

## Creating a NAS with Ubuntu Server

This setup guide will show you how to setup a Network Attached Storage (NAS) system with Ubuntu Server, create Samba / SMB shares, install Usenet file downloader, and install automated Usenet downloaders SickBeard and CouchPotato.

Use this guide along with my Youtube Video: [http://youtu.be/-5Z\\_-3EBIHE](http://youtu.be/-5Z_-3EBIHE)

Terminal commands are written in **Orange**

Settings you need to change for your system are written in **Purple**

Items Required:

- Wired Network Connection
- Small USB Flash drive (or CD) to install the OS
- 8GB (Minimum) Flash drive to run the OS
- A few spare hours

A NAS device can be created with minimal hardware, such as: Single core 1.0GHz CPU, 512 MB RAM, 160GB Hard Drive, and a USB 2.0 Flash drive.

For my NAS I'm using: AMD E-450 CPU (1.6GHz dual core), 8GB DDR3 RAM, 3x 3TB Seagate hard drives, and a 16GB USB3.0 Flash drive.

The system is running with RAID5 giving me a single 6TB drive mounted in /media. It has SMB shares setup, which allows shared access for all devices (computers, media player, and Android tablet) on my LAN.

I'm also using the system as a Media Downloader to download files from Usenet.

Majority of user interaction is done via Web GUI's, although there's remote access terminal available via SSH. Since everything can be done remotely over the network the NAS sits there running as a headless server (no keyboard or monitor connected).

## Installing Ubuntu Server

Download Ubuntu Server from: <http://www.ubuntu.com/download/server>

Either write it to the small USB Flash drive or CD

We are going to install the operating system to a USB Flash drive, this makes it easier to use the hard drives in a RAID array and keep the OS separate from data storage. Ubuntu Server is light enough to run on a Flash drive, and with enough RAM theres no appreciable delay in using the NAS.

It's certainly a cheaper alternative than installing the OS on a Solid State Disk.

Disconnect all hard drives and then install the Server OS onto the Flash drive.

Near the end of the install it asks you which components you would like to install, from that list choose SSH Server, and Samba.

Shutdown the computer, reconnect all hard drives, and then switch back on.

If all being well you will get a login screen. Login with your username and Password.

### **Firstly Check For Updates**

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo reboot
```

### **Set Static IP Address**

```
sudo nano /etc/network/interfaces
iface eth0 inet static
    address 192.168.1.100
    network 192.168.1.0
    netmask 255.255.255.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
    dns-nameservers 192.168.1.1 8.8.8.8
```

Adjust the IP addresses for your network.

8.8.8.8 is Google DNS Server, I find this a lot better than my ISP's DNS Server.

## Add NAS IP Address to your Local DNS Lookup

This is on any Linux computer that you would like to access the NAS from.

```
sudo echo "192.168.1.100 nas.local" >> /etc/hosts
```

The NAS can be accessed via SSH with the following terminal command:

```
ssh user@nas.local
```

## Install MDADM RAID Controller (Optional if using RAID)

```
sudo apt-get --no-install-recommends install mdadm
```

## Install Webmin

<http://www.webmin.com/download.html>

```
sudo apt-get install libapt-pkg-perl libnet-ssleay-perl
```

```
libauthen-pam-perl libio-pty-perl apt-show-versions
```

```
cd ~
```

```
wget http://prdownloads.sourceforge.net/webadmin/webmin_1.580_all.deb
```

```
sudo dpkg -i webmin*.deb
```

```
sudo reboot
```

## Setup RAID

Open browser and navigate to: <https://nas.local:10000>

Create and mount RAID via Webmin RAID setup utility.

Setup a mount point for the RAID in `/media/raidmount`

(It took approximately 36 hours to create a 9TB RAID5 array on my system, but the system was fully usable for this time)

## Alternatively Mount a Single Drive instead of RAID

```
sudo mkdir /media/disk1
```

```
sudo blkid
```

Take a note of the ID number for the drive you would like to mount

```
sudo nano /etc/fstab
```

Add new line:

```
UUID="ID From blkid" /media/disk1 ext4 rw,user,auto 0 0
```

*Note: for the remainder of the document I refer to the mounted drive as "raidmount", but this doesn't mean it has to be a RAID*

## Create Shared Folders

```
sudo su
cd /media/raidmount
mkdir Folder1
mkdir Folder2
```

Depending how you would like Samba / SMB shares to be accessed you have two choices:

1. Read/Write Access for one user, and Read Only Access for everyone else
2. Read/Write Access for everyone

Read/Write Access for one user:

```
chown user:user Folder1/
chown user:user Folder2/
```

Read/Write Access for everyone:

```
chmod a=rw -R /media/raidmount
chown nobody:nogroup Folder1/
chown nobody:nogroup Folder2/
```

## Setup Samba Shares

Backup old Config File:

```
sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.old
```

Example smb.conf file can be found at [Annex A](#) (at the end of this document)

Use Webmin > Servers > Samba Windows File Sharing > Edit Config File, to copy and paste example Config file in.

Click "Restart Samba Servers" button after editing Config file

## Usenet Components

### Install SABNZBD

```
sudo su
apt-get install python-software-properties add-apt-key
add-apt-repository ppa:jcfp/ppa
apt-get update
apt-get install javascript-common libjs-excanvas libjs-mochikit
libtidy-0.99-0 par2 python-chardet python-cheetah python-configobj
python-feedparser python-libxml2 python-openssl python-utidylib
python-yenc python-dbus zip unrar unzip python-markdown
python-pygments python-memcache
apt-get install sabnzbdplus sabnzbdplus-theme-plush
sabnzbdplus-theme-smpl
sed -i 's/USER=/USER=user/' /etc/default/sabnzbdplus
sed -i 's/HOST=/HOST=0.0.0.0/' /etc/default/sabnzbdplus
sed -i 's/PORT=/PORT=6000/' /etc/default/sabnzbdplus
sudo service sabnzbdplus restart
```

### Configure SABNZBD

Open web browser to <http://nas.local/6000>

Follow the setup wizard and fill in your Usenet Provider details.

Make the following changes in Config:

General:

API Key      \*Note this down for copying into SickBeard and CouchPotato

*The following changes will need to be made after you've installed SickBeard*

Folders:

Temporary Download Folder:/media/**raidmount**/Downloads

Completed Download Folder:/media/**raidmount**/New

Permissions for completed downloads: 755

Post-Processing Scripts Folder: /home/**user**/SickBeard/autoProcessTV

Categories:

movie

Script: Default

Folder: /media/**raidmount**/Downloads/Movies

tv

Script: sabToSickBeard.py

Folder: /media/**raidmount**/TV

## Install Sick Beard

Download from website: <http://sickbeard.com/downloads.html>

SSH Into the NAS via Nautilus

Unzip and copy into /home/**user**/SickBeard

```
cd ~/SickBeard/autoProcessTV
mv autoprocessTV.cfg.sample autoprocessTV.cfg
nano ~/SickBeard/autoProcessTV/autoprocessTV.cfg
```

Make the following Changes:

host=localhost

port=**6001**

## Create Sick Beard Init Script

Example Sickbeard Init Script can be found at [Annex B](#)

With Nautilus create new text file in your home folder. Copy in the example Init Script and make the necessary changes.

```
sudo cp ~/NewTextFile /etc/init.d/
cd /etc/init.d
sudo mv NewTextFile sickbeard
sudo chmod 755 sickbeard
sudo update-rc.d sickbeard defaults
sudo service sickbeard start
```

Open web browser to <http://nas.local:8081> and then make the following changes in SickBeard Config:

General:

Port: **6001**

Search Settings:

Download Propers: Yes

Search Frequency: 60

Usenet Retention: 1325

NZB Method: SABNZDB

SABNZDB URL: <http://localhost:6000>

API Key: Copy from SAB Config

Search Providers:

Newznab

Name: Your NZB Searcher, e.g. NZB.su  
Host: Their internet address, e.g. <https://nzb.su>  
API Key: Your RSS / API Key provided by your NZB Searcher

Post Processing:

TV Download Folder:	Blank
Keep Original Files:	No
Move Associated Files:	No
Rename Episodes:	Yes
Scan and Process:	No
Metadata Type:	XBMC
Show Name:	Yes
Episode Name:	Yes
Use Periods:	No
Quality:	No
Air-By-Date Format:	No
Separator Style:	Dash
Number Style:	S01E02

## Install CouchPotato

Download from website: <http://couchpota.to/>

SSH Into the NAS via Nautilus

Unzip and copy into /home/**user**/CouchPotato

## Create CouchPotato Init Script

Example CouchPotato Init Script can be found at [Annex C](#)

With Nautilus create new text file in your home folder. Copy in the example Init Script and make the necessary changes.

```
sudo cp ~/NewTextFile /etc/init.d/  
cd /etc/init.d  
sudo mv NewTextFile couchpotato  
sudo chmod 755 couchpotato  
sudo update-rc.d couchpotato defaults  
sudo service couchpotato start
```

Open web browser to <http://nas.local:5000> and then make the following changes in CouchPotato Config:

### General:

Host:	0.0.0.0
Port:	<b>6002</b>
Ignore Words:	HDTV
Preferred Words:	BluRay, DTS

### Downloaders:

Enable:	Yes
Retention:	1325 Days
Host:	localhost: <b>6000</b>
API Key:	Copy from SAB Config
Category:	movies
Torrents:	No

### Providers:

Newznab	
Name:	Your NZB Searcher, e.g. NZB.su
Host:	Their internet address, e.g. <a href="https://nzb.su">https://nzb.su</a>
API Key:	Your RSS / API Key provided by your NZB Searcher



## **Annex A: Example smb.conf file**

```
#===== Global Settings =====

[global]
    log file = /var/log/samba/log.%m
    load printers = no
    passwd chat = *Enter\snew\s*\spassword:* %n\n
*Retype\snew\s*\spassword:* %n\n *password\supdated\ssuccessfully* .
    obey pam restrictions = yes
    null passwords = yes
    map to guest = Bad User
    encrypt passwords = yes
    passdb backend = tdbsam
    passwd program = /usr/bin/passwd %u
    dns proxy = no
    server string = %h server (Samba, Ubuntu)
    unix password sync = yes
    workgroup = WORKGROUP
    debug level = 1
guest account = nobody
    os level = 20
    auto services = global
    syslog = 0
    usershare allow guests = yes
    panic action = /usr/share/samba/panic-action %d
    max log size = 1000
    pam password change = yes
    use sendfile = yes
    time server = no
    wins support = no

#===== Share Definitions =====

[homes]
    browseable = no
    comment = Home Directories
    available = no

[Folder1]
    read list = nobody, user
    write list = user, add nobody here for passwordless read/write access
    force directory mode = 0755
    store dos attributes = no
    create mask = 0755
    hide dot files = yes
```

```
user = nobody, user
public = yes
ea support = no
inherit acls = yes
browseable = yes
writeable = yes
inherit permissions = yes
printable = no
path = /media/raiddisk/Folder1
force create mode = 0755
comment = Folder 1
directory mask = 0755
valid users = nobody, user
```

## Annex B: Sickbeard Init Script

```
#!/bin/sh
# Edited by QuidsUp
# Based on script written by daemox http://www.ainer.org
# Modified to run on Debian and Ubuntu Server

### BEGIN INIT INFO
# Provides:          sickbeard
# Required-Start:    $local_fs $network $remote_fs
# Required-Stop:     $local_fs $network $remote_fs
# Should-Start:      $NetworkManager
# Should-Stop:       $NetworkManager
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: starts and stops sick beard
# Description:       Sick Beard is an Usenet PVR. For more
information see:
#                   http://www.sickbeard.com
### END INIT INFO

#Required -- Must Be Changed!
USER=user #Set Linux Mint, Ubuntu, or Debian user name here.

#Required -- Defaults Provided (only change if you know you need to).
HOST=127.0.0.1 #Set Sick Beard address here.
PORT=6001 #Set Sick Beard port here.

#Optional -- Unneeded unless you have added a user name and password
to Sick Beard.
SBUSR="" #Set Sick Beard user name (if you use one) here.
SBPWD="" #Set Sick Beard password (if you use one) here.

#Script -- No changes needed below.
case "$1" in
start)
#Start Sick Beard and send all messages to /dev/null.
#cd /home/$USER/.sickbeard
echo "Starting Sick Beard"
sudo -u user -EH nohup python /home/user/SickBeard/SickBeard.py -q >
/dev/null 2>&1 &
;;
```

```
stop)
#Shutdown Sick Beard and delete the index.html files that wget
generates.
echo "Stopping Sick Beard"
wget -q --user=$SBUSR --password=$SBPWD
"http://$HOST:$PORT/home/shutdown/" --delete-after
sleep 6s
;;
*)
echo "Usage: $0 {start|stop}"
exit 1
esac

exit 0
```

## Annex C: CouchPotato Init Script

```
#!/bin/sh
# Author: RuudBurger
# Edited by: QuidsUp

### BEGIN INIT INFO
# Provides:          CouchPotato application instance
# Required-Start:    $local_fs $network $remote_fs
# Required-Stop:     $local_fs $network $remote_fs
# Should-Start:      $NetworkManager
# Should-Stop:       $NetworkManager
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: starts instance of CouchPotato
# Description:       starts instance of CouchPotato using
start-stop-daemon
### END INIT INFO

#Required -- Must be changed!
USER=user #Set Ubuntu or Linux Mint username here.

#Required -- Defaults provided (only change if you know you need
to!).
DAEMON=/usr/bin/python
DAEMON_OPTS=" CouchPotato.py -q"
NAME=couchpotato
DESC=CouchPotato
PID_FILE=/home/user/.CouchPotoatoPID

#Script -- No changes should be needed below!
#test -x $DAEMON || exit 0

set -e

case "$1" in
  start)
    echo "Starting $DESC"
    /sbin/start-stop-daemon -d /home/user/CouchPotato -c $USER
--start --background --pidfile $PID_FILE --make-pidfile --exec
$DAEMON -- $DAEMON_OPTS
    ;;
```

```
stop)
    echo "Stopping $DESC"
    /sbin/start-stop-daemon --stop --pidfile $PID_FILE
    sleep 6s
    ;;

restart|force-reload)
    echo "Restarting $DESC"
    start-stop-daemon --stop --pidfile $PID_FILE
    sleep 15
    /sbin/start-stop-daemon -d /home/user/CouchPotato -c $USER
--start --background --pidfile $PID_FILE --make-pidfile --exec
$DAEMON -- $DAEMON_OPTS
    ;;

*)
    N=/etc/init.d/$NAME
    echo "Usage: $N {start|stop|restart|force-reload}" >&2
    exit 1
    ;;

esac

exit 0
```