“This isn’t really a whodunnit but how he did it”

“This is not a how to do it but how I did it”

Debian File Serve

A noobs experience

A little about me

- I’ve been using Linux for 9 months
  - Computational thinking and understanding the machine
  - During lockdown
  - OneDrive
- First distro Linux Mint
- Current Arch with Qtile WM

Overview

- Hardware
- Debian vs Ubuntu vs Other
- Rsnapshot
- SAMBA
- Music / Media Server and other uses
- Final thoughts

Hardware

Motherboard

Intel D525MW ~ Intel Atom and 1GB DDR3 RAM Motherboard Mini ITX Bundle + VGA

Harddrive

Seagate IronWolf 2TB NAS Hard Drive 3.5" SATA III 6GB's 5900RPM 64MB Cache

Case

Thermaltake Core V1 Snow Mini-ITX Cube Case With Side Window

Debian vs Ubuntu vs Other

Looking at the installers there seemed to be only a couple of differences that affected me.

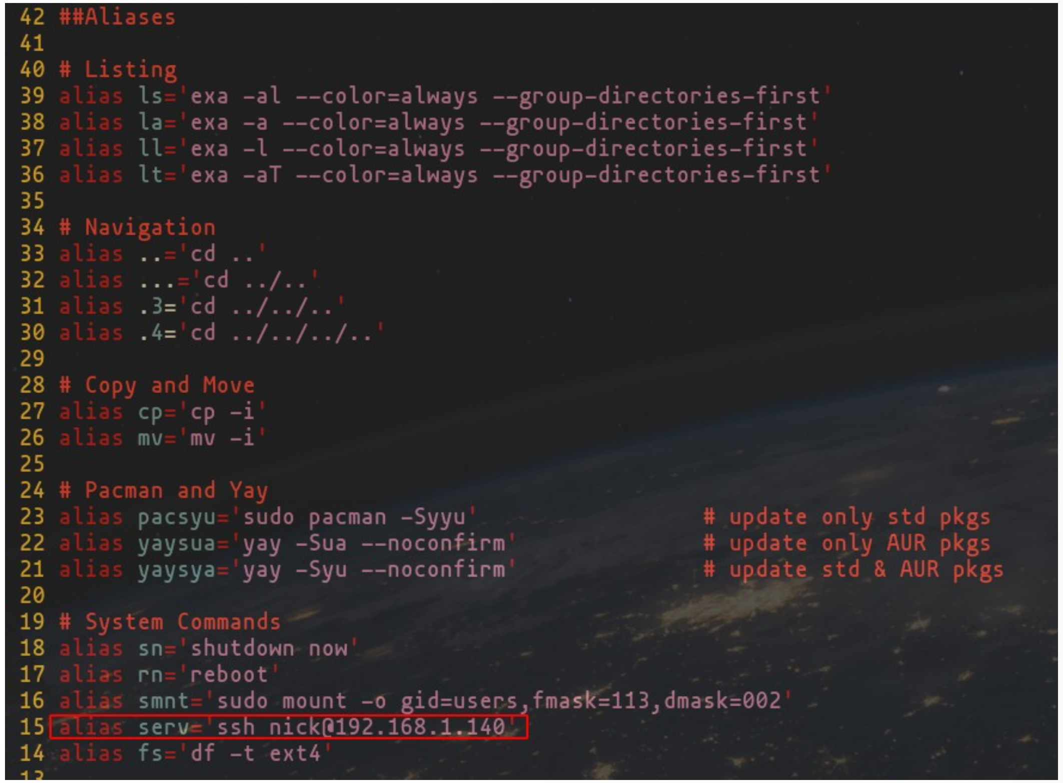
- Ubuntu offered a server install that inlcuded SAMBA
- Ubuntu offered automatic security updates
- There are of course many differences between Ubuntu server and vanilla Debian but …
- Why not CentOS or other?

Backing up

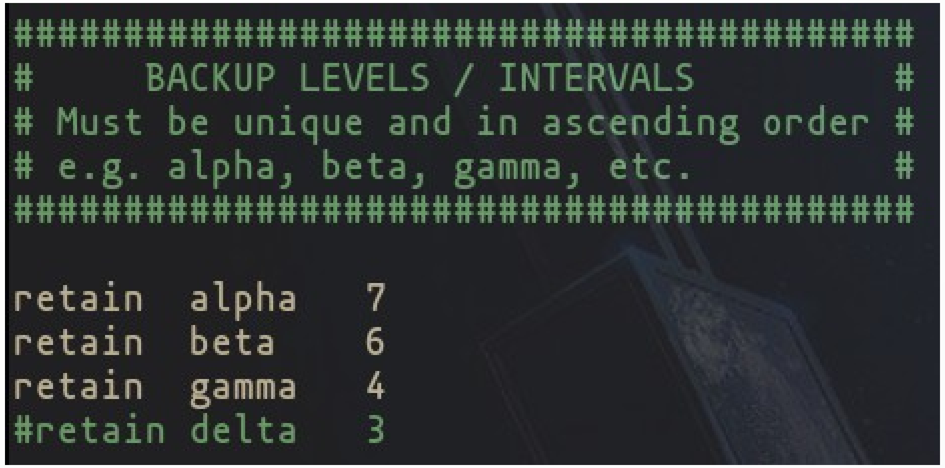
- Arch is a rolling release
- This gives it a reputation of instability
- Back ups allow you keep up to date with the ability to go back to a previous “version” if something breaks
- First attempt was to rsync via SAMBA

Rsnapshot

- Uses rsync via SSH
  - Install rsnapshot rysnc and openSSH
  - SSH passwordless keys
- Rotate files



```
##Aliases

# Listing
alias ls='exa -al --color=always --group-directories-first'
alias la='exa -a --color=always --group-directories-first'
alias ll='exa -l --color=always --group-directories-first'
alias lt='exa -aT --color=always --group-directories-first'

# Navigation
alias ..='cd ..'
alias ...='cd ../..'
alias .3='cd ../../..'
alias .4='cd ../../../..'

# Copy and Move
alias cp='cp -i'
alias mv='mv -i'

# Pacman and Yay
alias pacsyu='sudo pacman -Syyu'                 # update only std pkgs
alias yaysua='yay -Sua --noconfirm'              # update only AUR pkgs
alias yaysya='yay -Syu --noconfirm'              # update std & AUR pkgs

# System Commands
alias sn='shutdown now'
alias rn='reboot'
alias smnt='sudo mount -o gid=users,fmask=113,dmask=002'
alias serv='ssh nick@192.168.1.140'
alias fs='df -t ext4'
```

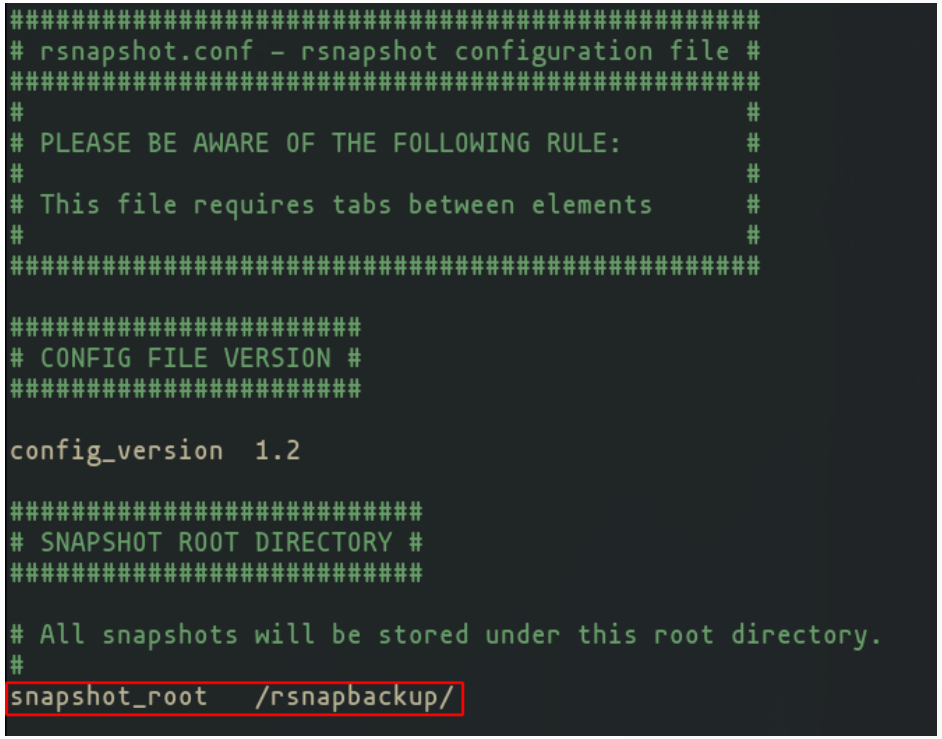


```
#########################################
#     BACKUP LEVELS / INTERVALS         #
# Must be unique and in ascending order #
# e.g. alpha, beta, gamma, etc.         #
#########################################

retain  alpha   7
retain  beta    6
retain  gamma   4
#retain delta   3
```

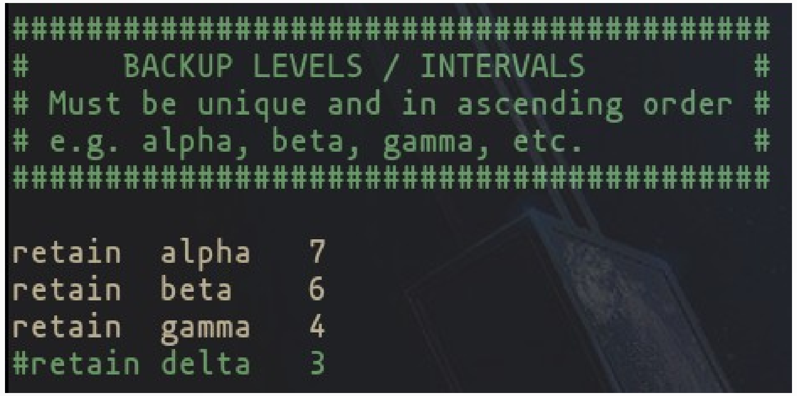
Rsnapshot Setup

https://ostechnix.com/setup-backup-server-using-rsnapshot-linux/

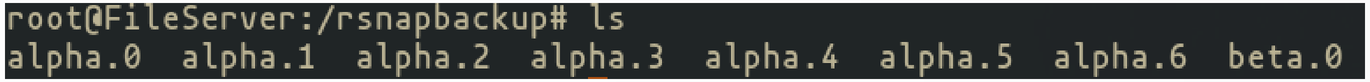
- On server
  - sudo mkdir */rsnapbackup/*
  - sudo cp /*etc*/rsnapshot.conf /*etc*/rsnapshot.conf.bak
  - sudo vim /*etc*/rsnapshot.conf



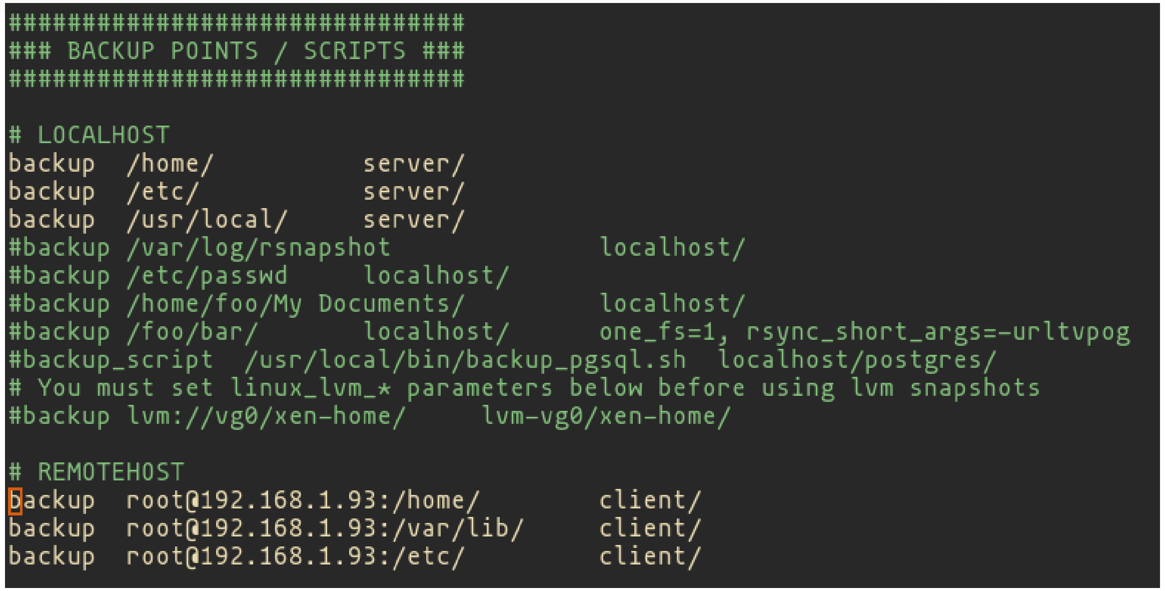
```
#################################################
# rsnapshot.conf - rsnapshot configuration file #
#################################################
#                                               #
# PLEASE BE AWARE OF THE FOLLOWING RULE:        #
#                                               #
# This file requires tabs between elements      #
#                                               #
#################################################

#######################
# CONFIG FILE VERSION #
#######################

config_version  1.2

###########################
# SNAPSHOT ROOT DIRECTORY #
###########################

# All snapshots will be stored under this root directory.
#
snapshot_root   /rsnapbackup/
```



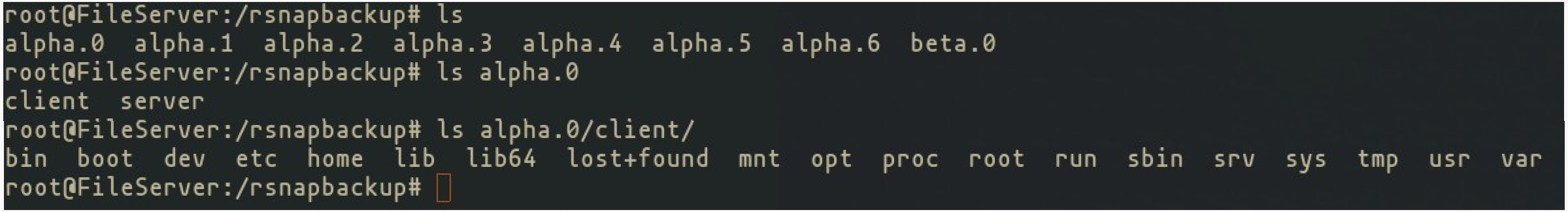
```
#########################################
#     BACKUP LEVELS / INTERVALS         #
# Must be unique and in ascending order #
# e.g. alpha, beta, gamma, etc.         #
#########################################

retain  alpha   7
retain  beta    6
retain  gamma   4
#retain delta   3
```



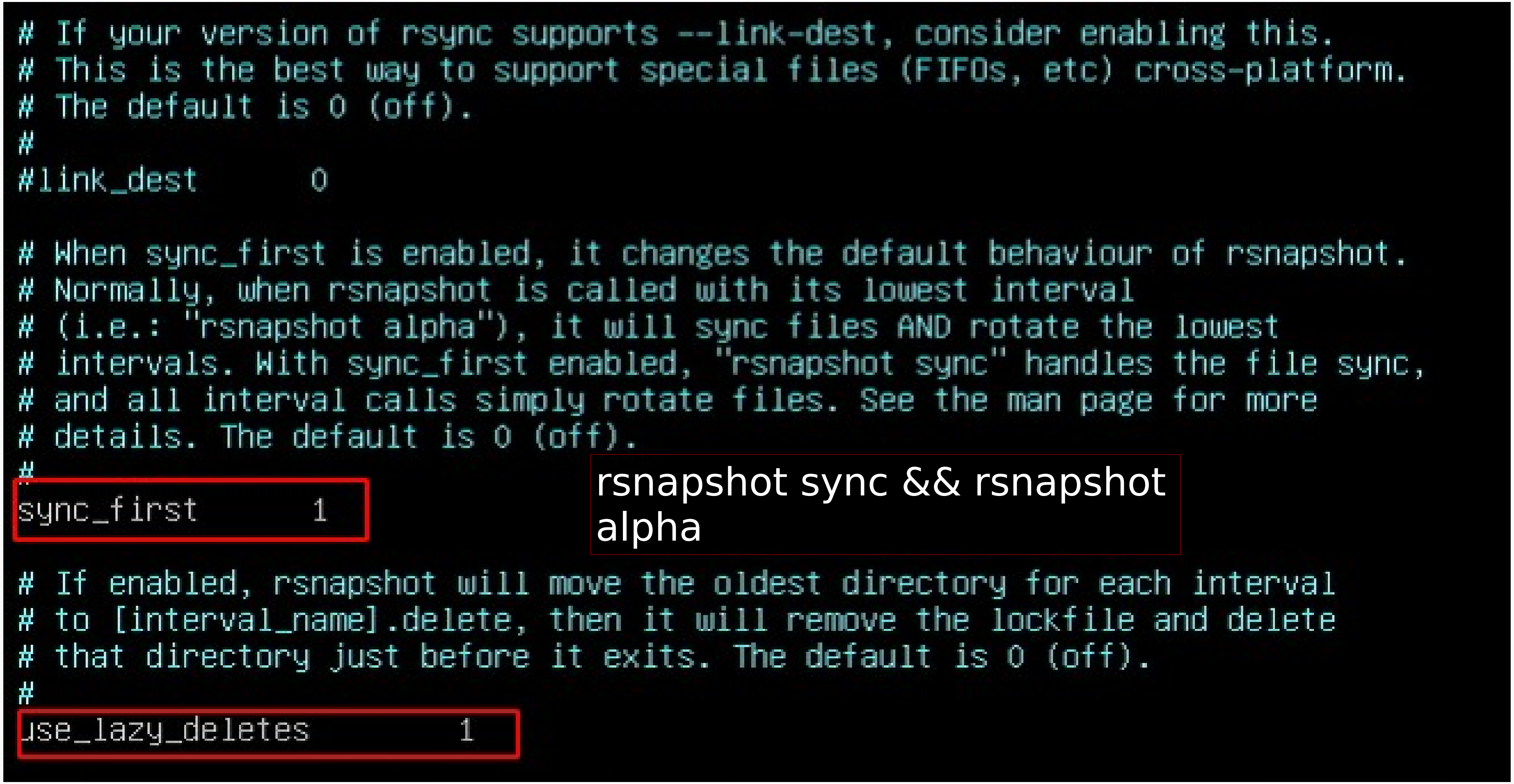
```
root@FileServer:/rsnapbackup# ls
alpha.0  alpha.1  alpha.2  alpha.3  alpha.4  alpha.5  alpha.6  beta.0
```



```
###############################
### BACKUP POINTS / SCRIPTS ###
###############################

# LOCALHOST
backup  /home/          server/
backup  /etc/           server/
backup  /usr/local/     server/
#backup /var/log/rsnapshot              localhost/
#backup /etc/passwd     localhost/
#backup /home/foo/My Documents/         localhost/
#backup /foo/bar/       localhost/      one_fs=1, rsync_short_args=-urltvpog
#backup_script  /usr/local/bin/backup_pgsql.sh  localhost/postgres/
# You must set linux_lvm_* parameters below before using lvm snapshots
#backup lvm://vg0/xen-home/     lvm-vg0/xen-home/

# REMOTEHOST
backup  root@192.168.1.93:/home/        client/
backup  root@192.168.1.93:/var/lib/     client/
backup  root@192.168.1.93:/etc/         client/
```

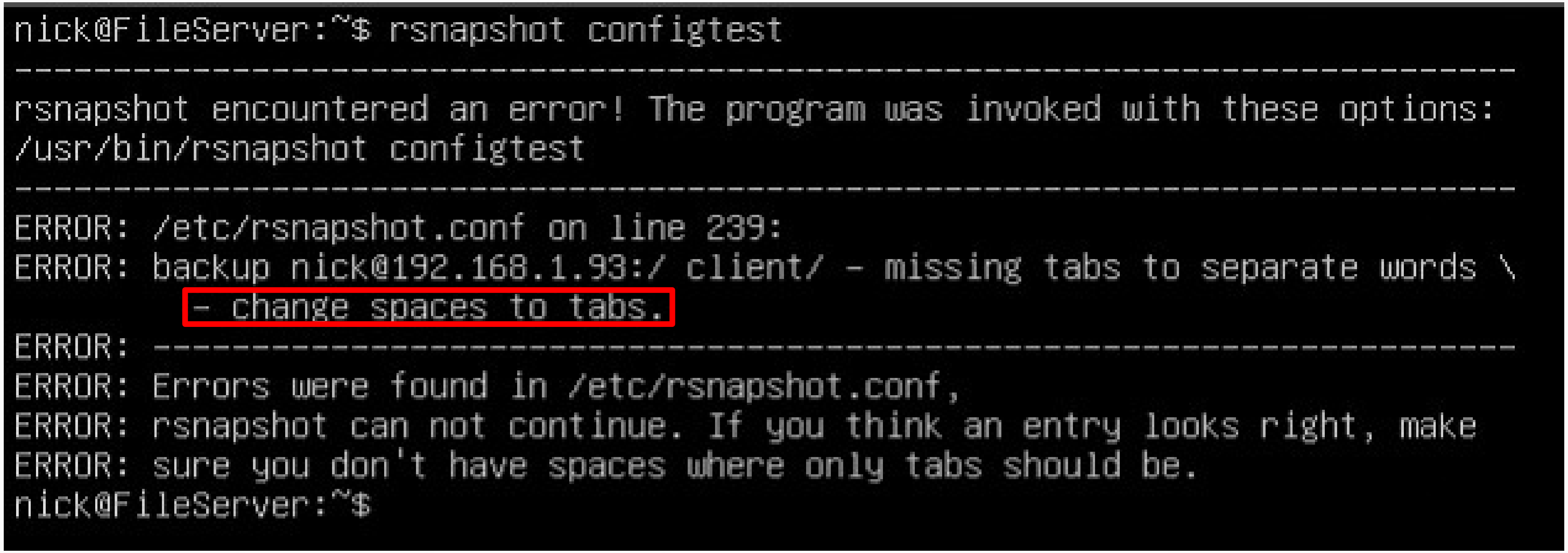


```
root@FileServer:/rsnapbackup# ls
alpha.0  alpha.1  alpha.2  alpha.3  alpha.4  alpha.5  alpha.6  beta.0
root@FileServer:/rsnapbackup# ls alpha.0
client  server
root@FileServer:/rsnapbackup# ls alpha.0/client/
bin  boot  dev  etc  home  lib  lib64  lost+found  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@FileServer:/rsnapbackup# 
```



```
# If your version of rsync supports --link-dest, consider enabling this.
# This is the best way to support special files (FIFOs, etc) cross-platform.
# The default is 0 (off).
#
#link_dest      0

# When sync_first is enabled, it changes the default behaviour of rsnapshot.
# Normally, when rsnapshot is called with its lowest interval
# (i.e.: "rsnapshot alpha"), it will sync files AND rotate the lowest
# intervals. With sync_first enabled, "rsnapshot sync" handles the file sync,
# and all interval calls simply rotate files. See the man page for more
# details. The default is 0 (off).
#
sync_first      1

# If enabled, rsnapshot will move the oldest directory for each interval
# to [interval_name].delete, then it will remove the lockfile and delete
# that directory just before it exits. The default is 0 (off).
#
use_lazy_deletes        1
```

rsnapshot sync && rsnapshot alpha



```
nick@FileServer:~$ rsnapshot configtest
----------------------------------------------------------------------------
rsnapshot encountered an error! The program was invoked with these options:
/usr/bin/rsnapshot configtest
----------------------------------------------------------------------------
ERROR: /etc/rsnapshot.conf on line 239:
ERROR: backup nick@192.168.1.93:/ client/ - missing tabs to separate words \
         - change spaces to tabs.
ERROR: ---------------------------------------------------------------------
ERROR: Errors were found in /etc/rsnapshot.conf,
ERROR: rsnapshot can not continue. If you think an entry looks right, make
ERROR: sure you don't have spaces where only tabs should be.
nick@FileServer:~$
```

What I wanted…

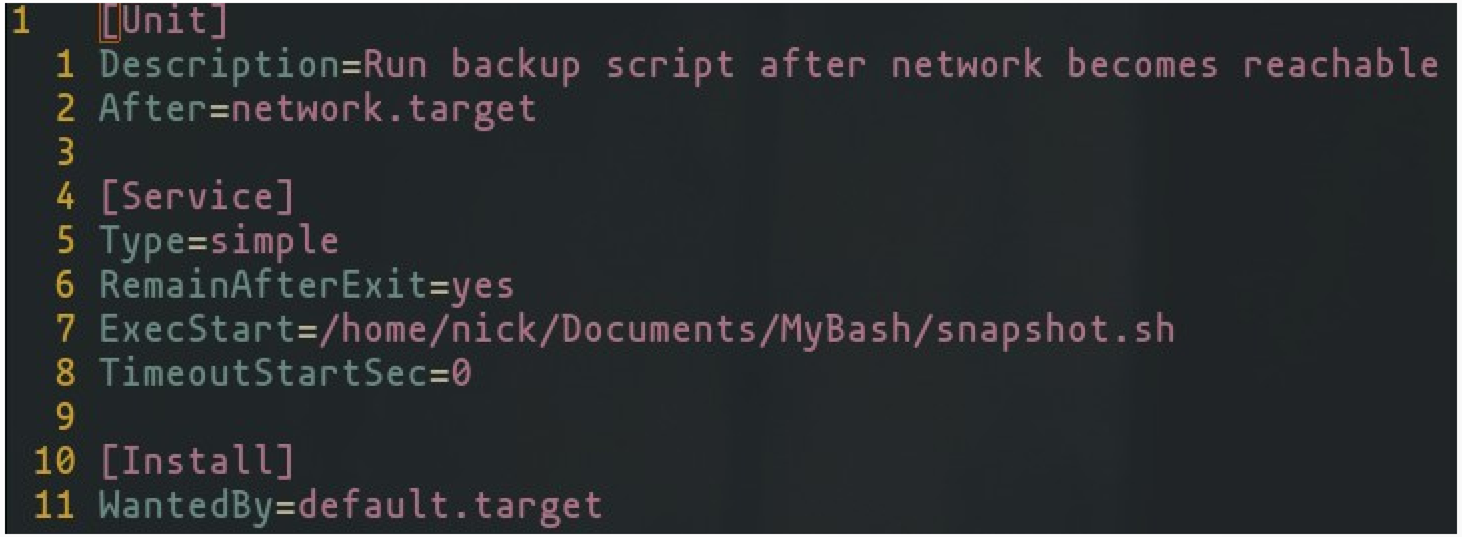
- For my laptop to tell my server to start the backup
- The first few tries
  - Kept being asked for passwords
  - Sudo
  - SSH keys
- Start from the end and work back

Thinking backwards (mostly)…

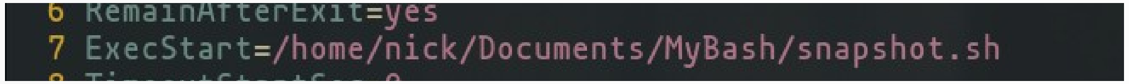
- Server: script that called rsnapshot
  - Then to ssh into my laptop update a log file and send-notify completion
- Laptop: script that ssh'd into my server and called the rsnapshot script
  - Check the log file and check if it should proceed
  - Call script from autostart.sh

A more linear explanation

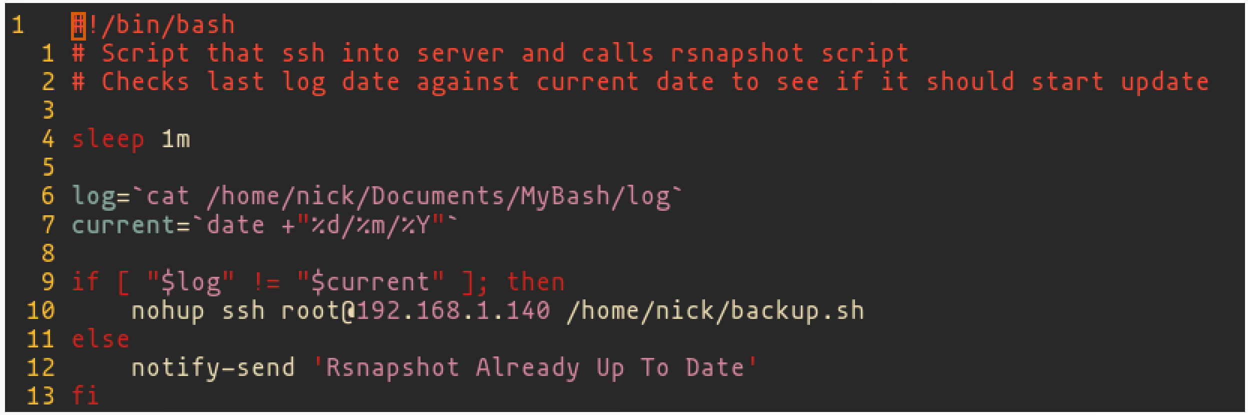
- In the end calling the script from autostart.sh worked but caused other issues.
- Made a systemd .service file
  - https://www.golinuxcloud.com/run-script-at-startup-boot-without-cron-linux/



```
[Unit]
Description=Run backup script after network becomes reachable
After=network.target

[Service]
Type=simple
RemainAfterExit=yes
ExecStart=/home/nick/Documents/MyBash/snapshot.sh
TimeoutStartSec=0

[Install]
WantedBy=default.target
```



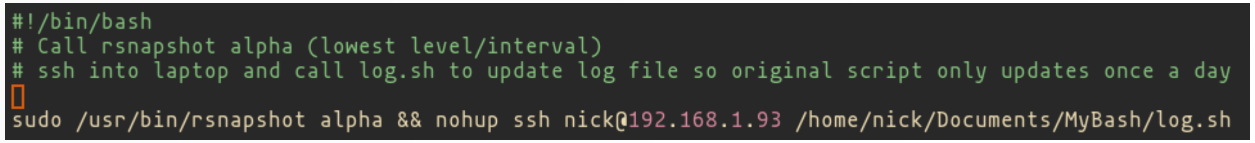
```
RemainAfterExit=yes
ExecStart=/home/nick/Documents/MyBash/snapshot.sh
TimeoutStartSec=0
```



```
#!/bin/bash
# Script that ssh into server and calls rsnapshot script
# Checks last log date against current date to see if it should start update

sleep 1m

log=`cat /home/nick/Documents/MyBash/log`
current=`date +"%d/%m/%Y"`

if [ "$log" != "$current" ]; then
    nohup ssh root@192.168.1.140 /home/nick/backup.sh
else
    notify-send 'Rsnapshot Already Up To Date'
fi
```

Laptop:
snapshot.service calls snapshot.sh

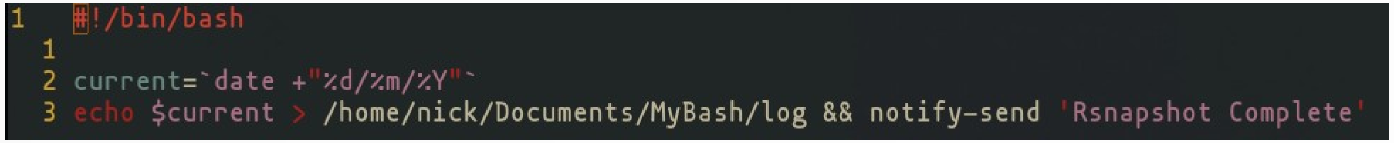
snapshot.sh checks log and if not todays calls backup.sh on server



```
#!/bin/bash
# Call rsnapshot alpha (lowest level/interval)
# ssh into laptop and call log.sh to update log file so original script only updates once a day

sudo /usr/bin/rsnapshot alpha && nohup ssh nick@192.168.1.93 /home/nick/Documents/MyBash/log.sh
```

Server:
backup.sh calls rnsapshot alpha

Then calls log.sh on laptop



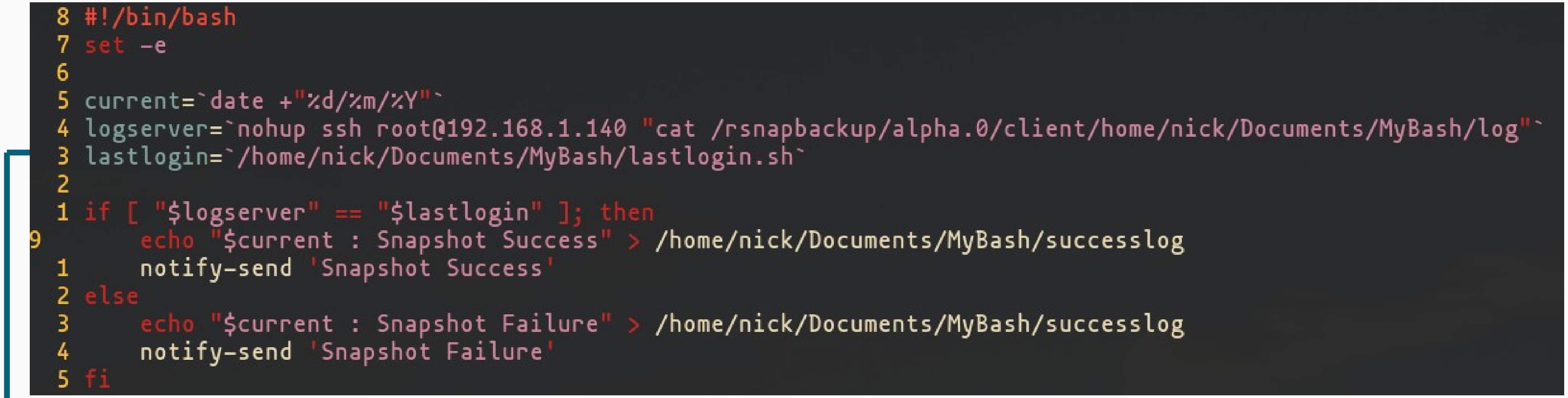
```
#!/bin/bash

current=`date +"%d/%m/%Y"`
echo $current > /home/nick/Documents/MyBash/log && notify-send 'Rsnapshot Complete'
```

Laptop:
log.sh calls updates log (text file) with todays date

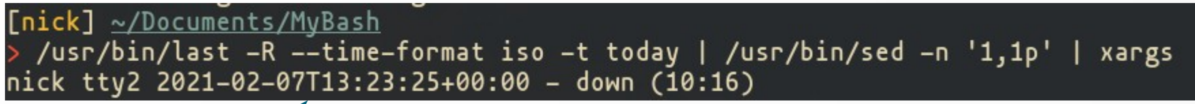
Then sends notification that it is complete

Rsnapshot && issue

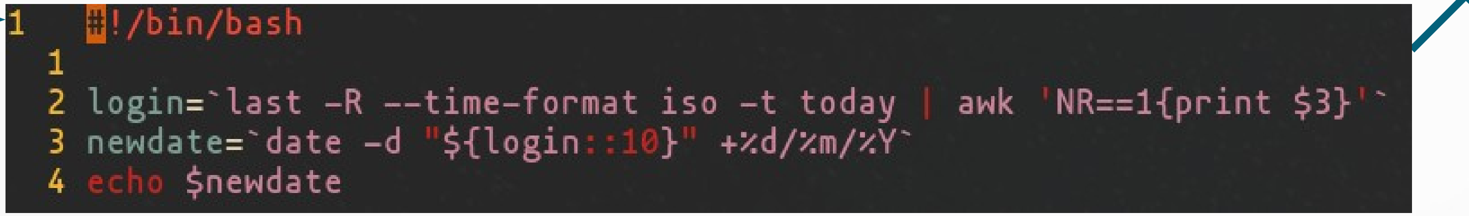
I originally had an issue with Rsnapshot not giving the correct exit code. This is why I changed to backing up as root and of specific folders.



```bash
#!/bin/bash
set -e

current=`date +"%d/%m/%Y"`
logserver=`nohup ssh root@192.168.1.140 "cat /rsnapbackup/alpha.0/client/home/nick/Documents/MyBash/log"`
lastlogin=`/home/nick/Documents/MyBash/lastlogin.sh`

if [ "$logserver" == "$lastlogin" ]; then
    echo "$current : Snapshot Success" > /home/nick/Documents/MyBash/successlog
    notify-send 'Snapshot Success'
else
    echo "$current : Snapshot Failure" > /home/nick/Documents/MyBash/successlog
    notify-send 'Snapshot Failure'
fi
```



```
[nick] ~/Documents/MyBash
> /usr/bin/last -R --time-format iso -t today | /usr/bin/sed -n '1,1p' | xargs
nick tty2 2021-02-07T13:23:25+00:00 - down (10:16)
```

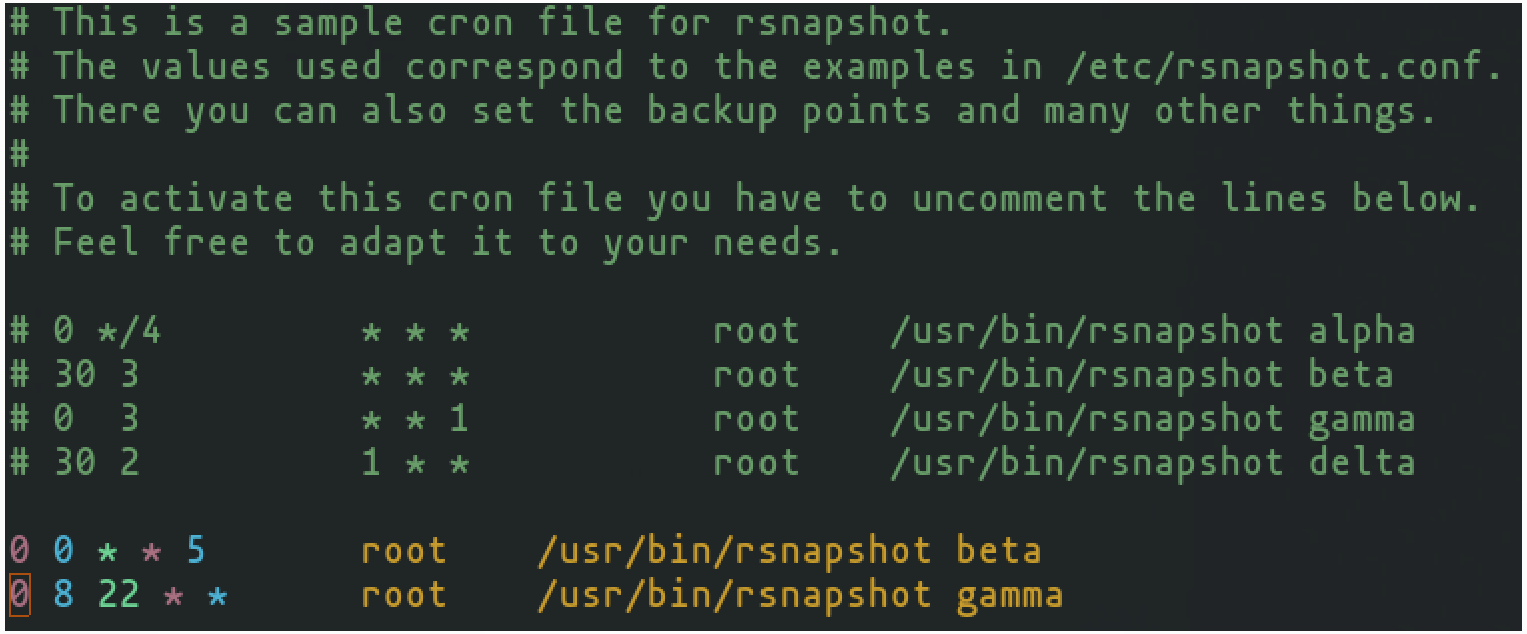


```bash
#!/bin/bash

login=`last -R --time-format iso -t today | awk 'NR==1{print $3}'`
newdate=`date -d "${login::10}" +%d/%m/%Y`
echo $newdate
```

Crontab on server

https://ostechnix.com/setup-backup-server-using-rsnapshot-linux/

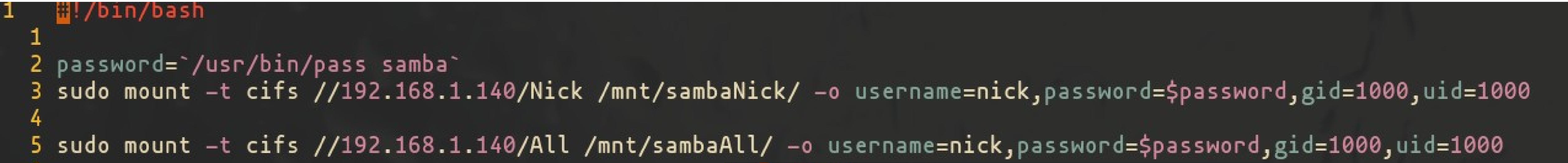
- sudo vim /*etc*/cron.d/rsnapshot



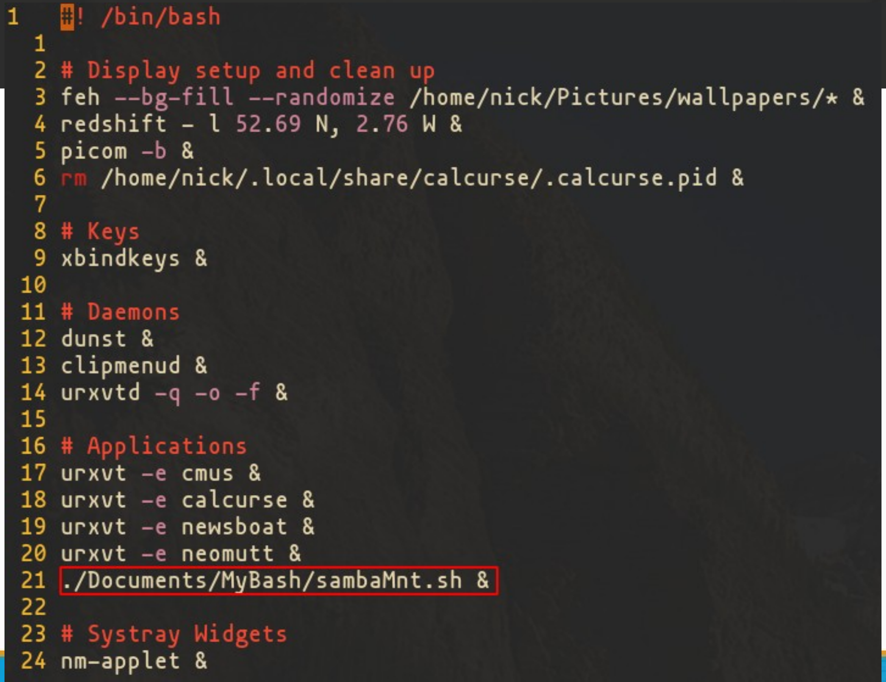
```
# This is a sample cron file for rsnapshot.
# The values used correspond to the examples in /etc/rsnapshot.conf.
# There you can also set the backup points and many other things.
#
# To activate this cron file you have to uncomment the lines below.
# Feel free to adapt it to your needs.

# 0 */4         * * *           root    /usr/bin/rsnapshot alpha
# 30 3          * * *           root    /usr/bin/rsnapshot beta
# 0  3          * * 1           root    /usr/bin/rsnapshot gamma
# 30 2          1 * *           root    /usr/bin/rsnapshot delta

0 0 * * 5       root    /usr/bin/rsnapshot beta
0 8 22 * *      root    /usr/bin/rsnapshot gamma
```

SAMBA

- Mainly a way for a windows machine to “backup” individual files
- You can have it mounted by fstab



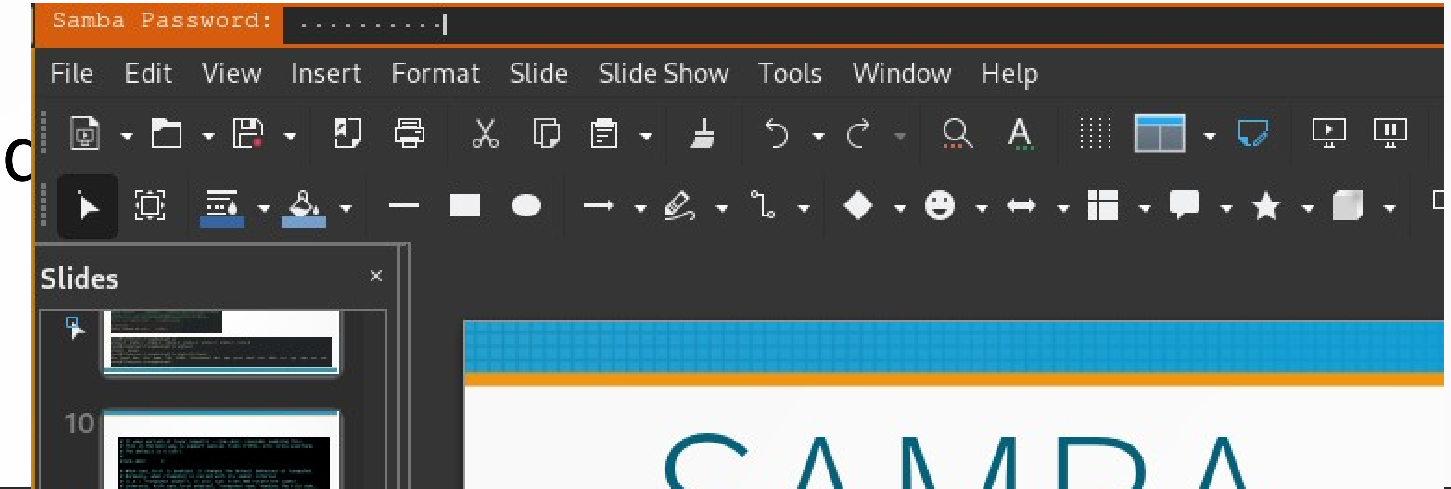
```
#!/bin/bash

password=`/usr/bin/pass samba`
sudo mount -t cifs //192.168.1.140/Nick /mnt/sambaNick/ -o username=nick,password=$password,gid=1000,uid=1000

sudo mount -t cifs //192.168.1.140/All /mnt/sambaAll/ -o username=nick,password=$password,gid=1000,uid=1000

notify-send 'Samba File Share Mounted'
```

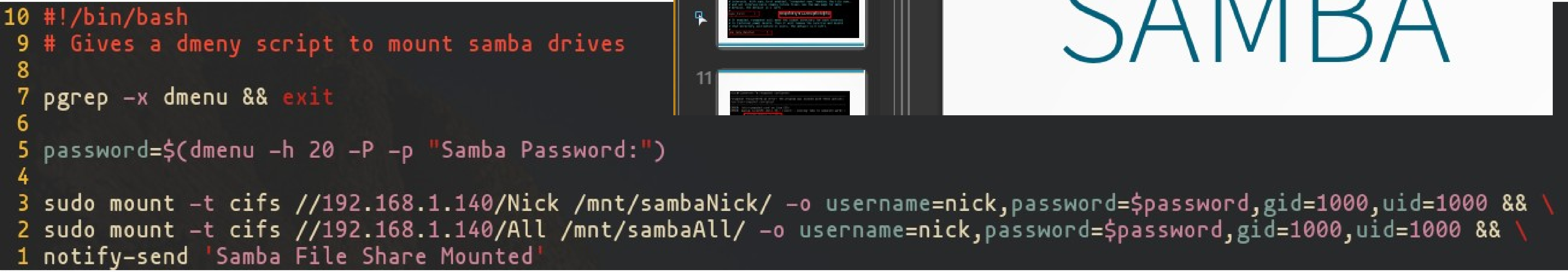


```
#! /bin/bash

# Display setup and clean up
feh --bg-fill --randomize /home/nick/Pictures/wallpapers/* &
redshift - l 52.69 N, 2.76 W &
picom -b &
rm /home/nick/.local/share/calcurse/.calcurse.pid &

# Keys
xbindkeys &

# Daemons
dunst &
clipmenud &
urxvtd -q -o -f &

# Applications
urxvt -e cmus &
urxvt -e calcurse &
urxvt -e newsboat &
urxvt -e neomutt &
./Documents/MyBash/sambaMnt.sh &

# Systray Widgets
nm-applet &
```

SAMBA

- If you don’t like storing passwords
- dmenu
  - Password patch need


Samba Password: ....



```
#!/bin/bash
# Gives a dmeny script to mount samba drives

pgrep -x dmenu && exit

password=$(dmenu -h 20 -P -p "Samba Password:")

sudo mount -t cifs //192.168.1.140/Nick /mnt/sambaNick/ -o username=nick,password=$password,gid=1000,uid=1000 && \
sudo mount -t cifs //192.168.1.140/All /mnt/sambaAll/ -o username=nick,password=$password,gid=1000,uid=1000 && \
notify-send 'Samba File Share Mounted'
```

Media Server

- Plex
- Volumio
- Camera (Raspberry Pi)

Final Thoughts

- I’ve learnt a lot
- I don’t think I would have started/finished it without SLUG
- Making a presentation on it meant I also had to know how I got it to work and made me acknowledge where it is still not finished.

Any Questions/Suggestions??