# SAMBA

***Samba*** is an open source implementation of the Server Message Block (SMB) protocol. It allows the networking of Microsoft Windows®, Linux, UNIX, and other operating systems together, enabling access to Windows-based file and printer shares. Samba's use of SMB allows it to appear as a Windows server to Windows clients.

**Samba daemons**

| | | |
|---|---|---|
| Samba is comprised of three daemons | - | **smbd, nmbd, and winbindd** |
| Three services | - | **smb,nmb and winbind** |

**Smbd**

The **smbd** server daemon provides file sharing and printing services to Windows clients. In addition, it is responsible for user authentication, resource locking, and data sharing through the SMB protocol. The default ports on which the server listens for SMB traffic are **TCP** ports **139**
The smbd daemon is controlled by the **smb** service.

**Nmbd**

The nmbd server daemon understands and replies to NetBIOS name service requests such as those produced by SMB/ *Common Internet File System* (CIFS) in Windows-based systems. The default port that the server listens to for NMB traffic is UDP port 137.
The nmbd daemon is controlled by the **nmb** service.

**Winbindd**

The winbind service resolves user and group information on a server running Windows NT, 2000, 2003 or Windows Server 2008. This makes Windows user / group information understandable by UNIX platforms. This is achieved by using Microsoft RPC calls, *Pluggable Authentication Modules* (PAM), and the *Name Service Switch* (NSS). This allows Windows NT domain users to appear and operate as UNIX users on a UNIX machine. Though bundled with the Samba distribution, the winbind service is controlled separately from the smb service.
The winbindd daemon is controlled by the winbind service and does not require the smb service to be started in order to operate.

---

**RHEL 6 basic file sharing samba require packages**

```
[root@server1 Desktop]# yum install samba*
[root@server1 Desktop]# rpm -qa samba*
samba-3.5.10-114.el6.x86_64
samba-winbind-3.5.10-114.el6.x86_64
samba-winbind-clients-3.5.10-114.el6.x86_64
samba-client-3.5.10-114.el6.x86_64
samba-common-3.5.10-114.el6.x86_64
```

```
[root@server1 Desktop]# rpm -qlc samba-common
/etc/samba/lmhosts
/etc/samba/smb.conf                    The default samba configuration file
/etc/sysconfig/samba
```

**Starting and stopping samba**

```
[root@server1 Desktop]# service smb {start|stop|restart|reload|configtest|status|condrestart
[root@server1 Desktop]# chkconfig smb --list
smb             0:off    1:off    2:off    3:off    4:off    5:off    6:off
[root@server1 Desktop]# chkconfig smb on
```

```
[root@server1 Desktop]# chkconfig nmb –list
nmb             0:off    1:off    2:off    3:off    4:off    5:off    6:off
[root@server1 Desktop]# chkconfig nmb on
```

---

**Samba configuration /etc/samba/smb.conf**

```
[root@server1 Desktop]# vim /etc/samba/smb.conf

workgroup = WORKGROUPNAME
server string = BRIEF COMMENT ABOUT SERVER
netbios name = MYSERVER
security = user
interfaces = lo eth0 192.168.12.2/24 192.168.13.2/24
hosts allow = 127. 192.168.12. 192.168.13.

[sharename]
comment = Insert a comment here
path = /home/share/
valid users = tfox carole
public = no
writable = yes
printable = no
create mask = 0765
```

The above example allows the users `tfox` and `carole` to read and write to the directory `/home/share`, on the Samba server, from a Samba client.

**Note**

All servers (including Samba) should connect to a WINS server to resolve **NetBIOS** names. Without WINS, browsing only occurs on the local subnet. Furthermore, even if a domain-wide list is somehow obtained, hosts cannot be resolved for the client without WINS.

**Anonymous Read-Only**

```
[global]
workgroup = DOCS
netbios name = DOCS_SRV
security = share

[data]
comment = Documentation Samba Server
path = /export
read only = Yes
guest only = Yes
```

**Anonymous Read/Write**

```
[global]
workgroup = DOCS
netbios name = DOCS_SRV
security = share

[data]
comment = Data
```

```
path = /export
force user = docsbot
force group = users
read only = No
guest ok = Yes
```

### The “testparm” command

The “testparm” program checks the syntax of the /etc/samba/smb.conf file.

**syntax**
**testparm *<options> <filename> <hostname IP_address>***

```
[root@server1 Desktop]# testparm
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]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

[global]
        workgroup = MYGROUP
        server string = Samba Server Version %v
        log file = /var/log/samba/log.%m
        max log size = 50
        cups options = raw

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

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

---

### Create new samba user

**Smbpasswd [options] [username]**

**[options]**

| | | |
|---|---|---|
| **-a** | **-** | **add user** |
| **-d** | **-** | **disable user** |
| **-e** | **-** | **enable user** |
| **-n** | **-** | **set no user password** |
| **-x** | **-** | **delete user** |

```
[root@server1 Desktop]# useradd -s /sbin/nologin babu
[root@server1 Desktop]# smbpasswd -a babu
New SMB password:
Retype new SMB password:
Added user babu.
```

---

# Samba Security level mode

There are two types of security level “user-level-security” and “share-level-security”
The “user-level-security” cab be implemented four way.
The “share-level-security” can be implemented one way.

## 1. User-level-security

User-level security is the default setting for Samba. Even if the security = user directive is not listed in the /etc/samba/smb.conf file, it is used by Samba. If the server accepts the client's username/password, the client can then mount multiple shares without specifying a password for each instance. Samba can also accept session-based username/password requests. The client maintains multiple authentication contexts by using a unique UID for each logon.

In the /etc/samba/smb.conf file, the security = user directive that sets user-level security is:

```
[GLOBAL]
security = user
```

## 2. Domain-level-security (user-level-security)

In domain security mode, the Samba server has a machine account (domain security trust account) and causes all authentication requests to be passed through to the domain controllers. The Samba server is made into a domain member server by using the following directives in the /etc/samba/smb.conf file:

```
[GLOBAL]
 security = domain
workgroup = MARKETING
```

## 3. Active-Directory-Security-mode (user-level-security)

If you have an Active Directory environment, it is possible to join the domain as a native Active Directory member. Even if a security policy restricts the use of NT-compatible authentication protocols, the Samba server can join an ADS using Kerberos. Samba in Active Directory member mode can accept Kerberos tickets.
In the /etc/samba/smb.conf file, the following directives make Samba an Active Directory member server:

```
[GLOBAL]
security = ADS
realm = EXAMPLE.COM
password server = kerberos.example.com
```

## 4. Server-Security-Mode (user-level-security)

Server security mode was previously used when Samba was not capable of acting as a domain member server.

**Avoid using the server security mode**
**It is highly recommended to *not* use this mode since there are numerous security drawbacks.**

In the /etc/samba/smb.conf, the following directives enable Samba to operate in server security mode:

```
[GLOBAL]
encrypt passwords = Yes
security = server
password server = "NetBIOS_of_Domain_Controller"
```

## 5. Share-level-security

With share-level security, the server accepts only a password without an explicit username from the client. The server expects a password for each share, independent of the username. There have been recent reports that Microsoft Windows clients have compatibility issues with share-level security servers. Samba developers strongly discourage use of share-level security.
In the /etc/samba/smb.conf file, the security = share directive that sets share-level security is:

```
[GLOBAL]
security = share
```

## Samba Firewall Port

| | |
|---|---|
| **Netbios –ns** | **-udp-137** |
| **Netbios-dgm** | **-udp-138** |
| **Netbios-ssn** | **-tcp-139** |
| **Microsoft-ds** | **-tcp-445** |

```
[root@server1 Desktop]# iptables -I INPUT -p udp --dport 137:138 -j ACCEPT
[root@server1 Desktop]# iptables -I INPUT -p tcp --dport 139 -j ACCEPT
[root@server1 Desktop]# iptables -I INPUT -p tcp --dport 445 -j ACCEPT
```

```
[root@server1 Desktop]# iptables -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:microsoft-ds
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:netbios-dgm
ACCEPT     udp  --  anywhere             anywhere            udp dpts:netbios-ns:netbios-dgm
```

---

## Samba server client side connection method

To connect to a Samba share from a shell prompt, type the following command:

```
Syntax:
smbclient //<hostname>/<sharename> -U <username>
```

Example:
[root@client1 Desktop]# **smbclient //server1.example.com/samba -U senthil**

### Mounting the samba share directory at client side

The **mount.cifs** utility is a separate RPM (independent from Samba). In order to use **mount.cifs**, first ensure the **cifs-utils** package is installed on your system by running, as root:

```
yum install cifs-utils
```

Note that the **cifs-utils** package also contains the **cifs.upcall** binary called by the kernel in order to perform kerberized CIFS mounts. For more information on **cifs.upcall**, refer to man cifs.upcall.

**syntax**

```
mount -t cifs //<servername>/<sharename> /mnt/point/ -o
username=<username>,password=<password>
Example
[root@client1 Desktop]# mount -t cifs //server1.example.com/samba /mnt/point/
-o username=senthil,password=1
```

### To check server side client system connecting status using “smbstatus” command

[root@server1 ~]# **smbstatus**

```
Unknown parameter encountered: "valid user"
Ignoring unknown parameter "valid user"
Samba version 3.5.10-114.el6
PID     Username      Group         Machine
-------------------------------------------------------------------
7018    senthil       admin         client1      (::ffff:192.168.1.101)

Service      pid     machine       Connected at
-------------------------------------------------------
samba        7018   client1       Wed Nov 21 10:56:56 2012
No locked files
```

## Samba Server and Client Setup Example

**Share the /common directory via SMB:**

- **Your SMB server must be a member of the STAFF workgroup**
- **The share' name must be common**
- **The shared share must be available to example.com domain clients only**
- **The shared share must be browse able**
- **Senthil, babu have read access to the share, authenticating with the password**
- **Kumar have read and write access to the share, authenticating with the password.**

**Install the required packages**

```
[root@server1 Desktop]# yum install samba*
```

**Check the installation status**

```
[root@server1 Desktop]# rpm -qa samba*
samba-3.5.10-114.el6.x86_64
samba-winbind-3.5.10-114.el6.x86_64
samba-winbind-clients-3.5.10-114.el6.x86_64
samba-client-3.5.10-114.el6.x86_64
samba-common-3.5.10-114.el6.x86_64

[root@server1 Desktop]# rpm -qa cifs-utils
cifs-utils-4.8.1-5.el6.x86_64                    default installed this package for mounting purpose

[root@server1 Desktop]# rpm -qlc samba
/etc/logrotate.d/samba
/etc/pam.d/samba
/etc/samba/smbusers
[root@server1 Desktop]# rpm -qlc samba-common
/etc/samba/lmhosts
/etc/samba/smb.conf                    samba configuration file
/etc/sysconfig/samba
```

**Create a shared directory**

```
[root@server1 Desktop]# mkdir /common
[root@server1 Desktop]# ll -dZ /common/
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /common/
[root@server1 Desktop]# semanage fcontext -a -t samba_share_t '/common(/.*)?'
[root@server1 Desktop]# restorecon -FRvv /common
restorecon reset /common context unconfined_u:object_r:default_t:s0-
>system_u:object_r:samba_share_t:s0
[root@server1 Desktop]# chmod 777 /common/
[root@server1 Desktop]# ls -dZ /common
drwxrwxrwx. root root system_u:object_r:samba_share_t:s0 /common
```

**Samba server configuration**

```
[root@server1 Desktop]# vim /etc/samba/smb.conf
:set nu

74      workgroup = STAFF
75      server string = DOCUMENT SHARE
```

```
76
77        netbios name = server1
78
79 ;      interfaces = lo eth0 192.168.12.2/24 192.168.13.2/24
80        hosts allow = 127. 192.168.1.

101        security = user
102        passdb backend = tdbsam

289 [common]
290 comment = skylark
291 path = /common
292 public = no
293 browseable = yes
294 writable = no
295 valid users = senthil babu kumar
296 write list = kumar
:wq!
```

```
[root@server1 Desktop]# service smb configtest
Syntax OK
[root@server1 Desktop]# service smb start
[root@server1 Desktop]# chkconfig smb on
[root@server1 Desktop]# testparm
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 "[common]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
[global]
        workgroup = STAFF
        server string = DOCUMENT SHARE
        log file = /var/log/samba/log.%m
        max log size = 50
        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
        browseable = No
[common]
        comment = skylark
        path = /common
        valid users = senthil, babu, kumar
```

```
            write list = kumar
```

**Create samba users**

```
[root@server1 Desktop]# useradd  –M -s /sbin/nologin senthil
[root@server1 Desktop]#useradd  –M -s /sbin/noloign babu
[root@server1 Desktop]#useradd  –M  -s /sbin/nologin kumar
 [root@server1 Desktop]#smbpasswd  –a senthil
[root@server1 Desktop]#smbpasswd  –a babu
[root@server1 Desktop]#smbpasswd  –a kumar
[root@server1 Desktop]# service smb restart
Shutting down SMB services:                          [  OK  ]
Starting SMB services:                               [  OK  ]
```

**Samba login client side**

```
[root@client1 Desktop]# smbclient //server1.example.com/common -U senthil
Enter senthil's password:
Domain=[STAFF] OS=[Unix] Server=[Samba 3.5.10-114.el6]
smb: \> dir
  .                           D        0  Wed Nov 21 15:12:24 2012
  ..                         DR        0  Wed Nov 21 14:04:01 2012

                59057 blocks of size 524288. 43459 blocks available
smb: \> mkdir test
NT_STATUS_MEDIA_WRITE_PROTECTED making remote directory \test
smb: \> exit
[root@client1 Desktop]# smbclient //server1.example.com/common -U kumar
Enter kumar's password:
Domain=[STAFF] OS=[Unix] Server=[Samba 3.5.10-114.el6]
smb: \> mkdir test
smb: \> dir
  .                           D        0  Wed Nov 21 15:24:46 2012
  ..                         DR        0  Wed Nov 21 14:04:01 2012
  test                        D        0  Wed Nov 21 15:24:46 2012

                59057 blocks of size 524288. 43459 blocks available
smb: \> exit
```

**Samba server connect mounting option**

```
[root@client1 Desktop]# mkdir /smbmount
[root@client1 Desktop]# rpm -qa cifs-utils
cifs-utils-4.8.1-5.el6.x86_64
[root@client1 Desktop]# mount -t cifs //server1.example.com/common /smbmount -o
username=kumar,password=1
[root@client1 Desktop]# vim /etc/fstab
//192.168.1.100/common        /smbmount            cifs      username=kumar,password=1  0 0
:wq!
[root@client1 ~]# mount | grep cifs
//192.168.1.100/common on /smbmount type cifs (rw)
```