





Set Up Samba Server on CentOS 8/RHEL 8 for File Sharing

 Last Updated: March 31st, 2020  Xiao Guoan (Admin)  4 Comments  CentOS, Redhat

In this tutorial, we're going to learn how to install and configure a samba server on CentOS 8/RHEL8 to share files on the local network. Samba is a free and open-source SMB/CIFS protocol implementation for Unix and Linux that allows for file and print sharing between Linux, Windows and macOS machines in a local area network.

Samba is usually installed and run on Linux. It comprises several programs that serve different but related purposes, the most important two of which are:

- **smbd**: provides SMB/CIFS service (file sharing and printing), can also act as a Windows domain controller.
- **nmdb**: This daemon provides NetBIOS name service, listens for name-server requests. It also allows the Samba server to be found by other computers on the network.

How to Install Samba Server on CentOS 8/RHEL8

Samba is included in most Linux distributions. To install Samba on CentOS 8/RHEL8, run the following command in terminal.

```
sudo dnf install samba
```



To check your Samba version, run

```
smbd --version
```

Sample output:

```
Version 4.10.4
```

Then issue the following command to start the `smbd` and `nmbd` service.

```
sudo systemctl start smb nmb
```

Enable auto-start at boot time.

```
sudo systemctl enable smb nmb
```

To check if Samba service is running, issue the following commands.

```
systemctl status smb nmb
```

Once started, the `smb` daemon will be listening on TCP port 139 and 445. The `nmbd` will be listening on UDP port 137 and 138.

- TCP 139: used for file and printer sharing and other operations.
- TCP 445: the NetBIOS-less CIFS port.
- UDP 137: used for NetBIOS network browsing.
- UDP 138: used for NetBIOS name service.

Run the following command to open the above ports in the firewall.

```
sudo firewall-cmd --permanent --add-service=samba
```



Reload firwall daemon for the change the take effect.

```
sudo systemctl reload firewalld
```

Create a Private Samba Share

In this section, we will see how to create a private Samba share that requires the client to enter username and password in order to gain access. The main Samba configuration file is located at: `/etc/samba/smb.conf`. You can edit it in terminal with a command line text editor like `nano`.

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

By default, there are 4 sections:

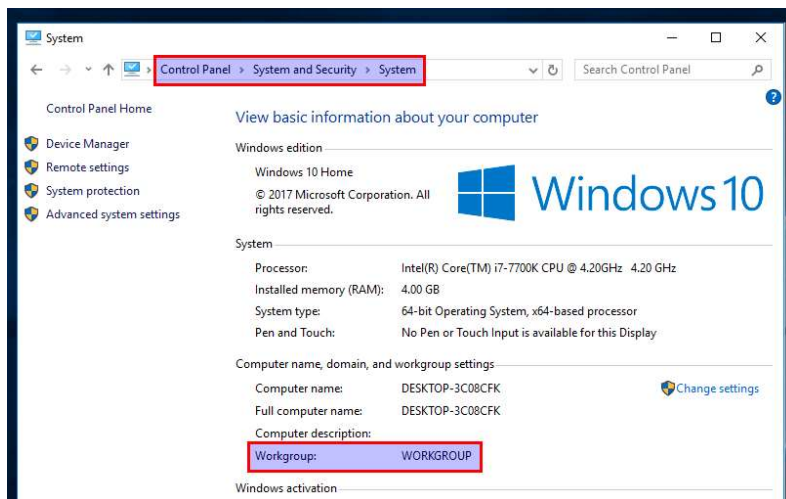
- `global`
- `homes`
- `printers`
- `print$`

In the `[global]` section, the `workgroup` is set to `SAMBA` by default.

```
workgroup = SAMBA
```

Make sure the value of `workgroup` is the same with the `workgroup` settings of Windows computers. You can find the setting on your Windows computer by going to `Control Panel > System and Security > System`.





In this article, I will use the Windows default workgroup name.

```
workgroup = WORKGROUP
```

It's recommended to use the `hosts allow` parameter to create an IP address whitelist to prevent unauthorized access. Add the following line in the `[global]` section, which will allow the localhost and clients in the `192.168.0.0/24` network to access Samba services. If you are using a different network range, replace `192.168.0` with your own.

```
hosts allow = 127. 192.168.0
```

Hint: You can also use the CIDR notation for network range like `hosts allow = 127.0.0.1 192.168.0.0/24`

Then scroll down to the bottom of the file. (In nano text editor, you can achieve that by pressing `CTRL+W` then `CTRL+V`.) Add a new section like below.

```
[Private]
    comment = needs username and
password to access
    path = /srv/samba/private/
    browseable = yes
    guest ok = no
```

```
writable = yes
valid users = @samba
```

Where:

- **Private** is the folder name that will be displayed on the Windows computer. You can use whatever name you like.
- The comment is a description for the shared folder.
- The path parameter specifies the path to the shared folder. I use `/srv/samba/private/` as an example.
- **browseable = yes**: Allow other computers in the network to see the Samba server and Samba share. If set to no, users have to know the name of the Samba server and then manually enter a path in the file manager to access the shared folder.
- **guest ok = no**: Disable guest access. In other words, you need to enter username and password on the client computer to access the shared folder.
- **writable = yes**: Grants both read and write permission to clients.
- **valid users = @samba**: Only users in the samba group are allowed to access this Samba share.

Save and close the file. (To save the file in nano text editor, press `Ctrl+O`, then press `Enter` to confirm the file name to write. To close the file, press `Ctrl+X`.) Now we need to create a Samba user. First, we need to create a standard Linux user account with the following command. Replace **username** with your desired username.

```
sudo adduser username
```

You will be prompted to set an Unix password. After that, you also need to set a separate Samba password for the new user with the following command:

```
sudo smbpasswd -a username
```



Create the samba group.

```
sudo groupadd samba
```

And add this user to the samba group.

```
sudo gpasswd -a username samba
```

Create the private share folder.

```
sudo mkdir -p /srv/samba/private/
```

The samba group needs to have read, write and execute permission on the shared folder. You can grant these permissions by executing the following command.

```
sudo setfacl -R -m "g:samba:rwx" /srv/samba/private/
```

We also need to label this directory with samba_share_t so that SELinux allows Samba to read and write to it.

```
sudo chcon -t samba_share_t /srv/samba/private/ -R
```

Next run the following command to check if there's syntactic errors.

```
testparm
```

Samba configuration files are automatically reloaded every minute, if they change. You can manually restart smbd and nmbd daemon for the changes to take effect immediately.

```
sudo systemctl restart smb nmb
```



How to Create a Samba Public Share

Without Authentication

To create a public share without requiring username and password, the following conditions must be met.

- Set `security = user` in the global section of Samba configuration file.
- Set `map to guest = bad user` in the global section of Samba configuration file. This will cause `smbd` to use a guest account to authenticate clients who don't have registered account on the Samba server. Since it's a guest account, Samba clients don't need to enter password.
- Set `guest ok = yes` in the share definition to allow guest access.
- Grant read, write and execute permission of the public folder to the `nobody` account, which is the default guest account.

Open and edit the Samba configuration file.

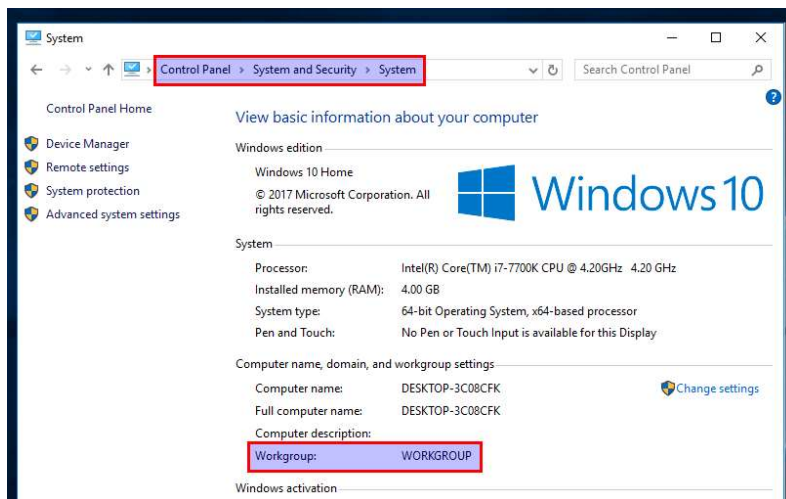
```
sudo nano /etc/samba/smb.conf
```

In the `[global]` section, make sure the value of `workgroup` is the same with the workgroup settings of Windows computers.

```
workgroup = WORKGROUP
```

You can find the setting on your Windows computer by going to `Control Panel > System and Security > System`.





Add the following line in the [global] section.

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

Then scroll down to the bottom of the file and paste the following lines.

```
[public]
    comment = public share, no need to enter username and password
    path = /srv/samba/public/
    browseable = yes
    writable = yes
    guest ok = yes
```

Save and close the file. Next, create the /srv/public/ folder.

```
sudo mkdir -p /srv/samba/public
```

You must change the permission of this folder to 777, if you want to allow write operation on the public share.

```
sudo chmod 777 /srv/samba/public/ -R
```

We also need to label this directory with samba_share_t so that SELinux allows Samba to read and write to it.




```
sudo chcon -t samba_share_t /srv/samba/public/ -R
```

Restart smbd and nmbd.

```
sudo systemctl restart smb nmb
```

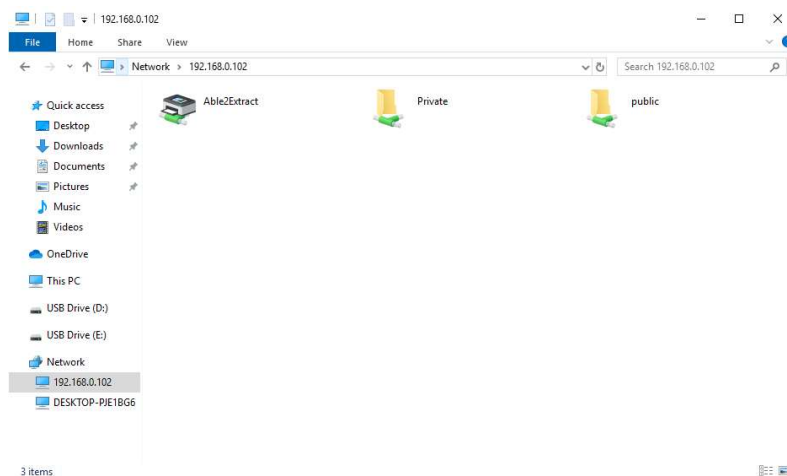
Accessing Samba Shared Folder From Windows

On a Windows computer that is in the same network, open File Explorer and click Network on the left pane. If you see the following message, then you need to click on the message and turn on network discovery and file sharing.

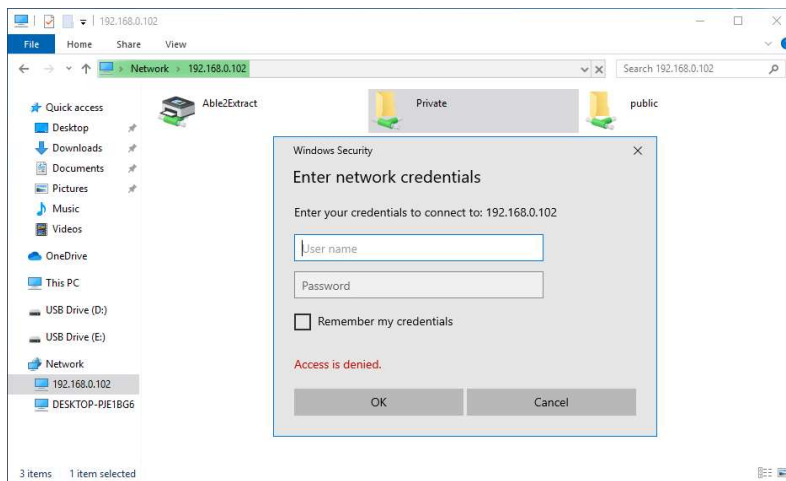
File sharing is turned off. Some network computers and devices might not be visible.

Next, enter \\ followed by the IP address of Samba server in the address bar of File Explorer, like this:

\\192.168.0.102. You will see a list of shared resources on the Samba server.



Then double-click the shared folder. To access private share, you need to enter the samba username and password. You don't need to do so to access public share.



Once connected, you can read, write and delete files in the Samba shared folder.

Connecting Error

If you get the following error:

```
You do not have permission to access
\\hostname\share-name. Contact your n
etwork administrator to request acces
s.
```

You can try connecting to the Samba share from the command prompt. Open up a command prompt, then run the following command to close current Samba session.

```
net use \\samba-server-ip\share-name
/delete
```

Next, connect to the Samaba share with the following command:

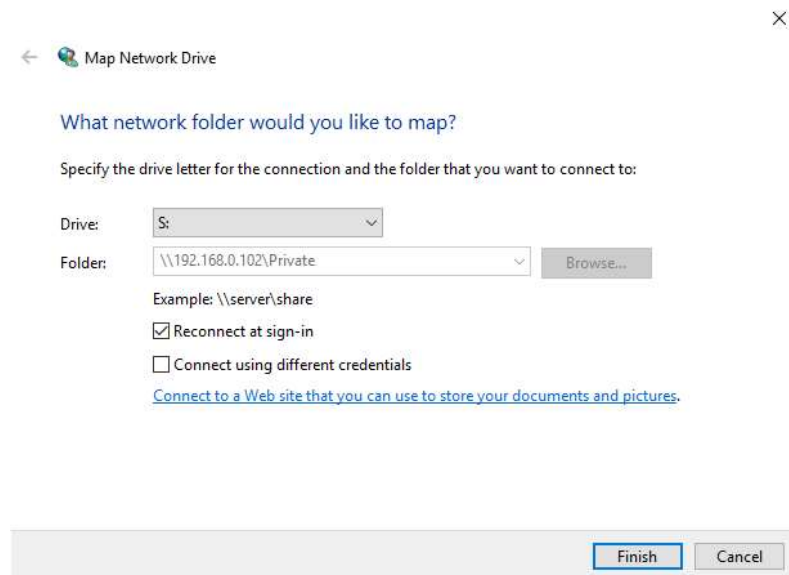
```
net use \\samba-server-ip\share-name
/user:samba-username password
```

Once the above command completed successfully, go to the Network tab in File Explorer and now you should be able to access the Samba share.



Drive Mapping

One feature of the Windows operating system is the capability to map a drive letter (such as S:) to a remote directory. To map the drive letter S: to the Samba share, right-click the Samba shared folder and select **Map network drive**. Then choose a drive letter and click Finish.

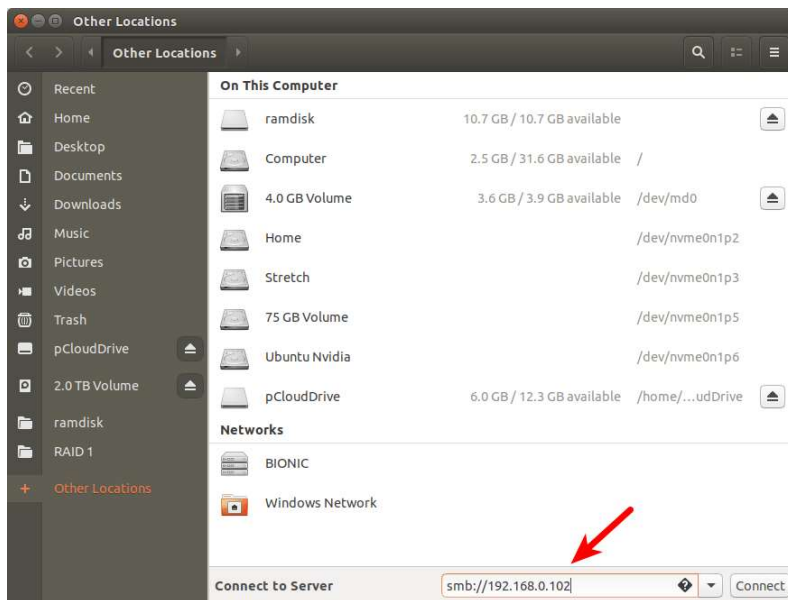


Once the drive mapping is established, applications can access the files in the Samba share through the drive letter S: . And this Samba share will be automatically mounted when you log in to your Windows computer.

Accessing Samba Share Folder in Nautilus File Manager on Linux

If you are using Nautilus file manager, then click **Other Locations** on the left pane. On the bottom you will see an option to **connect to server**. To access your Samba share, type in `smb : //` followed by the IP address of the Samba server and press Enter. For example:

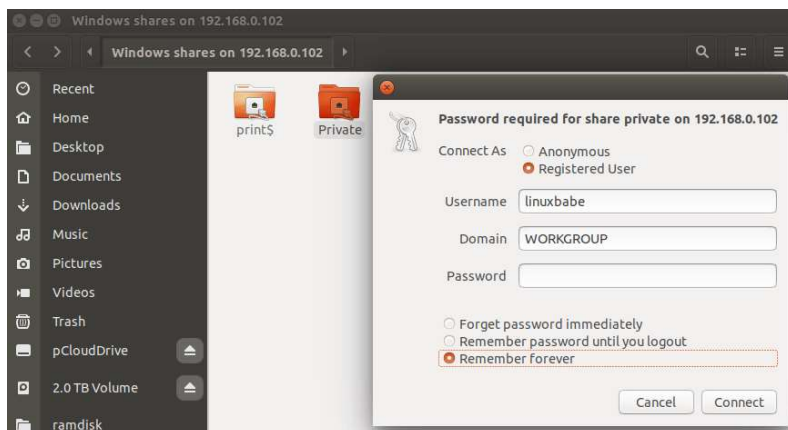
- `smb://192.168.0.102`



You will see a list of shared resources on the Samba server.



If you click the private shared folder, then you will need to enter the Samba username and password. If you click the public shared folder, then choose to connect as Anonymous.



Automatically Mount Samba Share From Command Line on Linux

If you need to automatically mount the Samba share at boot time, you can use command line to mount and then add an



entry in the `/etc/fstab` file. In order to do that, you need to install the `cifs-utils` package.

CentOS/RHEL

```
sudo dnf install cifs-utils
```

Debian/Ubuntu

```
sudo apt install cifs-utils
```

Then create a mount point for the Samba share.

```
sudo mkdir /mnt/samba-private
```

Now you can use the following command to mount a private shared folder.

```
sudo mount -t cifs -o username=your_samba_username //192.168.0.102/private /mnt/samba-private/
```

It will ask you to enter the Samba password. After that, it will be mounted at `/mnt/samba-private/` directory.

To automatically mount the Samba share, edit `/etc/fstab` file.

```
sudo nano /etc/fstab
```

Add the following line in the file.

```
//192.168.0.102/private /mnt/samba-private cifs x-systemd.automount, _netdev, credentials=/etc/samba-credential.conf, uid=1000, gid=1000, x-gvfs-show 0 0
```



Where:

- **//192.168.0.102/private**: the IP address of Samba server and the share name.
- **/mnt/samba-private**: mount point for the Samba share.
- **cifs**: filesystem type
- **x-systemd.automount**: This option tells systemd to create an automount unit for the file system. We use this because it gives us the ability to mount remote filesystem after there's network access.
- **_netdev**: This specifies that the mount requires network.
- **credentials=**: Linux should look for credentials in the `/etc/samba-credential.conf` file.
- **uid=1000,gid=1000**: By default the mounted filesystem would be owned by the root user. We use **uid** and **gid** to change the ownership of the filesystem. Normally you use your own **uid** and **gid**, which are both 1000 by default.
- **x-gvfs-show**: If you are using GNOME desktop environment or its derivatives, you can use this option to show the mounted file system in the file manager.

Save and close the file. Then create the credential file.

```
sudo nano /etc/samba-credential.conf
```

Add the following lines in the file.

```
username=your_samba_username  
password=samba_password  
domain=WORKGROUP
```

Save and close the file. Make sure only the root user can read this file.



```
sudo chmod 600 /etc/samba-credential.  
conf
```

If you restart your Linux computer now, the Samba share will be automatically mounted.

Troubleshooting Tip

If your Samba server is not working as expected, you can check the log files under `/var/log/samba/` directory. You can add the following line in the `[global]` section of `/etc/samba/smb.conf` file to increase the log level, if you want to log more information.

```
log level = 2
```





Wrapping Up



That's it! I hope this tutorial helped you set up Samba server on CentOS 8/RHEL8. As always, if you found this post useful, then [subscribe to our free newsletter](#). And you may also want to read the following article to share printer on the local network.

- [Set Up CUPS Print Server on CentOS 8/RHEL 8 \(Bonjour, IPP, Samba, AirPrint\)](#)

Rate this tutorial

 [Total: 3 Average: 4.7]

 CentOS  CentOS Server  Linux  Red Hat

 Red Hat Server  Samba

