**Samba installation on RHEL/CentOS 7**

**Samba** is a **client**/**server** system that implements**network resource sharing** for Linux and other UNIX computers. With **Samba**, UNIX **files** and **printers** can be **shared** with**Windows clients** and vice versa. **Samba** supports the **Session Message Block (SMB) protocol**. Nearly all Windows computers include **SMB** support with their internal network subsystems (NetBIOS in particular).  
With an appropriately-configured **Samba** server on **Linux**, **Windows clients** can map drives to the Linux filesystems. Likewise, the**Samba client** on UNIX can connect to **Windows** shares by their UNC name. Although differences among various operating systems (such as filesystem naming conventions, end-of-line conventions, and authentication) can limit interoperability, **Samba** offers a generally serviceable mechanism for resource sharing on a heterogenous network.  
In this tutorial we will show you h**ow to install and configure Samba server on RHEL and CentOS 7 linux.**

## Install and configure Samba on Rhel/CentOS 7

To install samba packages enter following command:

yum install samba samba-client samba-common -y

Now configure samba edit the file /etc/samba/smb.conf

mv /etc/samba/smb.conf /etc/samba/smb.conf.bkp

vi /etc/samba/smb.conf

and paste following line:

[global]

workgroup = WORKGROUP

server string = Samba Server %v

netbios name = centos

security = user

map to guest = bad user

dns proxy = no

#============================ Share Definitions ==============================

[Anonymous]

path = /samba/anonymous

browsable =yes

writable = yes

guest ok = yes

read only = no

Save the smb.conf file and restart the service:

mkdir -p /samba/anonymous

systemctl enable smb.service

systemctl enable nmb.service

systemctl restart smb.service

systemctl restart nmb.service

Add these Iptables rules, so that samba will work perfectly:

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

firewall-cmd --reload

Change permission for samba share:

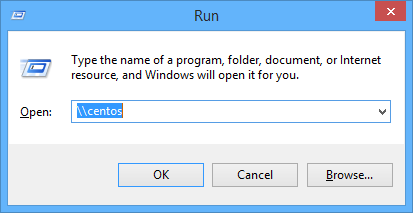
chmod -R 0755 anonymous/

chown -R nobody:nobody anonymous/

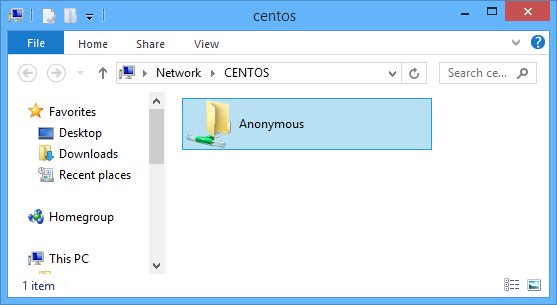
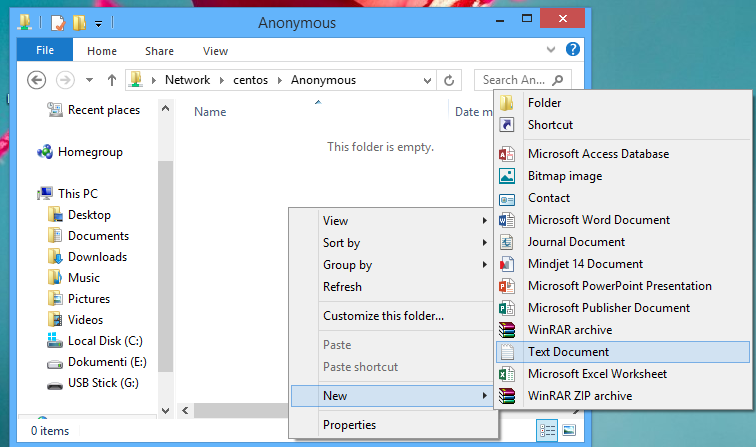
Further we need to allow the selinux for the samba configuration as follows:

chcon -t samba\_share\_t anonymous/

Now you can access the Centos 7.0 sharing in windows as follows, go to the Run prompt and type \centos :



*Access to samba share*

  
Now anonymous user can browse & create new text documents:  


### Secured samba server

For this I will create a group smbgrp & user rasho to access the samba server with proper authentication

useradd rasho

groupadd smbgrp

usermod -a -G smbgrp rasho

smbpasswd -a rasho

[root@localhost]# smbpasswd -a rasho

New SMB password: YOUR SAMBA PASS

Retype new SMB password: REPEAT YOUR SAMBA PASS

Added user rasho.

Create a new share, set the permission on the share:

mkdir /home/secure

chown -R rasho:smbgrp /home/secure/

chmod -R 0770 /home/secure/

chcon -t samba\_share\_t /home/secure/

Again edit the configuration file as :

vi /etc/samba/smb.conf

Add the newly created samba share in smb.conf file:

[Secure]

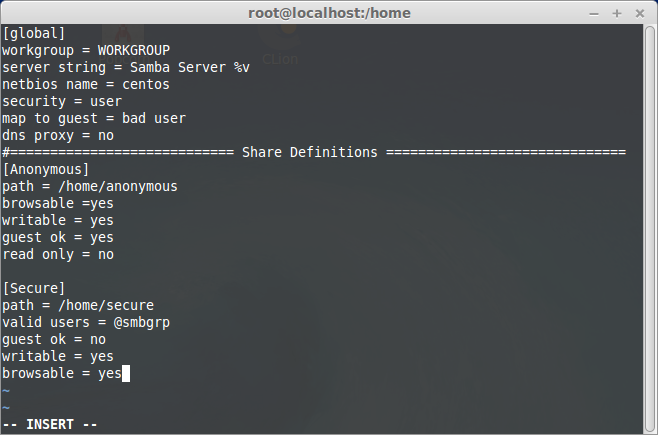
path = /home/secure

valid users = @smbgrp

guest ok = no

writable = yes

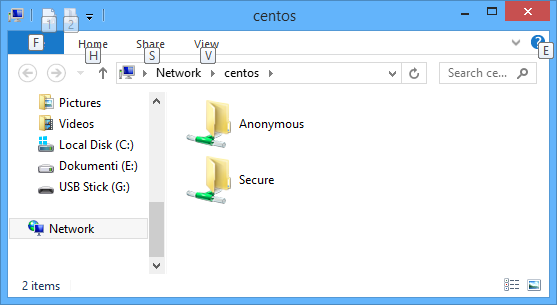
browsable = yes

  
Restart the samba service:

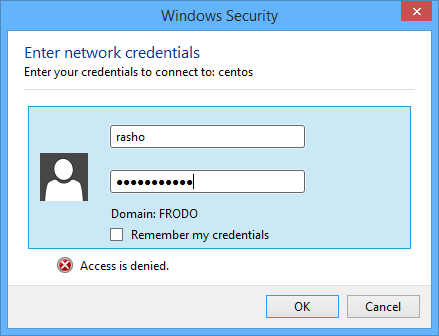
systemctl restart smb.service

systemctl restart nmb.service

Now at windows machine check the folder now with the proper credentials

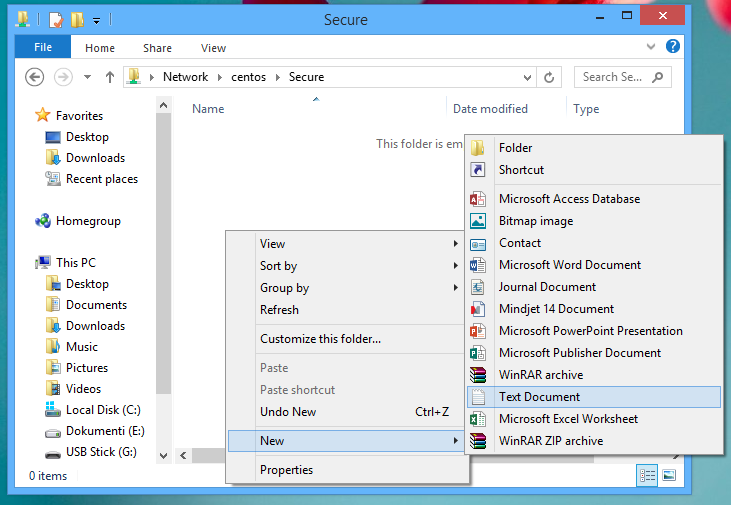


*Open samba sharing*



*Enter your samba user and passa*

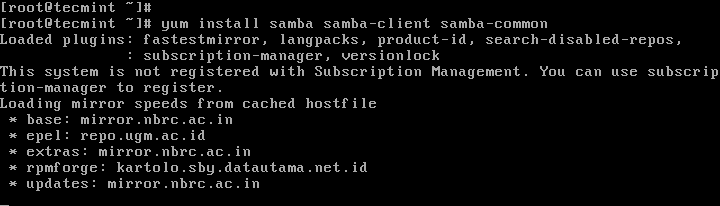
Create new text documents:



### Install Samba4 in CentOS 7

**1.** First install Samba4 and required packages from the default CentOS repositories using the yum package manager tool as shown.

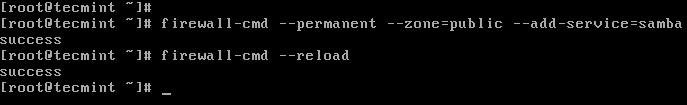
# yum install samba samba-client samba-common



**2.** After installing the samba packages, enable samba services to be allowed through system firewall with these commands.

# firewall-cmd --permanent --zone=public --add-service=samba

# firewall-cmd --reload



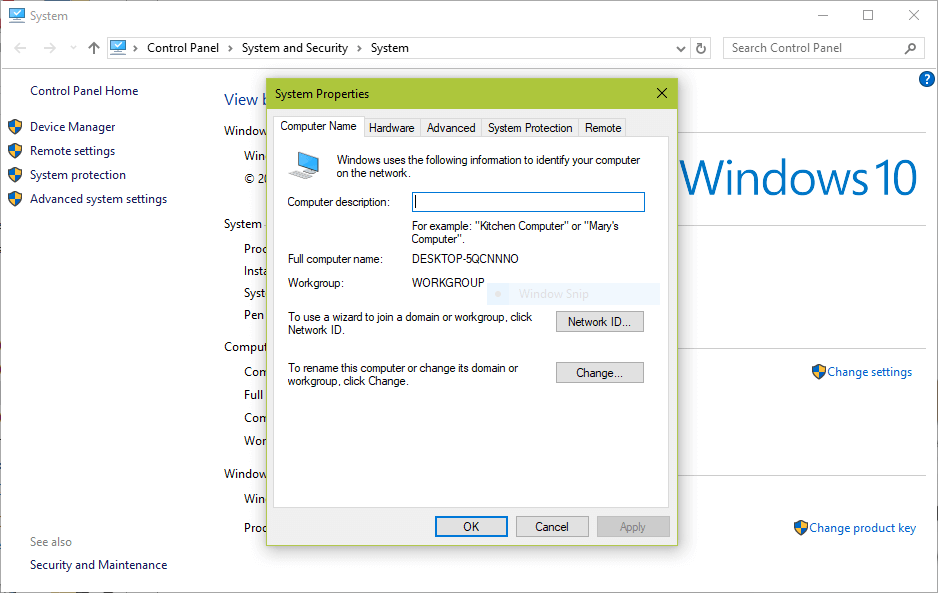
*Open Samba on Firewalld*

### Check Windows Machine Workgroup Settings

**3.** Before you proceed to configure samba, make sure the Windows machine is in the same **workgroup** to be configured on the CentOS server.

There are two possible ways to view the Windows machine workgroup settings:

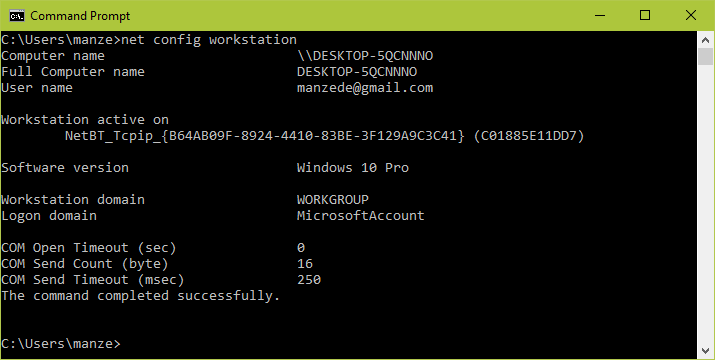
* Right clicking on “**This PC**” or “**My Computer**” → **Properties** → **Advanced system settings** → **Computer Name**.



*Check Windows WorkGroup*

* Alternatively, open the **cmd prompt** and run the following command, then look for **“workstation domain**” in the output as shown below.

>net config workstation

[](https://www.tecmint.com/wp-content/uploads/2017/06/Verify-Windows-WorkGroup.png)

*Verify Windows WorkGroup*

### Configuring Samba4 on CentOS 7

**4.** The main samba configuration file is **/etc/samba/smb.conf**, the original file comes with pre-configuration settings which explain various configuration directives to guide you.

But, before configuring samba, I suggest you to take a backup of the default file like this.

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

Then, proceed to configure samba for **anonymous** and **secure file sharing** services as explained below.

#### Samba4 Anonymous File Sharing

**5.** First create the shared directory where the files will be stored on the server and set the appropriate permissions on the directory.

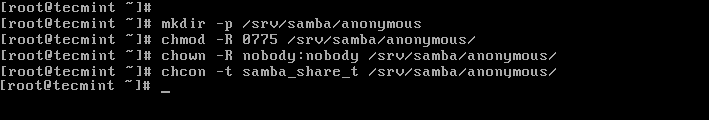
# mkdir -p /srv/samba/anonymous

# chmod -R 0775 /srv/samba/anonymous

# chown -R nobody:nobody /srv/samba/anonymous

Also, you need to change the SELinux security context for the samba shared directory as follows.

# chcon -t samba\_share\_t /srv/samba/anonymous

[](https://www.tecmint.com/wp-content/uploads/2017/07/Create-Samba-Shared-Directory.png)

*Create Samba Shared Directory*

**6.** Next, open the samba configuration file for editing, where you can modify/add the sections below with the corresponding directives.

# vi /etc/samba/smb.conf

**Samba Configuration Settings**

[global]

workgroup = WORKGROUP

netbios name = centos

security = user

[Anonymous]

comment = Anonymous File Server Share

path = **/srv/samba/anonymous**

browsable =yes

writable = yes

guest ok = yes

read only = no

force user = nobody

**7.** Now verify current samba settings by running the command below.

# testparm

**Verify Samba Current Configuration Settings**

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 "[print$]"

Processing section "[Anonymous]"

Loaded services file OK.

Server role: ROLE\_STANDALONE

Press enter to see a dump of your service definitions

# Global parameters

[global]

netbios name = centos

printcap name = cups

security = USER

idmap config \* : backend = tdb

cups options = raw

[homes]

comment = Home Directories

browseable = No

inherit acls = Yes

read only = No

valid users = %S %D%w%S

[printers]

comment = All Printers

path = /var/tmp

browseable = No

printable = Yes

create mask = 0600

[print$]

comment = Printer Drivers

path = /var/lib/samba/drivers

create mask = 0664

directory mask = 0775

write list = root

[Anonymous]

comment = Anonymous File Server Share

path = /srv/samba/anonymous

force user = nobody

guest ok = Yes

read only = No

**8.** Finally, start and enable samba services to start automatically at next boot and also apply the above changes to take effect.

# systemctl enable smb.service

# systemctl enable nmb.service

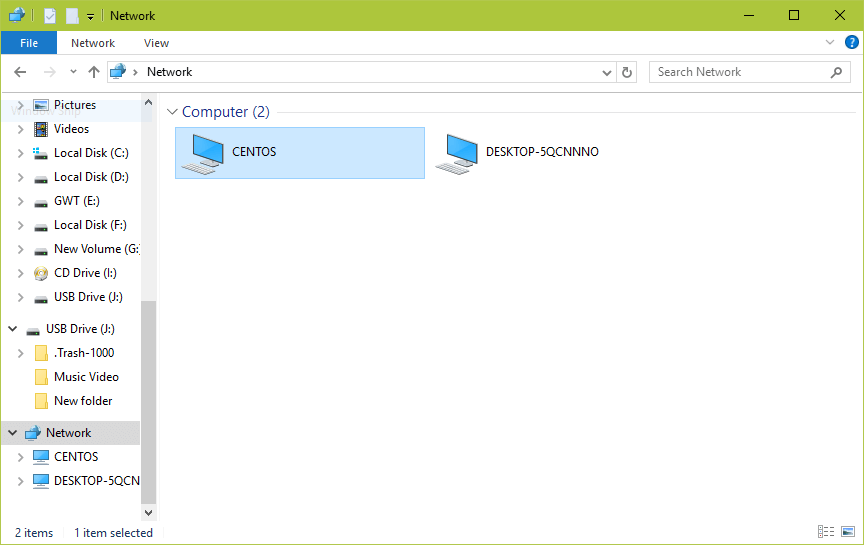
# systemctl start smb.service

# systemctl start nmb.service

#### Testing Anonymous Samba File Sharing

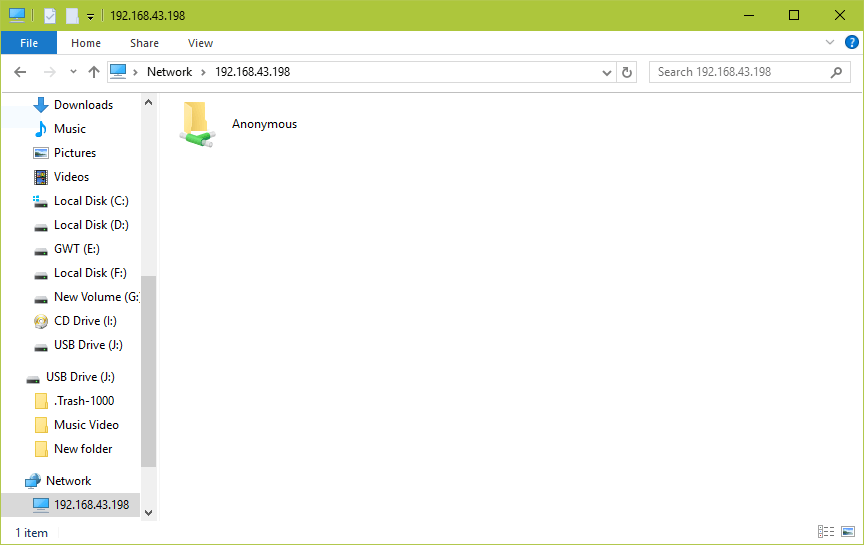
**9.** Now on the Windows machine, open “**Network**” from a Windows Explorer window, then click on the **CentOS**host, or else try to access the server using its IP address (use ifconfig command to get IP address).

e.g. \\192.168.43.168.

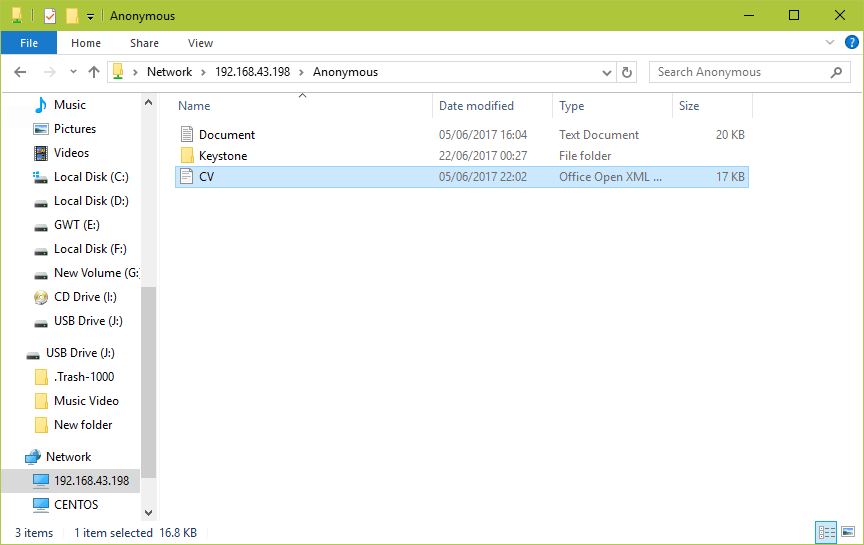


*Shared Network Hosts*

**10.** Next, open the **Anonymous** directory and try to add files in there to share with other users.

[](https://www.tecmint.com/wp-content/uploads/2017/07/Samba-Anonymous-Share.png)

*Samba Anonymous Share*

[](https://www.tecmint.com/wp-content/uploads/2017/07/Add-Files-to-Samba-Anonymous-Share.png)

*Add Files to Samba Anonymous Share*

### Setup Samba4 Secure File Sharing

**11.** First start by creating a samba system group, then add users to the group and set a password for each user like so.

# groupadd smbgrp

# usermod tecmint -aG smbgrp

# smbpasswd -a tecmint

**12.** Then create a secure directory where the shared files will be kept and set the appropriate permissions on the directory with SELinux security context for the samba.

# mkdir -p /srv/samba/secure

# chmod -R 0770 /srv/samba/secure

# chown -R root:smbgrp /srv/samba/secure

# chcon -t samba\_share\_t /srv/samba/secure

**13.** Next open the configuration file for editing and modify/add the section below with the corresponding directives.

# vi /etc/samba/smb.conf

**Samba Secure Configuration Settings**

[Secure]

comment = Secure File Server Share

path = **/srv/samba/secure**

valid users = @smbgrp

guest ok = no

writable = yes

browsable = yes

**14.** Again, verify the samba configuration settings by running the following command.

$ testparm

**Verify Secure Configuration Settings**

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 "[print$]"

Processing section "[Anonymous]"

Loaded services file OK.

Server role: ROLE\_STANDALONE

Press enter to see a dump of your service definitions

# Global parameters

[global]

netbios name = centos

printcap name = cups

security = USER

idmap config \* : backend = tdb

cups options = raw

[homes]

comment = Home Directories

browseable = No

inherit acls = Yes

read only = No

valid users = %S %D%w%S

[printers]

comment = All Printers

path = /var/tmp

browseable = No

printable = Yes

create mask = 0600

[print$]

comment = Printer Drivers

path = /var/lib/samba/drivers

create mask = 0664

directory mask = 0775

write list = root

[Anonymous]

comment = Anonymous File Server Share

path = /srv/samba/anonymous

force user = nobody

guest ok = Yes

read only = No

[Secure]

comment = Secure File Server Share

path = /srv/samba/secure

read only = No

valid users = @smbgrp

**15.** Restart Samba services to apply the changes.

# systemctl restart smb.service

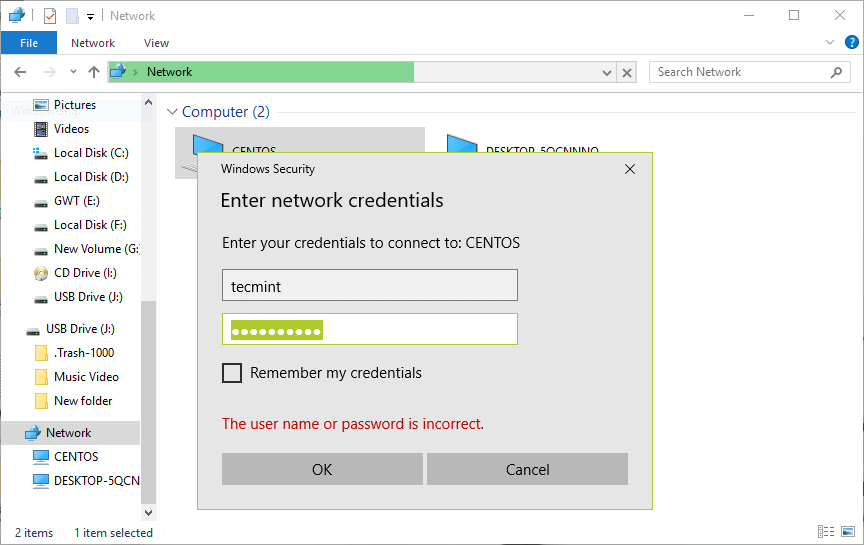
# systemctl restart nmb.service

#### Testing Secure Samba File Sharing

**16.** Go to Windows machine, open “**Network**” from a Windows Explorer window, then click on the **CentOS** host, or else try to access the server using its IP address.

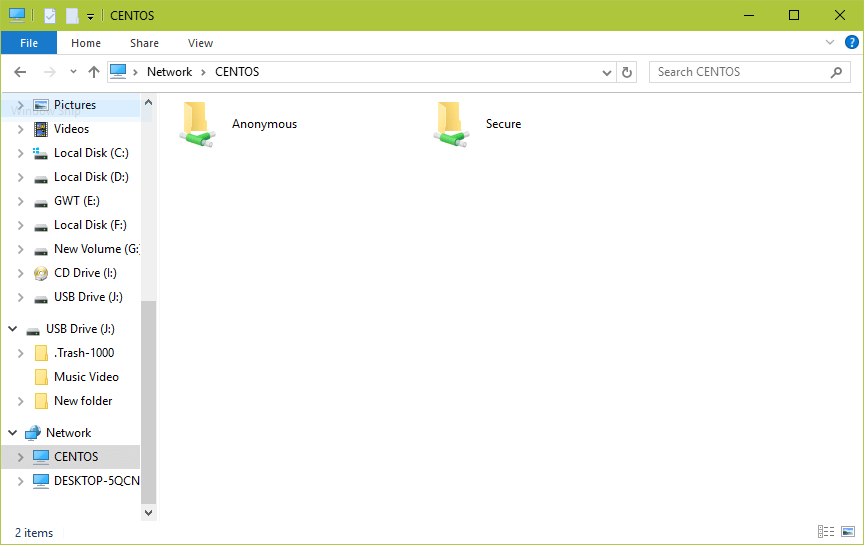
e.g. \\192.168.43.168.

You’ll be asked to provide your username and password to login the CentOS server. Once you have entered the credentials, click **OK**.



*Samba Secure Login*

**17.** Once you successfully login, you will see all the samba shared directories. Now securely share some files with other permitted users on the network by dropping them in **Secure** directory.



*Samba Secure Share Directory*

# How to Mount and Unmount Filesystem / Partition in Linux (Mount/Umount Command Examples)

he general mount command syntax to mount a device:

mount -t type device destination\_dir

### 1. Mount a CD-ROM

The device file for CD would exist under /dev directory. For example, a CD-ROM device will be mounted as shown below.

# mount -t iso9660 -o ro /dev/cdrom /mnt

In the above example, the option “-o ro” indicates that the cdrom should be mounted with read-only access. Also, make sure that the destination directory (in the above example, /mnt) exist before you execute the mount command.

### 2. View All Mounts

After you execute mount a partition or filesystem, execute the mount command without any arguments to view all the mounts.

In the example below, after mounting the USB drive on a system, the output of mount looks like the below. As seen below, the USB device (i.e:/dev/sdb) is mounted on /media/myusb, which is displayed as the last line in the mount command.

# mount

/dev/sda5 on / type ext4 (rw,errors=remount-ro)

proc on /proc type proc (rw,noexec,nosuid,nodev)

sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)

none on /sys/fs/fuse/connections type fusectl (rw)

none on /sys/kernel/debug type debugfs (rw)

none on /sys/kernel/security type securityfs (rw)

udev on /dev type devtmpfs (rw,mode=0755)

devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)

tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)

none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)

none on /run/shm type tmpfs (rw,nosuid,nodev)

/dev/sda6 on /mydata type ext2 (rw)

/dev/sda7 on /backup type vfat (rw)

gvfs-fuse-daemon on /home/bala/.gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev,user=bala)

/dev/sdb on /media/myusb type vfat (rw,nosuid,nodev,uid=1000,gid=1000,shortname=mixed,dmask=0077,utf8=1,showexec,flush,uhelper=udisks)

You can also use df command to view all the mount points.

# df

Filesystem 1K-blocks Used Available Use% Mounted on

/dev/sda5 195069136 128345036 56958520 70% /

udev 2008336 4 2008332 1% /dev

tmpfs 806244 928 805316 1% /run

none 5120 0 5120 0% /run/lock

none 2015604 228 2015376 1% /run/shm

/dev/sda6 17729076 176200 16657596 2% /mydata

/dev/sda7 11707200 573312 11133888 5% /backup

/dev/sdb 3910656 2807160 1103496 72% /media/myusb

### 3. Mount all the filesystem mentioned in /etc/fstab

The filesystems listed in /etc/fstab gets mounted during booting process. After booting, system administrator may unmount some of the partitions for various reasons. If you want all the filesystems to be mounted as specified in /etc/fstab, use -a option with mount as shown below:

Example /etc/fstab file entries:

# cat /etc/fstab

#

proc /proc proc nodev,noexec,nosuid 0 0

# / was on /dev/sda5 during installation

/dev/sda5 / ext4 errors=remount-ro 0 1

# /mydata was on /dev/sda6 during installation

/dev/sda6 /mydata ext2 defaults 0 2

# /backup was on /dev/sda7 during installation

/dev/sda7 /backup vfat defaults 0 3

Execute mount command with -a option to mount all the /etc/fstab entries.

# mount -a

# mount

/dev/sda5 on / type ext4 (rw,errors=remount-ro)

proc on /proc type proc (rw,noexec,nosuid,nodev)

sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)

none on /sys/fs/fuse/connections type fusectl (rw)

none on /sys/kernel/debug type debugfs (rw)

none on /sys/kernel/security type securityfs (rw)

udev on /dev type devtmpfs (rw,mode=0755)

devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)

tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)

none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)

none on /run/shm type tmpfs (rw,nosuid,nodev)

/dev/sda6 on /mydata type ext2 (rw)

/dev/sda7 on /backup type vfat (rw)

gvfs-fuse-daemon on /home/bala/.gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev,user=bala)

The same -a option can be used with umount to unmount all the filesystems mentioned in /etc/mtab

# umount -a

umount: /run/shm: device is busy.

(In some cases useful info about processes that use

the device is found by lsof(8) or fuser(1))

umount: /run: device is busy.

(In some cases useful info about processes that use

the device is found by lsof(8) or fuser(1))

umount: /dev: device is busy.

(In some cases useful info about processes that use

the device is found by lsof(8) or fuser(1))

umount: /: device is busy.

(In some cases useful info about processes that use

the device is found by lsof(8) or fuser(1))

Some filesystem are not unmounted as its busy or currently in use. Note that the files /etc/mtab and /proc/mounts contents would be similar.

### 4. Mount only a specific filesystem from /etc/fstab

When you pass only the directory name to mount, it looks for mount point entries, if not found, then search continuous for a device in /etc/fstab and gets mounted.

# mount | grep /mydata

# cat /etc/fstab | grep mydata

##########/mydata was on /dev/sda6 during installation##########

/dev/sda6 /mydata ext2 defaults 0 2

As seen above, /mydata directory is not a mountpoint, but it is present in /etc/fstab.

# mount /mydata

# mount | grep /mydata

/dev/sda6 on /mydata type ext2 (rw)

If you execute the same again, you would get the error message as follows:

# mount /mydata

mount: /dev/sda6 already mounted or /mydata busy

mount: according to mtab, /dev/sda6 is already mounted on /mydata

Here you may also pass the device name instead of directory name (to be picked up from /etc/fstab file).

# mount /dev/sda6

Note that the files /etc/mtab and /proc/mounts contents would be similar.

### 5. View all mounted partitions of specific type

It is possible to list only the specific type of filesystem mounted using the option -l with -t as shown below:

# mount -l -t ext2

/dev/sda6 on /mydata type ext2 (rw)

# mount -l -t ext4

/dev/sda5 on / type ext4 (rw,errors=remount-ro)

As seen above, /dev/sda6 is the only ext2 partition and /dev/sda5 is the only ext4 partition accordingly.

### 6. Mount a Floppy Disk

The device file for floppy disk would exist under /dev directory. For example, a floppy disk will be mounted as shown below.

# mount /dev/fd0 /mnt

# cd /mnt

After the successful mount, you would be able to access the contents of the floppy disk. Once you are done with it, use umount before you physically remove the floppy disk from the system.

# umount /mnt

### 7. Bind mount points to a new directory

The mountpoint can be binded to a new directory. So that you would be able to access the contents of a filesystem via more than one mountpoints at the same time.

Use -B option with olddir and newdir to be binded as follows,

# mount -B /mydata /mnt

Now the bind is done and you might verify it as follows,

# mount | grep /mydata

/dev/sda6 on /mydata type ext2 (rw)

/mydata on /mnt type none (rw,bind)

As seen above the bind is done properly. So when you do modification in filesystem in one place, you can see those reflection of it in other mount point as shown below:

# cd /mydata

# ls

test

# mkdir dir1

# mkdir dir2

# ls

test dir1 dir2

# cd /mnt

# ls

test dir1 dir2

### 8. Access contents from new mount point

Mount allows you to access the contents of a mount point from a new mount point. Its nothing but move a mounted tree to another place.

In the example below, the mount point /mydata will be accessed from /mnt using the option -M as shown below:

# mount -M /mydata /mnt/

Once its done, you cant use the old mount point as its moved to a new mount point and this can be verified as shown below:

# mount | grep /mydata

# mount | grep /mnt

/dev/sda6 on /mnt type ext2 (rw)

### 9. Mount without writing entry into /etc/mtab

During read only mount of /etc/, the /etc/mtab file entries cannot be modified by mount command. However, mount can be done without writing into /etc/mtab by using the option -n as follows,

# mount -n /dev/sda6 /mydata

You cannot see any entry for this /mydata in mount command output and as well from /etc/mtab file as follows:

# mount | grep /mydata

# cat /etc/mtab | grep /mydata

Access the contents of a mounted directory /mydata:

# cd /mydata

# ls

dir1 dir2 test

### 10. Mount filesystem with read or read/write access

To mount partition as read only, use -r option which is synonym to -o ro.

# mount /dev/sda6 /mydata -r

# mount | grep /mydata

/dev/sda6 on /mydata type ext4 (ro)

ext3 and ext4 filesystem would still allow you to do write operation when the filesystem is dirty. So, you may have to use “ro,noload” to prevent these kind of write operation.

# mount /dev/sda6 /mydata -t ext4 -o ro -o noload

# mount | grep /mydata

/dev/sda6 on /mydata type ext4 (ro,noload)

To mount a partition with read/write access, use -w option which is same as “-o rw” (i.e : default).

### 11. Remount the mounted filesystem

In order to mount the already mounted filesystem, use remount option and its normally used to remount the filesystem with read/write access when its previously mounted with read access.

The /mydata mount point is going to be remounted with read/write access from read access as shown below:

# mount | grep /mydata

/dev/sda6 on /mydata type ext4 (ro,noload)

# mount -o remount,rw /mydata

# mount | grep /mydata

/dev/sda6 on /mydata type ext4 (rw)

### 12. Mount an iso image into a directory

The iso image can be mounted as shown below:

# mount -t iso9660 -o loop pdf\_collections.iso /mnt

# cd /mnt

# ls

perl/ php/ mysql/

### 13. Unmount more than one mount points

Umount allows you to unmount more than mount point in a single execution of umount of command as follows:

# umount /mydata /backup

# mount | grep /mydata

# mount | grep /backup

### 14. Lazy unmount of a filesystem

This is a special option in umount, in case you want to unmount a partition after disk operations are done. You can issue command umount -l with that partition and the unmount will be done after the disk operations gets finished.

For instance, consider a scenario that a task (i.e: script or any other command) is doing a copy operation on a disk and at the same time you are allowed to issue a unmount with -l, so that unmount would be done once the copy is over (i.e: the disk operation).

# umount /mydata -l

### 15. Forcefully unmount a filesystem

umount provides the option to forcefully unmount a filesystem with option -f when the device is busy as shown below:

# umount -f /mnt

If this doesn’t work for you, then you can go for lazy unmount.

Meanwhile, you can also have a look at ps command output that which process is presently using the mountpoint as shown below:

# ps ajx | grep /mydata

2540 3037 3037 2468 pts/2 3037 D+ 0 0:00 cp -r /home/geekstuff/ProjectData/ /mydata

You can also execute fuser command to find out which process is holding the directory for operations.

# fuser -cu /mydata

/mydata: 3087(root)