */samba/restricted*. For the *all* share, we want it to be globally accessible by unauthenticated users, and these users should be able to write. For the *restricted* share, we must create the `restricted` group, add a user to that group, and add the user to the Samba database.

```
mkdir  -p  /samba/all
mkdir  -p  /samba/restricted
groupadd  restricted
usermod  -a  -G  restricted  username
cd  /samba
chmod  -R  1777  all
chmod  -R  0775  restricted
chown  -R  nobody:nobody  all
chown  -R  username:restricted  restricted
smbpasswd  -a  username
```

SELinux

```
chcon  -t  samba_share_t  all
chcon  -t  samba_share_t  restricted
```

Start the services

Firewall

```
firewall-cmd  --permanent  --zone=public  --add-service=samba
firewall-cmd  --reload
```

Enable & Start

```
systemctl  enable  smb.service
systemctl  enable  nmb.service
systemctl  restart  smb.service
systemctl  restart  nmb.service
```

Scenario

In this tutorial, I will be using two systems as described below.

Samba server:
Operating system: CentOS 7
Hostname : server.unixmen.local
IP Address : 192.168.1.101/24

Samba client:
Operating system : Windows 7 Professional
Hostname : client
IP Address : 192.168.1.102/24

Check for existing `samba` package if any using the following commands.

```
rpm -qa | grep samba
yum list installed | grep samba
```

If Samba is installed, remove it using the below command:

```
yum remove samba*
```

Now, install Samba using the following command.

```
yum install samba* -y
```

1. Configure a fully-accessible anonymous share

Now, let us create a fully-accessible anonymous share for the users. Any one can read/write in this share.

Create a directory called `/samba/anonymous_share` and set full permission. You can name this share as per your liking.

```
mkdir -p /samba/anonymous_share
chmod -R 1777 /samba/anonymous_share
```

Edit Samba configuration file;

```
vi /etc/samba/smb.conf
```

Find the following directives, and make the changes as shown below.

[...]

```
## Add the following lines under [global] section ##
unix charset = UTF-8
dos charset = CP932

## Change the to windows default workgroup ##
workgroup = WORKGROUP

## Uncomment and set the IP Range ##
hosts allow = 127. 192.168.1.

## Uncomment ##
max protocol = SMB2

## Uncomment, and change the value of 'Security' to 'user' ##
security = user

## Add the following line ##
map to guest = Bad User

## Add the following lines at the bottom ##
[Anonymous share]
path = /samba/anonymous_share
writable = yes
browsable = yes
guest ok = yes
guest only = yes
create mode = 0777
directory mode = 0777
```

Start Samba services, and enable them to start automatically on every reboot.

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

Test the Samba server configuration

We can test the Samba server configuration syntax errors using the command ‘`testparm`’.

```
testparm
```

Sample Output:

```
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Processing section "[homes]"
Processing section "[printers]"
Processing section "[Anonymous share]"
Loaded services file OK.
WARNING: You have some share names that are longer than 12 characters.
These may not be accessible to some older clients.
(Eg. Windows9x, WindowsMe, and smbclient prior to Samba 3.0.)
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

[global]
    dos charset = CP932
    netbios name = UNIXMEN SAMBA SERVER
    server string = Samba Server Version %v
    map to guest = Bad User
    log file = /var/log/samba/log.%m
    max log size = 50
    server max protocol = SMB2
    idmap config * : backend = tdb
    hosts allow = 127., 192.168.1.
    cups options = raw

[homes]
    comment = Home Directories
    read only = No
    browseable = No

[printers]
    comment = All Printers
    path = /var/spool/samba
    printable = Yes
    print ok = Yes
    browseable = No

[Anonymous share]
    path = /samba/anonymous_share
    read only = No
    create mask = 0777
    directory mask = 0777
    guest only = Yes
    guest ok = Yes
```

SELinux Configuration

Turn the `samba_enable_home_dirs` Boolean on if you want to share home directories via Samba.

```
setsebool -P samba_enable_home_dirs on
```

If you create a new directory, such as a new top-level directory, label it with `samba_share_t` so that SELinux allows Samba to read and write to it. Do not label system directories, such as `/etc/` and `/home/`, with `samba_share_t`, as such directories should already have an SELinux label.

In our case, we already have created an `anonymous` directory. So let us label it as shown below.

```
chcon -t samba_share_t /samba/anonymous_share/
```

If you don't want to mess with SELinux, just disable it as shown below, and continue.

To disable SELinux, edit file `/etc/sysconfig/selinux`,

```
vi /etc/sysconfig/selinux
```

Set SELinux value to **`disabled`**.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
# targeted - Targeted processes are protected,
# mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Restart the server for the changes to take effect.

Test Samba Shares

Now, go to any Windows client system. In this example, I am using Windows 7.

Click Start -> Run. Enter the Samba server IP in UNC format as shown below.

```
\\192.168.0.101
```

Now, you'll be able to access the fully-accessible Samba share.

You can create, modify or delete the files/folders inside the share.

On Linux, check the newly-created files or folders are present in the Samba server:

```
ls -l /samba/anonymous_share/
```

2. Create security enabled share in samba server

What we have seen so far is creating a fully-accessibly Samba share. Anyone can access that share folder, and can create, delete files/folders in that share.

Now, let us create a password-protected Samba share so that the users should enter a valid username and password to access the share folder.

Create a user called “`smbuser`” and a group called “`smbgroup`”.

```
useradd -s /sbin/nologin smbuser
groupadd smbgroup
```

Assign the user `smbuser` to `smbgroup`, and set the *Samba* password for that user.

```
usermod -a -G smbgroup smbuser
smbpasswd -a  smbuser
```

Create a new share called “`/samba/secure_share`” and set the permissions to that share.

```
mkdir /samba/secure_share
chmod -R 0755 /samba/secure_share
chown -R smbuser:smbgroup /samba/secure_share
```

Edit the Samba configuration file:

```
vi /etc/samba/smb.conf
```

Add the following lines at the bottom of the Samba configuration file:

```
[secure_share]
path = /samba/secure_share
writable = yes
browsable = yes
guest ok = no
valid users = @smbgroup
```

Test the Samba configuration for any errors.

```
testparm
```

Sample output:

```
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Processing section "[homes]"
Processing section "[printers]"
Processing section "[Anonymous share]"
Processing section "[secure_share]"
Loaded services file OK.
WARNING: You have some share names that are longer than 12 characters.
These may not be accessible to some older clients.
(Eg. Windows9x, WindowsMe, and smbclient prior to Samba 3.0.)
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

[global]
    dos charset = CP932
    netbios name = UNIXMEN SAMBA SERVER
    server string = Samba Server Version %v
    map to guest = Bad User
    log file = /var/log/samba/log.%m
    max log size = 50
    server max protocol = SMB2
    idmap config * : backend = tdb
    hosts allow = 127., 192.168.1.
    cups options = raw

[homes]
    comment = Home Directories
    read only = No
    browseable = No

[printers]
    comment = All Printers
    path = /var/spool/samba
    printable = Yes
    print ok = Yes
    browseable = No

[Anonymous share]
    path = /samba/anonymous_share
    read only = No
    create mask = 0777
    directory mask = 0777
    guest only = Yes
    guest ok = Yes

[secure_share]
    path = /samba/secure_share
    valid users = @smbgroup
    read only = No
```

Label the `/samba/secure_share/` with `samba_share_t` so that SELinux allows Samba to read and write to it.

```
chcon -t samba_share_t /samba/secure_share/
```

Fix permissions on the secured folder that allows all members of the `smbgroup` to read and write, and restricts access to group members only:

```
chown  -R  smbuser:smbgrp  /samba/secure_share/
chmod  -R  o-rwx  /samba/secure_share/
chmod  -R  g+w  /samba/secure_share/
chmod  -R  g+s  /samba/secure_share/
```

Restart Samba services.

```
systemctl restart smb
systemctl restart nmb
```

Test Samba shares

Now, go to the Windows client.

To make the Linux machine reachable in Windows by name, add the entry of your server IP address to the `hosts` file:

```
notepad C:\Windows\System32\drivers\etc\hosts
```

This will also speed up connection times to your server.

Double click to open the secured share. You’ll be asked to enter the user name and password to access the share.

That’s it. Now, you can access the secured samba share folder.

Bibliography

https://my.esecuredata.com/index.php?/knowledgebase/article/56/install-and-configure-samba-centos-7

https://www.unixmen.com/install-configure-samba-server-centos-7/

https://www.tecmint.com/install-samba4-on-centos-7-for-file-sharing-on-windows/

https://www.howtoforge.com/samba-server-installation-and-configuration-on-centos-7

https://lintut.com/easy-samba-installation-on-rhel-centos-7/

All content retrieved on September 26, 2017.