How to customize Calamares and some useful CLI commands

In this tutorial, how to customize Calamares, and familiarization of some useful CLI commands will be covered.

To do this, we will build a headless LAN server on a x86_64 computer.
The Operating System is installed on a small SATA SSD on the sata1 connector, and a large SATA SSD on the sata2 connector for data storage. Then for data backup, a SSD that is identical to the data SSD will be in a USB 3 enclosure. This separates the OS from the data, and the backup SSD is physically separate. Also it is assumed that an ethernet LAN will be utilized.

If you have a spare x86_64 system lurking about, you are ready to go.

How to customize Calamares

Since this is intended to be a headless server, Calamares needs to be customized for our use case.
On your x86_64 device, install a small SATA SSD on the SATA1 connector to receive the Operating System. A 128 GB SSD is more than enough as it will only contain the Operating System. If no spare SSD is available, and you just want to tinker around, the OS can be installed on a USB 3 thumb drive that is at least 32 GB for this tutorial. Another USB 3 thumb drive can be used for the DATA device.
This tutorial has been tested using 2 USB 3 Thumb drives, one for the OS, the other for data.

Boot up the Cassini Neo (or latest version) ISO USB Thumb drive on the target x86_64 device in UEFI mode. To customize Calamares, we need to exit the Welcome window. Then when we restart Welcome it will read in our changes.

Customization 1.
First we need to specify a list of extra packages for Calamares to install for us.
Open Thunar and find “user_pkglist.txt”.
Open with “Text Editor” and add the following, one item per line.

```
libusb-compat
libusbmuxd
usbmuxd
usbredir
neofetch
rsync
sed
parted
vi
wget
```

Save “user_pkglist.txt” close “Text Editor”
Any package in the Archlinux main repositories, Core, Community, or Extra that is not normally installed by Calamares can be listed here.

Customization 2.
In Thunar, find “user_commands.bash” and open with “Text Editor” and add the following at the bottom of the file.

```
systemctl enable --quiet sshd.service
systemctl enable --quiet firewalld.service
```

Save “user_commands.bash” close “Text Editor” and close Thunar

With the “user_commands.bash” file you can customize the installed system. For example:
- install and/or remove packages
- enable and disable systemd services

and more. See the user_commands.bash file for further details.

In a terminal window, use vi or nano to edit this conf file

```
sudo vi /etc/calamares/modules/pacstrap.conf
```

delete the following lines

```
base-devel
f2fs-tools
jfsutils
lvm2
mdadm
reiserfsprogs
s-nail
xfsprogs
```

Save “pacstrap.conf” close vi or nano and close the terminal window.
NOTE. Do not delete sudo or else the install will fail.

When Calamares runs pacstrap, all packages in the “/etc/calamares/modules/pacstrap.conf” file are ALWAYS installed irregardless of what options you chose from the Calamares GUI.
If you want to install from the AUR, do NOT delete base-devel. Most of the rest are packages for filesystem types. For a server, we are going with ext4, so most of the rest is just Bloat. If you never want to use BRTFS on your server, that is one more filesystem type that can be eliminated. Also, if you do not want to use the man pages on the server, you can delete man-db and man-pages.

Now we need to start the Welcome screen. On the panel there is an icon that looks like a CD ROM disk. Hovering the cursor over it should show “Install System Calamares --- System Installer”. The Welcome screen will launch and read in all the customization we provided.

START THE INSTALLER
- Choose Online

WELCOME SCREEN
- Choose the Language

LOCATION SCREEN
- Set the Time Zone

KEYBOARD SCREEN
- Set the keyboard

DESKTOP SCREEN

Select “No Desktop”

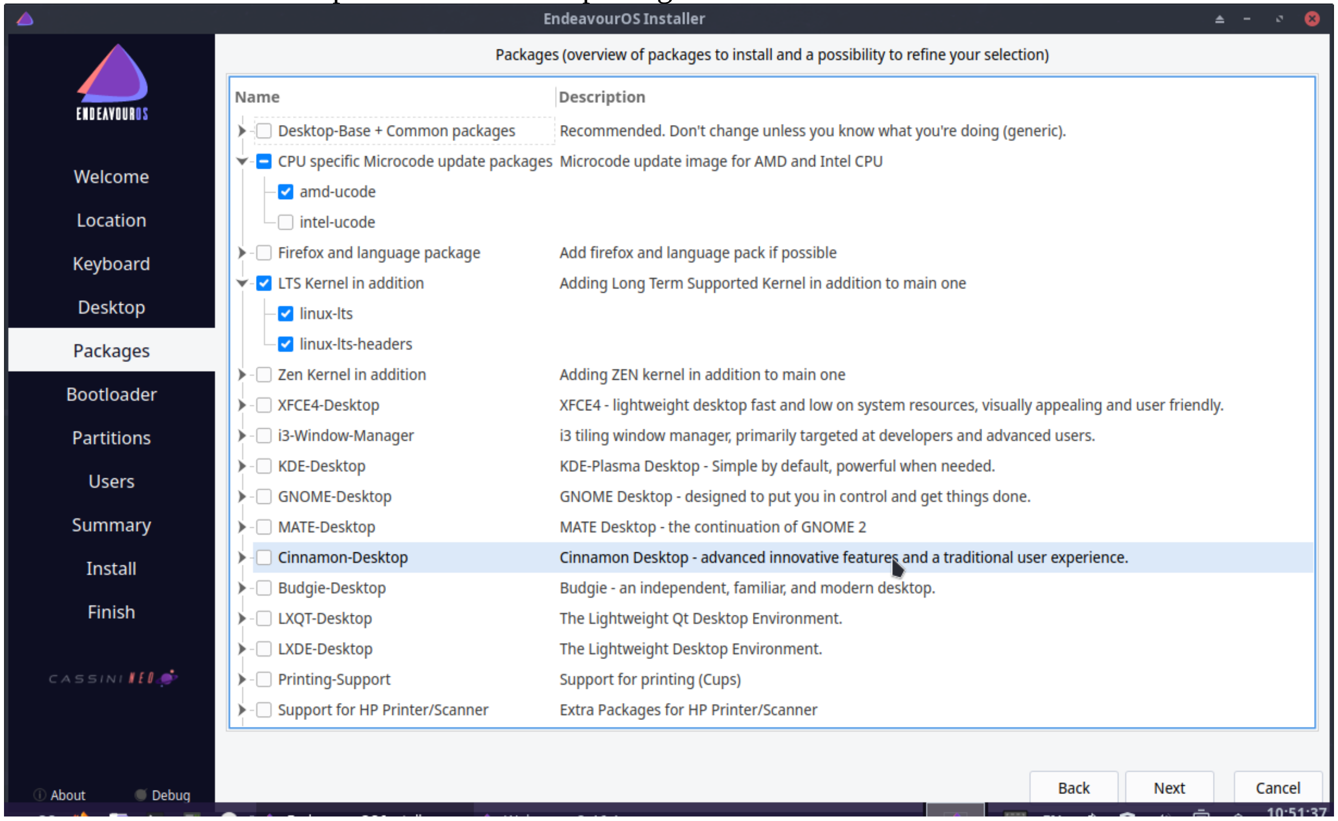
PACKAGES SCREEN

“CPU specific Microcode update packages”, deselect either intel-ucode or amd-ucode

De-select “Firefox and language package”

Select “LTS Kernel in addition”

De-select “Desktop-Base + Common packages”



Packages (overview of packages to install and a possibility to refine your selection)

Name	Description
☐ Desktop-Base + Common packages	Recommended. Don't change unless you know what you're doing (generic).
▬ CPU specific Microcode update packages	Microcode update image for AMD and Intel CPU
☑ amd-ucode	
☐ intel-ucode	
☐ Firefox and language package	Add firefox and language pack if possible
☑ LTS Kernel in addition	Adding Long Term Supported Kernel in addition to main one
☑ linux-lts	
☑ linux-lts-headers	
☐ Zen Kernel in addition	Adding ZEN kernel in addition to main one
☐ XFCE4-Desktop	XFCE4 - lightweight desktop fast and low on system resources, visually appealing and user friendly.
☐ i3-Window-Manager	i3 tiling window manager, primarily targeted at developers and advanced users.
☐ KDE-Desktop	KDE-Plasma Desktop - Simple by default, powerful when needed.
☐ GNOME-Desktop	GNOME Desktop - designed to put you in control and get things done.
☐ MATE-Desktop	MATE Desktop - the continuation of GNOME 2
☐ Cinnamon-Desktop	Cinnamon Desktop - advanced innovative features and a traditional user experience.
☐ Budgie-Desktop	Budgie - an independent, familiar, and modern desktop.
☐ LXQT-Desktop	The Lightweight Qt Desktop Environment.
☐ LXDE-Desktop	The Lightweight Desktop Environment.
☐ Printing-Support	Support for printing (Cups)
☐ Support for HP Printer/Scanner	Extra Packages for HP Printer/Scanner

The window should look like this, with the possible exception being intel-ucode selected

Scroll to the bottom of the window, and notice “Custom Packages”. Those are the packages added by the “user_pkglist.txt” file. Leave it selected.

Click the horizontal black arrow to open the “Desktop-Base + Common packages” list.

All categories should be unselected

“Network” category, click the black arrow to open, and select the following

dhclient
dnsmasq
dnsutils
ethtool
iwd
networkmanager
nss-mdns
openssh
usb_modeswitch

- "firewall" category, click on the box to select all three packages
- "packages management" category, click the black arrow to open, and select the following
  - downgrade
  - pacman-contrib
  - pkgfile
  - rebuild-detector
  - reflector
- "desktop integration" category, click the black arrow to open, and select the following
  - bash-completion
  - mlocate
- "filesystem" category, click the black arrow to open, and select the following
  - efitools
  - haveged
  - nfs-utils
  - ntp
  - unrar
  - xz
- "hardware" category, click the black arrow to open, and select the following
  - hdparm
  - hwdetect
  - lsscsi
  - mtools
  - sg3_utils
- Double check your selections, then Click on the "Next icon"

BOOTLOADER SCREEN
- Select "Systemd-boot" which is the default

PARTITIONS SCREEN
- At the top of the window, make sure the proper storage device is selected. SSD or USB 3
- Select the "Erase disk" option
- then select "Swap (no Hibernate) and ext4 with no encryption

USERS SCREEN
- What is your name [your name]
- What name do you want to use to log in [pshare]   (recommended for following how-to's)
- What is the name of this computer [enosServer]   (recommended for following how-to's)
- Enter the desired user password
- PLEASE use a different password for the administrator account (root password).
- Make sure the root password is a strong password. The root password is the most important for good security

Finish the Calamares Install
Shut Down the computer, and remove the liveISO USB thumb drive.

NOTE:
After Shut Down, since the Cassini Neo liveISO is not persistent, ALL changes we made above will be gone and the liveISO is exactly as it was before we started this exercise.

Configure the server

With the keyboard, mouse, and monitor still connected, boot the computer into the newly installed OS. It should boot into the LTS kernel by default. Now we will practice administrating the server with CLI commands.

Log in as user “root” and enter the root password you established in Calamares.

Create a Static IP Address for the server

```
[root@enosServer]# nmcli con
NAME                UUID                                    TYPE      DEVICE
Wired connection 1  xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx     ethernet  enp2s0

[root@enosServer]# ip route
default via 192.168.0.1 dev enp2s0 proto dhcp src 192.168.0.106 metric 100
192.168.0.0/24 dev enp2s0 proto kernel scope link src 192.168.0.106
```

Here is the information we need to change from a DHCP assigned IP to a STATIC IP
Wired connection 1 = the current network connection name
enp2s0 = the device name of the computer’s ethernet device
192.168.0.1 = The LAN side IP address of your router or switch
192.168.0.106 = The current IP address of the server computer assigned by DHCP
192.168.0.0/24 = The current range of the LAN’s domain, 192.168.0.0 through 192.168.0.255

Your information may be different, adjust as necessary. Write this information down.
Convention says, 192.168.0.0 and 192.168.0.255 are reserved and shouldn’t be assigned to anything.

Next we modify networkmanager connection named “Wired connection 1” or whatever is listed for your computer.
NOTE: The static IP address you choose should have the last triad between 150 and 250 for the best compatibility with most routers. In the next command, be sure to notice the back slash \ or absence thereof at the end of the commands.

```
[root@enosServer]# nmcli con mod “Wired connection 1” \
> ipv4.addresses “192.168.0.168/24” \         (your desired Static IP address)
> ipv4.gateway “192.168.0.1” \                ( IP address of the LAN side of router or switch)
> ipv4.dns “192.168.0.1,8.8.8.8” \            ( IP address for your desired DNS server(s))
> ipv4.method “manual”                        (NOTE: no back slash on this command)
```

ipv4.gateway = “The LAN side IP address of your router or switch” listed above
ipv4.dns = “The LAN side IP address of your router or switch” comma “8.8.8.8” = Google DNS
You can omit 8.8.8.8 Google DNS server or substitute another DNS server of your choice.

```
[root@enosServer]# systemctl reboot
```

Log back in as root.
[root@enosServer]# ip addr

should show new static IP address in “2: enp2s0” line 3 as “192.168.0.168”

The section lableled 2: should show your device name in the first line and the device is UP. Your new static IP address is on the third line.

```
2: enp2s0: <BROADCAST,MULTICAST, UP,LOWER_UP> mtu 1500 qdisc fq_code1 state UP etc.
    link/ether bc:xx:xx:xx:xx:xx brd 192.168.0.255 scope global noprefixroute enp2s0
    inet 192.168.0.168/24 brd 192.168.0.255 scope global noprefixroute enp2s0
```

Our static IP address is complete.

Now for the controversy. For a personal server where one person will be doing all the administrative work, I prefer to not have sudo installed. If someone hacks into your server as pshare (user) and sudo is active, they only need to know one password to access root privileges. With sudo not installed, they would need to know both the user password AND the root password. Assuming these two passwords are not the same. In ‘Configure SSD’, we will set a parameter to ‘PermitRootLogin no’ so no one can log in through SSH as root. One has to login as user. With no sudo, they would have to login as user, and then switch to root. It’s your server, do as you wish. To delete sudo, use su to become root:

```
[root@enosServer]# pacman -Rc sudo
```

Configure SSH

In either vi or nano, edit the file /etc/ssh/sshd_config

change #Port 22 to Port xxxx   (uncommet & choose a port number between 8000 and 48000)
change #PermitRootLogin prohibit-password to PermitRootLogin no
change #PasswordAuthentication yes to PasswordAuthentication yes
change # PermitEmptyPasswords no to PermitEmptyPasswords no

[root@enosServer]# systemctl status sshd.service
should show sshd is enabled and active (running) and Server is listening on port xxxx.

If not enabled and active (running)
[root@enosServer]# systemctl enable --now sshd.service
recheck systemctl status sshd.service

If active and running but not on your specified port
[root@enosServer]# systemctl restart sshd.service
recheck systemctl status sshd.service

Configure firewalld

```
[ root@enosServer]#  firewall-cmd --reload
success
[ root@enosServer]#  firewall-cmd --permanent --zone=public --service=ssh  --remove-port=22/tcp
success
[ root@enosServer]#  firewall-cmd --permanent --zone=public --service=ssh  --add-port=xxxx/tcp
success
[ root@enosServer]#  firewall-cmd --permanent --zone=public –remove-service=dhcpv6-client
success
[ root@enosServer]#  firewall-cmd --permanent --zone=public --add-source=192.168.0.0/24
success
[ root@enosServer]#  firewall-cmd --permanent --zone=public --remove-forward
success
[ root@enosServer]#  firewall-cmd --reload
success
[ root@enosServer]# firewall-cmd --list-all
public (active)
   target: default
   icmp-block-inversion: no
   interfaces: enp2s0
   sources: 192.168.0.0/24
   services: ssh
   ports:
   protocols:
   forward: no
   masquerade: no
   forward-ports:
   source-ports:
   icmp-blocks:
   rich-rules:
```

In --service=ssh –add-port=xxxx/tcp   xxxx equals the ssh port you assigned above.
In –-add-source=192.168.0.0/24   the IP address should be the "current range of the LAN's domain" listed above . Adjust the IP as necessary. The firewall now only allows in computers from your LAN.

[root@enosServer]# systemctl poweroff   (use systemctl poweroff to power down the computer)

Do not boot back into the server until prompted in the next section.

SETUP DATA STORAGE DEVICE

If the server computer is not Shut Down, then.

```
[root@enosServer]# systemctl poweroff
```

Install a SSD of sufficient size to sata port 2. If you are only tinkering around, you can use a USB 3 thumb drive for the Data storage device.

Power up the server computer. Let's get some experience on how to remotely administer your server.

In a Linux computer with a full blown Desktop Environment or Window Manager, launch your favorite terminal app. In the following command substitute the static IP address of the server, and the ssh port you set up during server configuration.

```
[Odin@Valhalla]$ ssh pshare@Server-Static-IP -p Server-USB-Port-Number
```

You should get the following on your screen.

```
The authenticity of host '[192.168.X.XXX]:XXXX ([192.168.X.XX]:XXXx)' can't be established.
ED25519 key fingerprint is XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? Yes
```

Answer with the full word "yes" and it should continue and ask for pshare's password.
On subsequent remote logins, it should simply ask for pshare's password.

Notice that the terminal's prompt changed from [Odin@Vallhala] to [pshare@enosServer]. This is the way to remember what computer you are currently in. Get in the habit of checking the prompt before entering commands. This habit will save a lot of grief.

The lions share of administration will be done as root, and I prefer to run personal servers with no sudo.
enter

```
[pshare@enosServer ~]$ su -   (the dash after su indicates become root and go to root's home directory)
Enter root password
[root@enosServer ~]#
```

Partition and Format DATA SSD

```
[root@enosServer]# lsblk -f
NAME   FSTYPE FSVER LABEL      UUID                                 FSAVAIL FSUSE% MOUNTPOINTS
sda
├─sda1 vfat   FAT32            0445-1E26                             856.8M    14% /efi
├─sda2 ext4   1.0   endeavouros af073cbe-a1d8-4e75-ad4b-442aa0679af5   46.2G     5% /
└─sda3 swap   1     swap       3d04b03c-ec2e-4c4a-888f-7998292e9c8b                [SWAP]
sdb
├─sdb1 vfat   FAT16 BOOT_EOS   FCCB-B66C
└─sdb2 ext4   1.0   ROOT_EOS   07266f80-f66e-4c7d-b7bd-6a3e4499f2dc
```

In this example, the OS was installed on a 60 GB SSD. Several things tell me sda is the 60 GB SSD the sda2 partition is “ext4”, has “endeavouros” for a lable, with 46.2GB available. Also, partition one is mounted at /efi and partition two is mounted at / (root).

Device sdb is the newly installed data 500 GB SSD. Partition and format the SSD.

```
[root@enosServer]# fdisk /dev/sdb          (change device name if necessary)
Command: g                                 (create an new empty GPT partition table)
Command: n                                 (add a new partition)
Partition number: 1                        (first partition)
First sector:                              (Press Enter to accept default)
Last sector:                               (Press Enter to accept default)
            If you get:
                Partition #1 contains a ext4 signature.
                Do you want to remove the signature? [Y]es/[N]o:
            Type in yes
 Command: p                                (print the partition table)
Command: w                                 (write table to disk and exit)
```

```
[root@enosServer]#  mkfs.ext4 -L DATA /dev/sdb1     (format the DATA partition)
```
If the prompt ‘Proceed anyway? (y/n)’ appears, type y

Edit /etc/fstab to mount data device at boot

```
[root@enosServer]# lsblk -f
NAME   FSTYPE FSVER LABEL      UUID              FSAVAIL FSUSE% MOUNTPOINTS
sda
├─sda1 vfat                 FAT32      0445-1E26     856.8M    14%       /efi
├─sda2 ext4   1.0   endeavouros af073cbe-a1d8-4e75-ad4b-442aa0679af5   46.2G   5%    /
└─sda3 swap   1     swap         3d04b03c-ec2e-4c4a-888f-7998292e9c8b                [SWAP]
sdb
└─sdb1 ext4   1.0   DATA         949d52ae-0ecd-4b56-b3c3-80c58aac946f
```

Highlight the UUID for the DATA storage device, then copy the UUID to the clipboard
using vi or nano, edit /etc/fstab and add the following line to the /etc/fstab file.
After typing in “UUID=” then paste the UUID number and finish the line.

```
UUID=949d52ae-0ecd-4b56-b3c3-80c58aac946f   /server    ext4   defaults,noatime 0 2
```

close file /etc/fstab

Create mount points for the DATA SSD, and future DATABKUP SSD

```
[root@enosServer]#  mkdir /server /serverbkup         (create mount points for SSD’s)
[root@enosServer]# id pshare
uid=1000(pshare) gid=1000(pshare) groups=1000(pshare),3(sys),998(wheel),982(rfkill)
      (Notice that pshare is not in group users)
[root@enosServer]# gpasswd -a pshare users
[root@enosServer]# id pshare
uid=1000(pshare) gid=1000(pshare) groups=1000(pshare),3(sys),998(wheel),984(users),982(rfkill)
      (Notice that pshare is now in 984(users)
[root@enosServer]# systemctl reboot                  (wait about one minute)

[Odin@Valhalla]$ ssh pshare@Server-Static-IP -p Server-USB-Port-Number
[pshare@enosServer]$ su -
[root@enosServer]# chown root:users /server /serverbkup     (set ownerships)
[root@enosServer]#  chmod 774 /server /serverbkup            (set permissions)
[root@enosServer]# ls -al /
drwxrwxr--   3 root users  4096 Feb  9 16:37 server
drwxrwxr--   2 root users  4096 Feb  9 16:49 serverbkup
```

Check that the permissions and ownerships agree with the above.

```
[root@enosServer]# systemctl reboot
```

The terminal window will revert back to [Odin@Valhalla]$

Your headless server is now configured and ready to use.
After a minute, in a Linux Client terminal window ssh into enosServer. ssh into the server,
Log in as user **pshare**

```
[pshare@enosServ]$ ls -al /
drwxrwxr--   3 root users  4096 Feb  9 16:37 server
drwxrwxr--   2 root users  4096 Feb  9 16:49 serverbkup
```

Make sure the persmissions and ownershipts agree with the above.

```
[pshare@enosServer]# ls -al /server
drwx------  2 root root  16384 Feb  9 16:37 lost+found
```

you should see the directory ‘lost+found’ created as root when the partition was formatted with ext4

```
[pshare@enosServer]$ cat > /server/test       (type in the two following lines then press <Ctrl d>)
This is a test.
The quick brown fox jumped over the lazy dog’s back.
<Ctrl d>

[pshare@enosServer]$ ls -al /server
drwx------  2 root   root   16384 Feb  9 16:37 lost+found
-rw-r--r--  1 pshare pshare    69 Feb  9 17:45 test

[pshare@enosServer]$ cat /server/test
This is a test.
The quick brown fox jumped over the lazy dog's back.
```

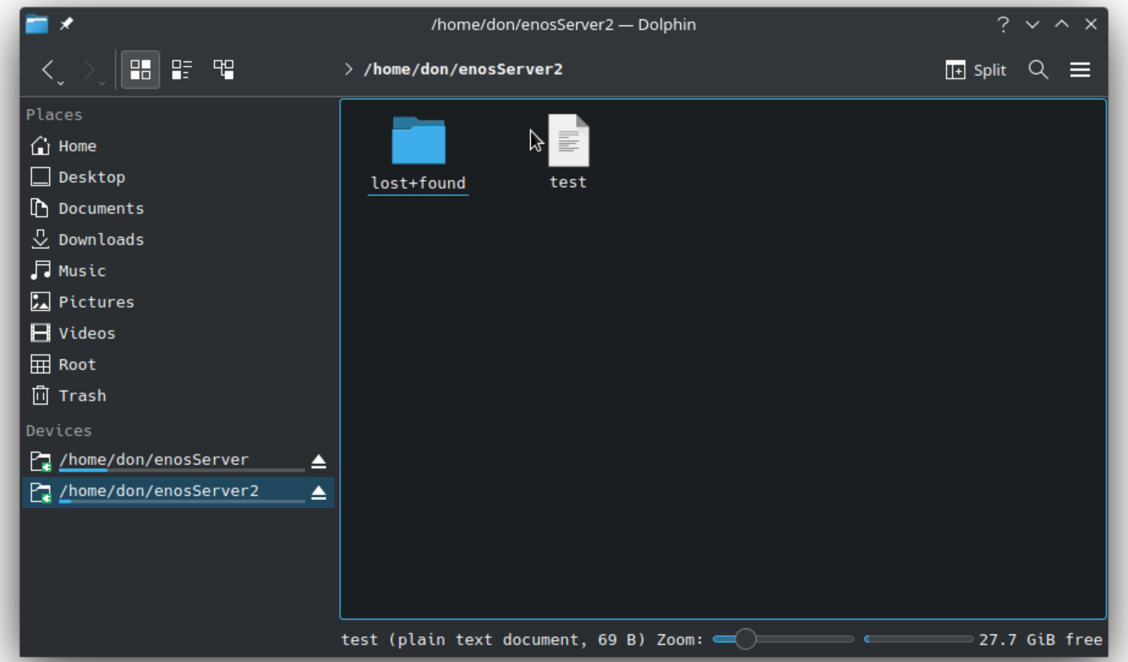
Notice that /server/test has pshare as ownership for both user and group ownerships.
The lost+found directory should be the only thing in /server owned by root. Everything else should be owned by pshare. Later, when we manipulate files remotely, it will be done as a user.
In the Linux client computer, open your favorite terminal.

```
[odin@vallhala]$ pacman -Q sshfs     (should list sshfs, if not install sshfs)
```

In your home directory, make a directory name enosServer

```
[odin@vallhala]$ mkdir /home/odin/enosServer
[odin@vallhala]$ sshfs -p port-number pshare@ServerIP:/server   /home/odin/enosServer
```

Open your favorite file manager, and you should see something similar to this.



You now have a personal LAN server that has no third party software. Every package is available in the Archlinux repositories. No third party software means you are in complete control.

Your personal LAN server now needs a Linux computer setup as a client. Go to
https://discovery.endeavouros.com/category/arm/
and follow these tutorials. They are in the ARM section, but they work the same for x86_64 devices

HOMESERVER 1 Set up a Linux client computer
HOMESERVER 2 Use FUSE and SSHFS to view server data in a file manager
HOMESERVER 3 Automatically mount server at log-in

HOMESERVER 4 installs SAMBA on the enosServer to serve a Windows Client computer
HOMESERVER 5 configures a Windows Client computer for SAMBA
HOMESERVER 6 installs and configures a miniDLNA server
HOMESERVER 7 sets up an USB 3 storage device to backup the DATA on.