How to make a SMB server using Raspberry PI

This document will guide a beginner understand how to create a server using Raspberry pi and the necessary background knowledge one should have before someone starts the project.

The videos and links shown below helps even an “amateur” understand the very basics of how to proceed.

Basics to understand before starting this project:

Introduction to File Sharing between machines:

https://www.youtube.com/watch?v=uFQhawnWOrI

Ownership in a Linux File system:

https://www.youtube.com/watch?v=P3DHLMEU5lo

Chronjob:

https://www.youtube.com/watch?v=rErAOjACT6w

By following the below given links and videos one can easily implement SMBServer on the Raspberry pi.

Websites:

The below websites provide a basic very clear explanation on how to proceed:

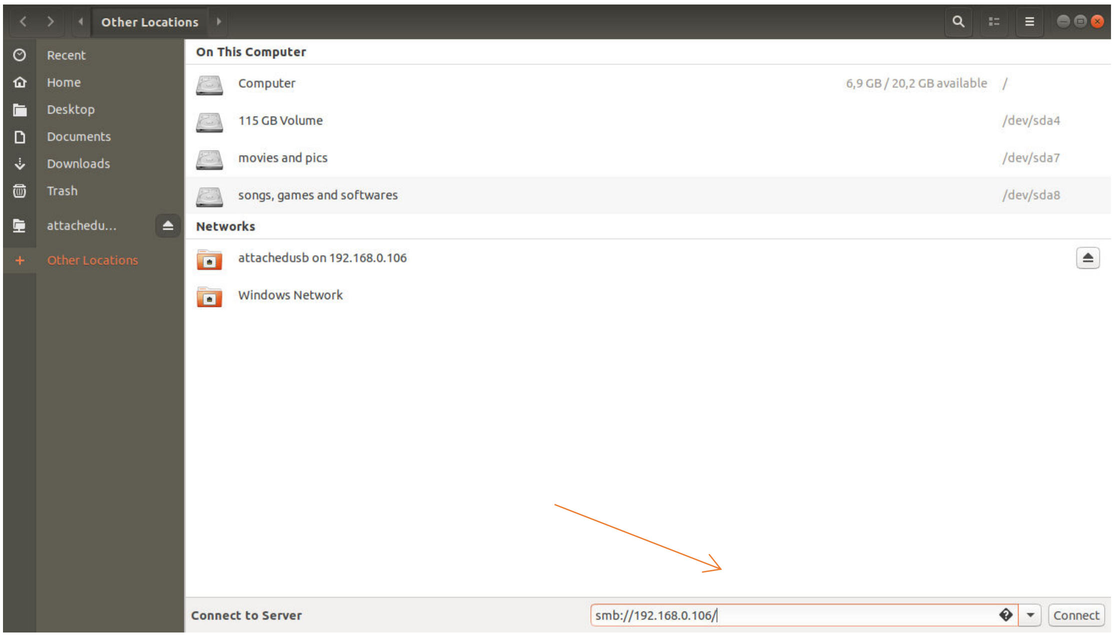
https://magpi.raspberrypi.org/articles/samba-file-server

https://raspberrytips.com/raspberry-pi-file-server/

Video:

https://www.youtube.com/watch?v=0vEK4A8FOX0

Connect to the SMB Server on Raspberry pi using a Ubuntu 18.04 Machine can be done as shown in the below picture



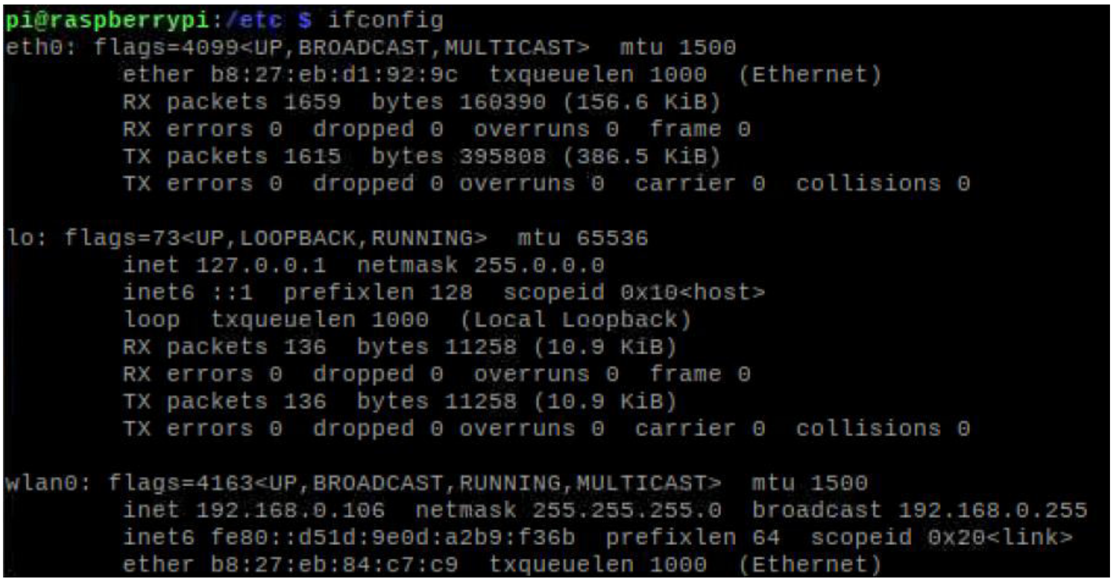
Connect to Server: smb://192.168.0.106/

Once you have implemented your SMB server in Raspberry pi, Raspberry pi goes offline from the Wi-Fi connection at regular intervals of time.

Solution:

Due to this reason we need to set verify the network connection in the PI at regular intervals of time.

1. It can be seen that “**ifconfig**” command shows 3 interfaces(Physical Interfaces present on Raspbery PI) and we are bothered only about “wlan0” which represents Wi-Fi connection interface.



```
pi@raspberrypi:/etc $ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether b8:27:eb:d1:92:9c  txqueuelen 1000  (Ethernet)
        RX packets 1659  bytes 160390 (156.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1615  bytes 395808 (386.5 KiB)
        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 136  bytes 11258 (10.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 136  bytes 11258 (10.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.106  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::d51d:9e0d:a2b9:f36b  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:84:c7:c9  txqueuelen 1000  (Ethernet)
```

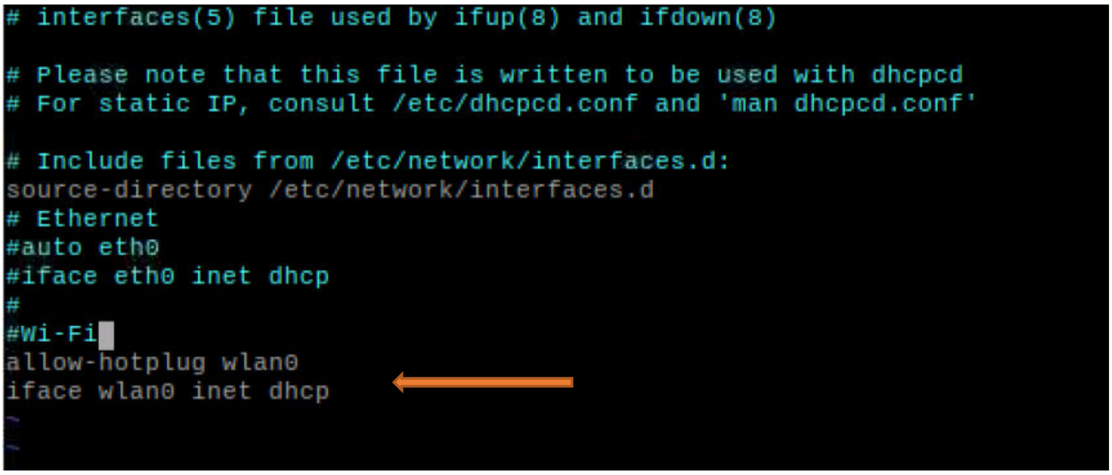
2. Enter the following commands:

cd /etc/network

Sudo vim interfaces

3. Add the following lines for Wi-fi. If you have some other “interface name” other than “wlan0”. Replace wlan0 with that interface name accordingly.

To know more about interfaces file, please visit: https://linuxconfig.org/etcnetworkinterfacesto-connect-ubuntu-to-a-wireless-network



```
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d
# Ethernet
#auto eth0
#iface eth0 inet dhcp
#
#Wi-Fi
allow-hotplug wlan0
iface wlan0 inet dhcp
~
~
```

4. One should create a bash script (using vim, gedit, etc ...) to check the wifi connectivity for a regular intervals of time. This is provided in the following link:

https://raspberrypi.stackexchange.com/questions/13473/connected-to-the-internet-but-cant-ssh-or-ping

```
#!/bin/bash

#Script to check the network connection


#Check network connection

if /sbin/ifconfig wlan0 | grep -q "inet addr:" ; then

    #Connection is good; do nothing

    echo "$(date "+%D [%H:%M:%S]") Connection is up"

else

    echo "$(date "+%D [%H:%M:%S]") Network connection down. Attempting to reconnect..."

    sudo /sbin/ifup --force wlan0

fi
```

5. Save the above script as as network-monitor.sh

6. Make it file executable using “chmod” command

```
Sudo chmod +x ./network-monitor.sh
```

7. Run the net work-monitor.sh executable now

```
./network-monitor.sh
```

8. **Almost done!!! Now this script needs to be run at regular intervals of time to keep the Wi-Fi network in check. This can be done by using chrontab.**

https://www.youtube.com/watch?v=rErA0jACT6w