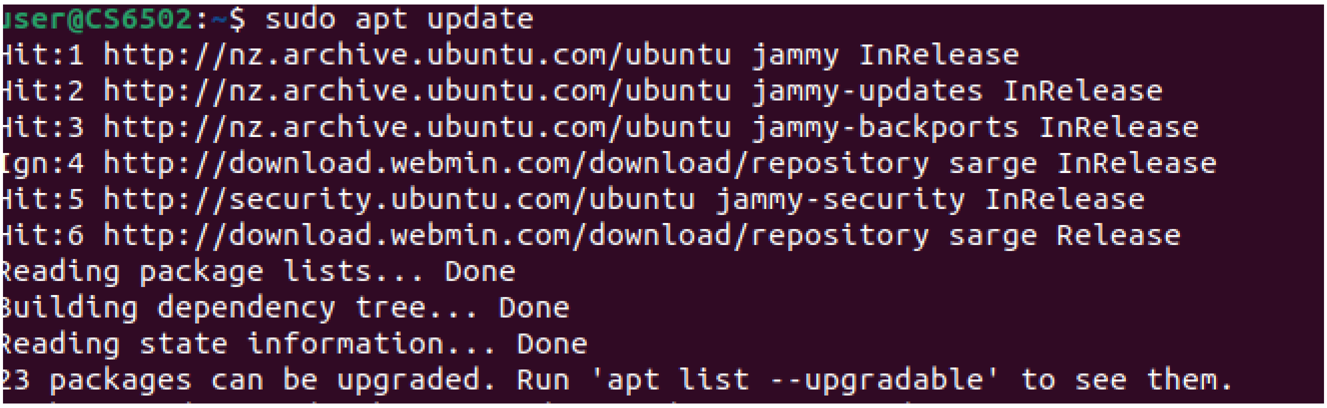
Samba

Install Samba

As usual, start by updating the package registry:

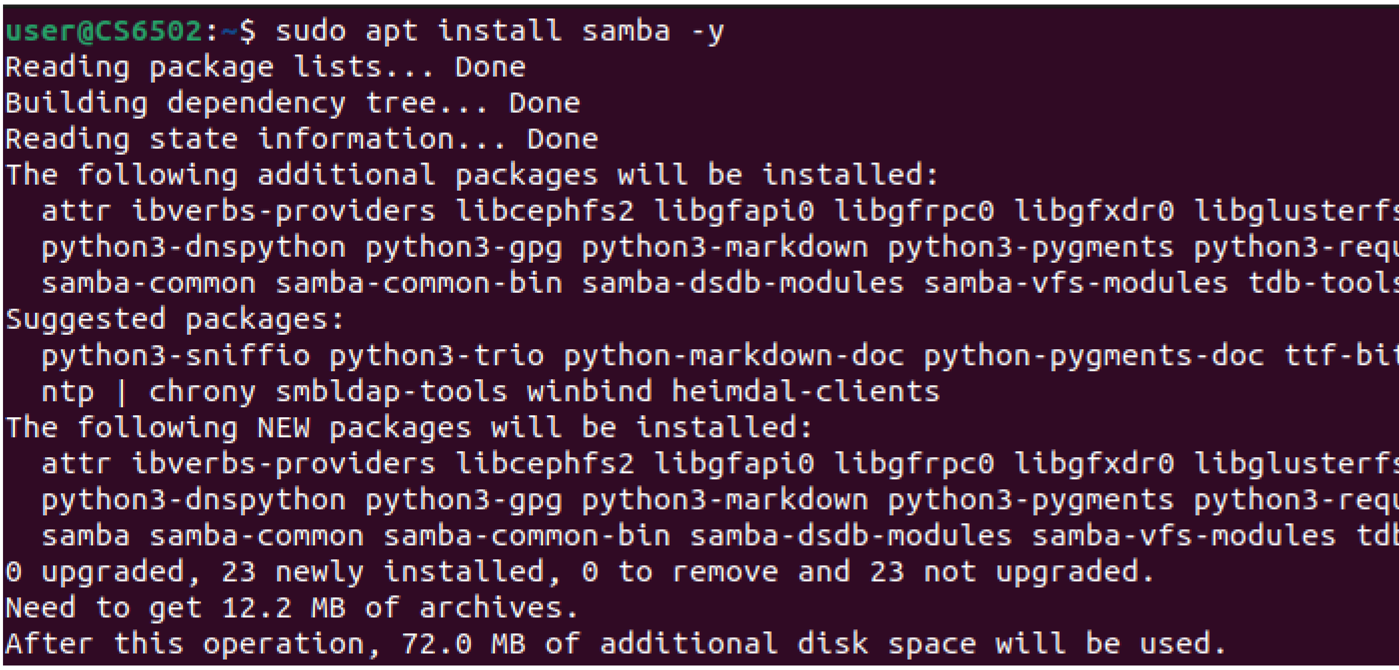
sudo apt update



```
user@CS6502:~$ sudo apt update
Hit:1 http://nz.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://nz.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://nz.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://download.webmin.com/download/repository sarge InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:6 http://download.webmin.com/download/repository sarge Release
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
23 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Next, use apt to install Samba. Include the **-y** argument to auto-approve any queries during the installation:

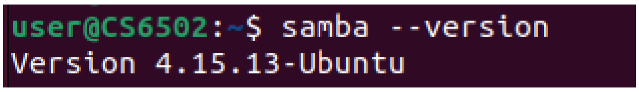
sudo apt install samba -y



```
user@CS6502:~$ sudo apt install samba -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  attr ibverbs-providers libcephfs2 libgfapi0 libgfrpc0 libgfxdr0 libglusterfs
  python3-dnspython python3-gpg python3-markdown python3-pygments python3-requ
  samba-common samba-common-bin samba-dsdb-modules samba-vfs-modules tdb-tools
Suggested packages:
  python3-sniffio python3-trio python-markdown-doc python-pygments-doc ttf-bit
  ntp | chrony smbldap-tools winbind heimdal-clients
The following NEW packages will be installed:
  attr ibverbs-providers libcephfs2 libgfapi0 libgfrpc0 libgfxdr0 libglusterfs
  python3-dnspython python3-gpg python3-markdown python3-pygments python3-requ
  samba samba-common samba-common-bin samba-dsdb-modules samba-vfs-modules tdb
0 upgraded, 23 newly installed, 0 to remove and 23 not upgraded.
Need to get 12.2 MB of archives.
After this operation, 72.0 MB of additional disk space will be used.
```

Verify the installation with:

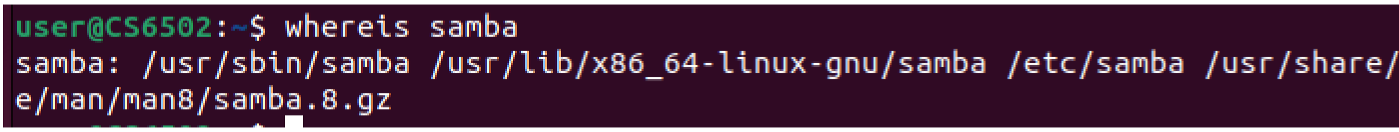
samba --version



```
user@CS6502:~$ samba --version
Version 4.15.13-Ubuntu
```

You can also check the installation withy the 'whereis' command

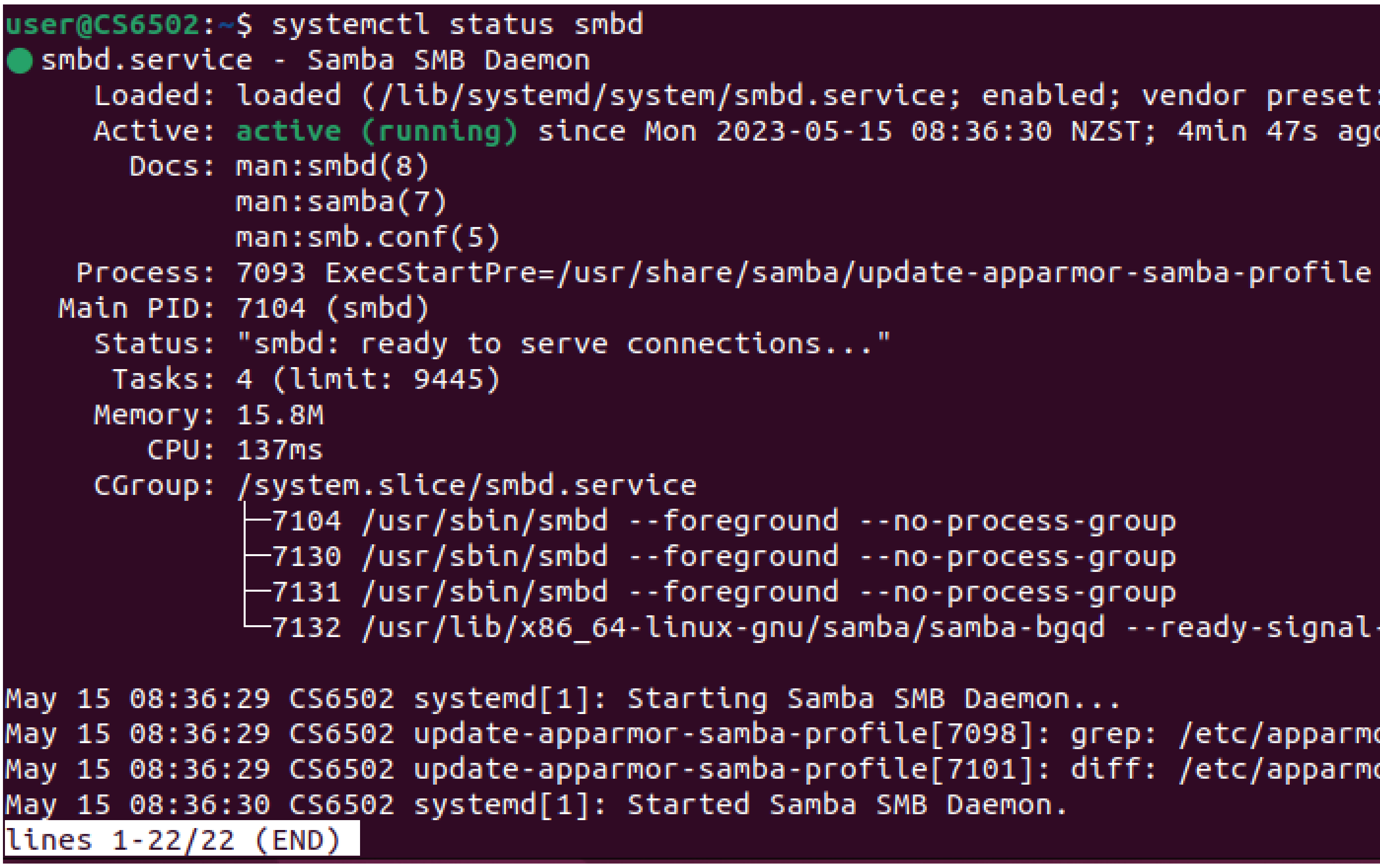
whereis samba



```
user@CS6502:~$ whereis samba
samba: /usr/sbin/samba /usr/lib/x86_64-linux-gnu/samba /etc/samba /usr/share/
e/man/man8/samba.8.gz
```

Confirm that the SMBD (daemon) is running with:

systemctl status smbd

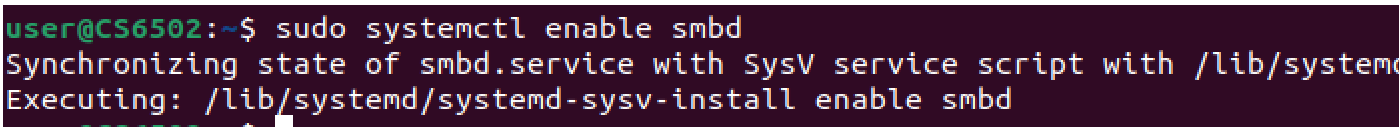


```
user@CS6502:~$ systemctl status smbd
● smbd.service - Samba SMB Daemon
     Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset:
     Active: active (running) since Mon 2023-05-15 08:36:30 NZST; 4min 47s ago
       Docs: man:smbd(8)
             man:samba(7)
             man:smb.conf(5)
    Process: 7093 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile
   Main PID: 7104 (smbd)
     Status: "smbd: ready to serve connections..."
      Tasks: 4 (limit: 9445)
     Memory: 15.8M
        CPU: 137ms
     CGroup: /system.slice/smbd.service
             ├─7104 /usr/sbin/smbd --foreground --no-process-group
             ├─7130 /usr/sbin/smbd --foreground --no-process-group
             ├─7131 /usr/sbin/smbd --foreground --no-process-group
             └─7132 /usr/lib/x86_64-linux-gnu/samba/samba-bgqd --ready-signal-

May 15 08:36:29 CS6502 systemd[1]: Starting Samba SMB Daemon...
May 15 08:36:29 CS6502 update-apparmor-samba-profile[7098]: grep: /etc/apparmo
May 15 08:36:29 CS6502 update-apparmor-samba-profile[7101]: diff: /etc/apparmo
May 15 08:36:30 CS6502 systemd[1]: Started Samba SMB Daemon.
lines 1-22/22 (END)
```

The output shows that the **smbd** service is enabled and running.

Make sure that smbd is enabled at startup

sudo systemctl enable smbd



```
user@CS6502:~$ sudo systemctl enable smbd
Synchronizing state of smbd.service with SysV service script with /lib/system
Executing: /lib/systemd/systemd-sysv-install enable smbd
```

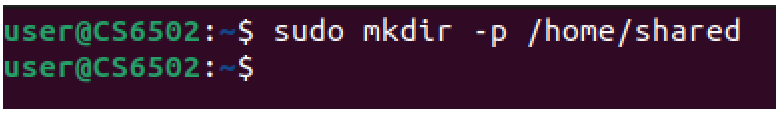
Step 2: Create a Shared Directory

To share files with Samba, create a directory containing files for sharing.

Use mkdir **-p** to create the directory under **/home**

For example, make a directory called ***sharing*** with:

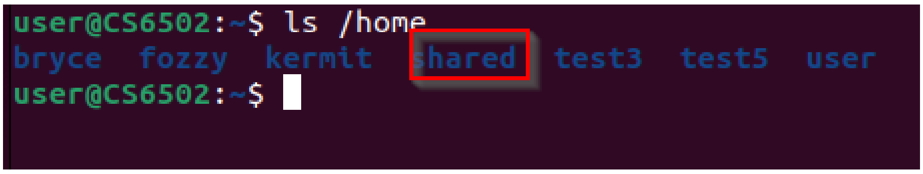
sudo mkdir -p /home/shared



```
user@CS6502:~$ sudo mkdir -p /home/shared
user@CS6502:~$
```

2. Use ls to verify the outcome.

ls /home



```
user@CS6502:~$ ls /home
bryce  fozzy  kermit  shared  test3  test5  user
user@CS6502:~$
```

Step 3: Configure Samba's Global Options

Configure Samba by editing the ***smb.conf*** file located in */etc/samba/smb.conf*.

Edit the file

sudo vi /etc/samba/smb.conf

Next, scroll down to the ***Global Settings*** section.

The key parameters to consider are in the following subsections:

Browsing/Identification

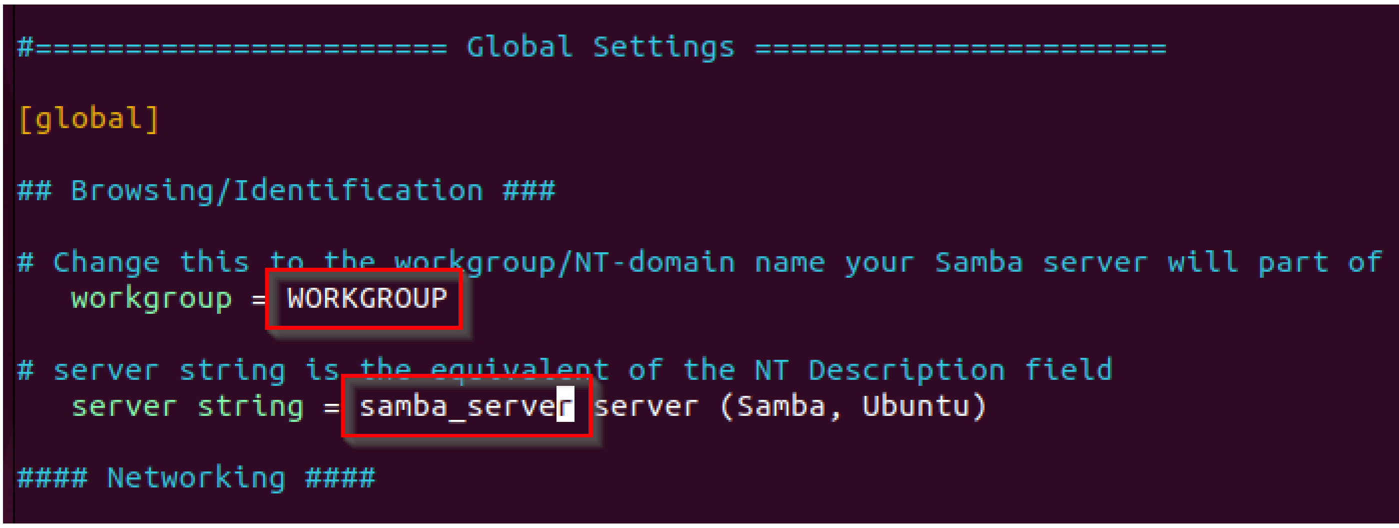
The ***Browsing*** subsection contains the **workgroup** and **server string** parameters:

- The **workgroup** parameter enables file sharing between a group of computers over a local area network. Ensure the **workgroup** settings correspond to the ones on Windows.
- The **server string** setting identifies the Samba server. In our example, the server is named *samba_server*.

Note: To set the workgroup settings on Windows 10, open the **Control Panel** and access the **System and Security** settings. The **workgroup** parameter is under the *System* section.
To configure the identification settings, uncomment the **workgroup** and **server string** parameters and add these values:

workgroup = WORKGROUP

server string = samba_server



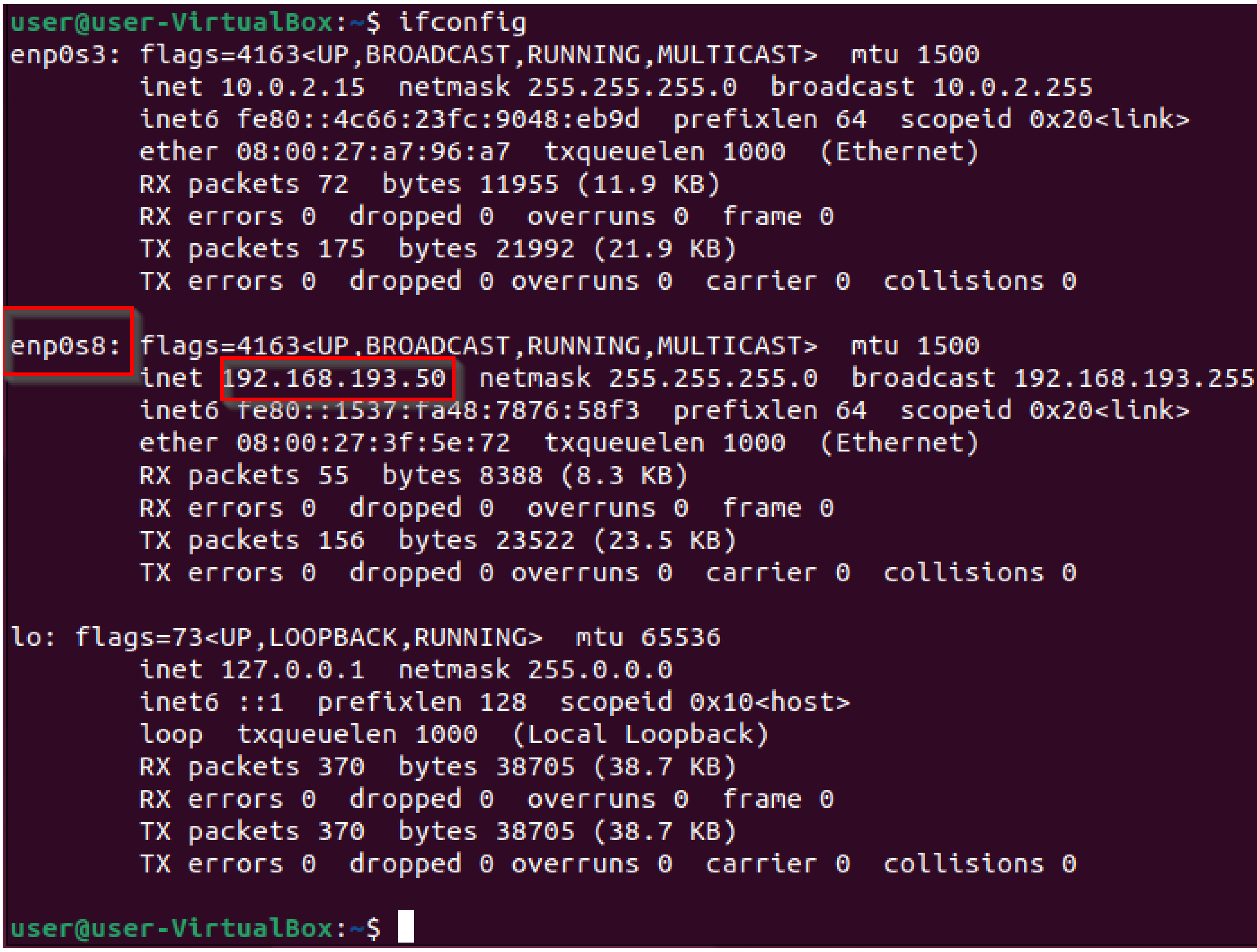
```
#======================= Global Settings =======================

[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba server will part of
   workgroup = WORKGROUP

# server string is the equivalent of the NT Description field
   server string = samba_server server (Samba, Ubuntu)

#### Networking ####
```

Networking

Use the ***Networking*** subsection to configure network interfaces that Samba binds to. ***Networking*** contains two parameters:

- The first parameter, **interfaces**, sets the network interface for Samba to bind to.
- The second parameter, **bind interfaces only**, ensures that Samba only binds to the interfaces listed in the file. The parameter should always be set to **yes**.

To set the **interfaces** parameter, first check the available interfaces with the ifconfig



```
user@user-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::4c66:23fc:9048:eb9d  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a7:96:a7  txqueuelen 1000  (Ethernet)
        RX packets 72  bytes 11955 (11.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 175  bytes 21992 (21.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.193.50  netmask 255.255.255.0  broadcast 192.168.193.255
        inet6 fe80::1537:fa48:7876:58f3  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:3f:5e:72  txqueuelen 1000  (Ethernet)
        RX packets 55  bytes 8388 (8.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 156  bytes 23522 (23.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 370  bytes 38705 (38.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 370  bytes 38705 (38.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

user@user-VirtualBox:~$ 
```

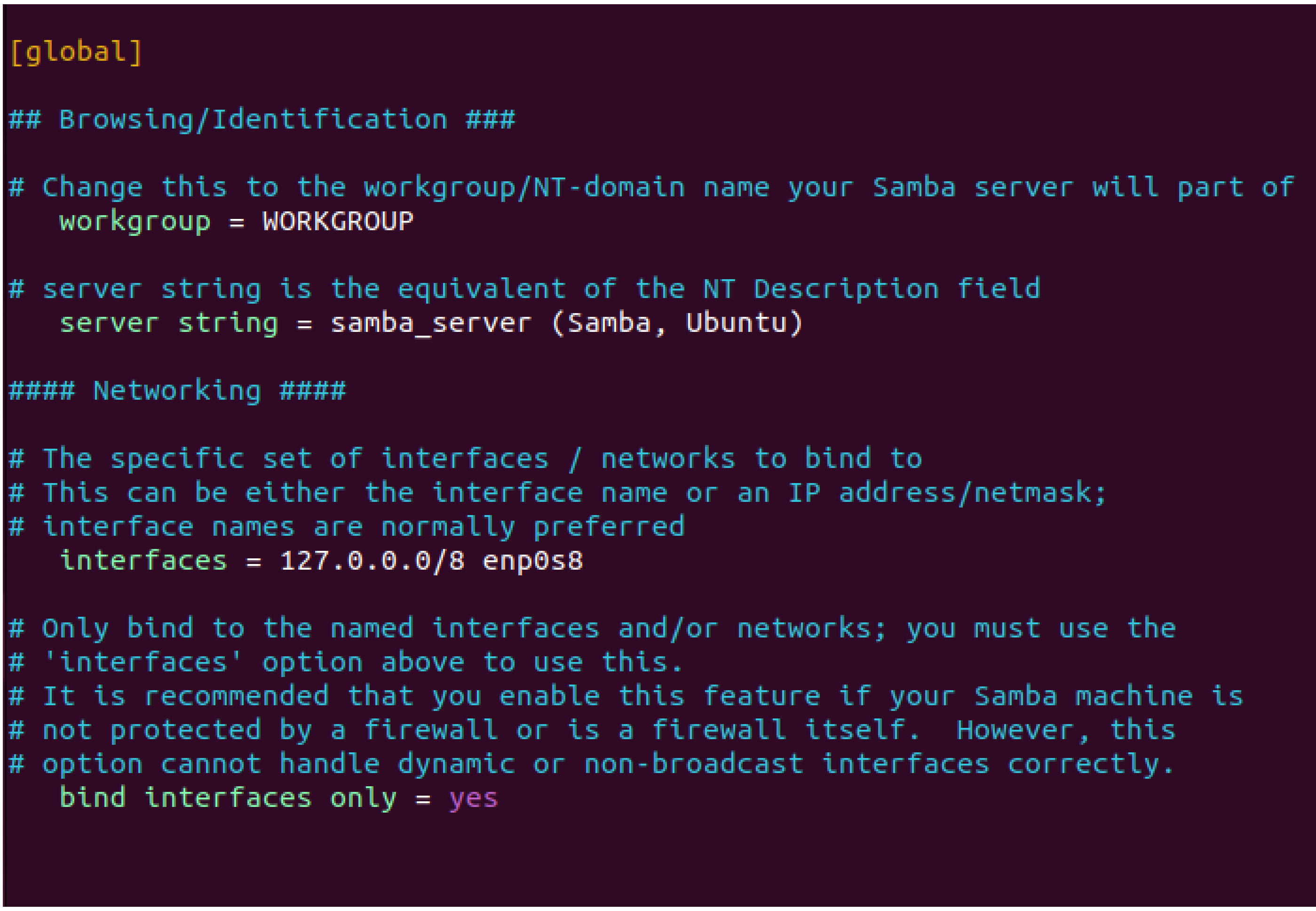
The example output indicates Samba need to bind to two interfaces: ***lo***, the loopback interface, and ***enp0s8***, the network interface.

For example, in this case, the settings are:

interfaces = lo enp0s8

bind interfaces only = yes

Note: The network interfaces Samba binds to may differ from one machine to another.



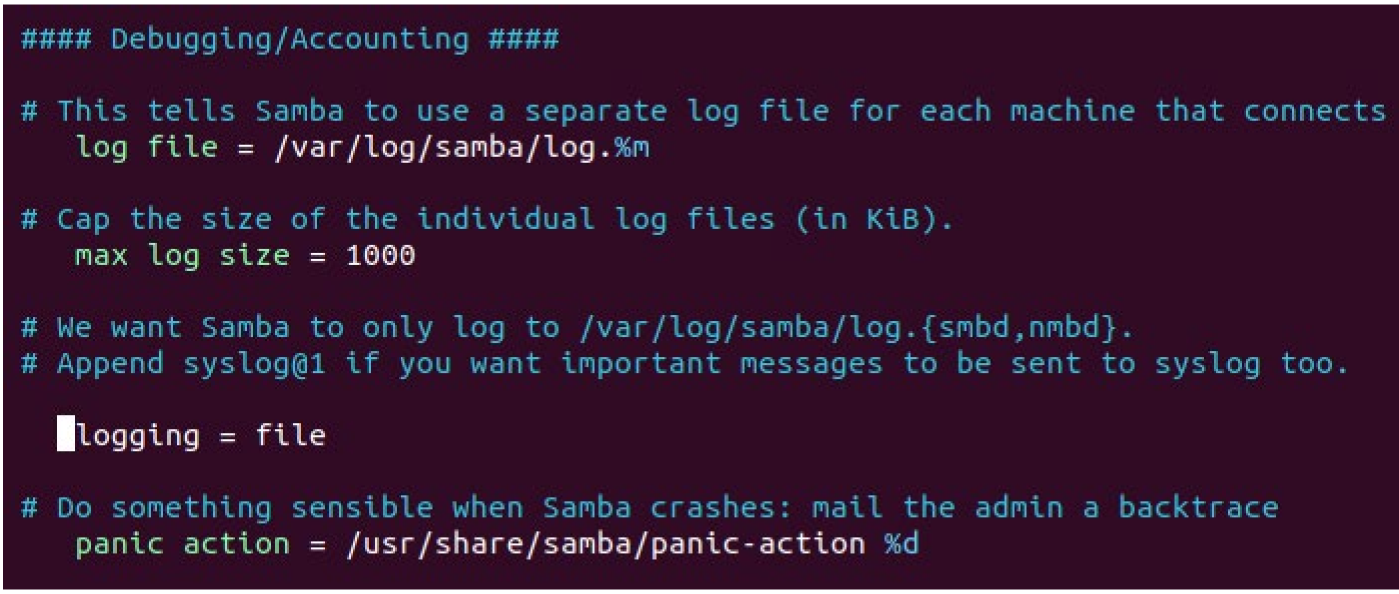
```
[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba server will part of
   workgroup = WORKGROUP

# server string is the equivalent of the NT Description field
   server string = samba_server (Samba, Ubuntu)

#### Networking ####

# The specific set of interfaces / networks to bind to
# This can be either the interface name or an IP address/netmask;
# interface names are normally preferred
   interfaces = 127.0.0.0/8 enp0s8

# Only bind to the named interfaces and/or networks; you must use the
# 'interfaces' option above to use this.
# It is recommended that you enable this feature if your Samba machine is
# not protected by a firewall or is a firewall itself.  However, this
# option cannot handle dynamic or non-broadcast interfaces correctly.
   bind interfaces only = yes
```

Debugging

The ***Debugging*** subsection has four parameters.

Set them as follows:

log file = /var/log/samba/log.%m
max log size = 1000
logging = file
panic action = /usr/share/samba/panic-action %d



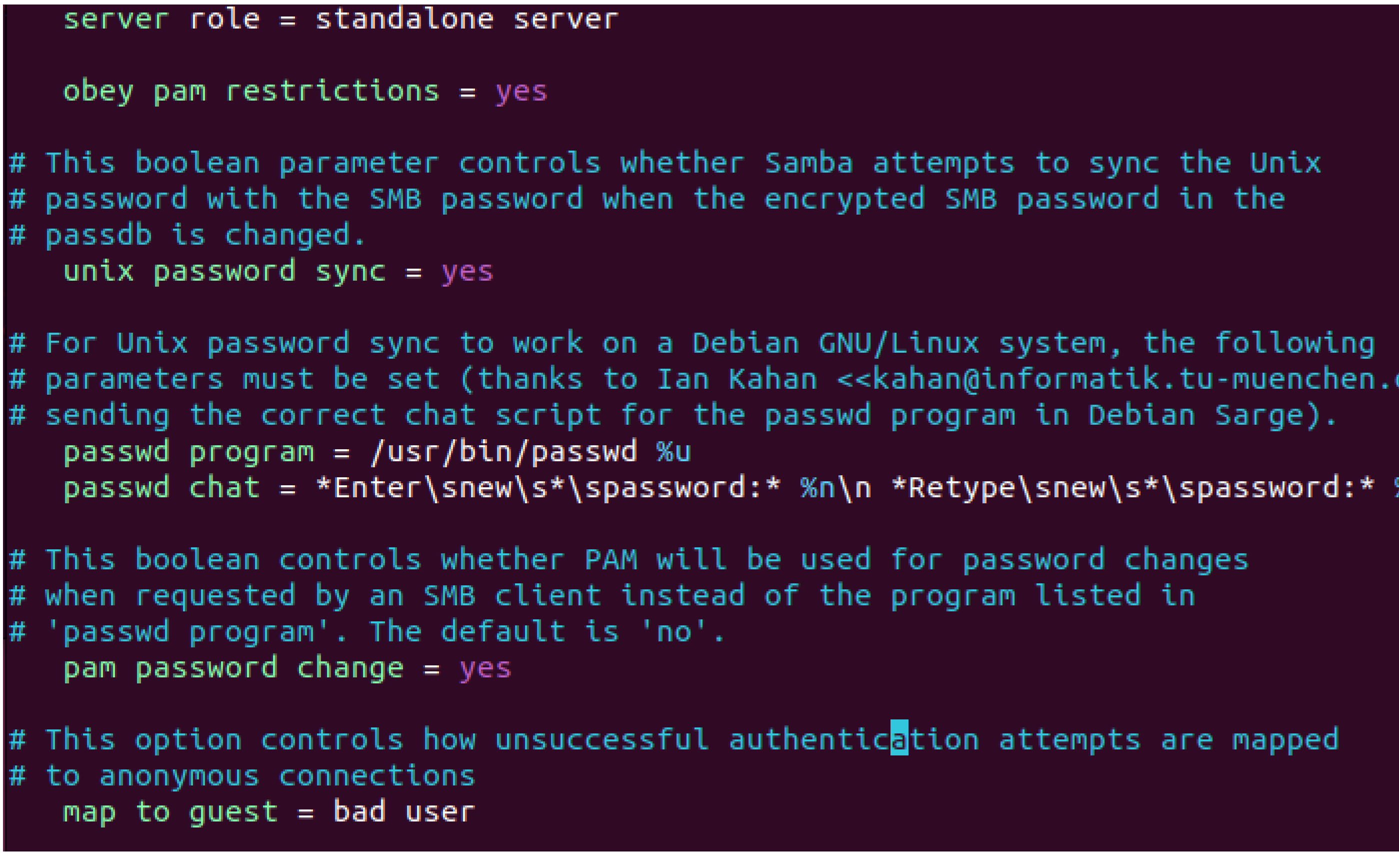
```
#### Debugging/Accounting ####

# This tells Samba to use a separate log file for each machine that connects
   log file = /var/log/samba/log.%m

# Cap the size of the individual log files (in KiB).
   max log size = 1000

# We want Samba to only log to /var/log/samba/log.{smbd,nmbd}.
# Append syslog@1 if you want important messages to be sent to syslog too.

   logging = file

# Do something sensible when Samba crashes: mail the admin a backtrace
   panic action = /usr/share/samba/panic-action %d
```

Authentication, Domain, and Misc

The most significant ***Authentication*** parameter is **server role**. This parameter determines the server type for Samba.

Set Samba as a standalone server:

server role = standalone server



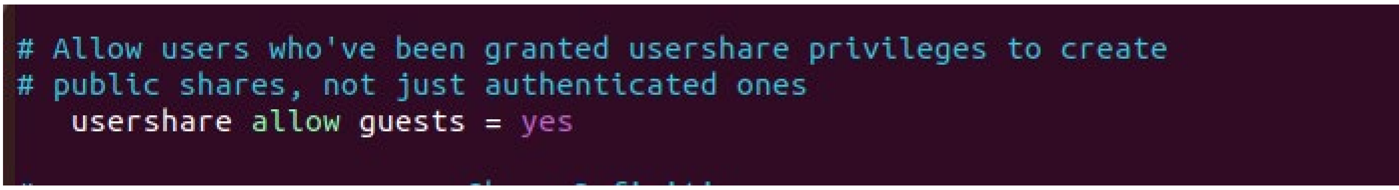
```
   server role = standalone server

   obey pam restrictions = yes

# This boolean parameter controls whether Samba attempts to sync the Unix
# password with the SMB password when the encrypted SMB password in the
# passdb is changed.
   unix password sync = yes

# For Unix password sync to work on a Debian GNU/Linux system, the following
# parameters must be set (thanks to Ian Kahan <<kahan@informatik.tu-muenchen.
# sending the correct chat script for the passwd program in Debian Sarge).
   passwd program = /usr/bin/passwd %u
   passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:*

# This boolean controls whether PAM will be used for password changes
# when requested by an SMB client instead of the program listed in
# 'passwd program'. The default is 'no'.
   pam password change = yes

# This option controls how unsuccessful authentication attempts are mapped
# to anonymous connections
   map to guest = bad user
```

The following is an extensive list of other authentication settings:

obey pam restrictions = yes
unix password sync = yes
passwd program = /usr/bin/passwd
passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:* %n\n *password\supdated\ssuccessfully* .
pam password change = yes
map to guest = bad user

Do not change any settings in the **Domain** subsection, but scroll down to **Misc** and set the following:

usershare allow guests = yes



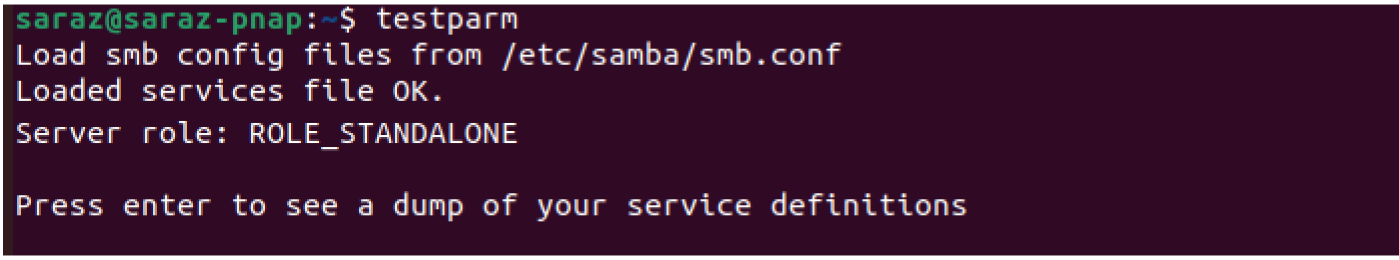
```
# Allow users who've been granted usershare privileges to create
# public shares, not just authenticated ones
   usershare allow guests = yes
```

Keep all other ***Global Settings*** unchanged.

Save and exit

Run the Samba utility **testparm** to check for syntax errors:

testparm



```
saraz@saraz-pnap:~$ testparm
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Server role: ROLE_STANDALONE

Press enter to see a dump of your service definitions
```

The output shows the ***Loaded services file OK*** message, which signifies no syntax errors. With ***Global Settings*** configured, the Samba server is ready to use.
Still, not configuring the users and the directory limits the Samba functionality.

Step 4: Set Up a User Account

To create a user account, set a username and password with:

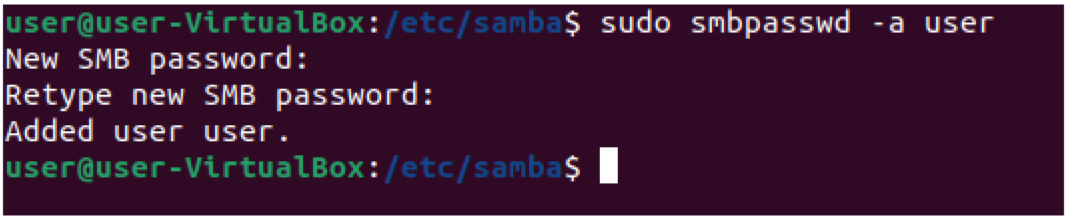
sudo smbpasswd -a <username>

Note that the username should belong to a system user.

For instance, in this example, the system account on the Ubuntu system is **user**.

Hence, the username is the same:

sudo smbpasswd -a user

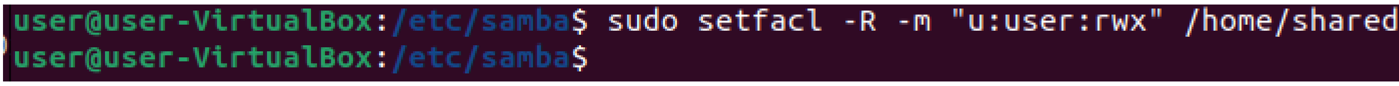


```
user@user-VirtualBox:/etc/samba$ sudo smbpasswd -a user
New SMB password:
Retype new SMB password:
Added user user.
user@user-VirtualBox:/etc/samba$ 
```

Next, users need to have read, write and execute access to the *sharing* directory.

To grant read, write, and execute permissions to the *sharing* directory, run **setfacl**:

sudo setfacl -R -m "u:user:rwx" /home/shared



```
user@user-VirtualBox:/etc/samba$ sudo setfacl -R -m "u:user:rwx" /home/shared
user@user-VirtualBox:/etc/samba$
```

https://www.golinuxcloud.com/setfacl-getfacl-command-in-linux/

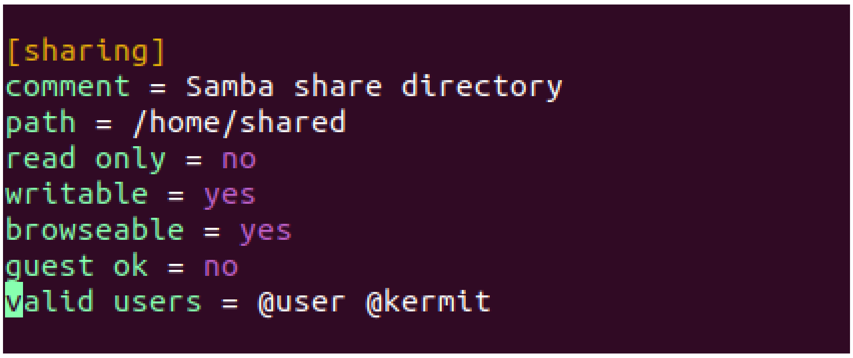
The command doesn’t produce any output.

Step 5: Configure Samba Share Directory Settings

Access the configuration file once again to add the previously made *sharing* directory.

Go to the end of the file and add:

[sharing]
comment = Samba share directory
path = /home/shared
read only = no
writable = yes
browseable = yes
guest ok = no
valid users = @user @kermit



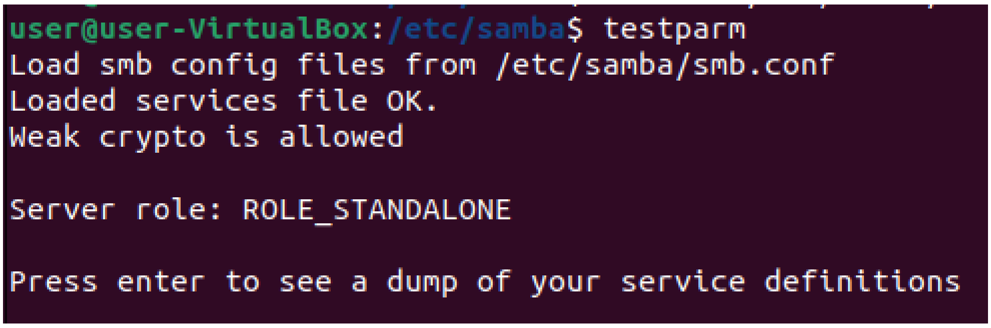
```
[sharing]
comment = Samba share directory
path = /home/shared
read only = no
writable = yes
browseable = yes
guest ok = no
valid users = @user @kermit
```

Each line grants specific permissions to access the directory. For instance:

- **[sharing]** Represents the parameters for sharing.
- **comment.** Serves as a shared directory description.
- **path.** This parameter specifies the shared directory location. The example uses a directory in */home*, but users can also place the shared files under */samba*.
- **read only.** This parameter allows users to modify the directory and add or change files when set to **no**.
- **writeable.** Grants read and write access when set to **yes**.
- **browseable.** This parameter allows other machines in the network to find the Samba server and Samba share when set to **yes**. Otherwise, users must know the exact Samba server name and type in the path to access the shared directory.
- **guest ok.** When set to **no**, this parameter disables guest access. Users need to enter a username and password to access the shared directory.
- **valid users.** Only the users mentioned have access to the Samba share.

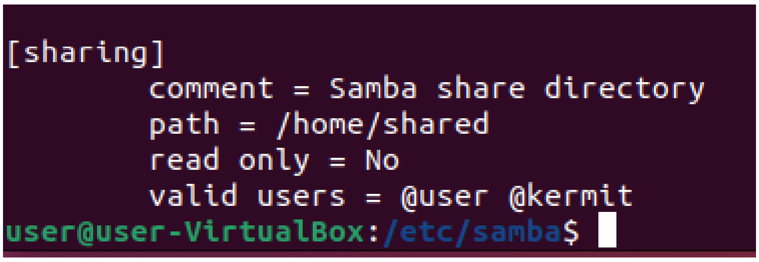
Save the changes and exit the file.

Rerun **testparm**:



```
user@user-VirtualBox:/etc/samba$ testparm
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
```

The output confirms that the Samba is adequately configured. For a more verbose output, hit **enter**:



```
[sharing]
        comment = Samba share directory
        path = /home/shared
        read only = No
        valid users = @user @kermit
user@user-VirtualBox:/etc/samba$
```

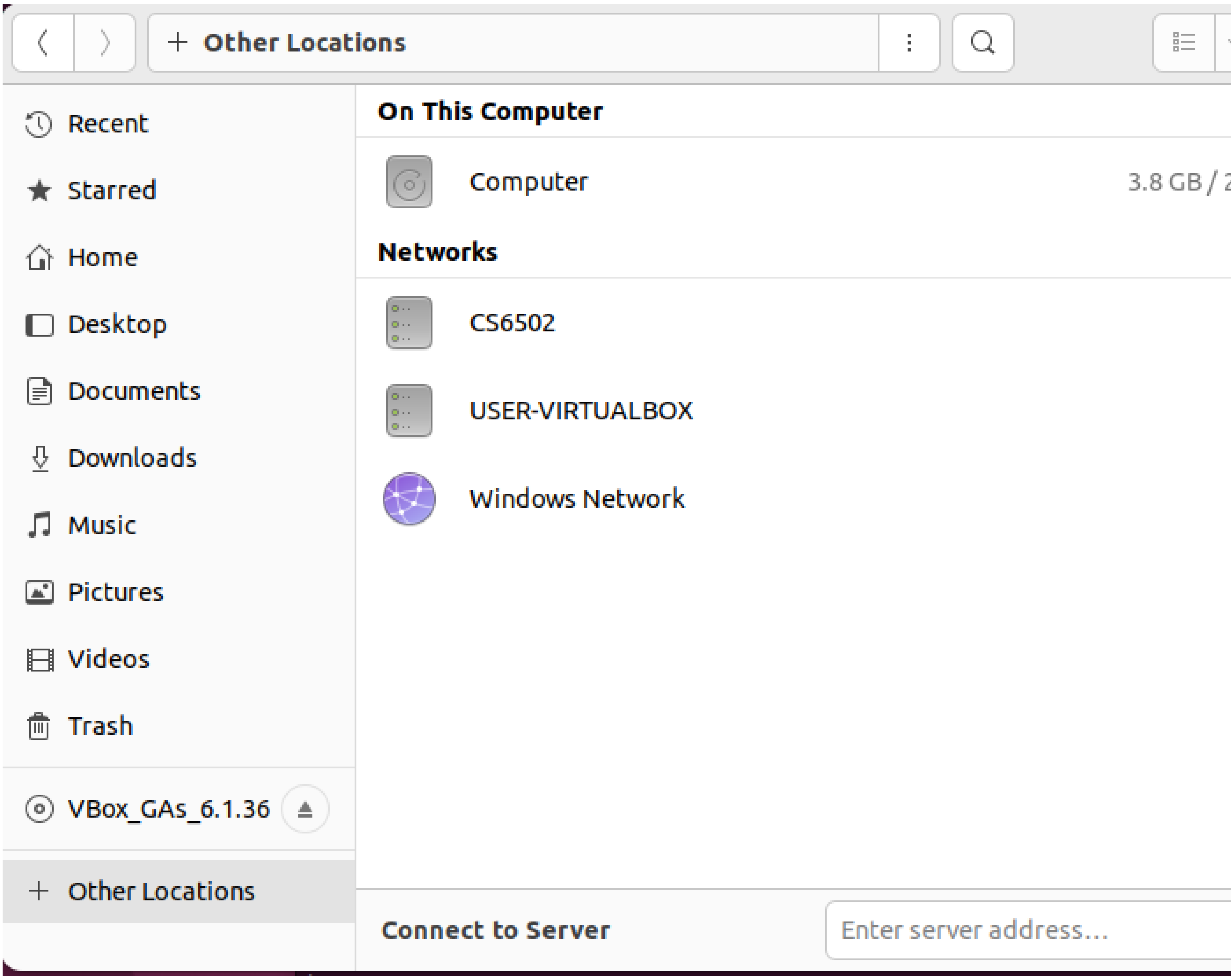
Step 6: Connect to the Shared Directory

Before connecting to the Samba server, restart the services with:

```
sudo systemctl restart smbd
```

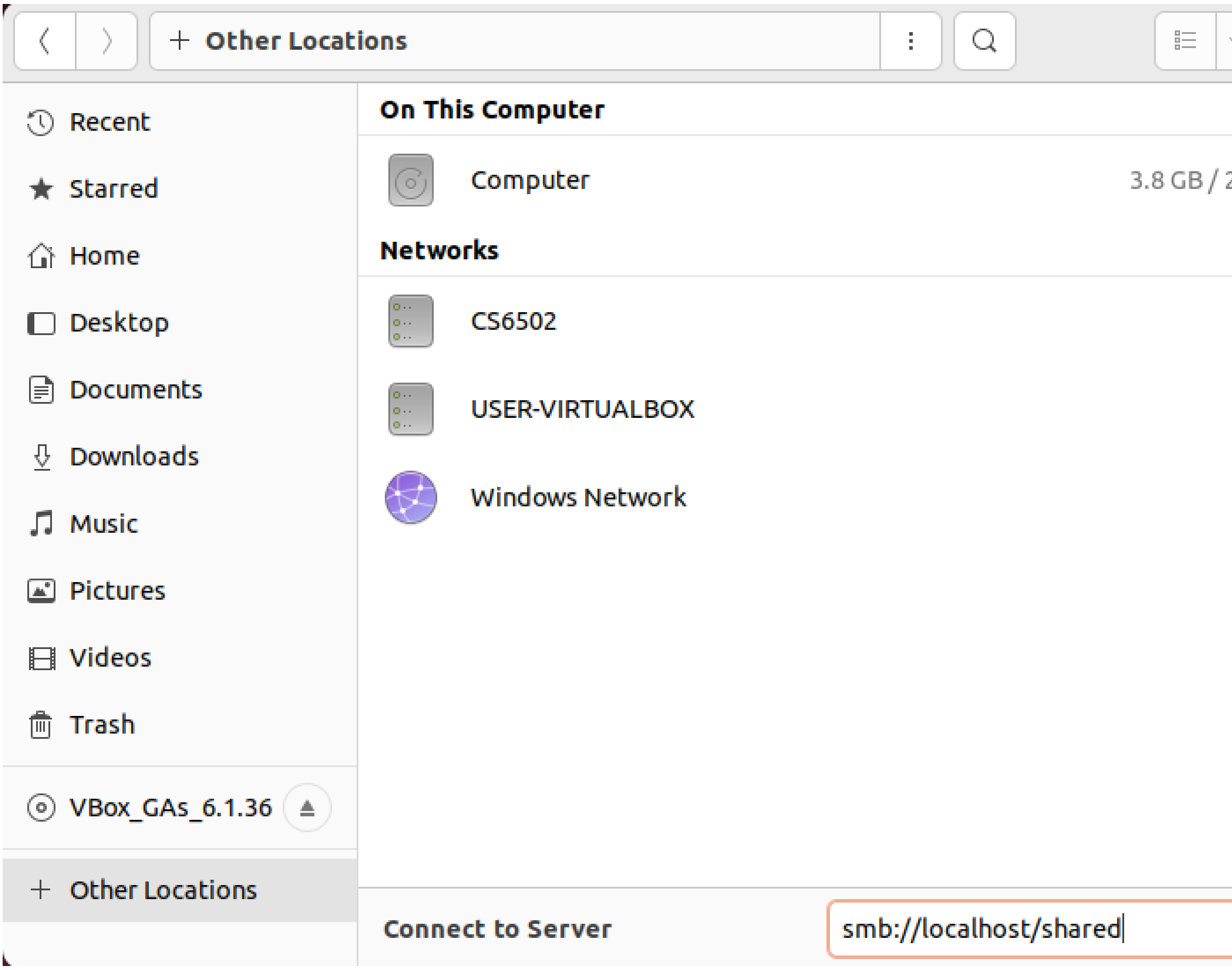
The command prints no output.

To connect to the shared directory via GUI, access the default file manager and choose the **Other Locations** option:



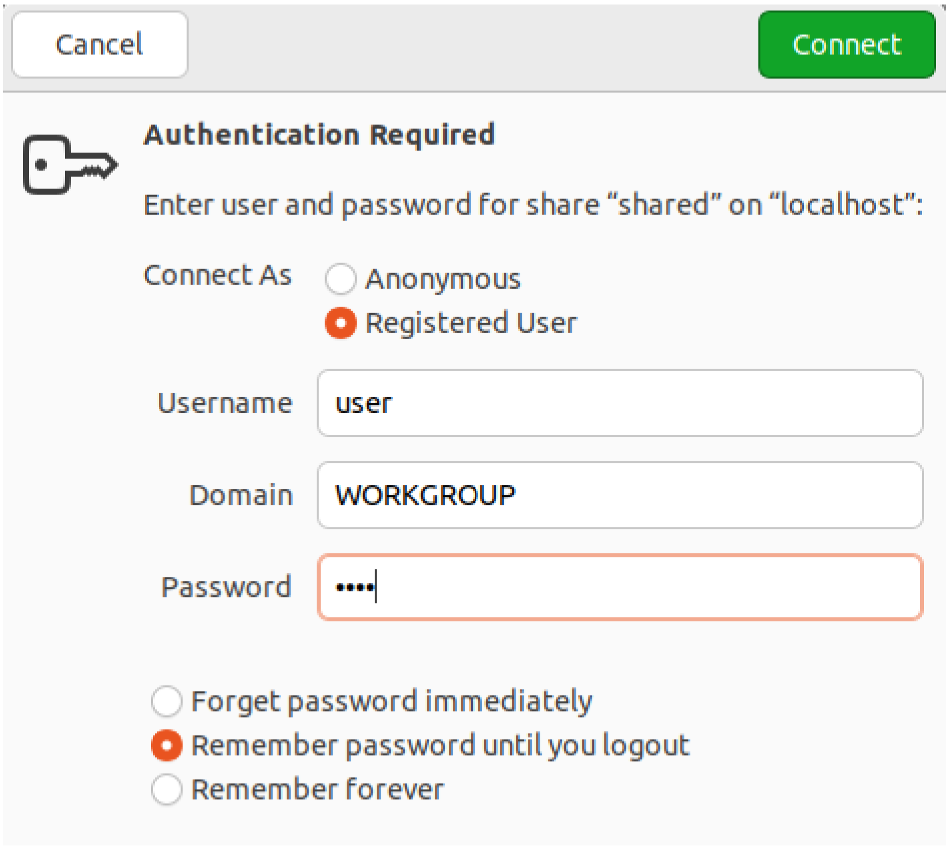
Type the following into the **Enter server address...** box and select **Connect**:

```
smb://ip-address/sharing
```



smb://localhost/shared

The system asks for a **Username** and **Password**. Provide the requested info and click **Connect** again:



Authentication Required

Enter user and password for share “shared” on “localhost”:

Connect As ○ Anonymous
● Registered User

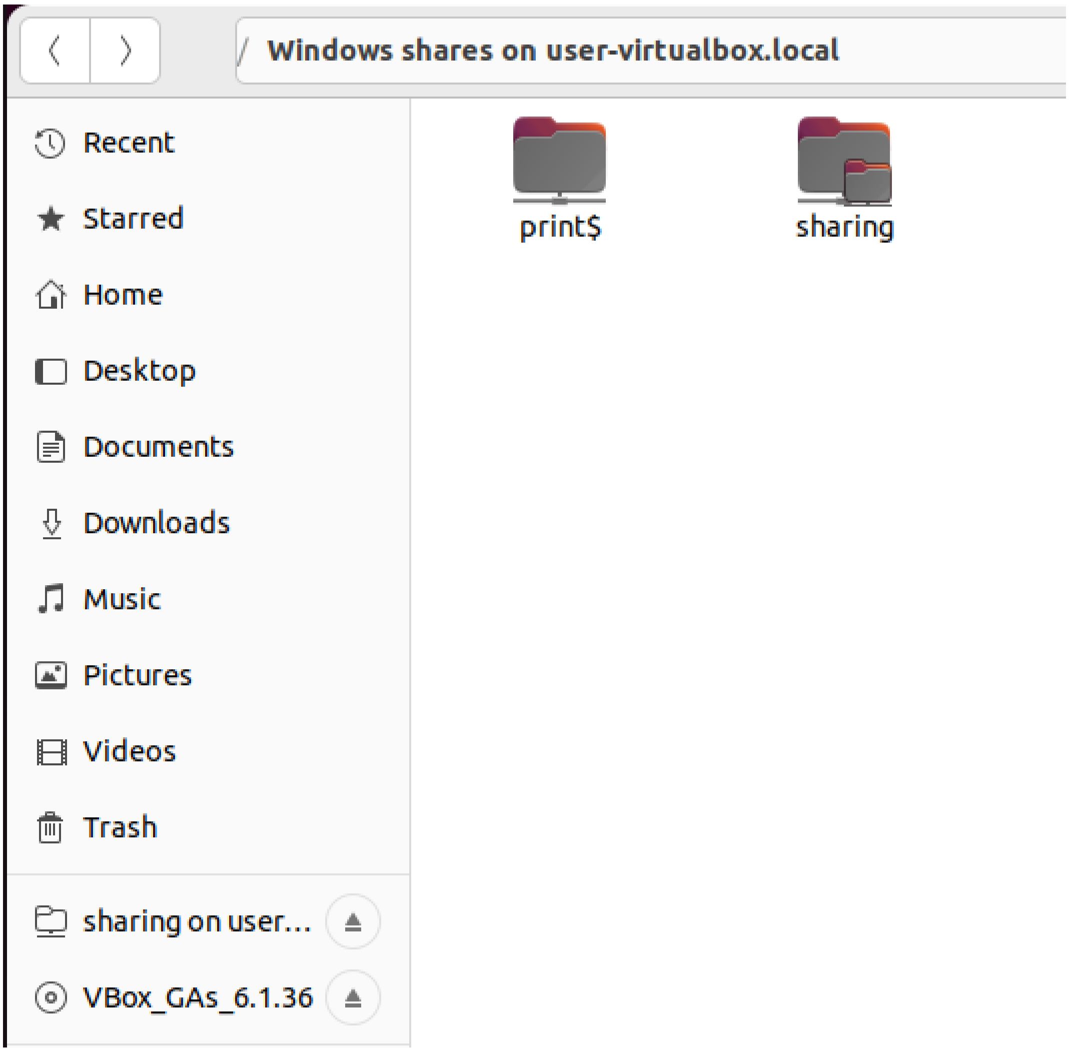
Username: user

Domain: WORKGROUP

Password: ••••

○ Forget password immediately
● Remember password until you logout
○ Remember forever

5. This adds the *sharing* directory to the *Windows shares* location:

Windows shares on user-virtualbox.local
Recent
Starred
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
sharing on user...
VBox_GAs_6.1.36
print$
sharing