A Fedora based Simple Home Server.

This type of server can be run on any minimal Gnu/Linux operating system. By “minimal” I mean no Xorg graphics server, and no Desktop Environments. Just good old text only Command Line Instructions in a console window. I chose Fedora because it is the only distribution I know of that includes SELinux and a good firewall out of the box as part of a minimal install. Also, Fedora puts a lot of development into their Fedora Server Edition. I just feel more comfortable with Fedora.

This simple server will use a combination of SSH, FUSE, and SSHFS for Gnu/Linux clients. It will utilize SAMBA for Windows clients. MINIDLNA will be used for any DLNA enabled devices such as Smart TVs or AV receivers. That will suffice for most home servers.

This guide installs the Fedora Server Edition on a 60 GB SSD, and uses an internal 500 GB SSD for data storage and an identical external 500 GB SSD for data backup to create a simple home LAN file/DLNA server. Having the OS on one physical device (60 GB SSD), and having your data on another physical device (l500 GB SSD), and having an external device (500 GB External SSD) for backup is the best setup for when something happens. And we all know that eventually something WILL happen. If something happens to the OS SSD, having the OS by itself on a separate SSD really simplifies that process. If the data SSD craps out, having an identical external SSD that contains a back up of your data makes it easy.

Since most servers are on line 24 X 7, the ideal hardware for this home server would be low power. However, you can use just about any hardware you have laying around.

I have an ASRock Q1900-ITX Motherboard. Which has an Intel Quad Core J1900 processor at 2.00 GHz, with 8 GB of RAM, and a Samsung 500 GB SSD drive. Which is overkill for what I use it for, but it’s very efficient power wise. As measured with a Kill A Watt meter, at idle the server pulls 12 Watts at the 120 VAC wall receptacle. Running this server 24 hours a day for 30 days at 12 cents per Kilowatt hour would be an approximate cost of $1.03 per month.

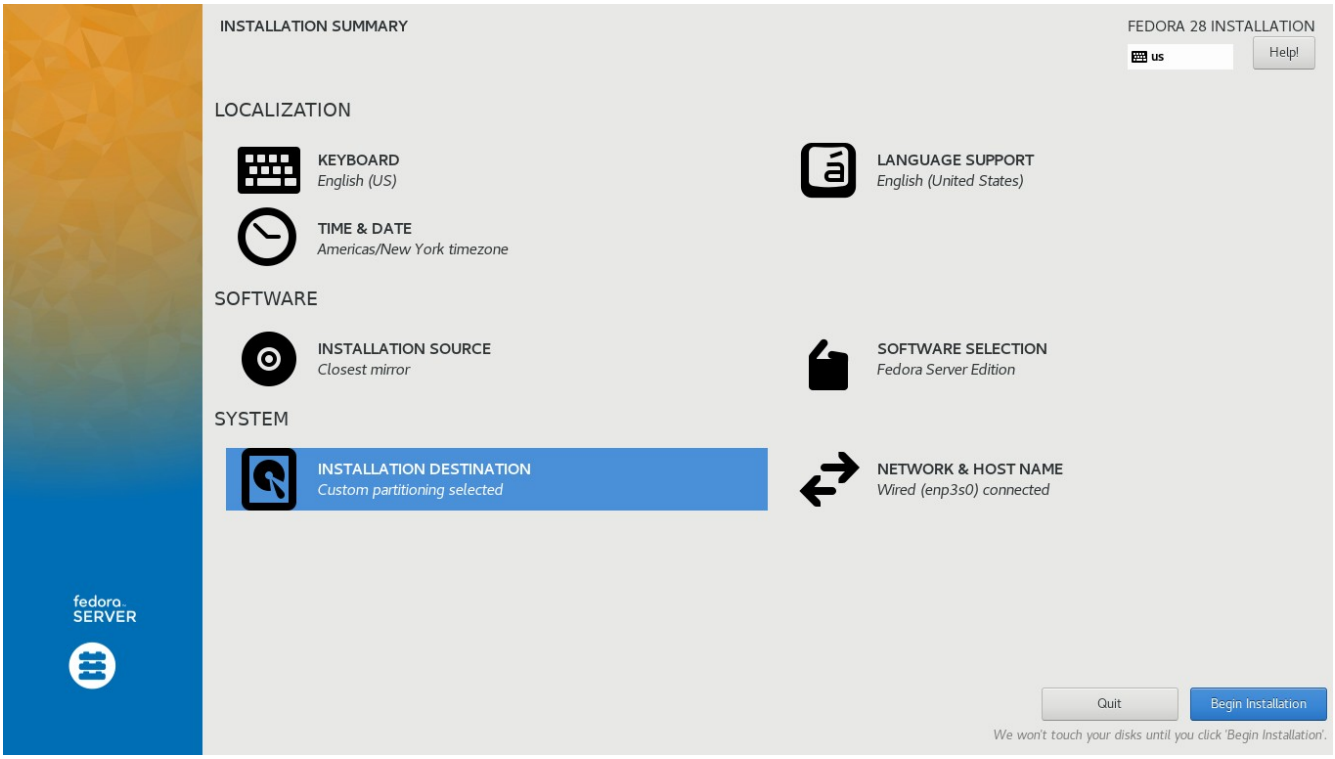
The above is my hardware setup. Of course you can use anything you want. You can use an old computer that’s laying around. You can use Hard Drives instead of SSDs. Your data requirements may need more or less storage than the 500 GB I decided on.

This guide requires the use of vi or nano for editing configuration files. If you don’t know vi or nano, then Google “Linux vi tutorial” or “Linux nano turorial” and learn before starting this Guide. This guide requires the server computer to have a monitor, mouse, and keyboard temporarily attached during installation. After installation the monitor, mouse, and keyboard can be removed, and the server can be run headless and administered from a Gnu/Linux Desktop Client. It is also possible to use PuTTY on a Windows machine for administration. This server is not intended to be accessed from the internet and should be connected directly by Lan cable to a router or to a switch connected to the router. IMHO a wireless server is NOT a good idea at all.

Download the 64 bit Fedora Server NETINSTALL IMAGE under the OTHER DOWNLOADS section. Burn the iso to a DVD or USB stick according to the instructions on the Fedora Web site.

ON THE SERVER COMPUTER.
Install a SSD to the SATA 1 connector on the motherboard to receive the Fedora OS. Install Fedora onto the SSD using the Net Install CD/DVD or USB stick.
WARNING:: ANYTHING ON THIS SSD WILL BE DESTROYED. Do not have the DATA SSD connected at this point, it just tends to confuse things. Configure your computer's BIOS to boot from the DVD/CD optical drive or USB stick and boot the computer.

After selecting a language, click on continue. You should get the INSTALLATION SUMMARY screen that looks similar to below. Installation is straight forward so I will only discuss two menu items: NETWORK & HOSTNAME and INSTALLATION DESTINATION.

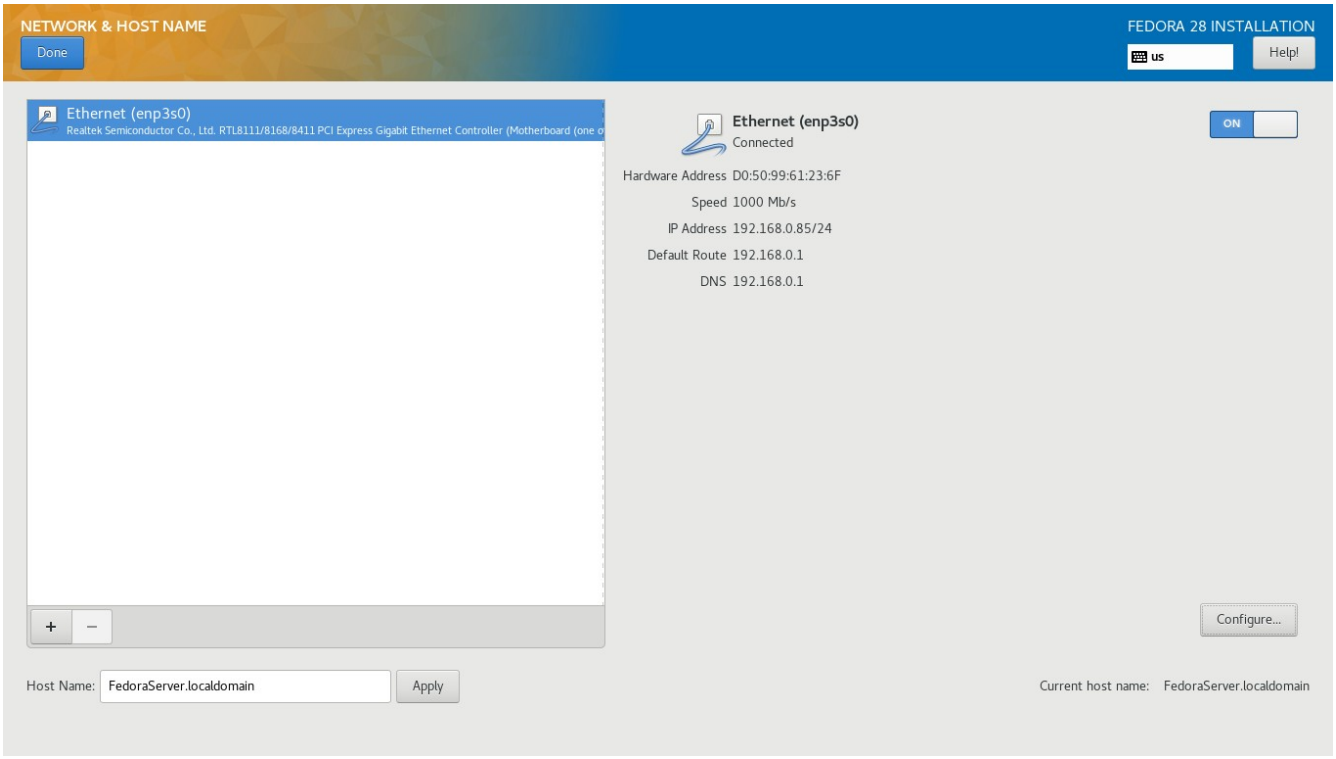


Wait until INSTALLATION SOURCE changes from “Checking Software Dependencies” to “Closest Mirror” and SOFTWARE SELECTION should change to “Fedora Server Edition” before going any further.

Click on DATE & TIME to set time zone, date, and time if necessary.
Click on KEYBOARD to change keyboard layout if necessary.
Click on LANGUAGE SUPPORT to change language if necessary.

Leave INSTALLATION SOURCE and SOFTWARE SELECTION as is: “Closest Mirror” and “Fedora Server Edition”

Servers should always run with a static IP address instead of a dynamic IP address (DHCP). Click on NETWORK & HOSTNAME. The following screen should appear



Go to the lower left to HOSTNAME and change to FedoraServer**.**localdomain Click on Apply

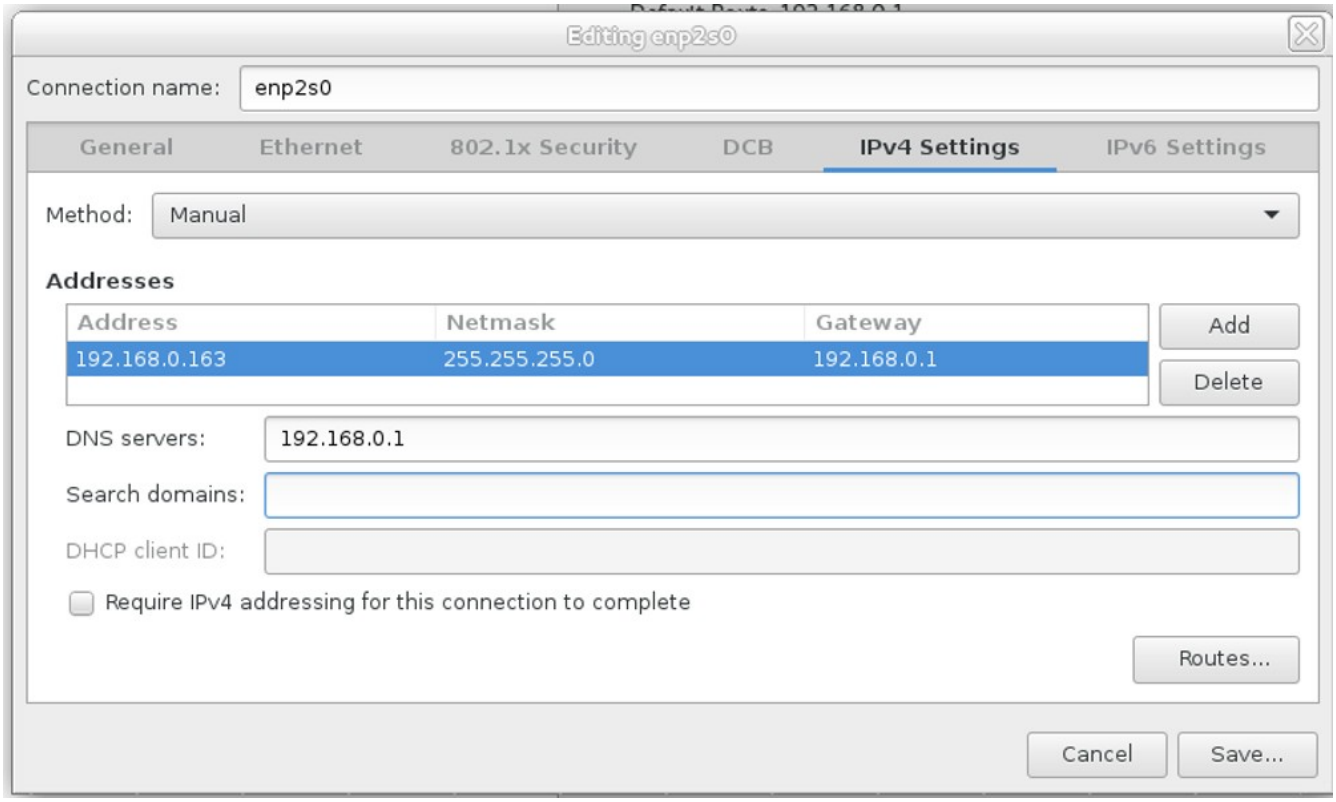
Go to the upper middle of the screen and write down the DHCP information that your router has determined for the server computer. In this case:

IP Address 192**.**168**.**0**.**102/24

Default Route 192**.**168**.**0.1

DNS 192**.**168**.**0.1

Your information may differ.
Go to lower right and Click on Configure and you should get the following.



Click on the "IPv4 Settings" Tab. Click on Method and change to Manual

Click on the "Add" button, then click in the Address entry field and enter the desired static IP address. I will be using 192.168.0.85 in this guide. Notice that all I changed is the last number from 102 to 85. You can set the last number to anything be 5 and 250 as desired.

Press TAB to enter the Netmask entry field. It should be 24. If not change it to 24.
Press TAB to enter the Gateway entry field, and enter what the previous screen listed as "Default Route"
Click in the DNS servers entry field and enter what the previous screen listed as "DNS"

The SAVE button should not be grayed out anymore, click on Save.
When you get to the NETWORK CONFIGURATION screen, click on Done and you will return to the INSTALLATION SUMMARY screen.

We need to partition and format the SSD or Hard Drive for the OS. There are two options.
1. Let the Anaconda installer do it for you, which will utilize LVM
2. Do a custom partitioning scheme, without LVM.

For my simple home server, I do not need a LVM (Logical Volume Manager). LVM is great for enterprise servers that need to be scalable, etc. For my simple web server, LVM is just another software layer affecting the filesystem. Not using LVM just means one less thing to go wrong. On my 60 GB SSD, I do custom partitioning using four standard partitions.

/boot boot partition
/ root partition
/home home partition
swap swap partition

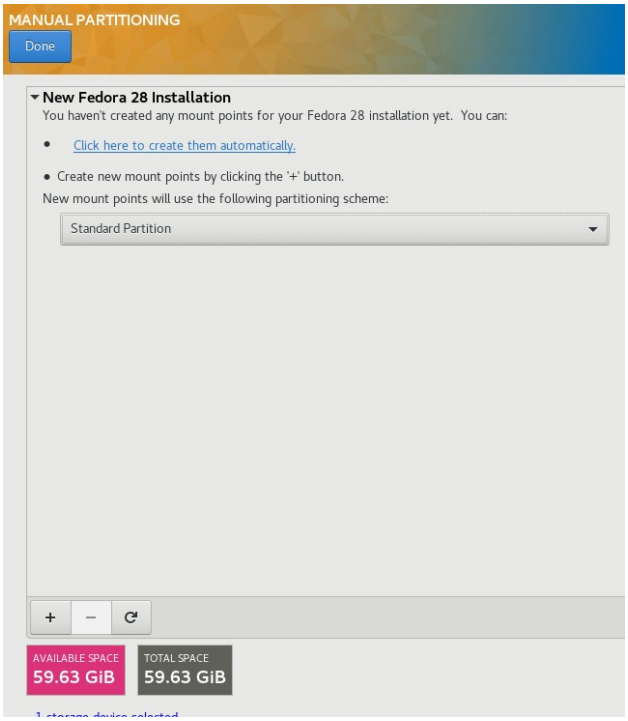
Letting Anaconda installer set up the partitioning with LVM is simple enough. If help is necessary, there are many Fedora installation tutorials on the internet for assistance.

If you want to consider doing a Custom or Manual partitioning, go here:

https://docs-old.fedoraproject.org/en-US/Fedora/26/html/Installation_Guide/sect-installation-gui-manual-partitioning.html

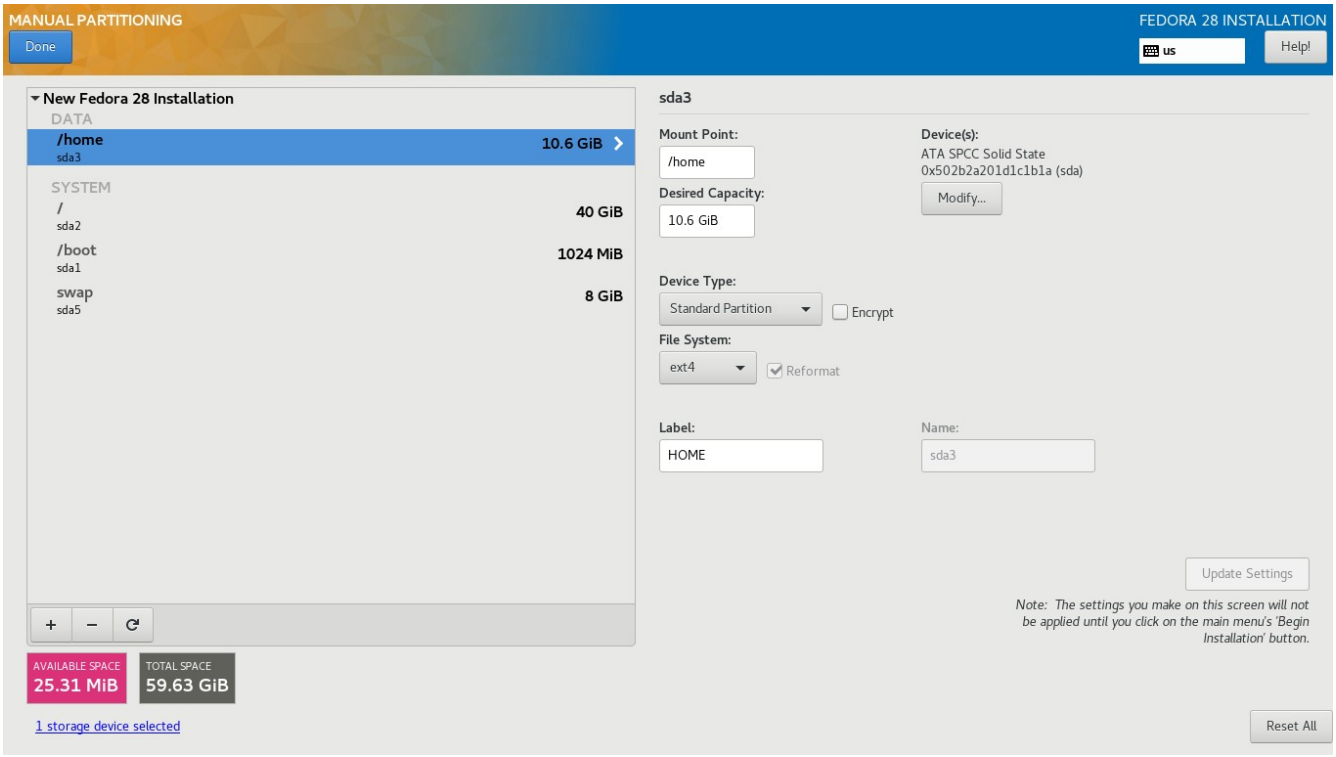
Here is how I manually partition a SSD for the OS.

Click on the INSTALLATION DESTINATION button, select the drive to be partitioned, click on “custom”, then click on “Done”.



Existing partitions will be listed under the drop down menu. Highlight a partition then delete it by using the minus icon until all partitions are deleted. Change the drop down menu to “Standard Partition”.

When the left side of your Manual Partition window looks like this, start adding partitions using the plus icon. Add the /boot partition first so it ends up being the first partition, such as sda1 or sdb1. When all partitions are added, it should look something like below.



Double check the information on the right for each partition. Check Mount Point, Desired Capacity, Device type (Standard Partition), Format type with Reformat checked, and give each partition an appropriate Label.

When finished, click on “Done” then click on Accept Changes and you should return to the “Installation Summary window”.

Once you are happy with your configurations, and there are no Orange Triangle warnings remaining, click on “Begin Installation”.

While Fedora is being installed, click on “ROOT PASSWORD” and set the root password. Click on Done when finished.

Next, click on USER CREATION and in the FULL NAME entry field enter “Public Share”. The USERNAME field should default to “pshare” if it doesn't, enter “pshare” for username then enter a password. Click on Done when finished.

After installation, be sure the CD/DVD or USB stick is removed. Enter BIOS and set the appropriate drive as the primary boot device if necessary. Boot the computer. You should get a login screen that looks similar to this:

```
Fedora 28 (Server Edition)
Kernel 4.17.6-200.fc28.x86_64 on an x86_64 (tty1)

Admin Console: https://192.168.0.85:9090 or https://[fe80::d6c2:3529:8940:e40]:9090

FedoraServer login:
```

At this point you can either continue locally using the keyboard and monitor attached to your server, or use the “Admin Console” otherwise known as Cockpit. The system login screen gives the adresses in IPv4 and IPv6 formats for Fedora’s Cockpit browser interface. See the very last page for information on how to access Cockpit.

Whether you continue using the keyboard and montitor locally on the server OR once you are logged into a Cockpit Terminal as root from a browser, the following commands will be the same.

Local at the server, Log in as root by typing in root for the username, and then enter your root password. Or you can use a Cockpit terminal after loggin in as user and su to root.

Check that installation set your IP address and host name corectly

ip addr

```
<snip>
  2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
   group default qlen 1000
   link/ether d0:50:99:71:1d:6e brd ff:ff:ff:ff:ff:ff
   inet 192.168.0.85/24 brd 192.168.0.255 scope global dynamic enp3s0
<snip>
```

Look for the above snippet. In this case enp3s0 (in red) is the name of the Ethernet interface. Your interface name may be different.

inet should be the Static IP address set during install. If not use vi or nano to edit ifcfg-enp3s0

vi /etc/sysconfig/network-scripts/ifcfg-enp3s0 (change if necessary)

then change the ip address.

Check the hostname for your server

```
# hostnamectl status
```

```
 Static hostname: FedoraServer.localdomain
       Icon name: computer-desktop
         Chassis: desktop
      Machine ID: 7185707552184fe1bb1c22404c1ca8d4
         Boot ID: b3864d9e2ebf4175b8bb8373e47e434f
Operating System: Fedora 28 (Server Edition)
     CPE OS Name: cpe:/o:fedoraproject:fedora:28
          Kernel: Linux 4.17.6-200.fc28.x86_64
    Architecture: x86-64
```

If the "Static hostname" isn't FedoraServer**.**localdomain execute the following

```
# hostnamectl set-hostname FedoraServer.localdomain --static
# hostnamectl status   (recheck if change was accepted)
```

Reboot the server if you changed the hostname.

add your user to group users

```
# gpasswd -a pshare users
# id pshare
```

```
uid=1000(pshare)  gid=1000(pshare)  groups=1000(pshare),100(users)
```

SETUP THE SSH SERVER

For security reasons, DO NOT use the default SSH port 22. There are 65,000 plus port numbers available. Go to Wikipedia's TCP port list or other such list. Scroll down to the 9000 – 10,000 range. Some of these ports are not assigned to any thing. Choose one of these ports to use for your SSH Local Area Network.

As root, edit sshd_config file

```
# vi /etc/ssh/sshd_config    & Change following lines:
        FROM #Port 22   TO  Port 9XXX
        From   #PermitRootLogin yes   TO   PermitRootLogin no
        From #PermitEmptyPasswords yes  TO   PermitEmptyPasswords no
        From #PasswordAuthentication no    TO  PasswordAuthentication  yes
```

use su (Switch User) to change to the user account

```
# su pshare
$ cd         (To ensure you are in pshare home directory)
```

create a ssh key for user pshare

```
$ ssh-keygen -t rsa -b 2048
```

accept defaults by Hitting Enter 3 times to get back to prompt.

```
$ ls -al      (  .ssh should be drwx------  if not  $ chmod 700 .ssh  then recheck)
```

Exit from user back to root

```
$ exit
#
```

Next we need to configure the firewall for ssh. First a discussion of firewalld and firewall-cmd. If you are not interested in this, skip to the next page.

`firewalld` uses the concepts of *zones* and *services*, that simplify the traffic management. Zones are predefined sets of rules. Network interfaces and sources can be assigned to a zone. The traffic allowed depends on the network your computer is connected to and the security level this network is assigned. Here is a list of Fedora's standard predefined zones.

firewall-cmd - -get-zones (list of all supported zones)
FedoraServer FedoraWorkstation block dmz drop external home internal public trusted work

Obviously, your home network is fairly trusted, so the home zone security is loose. A public network is not trusted at all, so the public zone has much tighter security.

In most distros, the default zone is public. Fedora Server Edition is designed for enterprize use. The Fedora Server Edition is set up with a custom zone and the Fedora Server developers made it the default zone. By the way, firewall-cmd options are prefaced by –(dash dash). The dash dash often looks like one horizontal line in word processors, so beware. To find what is the default zone:

firewall-cmd --get-default-zone
FedoraServer

The default zone is "FedoraServer". Your Fedora Server has it's own customized zone that the Fedora Server Edition developers created for better security (I assume).

A service can be a list of local ports, protocols, source ports, and destinations, if a service is enabled. Using services saves users time because they can achieve several tasks, such as opening ports, defining protocols, enabling packet forwarding and more, in a single step, rather than setting up everything one after another. To find a list of all predefined services

firewall-cmd - -list-services

For further information, go to the Red Hat Security Guide and start at section 5.1
RHEL Security Guide

I brought this up because if you are using Fedora Server Edition and go looking on the internet for help, this can be confusing. Unless the help obtained from a Fourm or tutorial is specifically for Fedora Server they will offer commands in the format:
firewall-cmd --permanent --zone=public --add-port=22/tcp
If you follow it exactly, you will be putting this rule in a zone where you don't want it. If you don't specify a zone, the default zone is used. In the case of Fedora Server the default zone is FedoraServer. So just leave out –zone=public when using internet advice.

See the addendum at the end of this how to for some useful firewall-cmd

Change firewall setting to disable the default ssh port of 22 and add port 9XXX

```
  # firewall-cmd  --query-service=ssh      ( “yes” if ssh is enabled or “no” if not enabled)
```
IF ssh is enabled (yes) skip the highlighted command.
IF ssh is not enabled (no) then run the following highlighted command
```
  # firewall-cmd  --permanent  --add-service=ssh
  # firewall-cmd  --permanent  --service=ssh --remove-port=22/tcp  (may be disabled already)
  # firewall-cmd  --permanent  --service=ssh --add-port=9XXX/tcp
  # firewall-cmd  --reload
```
inform SELinux that we have changed SSH port
```
  # semanage port  -a  -t  ssh_port_t -p tcp 9XXX     (This may take a while)
```

If semanage gives error “command not found” then
```
   # dnf install  policycoreutils-python-utils
```
and try again.

reboot the server for the changes to take effect.
```
  # reboot
```
Then log back into the server after reboot completes.

SET UP THE DATA DRIVE

set up /server & /serverbkup mount points
```
   # cd /   (Change to root directory)
   # mkdir /server   /serverbkup
   # chown root:users  /server  /serverbkup
   # chmod 774 /server   /serverbkup
```

You should now have something similar to this snippet

```
# ll
total 2
drwxrwxr--.  46 root users 4096 Aug 15 22:15 server
drwxrwxr--.   2 root users    6 Aug 19 16:29 serverbkup
```

The /server and /serverbkup are directories used for mounting hard drive partitions. You should never put any files or subdirectories in either one of these reserved directories.

```
# poweroff     (power off the computer)
```

Now determine how much DATA storage capacity you need and obtain a SSD or Hard Drive. With the server computer powered down, connect SSD or Hard Drive to the SATA 2 port on the server's motherboard. Then boot the server computer up. Log in as "root"

blkid

```
/dev/sda1: LABEL="BOOT" UUID="680ccd34-6234-453a-be44-92ec9b2f0262" TYPE="ext4"
/dev/sda2: LABEL="SWAP" UUID="5b247ec7-c3f3-4895-a651-1736e2013beb"
TYPE="swap"
/dev/sda3: LABEL="ROOT" UUID="ea93b6f3-cc33-4f8f-a2b6-d5877d35fa5e" TYPE="ext4"
/dev/sda5: LABEL="HOME" UUID="f1b1e022-56b4-4749-b553-b0f051691f58" TYPE="ext4"
/dev/sdb1: LABEL="####" UUID="f0f0a38a-11c2-4608-adc7-68d88c1863b6" TYPE="####"
```

The output of the blkid command gives us clues as to the dev assignment of the drives. The 4 partition labels listed for /dev/sda shows our four partitions that are obviously the drive that has our Operating System on it. In this case, /dev/sda. So /dev/sdb is our data drive we just connected. The output for /dev/sdb may vary depending on whether that SSD has ever been formatted, such as a brand new SSD Drive. We have determined in this case that the newly added Data Drive is /dev/sdb. So let's partition and format /dev/sdb

dd if=/dev/zero of=/dev/sdb bs=1M count=8 (zeros out the GPT or MBR)

fdisk /dev/sdb

a. Type in 0 (the lower case letter o not zero) to clear out any left over partitions
b. Type p to list partitions, there s/b no partitions left
c. Type in n for new partition, then p for a primary partition, then 1 for the first partition, then 2048 for the first sector, then press ENTER to accept the default last sector.
d. Type in w to write the partition and exit.

We just made the entire disk a single primary partition which will be /dev/sdb1. Now to format this partition.

mkfs.ext4 /dev/sdb1 -L DATA (the -L option sets the volume label for the partition)

We just formated the single partition to ext4. Part of that formatting includes assigning a new UUID number to the drive. Perform another blkid to find it's new UUID

blkid

```
/dev/sda1: LABEL="BOOT" UUID="680ccd34-6234-453a-be44-92ec9b2f0262" TYPE="ext4"
/dev/sda2: LABEL="SWAP" UUID="5b247ec7-c3f3-4895-a651-1736e2013beb"
TYPE="swap"
/dev/sda3: LABEL="ROOT" UUID="ea93b6f3-cc33-4f8f-a2b6-d5877d35fa5e" TYPE="ext4"
/dev/sda5: LABEL="HOME" UUID="f1b1e022-56b4-4749-b553-b0f051691f58" TYPE="ext4"
/dev/sdb1: LABEL="DATA" UUID="554ac9cc-8df5-44df-b1bd-dc489cc1f9c7" TYPE="ext4"
```

Write down the UUID of /dev/sdb1

Note in the above how judicious use of Labels during partitioning pays off in a blkid command.

Set up to mount Data SSD to mount point /server at boot up.

Your /etc/fstab file should look something like this, except I omitted the PARTUUID for brevity.

cat /etc/fstab (This lists the contents of fstab, I omitted comments)

```
UUID=ea93b6f3-cc33-4f8f-a2b6-d5877d35fa5e      /          ext4    defaults        1 1
UUID=680ccd34-6234-453a-be44-92ec9b2f0262   /boot    ext4    defaults        1 2
UUID=f1b1e022-56b4-4749-b553-b0f051691f58    /home   ext4    defaults        1 2
UUID=5b247ec7-c3f3-4895-a651-1736e2013beb   swap    swap  defaults        0 0
```

Add the following line to /etc/fstab using the new UUID as in the example above
vi /etc/fstab

UUID= YourNewUUID /server ext4 defaults 1 2

Be sure you specify the format you used when partitioning the Data Drive. If unsure, enter
blkid
to list the formats of connected drives.

Your /etc/fstab file should look something like this now:
cat /etc/fstab

```
UUID=ea93b6f3-cc33-4f8f-a2b6-d5877d35fa5e /                ext4    defaults        1 1
UUID=680ccd34-6234-453a-be44-92ec9b2f0262 /boot          ext4    defaults        1 2
UUID=f1b1e022-56b4-4749-b553-b0f051691f58 /home         ext4    defaults        1 2
UUID=5b247ec7-c3f3-4895-a651-1736e2013beb swap          swap   defaults        0 0
UUID=554ac9cc-8df5-44df-b1bd-dc489cc1f9c7  /server        ext4    defaults        1 2
```

Note the additional line at the end of the above listing, that is the Data Drive
Reboot the computer

After reboot, the SATA Data Drive should be mounted as /server.
Login as root.
ll /server

```
drwx------. 2 root root 16384 Feb  7 14:31 lost+found
```

should list a directory created at formatting named lost+found.

We are done with the FedoraServer computer for a while.
ON A CLIENT LINUX COMPUTER with Gnome, open a Terminal window and change to root.
Now that we are in a full blown Linux computer with a Gnome Desktop, you can use gedit to edit your config files or still use vi, your choice.
Add the following line to /etc/hosts to establish a route to the server

vi /etc/hosts
192.168.0.85 FedoraServer.localdomain FedoraServer

If you used a different network address from 192.168.0.85 use it instead.

As root:
cd /etc/ssh
ll
you should see two or more files which include
-rw-r--r--. 1 root root 2121 Dec 22 11:01 ssh_config
-rw-------. 1 root root 4424 Dec 11 07:31 sshd_config
If you don't have these files, install openssh on the client computer.
vi or gedit /etc/ssh/ssh_config and change the following lines.
  FROM #Port 22 TO Port 9XXX (Use the same port # as in the server)
  FROM #Protocol 2,1 TO Protocol 2 (this line may be absent in newer versions)
As User: (# su username)
$ cd ~/.ssh
$ ll
-rw-------. 1 don don 1675 Dec 22 11:04 id_rsa
-rw-r--r--. 1 don don 402 Dec 22 11:04 id_rsa.pub

If you see these 2 files and possibly more, you already have SSH keys.
If you don't have these files
$ ssh-keygen -t rsa -b 2048
accept defaults by hitting Enter 3 times to get back to prompt.

Restart the CLIENT COMPUTER.
Login as user and open a terminal window
Try to connect to FedoraServer
$ ssh pshare@FedoraServer (you did create a user named pshare on the server, right?)
If you successfully communicated with FedoraServer, you will get the following

The authenticity of host '[fedoraserver]:9XXX ([192.168.0.163]: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)?

Type in “yes”, the connection will be completed and you will be asked for pshare's password.
You should then have a terminal prompt of
[pshare@FedoraServer ~]$

You are now logged into FedoraServer as user pshare. Pay attention to the prompt, it will always let you know which computer you are in and the user name. Anything you type in the Terminal window will now be executed in the FedoraServer computer. Type in
$ exit
and you should log out of FedoraServer and return to your local prompt.

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 both the server and client computers.

IN THE CLIENT COMPUTER as user
$ cd ~/.ssh
$ ll
-rw-------. 1 don don 1675 Dec 22 11:04 id_rsa
-rw-r--r--. 1 don don 402 Dec 22 11:04 id_rsa.pub
-rw-r--r--. 1 don don 605 Feb 6 17:23 known_hosts

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 server

$ ssh-copy-id -i ~/.ssh/id_rsa.pub pshare@FedoraServer
enter pshare's password if requested. Now try to log into FedoraServer
$ ssh pshare@FedoraServer

You should now be logged into FedoraServer as pshare without having to enter your password. Once in FedoraServer change to root and do most anything you want from a nice GUI terminal window with mouse, scroll bars, cut and paste, etc.

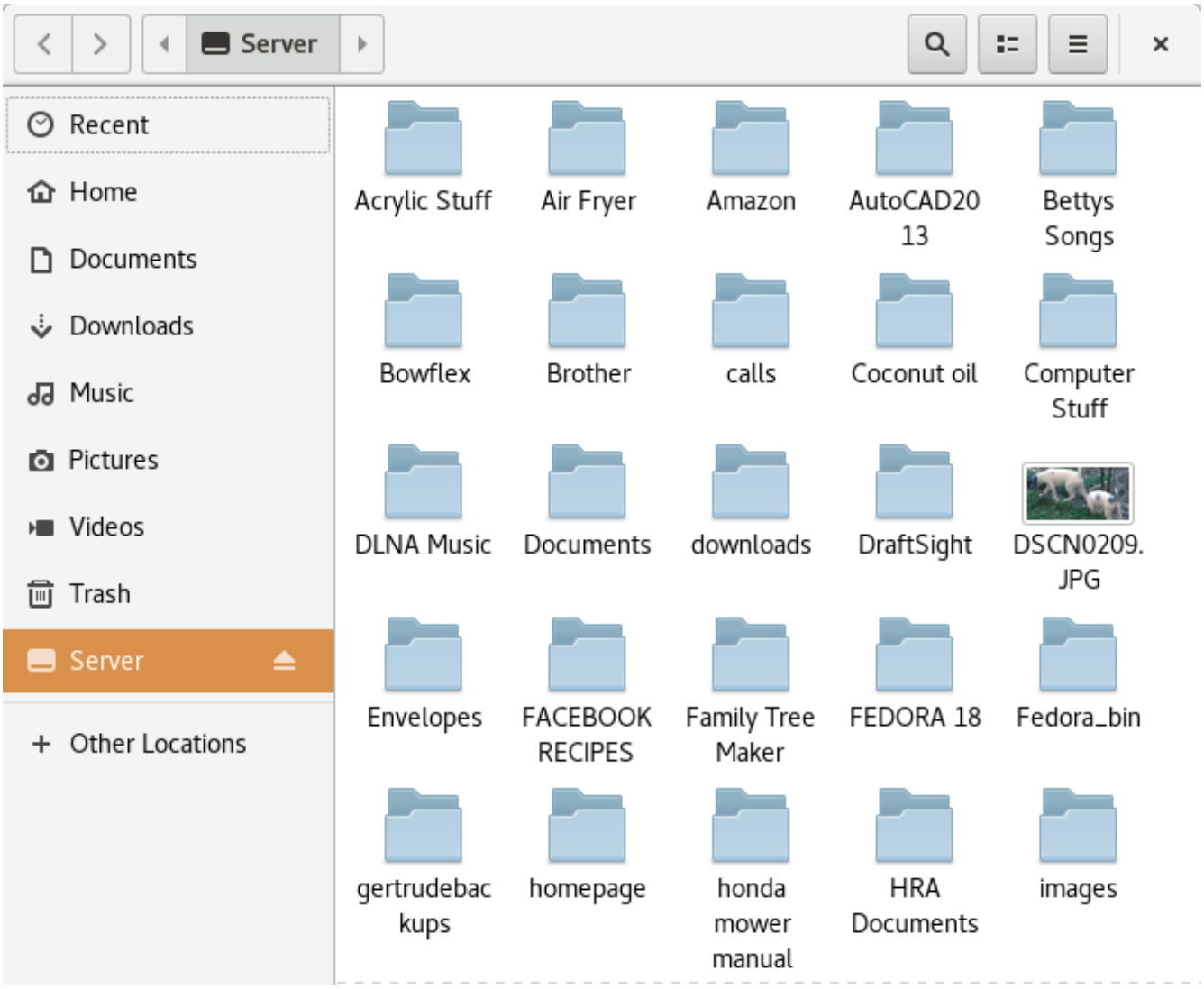
Also, from this point on you can run the server headless if you want. Just ssh into FedoraServer from a client terminal window or use Cockpit from a browser (see last page).

Now to tie this up and make this really neat and convenient to use, we will set up FUSE and sshfs to utilize a Nautilus window to access the files on the FedoraServer:/server partition.

IN THE CLIENT COMPUTER, ensure that fuse, sshfs, and all associated packages for fuse are installed as per your Linux distribution. In Fedora Workstation: #dnf install sshfs

IN THE CLIENT COMPUTER, in a Terminal window as user
$ mkdir ~/Server
Note the Upper Case S in ~/Server This is done on purpose to avoid confusion between directories named server. /server is a mount directory in the FedoraServer computer, /home/username/Server is a mount directory in the Client computer. Again, /home/username/Server in the client computer is a mount directory and should never contain any files or sub-directories. /home/username/Server is used as a mount point only.

As user
$ sshfs pshare@FedoraServer:/server ~/Server
$
It should complete without any errors. Launch Nautilus (also known as Files) and click on Server om the lower left. You should see the file Lost+Found listed, or something like this



sshfs uses FUSE to fool your client computer into thinking the /server directory on FedoraServer is a local directory named /home/username/Server. Anything you can do on a local directory, you can now do on the remote computer's directory.
Using Nautilus, make a new directory on /home/username/Server and copy a few files to it. Open LibreOffice and create a new document, then save it to /home/username/Server/YourNewDirectory.

As user:
$ fusermount -u ~/Server (dismounts the FedoraServer)
Direct Nautilus to the /home/username/Server directory, and it should be empty. Because there isn't a remote directory mounted there anymore.

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? Assuming you are using Fedora with Gnome, let's automate this process some. Other distributions with a Gnome desktop should work, but no guarantees. With any other desktop besides Gnome, you are on your own.

Automatically mount FedoraServer at log in.

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

```
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 vi or gedit, create a file named AutoMountServer and add these 3 lines
 $ vi AutoMountFedoraServer

```
#!/bin/bash      (the first two characters are # then an exclamation mark)
sshfs pshare@FedoraServer:/server  /home/$USER/Server
exit
```

$ chmod 754 AutoMountServer
$ ll

```
-rwxr-xr-x 1 pshare pshare   160 Apr 14 19:45 AutoMountServer
```

To have AutoMountServer run every time you log in.
In your Client Computer Terminal Window as root:
  # dnf install alacarte gnome-tweak-tool

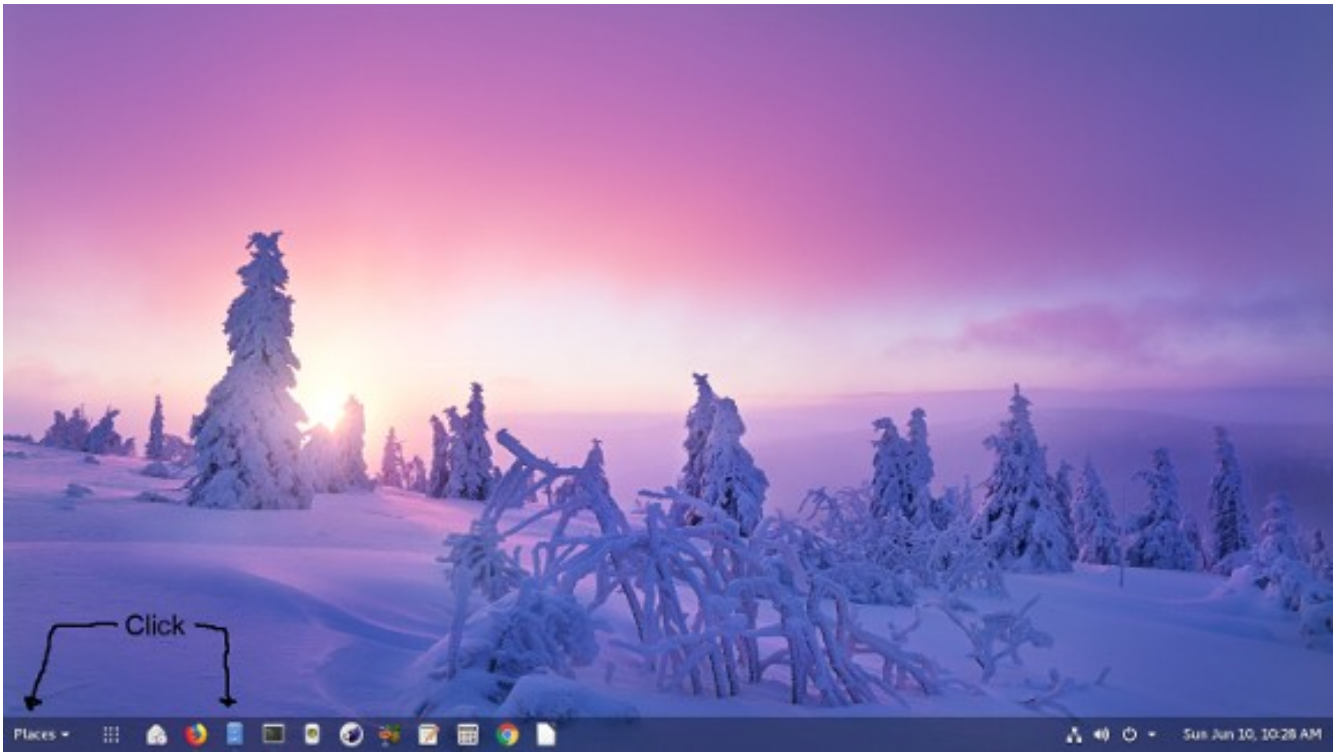
Go to the desktop, bump left upper corner, or click on “Show Applications” then click on “Sundry” then click on “Main Menu”. On the bottom of left hand column, click on “System Tools”. Click on “New Item” Give the Launcher a Name: Automount Server
Then browse to /home/username/bin/AutoMountServer
Comment: Automounts FedoraServer at login
Click on the Icon box in the upper left, browse to
“Other Locations” then “computer” then
/usr/share/icons/Adwaita/48x48/devices/drive-hardisk.png and click OK

Now get to “Show Applications” and you should see a new icon for Automount Server.
In “Show Applications” click on “Utilities” then on “Tweak Tool”
On the left, click on “Startup Applications” then click on the +
Scroll to Automount Server, highlight it, and click on “Add Application”
Automount Server should be listed as a startup application. Click on it.

Log out and log back in, or reboot.. FedoraServer 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 FedoraServer is ready to use.

Now configure any other Linux machines you have exactly like we did the first 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 simple 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 192.168.0.163. 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.

The above works for a Fedora Gnome desktop install on the Linux Client computer..
I assume you can set this up in a similar fashion in Ubuntu Unity, but I don't know for sure.
Other desktops may take some experimentation.



Here is a screenshot of my Gnome 3 desktop. Notice that if you have used Tweak to install the “Dash to Panel” extension and turned on the “Places Status Indicator” you can either click on the “Places” icon or the “Files” icon to access the FedoraServer.

SAMBA

Time to take care of any Windows computers by setting up a SAMBA server.

Either get into the server computer via it's connected keyboard and monitor, OR SSH into it from a from a Terminal window on a Client Linux computer, then change to root using su

IN THE SERVER COMPUTER as root.
dnf install samba samba-client

We need to inform SELinux that we are going to use samba
chcon -t samba_share_t /server/*
#semanage fcontext -a -t samba_share_t "/server(/.*)?"
restorecon -R -v /server

The first thing necessary is to obtain the work group your Windows computers are assigned to. In a Windows 7 computer, Start Button --> Control Panel --> System and Security --> System

This window will display this computer's work group. If I remember correctly, MYGROUP is the default work group. On anything that involves security, I hate using any defaults. I am going to call my work group THREEAMIGOS All Windows computers wanting to access the FedoraServer should have the same work group name. To change a work group name in a Windows 7 computer:
In the Control Panel --> System and Security --> System dialog box, click on "Change Settings".
In the System Properties dialog box, make sure the "Computer Name" Tab is selected.
To change the work group, click on the "Change" button.
In the "Computer Name/Domain Changes" dialog box, select the Workgroup entry field and type in
your work group name. See this document for the work group naming rules.
http://compnetworking.about.com/od/windowsnetworking/qt/workgroupnaming.htm

vi /etc/samba/smb.conf
(under GLOBAL change the work group name to the one you chose)
workgroup = THREEAMIGOS
go down to the end of the GLOBAL section and add the following line
hosts allow = 127. 192.168.0.

(The 192.168.0. is the static IP address we assigned to This FedoraServer computer, minus the numbers after the third decimal point. This means any computer with an IP address that starts with

192.168.0. is allowed to connect. So any computer not on our LAN block is not allowed.)

(The 127. allows you to use local loopback for testing)

(Add the following lines at the end of this smb.conf file)

```
    [SMBshare]
    comment = Samba Share
    path = /server
    valid users = pshare
    public = no
    writeable = yes
    printable = no
    create mask =0765    (End of editing, close file)
```

We set up a Samba share named SMBshare, told it which directory we are sharing, and allowed ONLY user pshare to be a valid user. Next we need to set up a Samba account for user pshare. User pshare must be a valid Linux user on the FedoraServer computer.

Still in the server computer as root

```
# smbpasswd -a pshare
   New SMB password:  EnterAPassword
   Retype new SMB password:  Re-enter the same password
```

Since samba wasn't part of the original server install, we installed the samba packages. Now we must let systemctl know we want the samba services initiated at boot up.

```
# systemctl enable nmb.service
# systemctl enable smb.service
# systemctl start nmb.service
# systemctl start smb.service
```

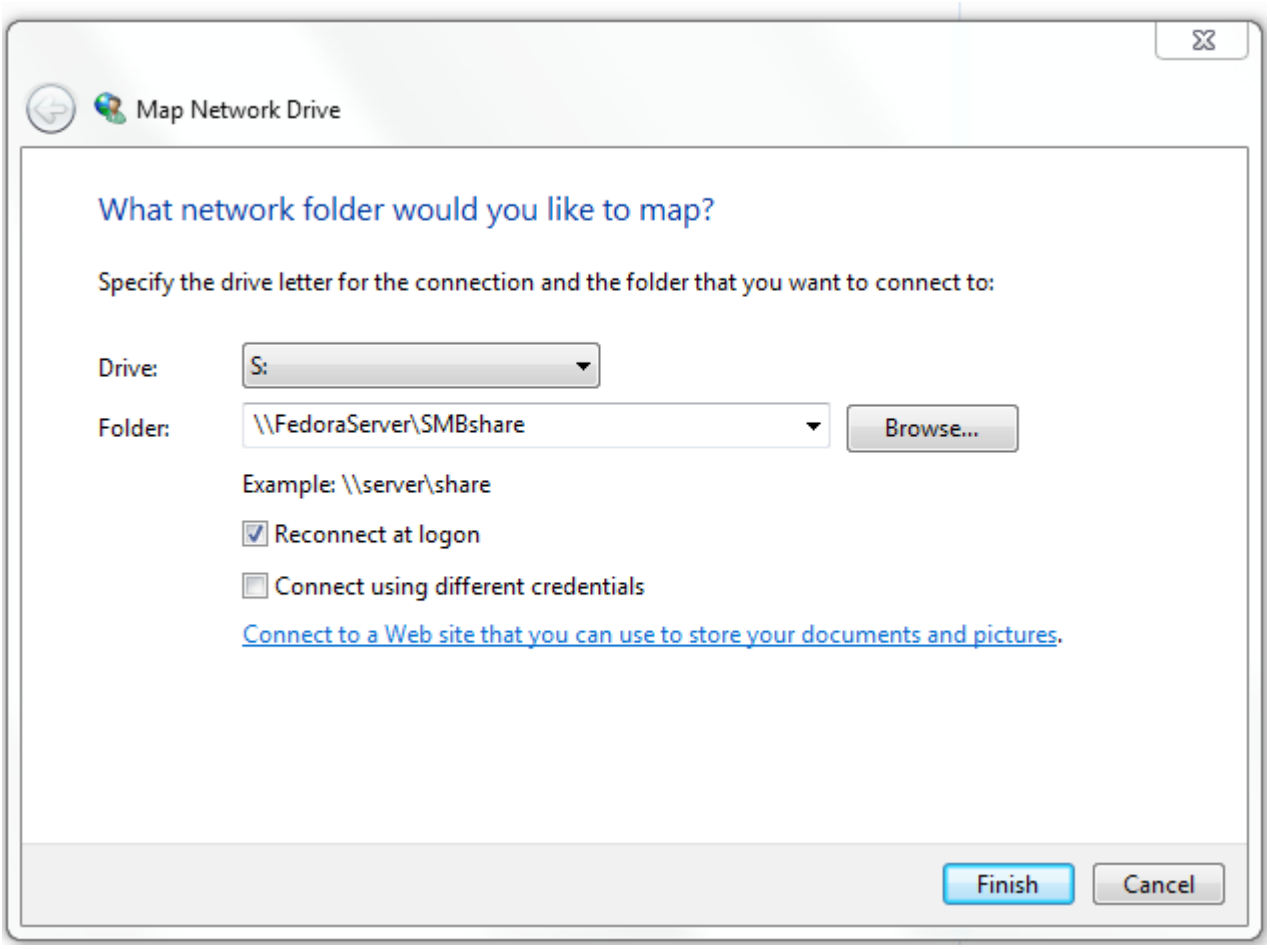
Next we set up the server's firewall for samba.

```
# firewall-cmd  --query-service=samba
   no     (If yes, skip next command)
#  firewall-cmd --permanent –add-service=samba
   success
#  firewall-cmd  --reload
#  firewall-cmd  --list-all    (should show samba as an active service)
```

Restart the server computer

```
# reboot
```

IN A WINDOWS 7 COMPUTER (Other versions of Windows should be similar)
Click on the “Computer” icon, Then click on “Map Network Drive” to get the following:



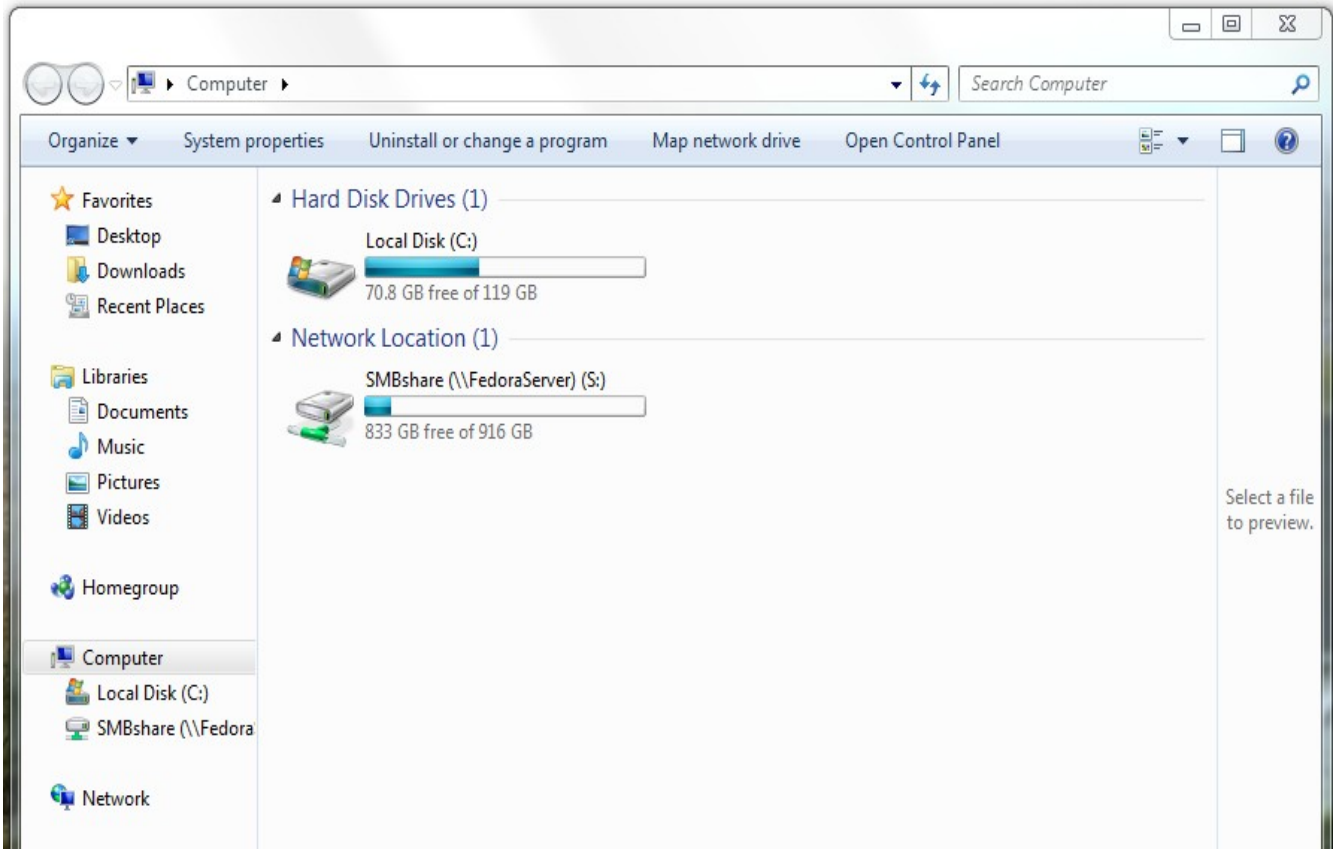
in the “Drive” pull down, select S: (S for server) for the drive letter to mount Samba Share to.

In the “Folder” pulldown enter \\FedoraServer\SMBshare (SMBshare is case sensitive)
Make sure “Reconnect at logon” is checked
Click Finish

You will get a dialog box saying Trying to connect

Then a box asking for username and password. Enter pshare for user and pshare's Samba password that we entered with # smbpasswd -a pshare
Make sure “Remember my credentials” is checked.

Go back to “Computer” and in addition to Local Disk (C:) you should see the SMBshare mounted as Drive S:



If you have the SMBshare mounted and listed as above. Power off and restart Windows and see if the SMBshare is automatically mounted after logging in. If it automatically mounts at boot up, we are done with Samba. Now Setup any other Windows machines in the exact same manner.

MINIDLNA

Lastly we will install and set up a minidlna server to play your MP3s, display Pictures, and view videos on your Smart TV or any other device which is DLNA enabled.

IN A LINUX CLIENT computer, from a Terminal window, SSH into the FedoraServer and become root.
Open a browser and go to http://rpmfusion.org/Configuration

Scroll down to:

Command Line Setup using rpm

To enable access to both the free and the nonfree repository use the following command:

•Fedora 22 and later:

```
dnf install https://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-$(rpm -E %fedora).noarch.rpm https://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-$(rpm -E %fedora).noarch.rpm
```

Highlight the command under “Fedora 22 and later” as shown above. Right click in the highlighted area and click Copy.

Now go back to your terminal window that is connected to FedoraServer as root and right click next to the # prompt and Click Paste. The command from the web page should be pasted into the Terminal Window.

dnf install https://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-$(rpm -E %fedora).noarch.rpm https://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-$(rpm -E %fedora).noarch.rpm

Press Enter and the command will install the rpmfusion repositories into FedoraServer's repositories list. Now that the rpmfusion repository is available to FedoraServer we can install minidlna. FedoraServer does not have a GUI interface, browsers, or cut and paste. You could have logged into FedoraServer directly with a connected Monitor and keyboard and typed in that extremely long command. Or, you can cheat by logging into a Linux Client that does have a GUI desktop and a browser and copy from the browser and paste into the Terminal window that is connected via ssh to FedoraServer.

Now we have the rpmfusion repositories available. Back in the Client computer Terminal Window that is logged into FedoraServer as root

```
# dnf install minidlna
# firewall-cmd  --query-service=minidlna
   no                                  (if yes, skip next step)
# firewall-cmd  --permanent  –add-service=minidlna    (should return success)
# firewall-cmd  --reload      (should return success)
# firewall-cmd  --list-all     (should show minidlna as an active service)

# vi /etc/minidlna.conf
  (make sure port is set to 8200
     port=8200
  (There may be some directories with music files that you do not want to share on
  FedoraServer.  So you can restrict minidlna to what directories it will share.
 Scroll to the section # set this to the directory you want scanned.
 There are directions and examples given.  Here is how I set up my directories)
    media_dir=A,/server/MP3
    media_dir=P,/server/Pictures
    media_dir=V,/server/Videos
(Right under this section is the Friendly Name section I changed it to)
    friendly_name=Fedora DLNA server

# gpasswd -a minidlna  pshare
# gpasswd -a minidlna users
# groups minidlna
minidlna : minidlna users pshare

# systemctl enable minidlna.service
Reboot the FedoraServer computer.
```

Using a Smart TV, Windows MediaPlayer, or other DLNA enabled device on your LAN, and see if it will recognize the Fedora DLNA Server. Minidlna works on my Samsung Smart TV, and my ONKYO AV Reciever. It will also work on Gnome's Rhythmbox if the Grilo plugin is enabled. Make sure there are some MP3s in the directory listed in minidlna.conf.
If a desktop or laptop computer is used as the minidlna client, be sure to open port 1900/UDP incoming on any installed firewalls.

To use DLNA with Rhythmbox, with Rhythmbox running, go to the options menu and click on “Plugins”. Then put a check in the “Grilo Media Browser” box. Restart Rhythmbox and it should show “Fedora DNLA Server” in the lower left under “Shared”.

Click on ‘Fedora DLNA Server’ then click the arrow on Music, then click on ‘All Music’. Fedora will talk to the DLNA server and list all music in the folders you designated in the MiniDLNA config file.

Backing up FedoraServer.

Remember when we made two directories right off root named /server and /serverbkup? Well /serverbkup is for mounting an external USB hard drive enclosure to backup FedoraServer.

Back on page 12, we used a Gparted Live DVD to partition and format a SATA SSD drive for our DATA drive. Now we will use the same procedure to partition and format an external USB hard drive for a backup hard drive. Ideally the USB SSD drive should be the same size as the SATA Data hard drive. In my case both are 500 GB. We are going to format the USB SSD drive exactly like the Data drive. One single partition formatted so that it has the same format (ext4) as the SATA Data Drive's format. There is a reason for all this. IF the external USB hard drive enclosure has a SATA hard drive inside the enclosure, and IF the SATA Data Drive goes bad, all that needs to be done is take the hard drive out of the USB enclosure and swap it with the defective Data hard drive. Then edit the /etc/fstab file and change the UUID of the old Data Hard drive to the UUID of the backup hard drive. Voila, you are back up and running in about 20 to 30 minutes.

Connect a USB external hard drive to the server computer. For safety, I make sure that the hard drive I intend to partition and format is the ONLY hard drive connected by temporarily unplugging the installed OS SSD drive and the internal SSD Data drive. During boot up, change BIOS so it boots from the Gparted DVD and partition and format the backup USB hard drive. Power off the computer and the USB hard drive and restore the computer to it's original condition.

Even though USB devices can be hot plugged, I still prefer to connect the USB external enclosure with the server powered down. I leave the USB external enclosure connected and powered down except when doing a back up. To do a backup

In the FedoraServer as root:
dnf install rsync (If you didn't install rsync earlier)
fdisk -l

Determine which devices are being used. Most likely /dev/sda and /dev/sdb
Power up the external USB hard drive

fdisk -l

Determine the new device that appeared after power up. Most likely the additional device will be /dev/sdc so in this example I will use /dev/sdc1

# mount -t ext4 /dev/sdc1 /serverbkup (be sure to use the proper format type ext3 or ext4)
# su pshare (preform the backup as user pshare)
$ rsync -av --delete /server/ /serverbkup (the / at the end of /server/ is important)
$ exit
# umount /serverbkup (then power off the external USB hard drive)

Fedora Sever - COCKPIT

For those of you who wish to play with Fedora's Admin Console interface (Cockpit), here is how to access it.

In the client computer:

open a browser (I use Firefox) and enter the IP address:
https://192.168.0.85:9090 or https://[fe80::d6c2:3529:8940:e40]:9090

You will get a dialog box saying
“Your Connection is not secure”
Click on “Advanced”
Click on “Add exception”
make sure “Permanently store this exception” is checked
Click on “Confirm Security Exception”

Then you should get the Cockpit interface.

Log in using your server user account (pshare) and password.

After loging in, on the lower left click on “Terminal”

You are now in a terminal window.
Then become root for administration of the server.

$ su
root_password
#

Exit root when finished with the terminal,
exit
$

Be sure to logout of Cockpit when finished with it.

I hope this guide was useful and maybe even educational. I call this my Nisson XTERRA server. Everything I need and nothing I don't.. Enjoy your new Fedora based simple home server.

Addendum

Some useful firewall-cmd commands:
firewall-cmd --get-default-zone (lists the default zone)
firewall-cmd --set-default-zone=public (sets a new default zone)
firewall-cmd --list-all (Lists info for the default zone)
firewall-cmd --list-services (lists active predifined services)
firewall-cmd --list-all –zone=FedoraServer (or other zones such as public)
firewall-cmd --reload (makes any changes active – see permanent and runtime settings)
firewall-cmd --query-service=ssh (list if a service such as ssh is enabled)
firewall-cmd --get-zones (List of all supported zones)
firewall-cmd --state (get state of firewall)
firewall-cmd --get-services (get list of all supported services)
firewall-cmd --list-all-zones (list all zones with the enabled features)
firewall-cmd --get-active-zones
firewall-cmd --permanent --zone=FedoraServer –add-service=http

firewall-cmd --help (the ultimate firewall-cmd)

USER'S NOTES: