An EndeavourOS based Simple Home Server.

SET UP A LINUX CLIENT COMPUTER

CONFIGURE SSH

ON A CLIENT LINUX COMPUTER with XFCE or other desk top environment, open a Terminal window and change to root. Use the static IP address you established for the server and the host name you decided on. Add the following line to the end of /etc/hosts. This will establish the route to the Server.

```
# vi or gedit  /etc/hosts
   192.168.0.150    enosServer.localdomain      enosServer
```

This establishes a relationship between the host name of enosServer and it's IP address, kind of like a DNS service. Otherwise you have have to address the server by it's IP address and port number, which can be awkward and hard to remember.

Since the SSH port that the enosServer listens on was changed from the default port 22, the Client computer needs to know which port to use.

```
# vi  /etc/ssh/ssh_config              (change the following line.)
      FROM  #Port 22          TO   Port 9XXX      (Use the same port # as in the server)
```

In the previous section, we already set up the SERVER's firewall to accept requests from any CLIENT computer on our LAN at Port 9XXX. This limits the amount of requests coming to the server. Once the request gets through the firewall, the service assigned to port 9XXX (SSH in this case) will check to see if the request is a valid SSD request. If the request is not a valid SSH request, SSH returns an error message. All necessary configuration to communicate from the Client to the Server is now in place.

CONFIGURE A FIREWALL

If you already have a firewall installed, skip this section. If not:

```
# pacman -S ufw                        (install ufw firewall)
# ufw status                           (check status of ufw)
  Status: inactive
# ufw logging off           (otherwise logging appears on screen & makes a mess of DMESG)
# ufw default deny          ( denies all incoming UNSOLICITED traffic, solicited traffic passes)
# ufw enable
# systemctl enable ufw.service
# systemctl start ufw.service
# reboot
# ufw status
  Status: active
```

Connect to enosServer with SSH

In a terminal window, as a user try to connect to enosServer

```
# su username                   (if you happen to be at root)
$ ssh pshare@enosServer
```

If you successfully communicated with enosServer, you will get something similar to:

```
The authenticity of host '[enosserver]:9XXX ([192.168.0.XXX]:9XXX)' can't be established.
ECDSA key fingerprint is 54:fa:20:25:c1:91:d3:3d:4c:8c:47:02:32:f2:5e:8e.
Are you sure you want to continue connecting (yes/no)?
```

The above dialog is a one time thing. Type in “yes”, the connection should be completed and you will be asked for pshare's password. You should then have a terminal prompt of
[pshare@enosServer ~]$
You are now logged into enosServer as user pshare. Pay attention to the prompt, it will always let you know which computer you are in and the user name. Remember we set up the SSH server on enosServer to NOT allow login as root. If you need to use root for administration, type in su and enter your server's root password. Anything you type in the Terminal window will now be executed in the enosServer computer.

Type in

```
# exit       (if you are in as root)
$ exit       (exit again as user.  This is the proper way to log out of a network connection.)
```

You should be back to your client computers prompt.

On the LINUX CLIENT COMPUTER, do not terminate your enosServer session by clicking on the X (close) button at the top right of your terminal window. After properly disconnecting from the enosServer you should return to your local prompt in the terminal window. Now you can close your Client side terminal window by typing in exit once more.

SSH into enosServer and bypass the password

This is all and good, but it is a pain to have to enter the password all the time. Let's use the SSH keys we generated with ssh-keygen in the server. Now ssh keys may need to be generated in our client computer as user.

```
$ ll .ssh
   -rw-------. 1 don don 1831 Jun  8  2019 id_rsa
   -rw-r--r--. 1 don don  406 Jun  8  2019 id_rsa.pub
```

If the ll command produces the above results, you already have ssh keys. You can skip the instructions in the box. If you don't have ssh keys, enter the following commands

```
$ ssh-keygen -t rsa -b 2048   (create  ssh keys for user, hit enter 3 times for defaults )
$ ls -al                      (  .ssh should be drwx------  if not  $ chmod 700 .ssh  then recheck)
  drwx------  don don  4096 Jan  5  1309  .ssh
$ ll .ssh
      -rw-------. 1 pshare pshare 1831 Jun  8  2019 id_rsa
      -rw-r--r--. 1 pshare pshare  406 Jun  8  2019 id_rsa.pub
```

The id_rsa file is your PRIVATE SSH Key and you should never do anything with it. Don't copy, move, or otherwise mess with it. Just leave it alone. The id_rsa.pub file is your PUBLIC SSH Key. We need to export your PUBLIC SSH Key to the enosServer.

As user

```
$ ssh-copy-id  -i  ~/.ssh/id_rsa.pub  pshare@enosServer
  enter pshare's password when requested.
```

You will be returned to your client's prompt. Now try to log into enosServer

```
$ ssh pshare@enosServer
```

You should now be logged into enosServer as pshare without having to enter your password. To do administration, just ssh into the Server from a client terminal window Once in enosServer change to root and do most anything you want from a nice GUI terminal window with mouse, scroll bars, cut and paste, etc. The server and client are now configured to remotely administer the server.

```
[pshare@enosServer ~]$ exit          repeat exit until back in the client computer.
```

Use FUSE and SSHFS to view Server data in a file manager

Now to tie this up and make this really neat and convenient to use, we will set up FUSE and sshfs to utilize a Thunar (or other file manager) window to access the files on the enosServer**:**/server partition.

IN THE CLIENT COMPUTER as root install sshfs.

```
# pacman -S sshfs
```

IN THE CLIENT COMPUTER, in a Terminal window as user

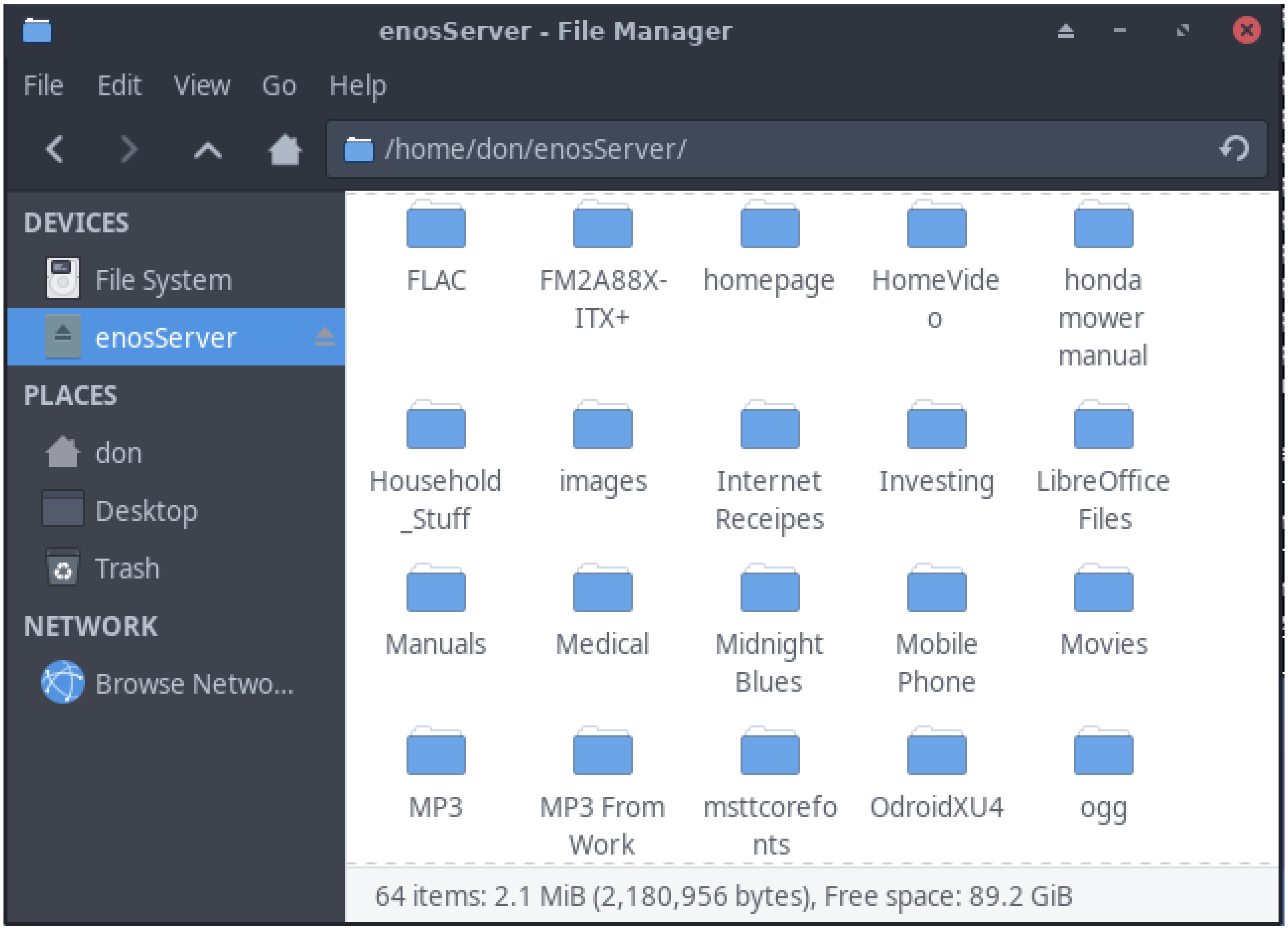
```
# su username
$ cd                          (ensure you are in user's home directory)
$ mkdir  enosServer        (make a mount point for the enosServer)
```

The /home/$USER/enosServer directory is a Mount Point for the enosServer. You should never add any directories or files locally in this directory. If enosServer is not mounted, the enosServer directory should be empty.

Drum Roll. Here is the part you've been waiting for. Still in the CLIENT LINUX COMPUTER As user

```
$ sshfs pshare@enosServer:/server  /home/$USER/enosServer
$
```

It should complete without any errors. Launch Thunar, then Click on Home and click on enosServer. You should see the file "test" that was created when the DATA SSD was added. Ideally, the enosServer folder should have nothing but sub folders to organize your data. Eventually as the enosServer folder is filled up,it could look something like this.



64 items: 2.1 MiB (2,180,956 bytes), Free space: 89.2 GiB

Also notice in the left column under the DEVICES, the enosServer Icon. Click on that, or click on username under PLACES then click on enosServer folder.

Under DEVICES the enosServer Icon also has a unmount icon at the right side. DO NOT try to unmount the enosServer by clicking on this.

sshfs uses the FUSE kernel module to fool your client computer into thinking the /server directory on enosServer is a local directory named /home/username/enosServer. Anything you can do on a local directory, you can now do on the remote computer's directory. This is how you will manage the content on the enosServer's DATA drive. Add files, delete files, add/delete folders just like you do on your local Drive. So, manage data files with sshfs, and do maintenance on the Operating System by using ssh pshare@enosServer. Once set up maintenance is usually just doing updates with pacman -Syu and data backups.

Use the following to unmount enosServer. In the Client computer type in the following:

```
$ fusermount -u /home/$USER/enosServer
```

Direct Thunar to the /home/username/enosServer directory, and it should be empty. Because there isn't a remote directory mounted there anymore.

AUTOMATE THE MOUNTING PROCESS

This is all fine and dandy, but who wants to type in these commands all the time, much less remember the exact Syntax for the commands? Let's automate this process some. Here are the instructions for XFCE, it should be doable in any Desktop Environment.

Mount enosServer with a desktop launcher

IN THE CLIENT COMPUTER as user in user's home directory

```
  $ ls -l
  drwx------. 2 username username 4096 Jan 31 20:34 bin
```

Look for a directory named bin. If bin doesn't appear, use mkdir to create one

```
$ mkdir bin
```

After creating bin, you may want to use

```
  $ chmod 700 bin
```

to change the permissions as above for security reasons.

```
 $ cd bin
```

Using your favorite text editor, create a file named AutoMountServer.sh and add these 3 lines

```
  $ vi AutoMountServer.sh
     #!/bin/bash       (the first two characters are # then an exclamation mark)
     sshfs pshare@enosServer:/server   /home/$USER/enosServer
     exit
```

```
 $ chmod 754 AutoMountServer.sh     (make AutoMountServer executable)
 $ ll
   -rwxr-xr- -   don don  47 Jan 8   AutoMountServer.sh
```

Open Thunar, and navigate to /home/user/bin
Right click on AutoMountServer.sh, then "Send to" then "Desktop (Create Link)"
There should be a launcher on the Desktop. To tidy things up, Right click on the desktop launcher, then Rename, and trim the launcher name to simply "Mount".

Reboot the computer. Open Thunar. It shouldn't show enosServer. Click on the desktop Icon "Mount" and enosServer should appear in Thunar under DEVICES.

View enosServer In Thunar with a desktop launcher

In /home/user/bin Make an additional file

```
  $ vi ViewServer.sh
    #!/bin/bash
    thunar /home/$USER/enosServer &
    exit
```

```
$ chmod 754 ViewServer.sh
$ ll
    -rxwr-xr--     ViewSherver.sh
```

In Thunar, navigate to /home/user/bin
Right click on ViewServer.sh, then "Send to" then "Desktop (Create Link)
Tidy up and change the Desktop Launcher name to ViewServer.
Close Thunar. Click on the desktop launcher and Thunar will open and go directly to the enosServer folder.

Automatically Mount Server at Log in

It would be nice to have AutoMountServer run every time you log in.

Go to "EndeavourOS" in the panel, click on "Settings", click on "Session and Startup"
Select "Application Autostart" and hit the +Add.
Name: MountServer
Description: Auto Mount enosServer
Command: Navigate to /home/$USER/bin/AutoMountServer.sh
Trigger: on login

Reboot, and enosServer should be automatically mounted at login. Extremely simple for any level of Linux user. Once set up for a user, all that user has to do is log in and enosServer is ready to use. Check Thunar and the enosServer should be mounted.

Unmount server with a desktop launcher

If unmounting the server is necessary, create a script UnMountServer.sh

```
 $ cd ~/bin
```

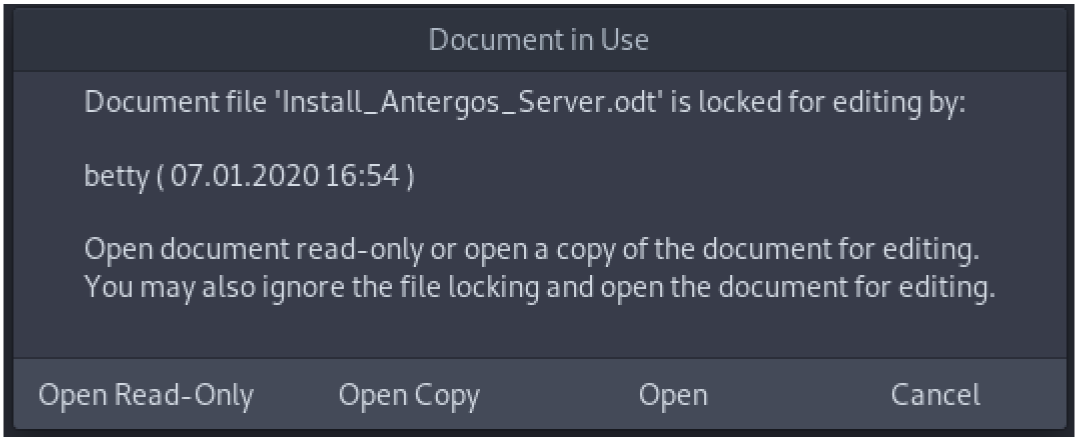
Using your favorite text editor, create a file named UnMountServer.sh and add these 3 lines

```
  $ vi UnMountServer.sh
     #!/bin/bash       (the first two characters are # then an exclamation mark)
     fusermount -u /home/$USER/enosServer
     exit
 $ chmod 754 UnMountServer     (make UnMountServer executable)
 $ ll   -rwxr-xr--        UnMountServer
```

In Thunar, navigate to /home/$USER/bin.
Right click on UnMountServer.sh, then "Send to", then "Desktop (Create Link)"
Tidy up and change the Desktop Launcher name to UnMount

Now if for any reason you need the server unmounted, double click on the UnMountServer desktop Icon. The nice thing about this is, if you have a file on the enosServer open, it will not unmount the server. That way if you have a file open in a text editor and then you click on UnMountServer, it will not unmount and loose any unsaved data in the text editor. All files on the enosServer must be closed to unmount the server.

Also, at least in LibreOffice, if someone else has a file open when you try to open it you get:



Document in Use

Document file 'Install_Antergos_Server.odt' is locked for editing by:

betty (07.01.2020 16:54)

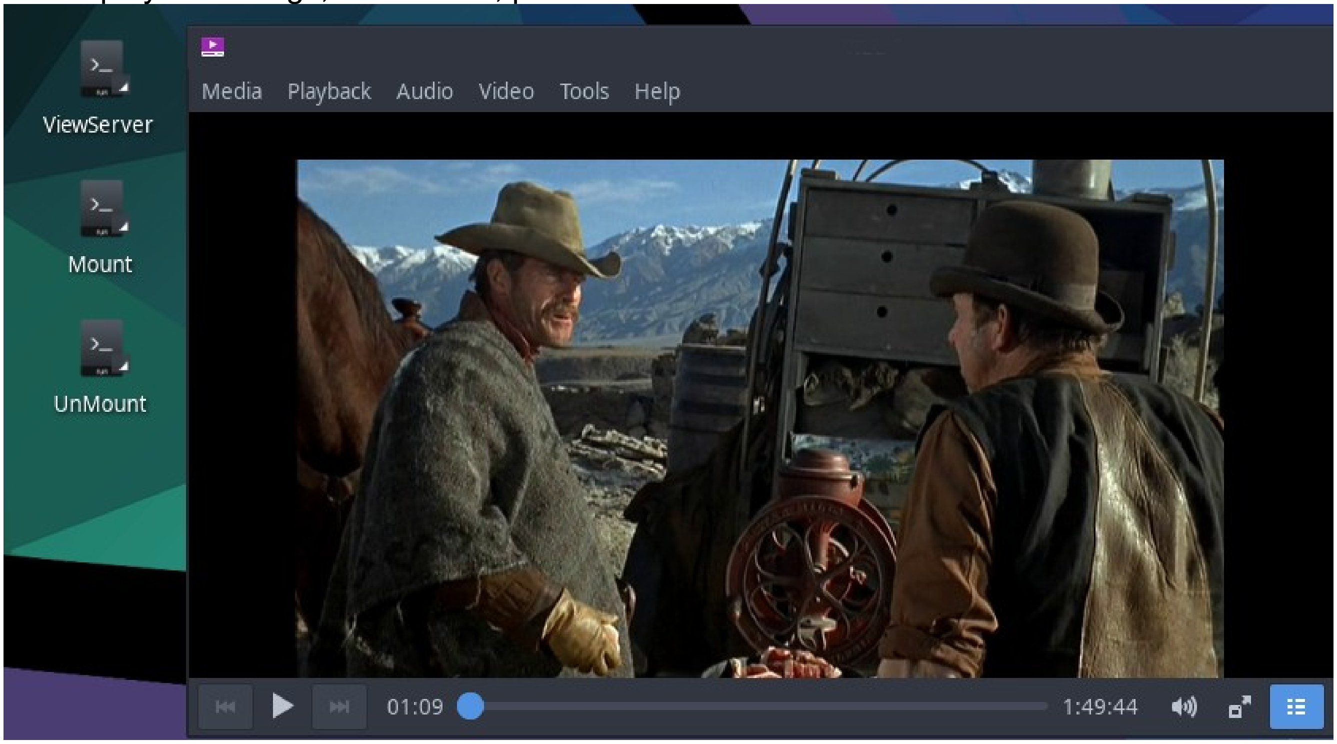
Open document read-only or open a copy of the document for editing.
You may also ignore the file locking and open the document for editing.

Try to avoid shutting down the computer with a file open on the server. Next time you try to open that file, you may get a similar warning as above saying the file is locked for editing by you. Improper shut downs can leave a lot of things hanging.

Now there are three Destop Launchers. If you change the setting in "Desktop" "Icons" and enable "Single Click to activate items" then you have single click access to manage your enosServer.

NOTE:
One thing I have noticed with Thunar. If you are in the /home/$USER/enosServer folder and the server is not mounted, it shows nothing as it should. If you then click on the desktop "Mount" Icon and mount the server, the enosServer Icon shows up under DEVICES but still nothing is shown. If you click on "View" then "Reload" the files appear. It says Ctrl+R is the short cut for Reload, but I can't get it to work. You can also in PLACES click on the username Icon then in DEVICES click on the enosServer icon and it shows the files. Another form of Reload I assume. Nothing major, just a quirk I ran across.

Below is a snippet of a screenshot in XFCE playing a movie on the enosServer with Parole media player. No lags, no artifacts, perfect.



If you do not want the server mounted at login, in the "Settings" "Session and Startup" Select "Application Autostart" and highlight MountServer and Unselect the check mark. Now the desktop Icon is used to mount the sever.

Configure any other Linux machines you have exactly like we did this one. Yes, all Linux Client machines will mount the server as user pshare. You could create different users on the server that match the user names of all users accessing the server. In fact, that is what you would want to do in an Enterprise environment.   But that can make administration a nightmare, plus this can cause a lot of permission problems between users. We want to keep our home file server as KISS as possible. This way every file and every directory on the server will have pshare as the owner and pshare as group. When we set up the Samba server, we will mount SMB shares as user pshare. That way when someone writes files from a Windows machine, the files will still have pshare as the owner and pshare as group. Everything will be very homogeneous on the Data Drive. Out of the box, SSH will allow up to ten simultaneous connections per SSH server IP address, in this case 192.168.0.1.150 most households don't have ten Linux boxes. But if you need more, edit the sshd_config file on the server and change MaxSessions to something higher than ten. I don't know how many simultaneous SSH sessions a single user (pshare) can have, but I have never exceeded it so far. As an experiment, I had 3 Windows computers and 4 Linux computers simultaneously on the server, all 7 streaming music, and performing other server tasks. The server never missed a beat. If considering this for a SOHO server, I don't know what the limits would be for simultaneous connections. For a simple home server, it works great.