

# Configure Samba Share on Debian

Normally when you launch the system's file manager on Linux/Windows system, you will see network shares advised on your network. These shares are only allowed if remote users are connected to the machine.

**Samba** is a free software that enables one to share files across the network using the SMB(Server Message Block) protocol. This tool was developed by **Andrew Tridgell** in December 1991 and January 1992.

The cool features associated with Samba are:

- It is easy and quick to deploy
- It offers secured data transfer
- Multichannel technology
- Message signing-with digital signing - users who receive the data packets are assured of the origin point authenticity.
- Allows concurrent operations.(simultaneous access to the files)
- It offers good performance under heavy loads.
- Samba supports POSIX extensions for CIFS/SMB
- Supports NetBIOS over TCP/IP (NBT)
- It supports the NT-style printing service (SPOOLSS)

Samba is supported on various platforms such as Windows and Unix operating systems i.e Solaris, Linux, AIX, and BSD variants.

This guide will equip you with the required knowledge on how to configure Samba Share on Debian.

## 1 – Install Samba Packages

We will start off by installing Samba on Debian Linux system. This is easy since it is available in the default Debian repositories.

```
sudo apt update
sudo apt install samba smbclient cifs-utils
```

Dependency tree:

```
...
```

The following NEW packages will be installed:

```
attr cifs-utils ibverbs-providers keyutils libcephfs2 libgfapi0 libgfrpc0
libgfxdr0 libglusterfs0 libibverbs1 librados2 librdmacm1
python3-cffi-backend python3-cryptography python3-dnspython python3-gpg
python3-markdown python3-pygments python3-requests-toolbelt python3-samba
python3-tdb python3-yaml samba samba-common samba-common-bin
samba-dsdb-modules samba-vfs-modules smbclient tdb-tools
```

0 upgraded, 29 newly installed, 0 to remove and 0 not upgraded.

Need to get 24.4 MB of archives.

After this operation, 84.7 MB of additional disk space will be used.

Do you want to continue? [Y/n] y

## 2 – Set the Samba Global settings

The Samba configuration file is located under ***/etc/samba/smb.conf***. In this file, there are several changes we need to make. Although Debian is intelligent enough to provide default configurations, it is also good to verify this.

Open the file using a preferred editor.

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

In the file, make make adjustments as you deem fit, example for workgroup.

```
workgroup = WORKGROUP
```

## 3 – Create Shared Samba Directory

Here, you can share both public and private directories. So we will create the two directories as below.

```
sudo mkdir /public
sudo mkdir /private
```

Now edit the Samba conf and add the two directories.

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

At the end of the file, add the shares and authentication methods to access it.

```
[public]
  comment = Public Folder
  path = /public
  writable = yes
  guest ok = yes
  guest only = yes
  force create mode = 775
  force directory mode = 775
[private]
  comment = Private Folder
  path = /private
  writable = yes
  guest ok = no
  valid users = @smbshare
  force create mode = 770
  force directory mode = 770
  inherit permissions = yes
```

## 4 – Create Samba User and Group

We need the Samba share user group to access the Private share as specified in the conf above. So we will create the group as below.

```
sudo groupadd smbshare
```

Add the necessary permissions for the private share.

```
sudo chgrp -R smbshare /private/
sudo chgrp -R smbshare /public
```

Set the right directory permissions.

```
sudo chmod 2770 /private/
sudo chmod 2775 /public
```

In the above command, the value 2 at the beginning, stands for the SGID bit. This allows newly created files to inherit the parent group.

Next, create a no login local user to access the private share.

```
sudo useradd -M -s /sbin/nologin sambauser
```

Add the user to the Samba share group created above.

```
sudo usermod -aG smbshare sambauser
```

Now create an SMB password for the user.

```
sudo smbpasswd -a sambauser
```

Enable the created account:

```
sudo smbpasswd -e sambauser
```

## 5 – Verify the Samba configuration

Once changes have been made to the config file, it is recommended that you test it using the below command:

```
sudo testparm
```

Execution output:

```
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Weak crypto is allowed
Server role: ROLE_STANDALONE

Press enter to see a dump of your service definitions

# Global parameters
[global]
  interfaces = 192.168.205.0/24 eth0
  log file = /var/log/samba/log.%m
  logging = file
```

```

[]map to guest = Bad User
[]max log size = 1000
[]obey pam restrictions = Yes
[]pam password change = Yes
[]panic action = /usr/share/samba/panic-action %d
[]passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:* %n\n
*password\supdated\ssuccessfully* .
[]passwd program = /usr/bin/passwd %u
[]server role = standalone server
[]unix password sync = Yes
[]usershare allow guests = Yes
[]idmap config * : backend = tdb
.....
[public]
[]comment = Public Folder
[]force create mode = 0775
[]force directory mode = 0775
[]guest ok = Yes
[]guest only = Yes
[]path = /public
[]read only = No

[private]
[]comment = Private Folder
[]force create mode = 0770
[]force directory mode = 0770
[]inherit permissions = Yes
[]path = /private
[]read only = No
[]valid users = @smbshare

```

The above output shows that everything is configured appropriately. Now proceed as below.

Create demo files in the Samba shares:

```

sudo mkdir /private/demo-private /public/demo-public
sudo touch /private/demo1.txt /public/demo2.txt

```

Restart the Samba service for the changes to apply.

```
sudo systemctl restart nmbd
```

If you have a firewall running, you need to allow remote access from the specified IP range:

```
sudo ufw allow from 192.168.205.0/24 to any app Samba
```

---

Created 2025-03-11 02:28:15 UTC by Nicolas

Updated 2025-03-11 02:30:42 UTC by Nicolas