Useful Commands

This page is to share commands / arguments that makes life easier.

Rsync

```
rsync -vaopHDS --stats --ignore-existing -P (Source) (Destination)
```

> -v, --verbose
> -a, --archive (It is a quick way of saying you want recursion and want to preserve almost everything.)
> -o, --owner
> -H, --hard-links
> -D, --devices (This option causes rsync to transfer character and block device information to the remote system to recreate these devices.)
> -S, --sparse (Try to handle sparse files efficiently so they take up less space on the destination.)
> -P (The -P option is equivalent to --partial --progress.)

Fixing perms for a website

```
find /home/USERNAME/public_html/ -type f -exec chmod 644 {} \; && find
/home/USERNAMER/public_html/ -type d -exec chmod 755 {} \;
```

DDrescue

```
ddrescue -f -n -r3 /dev/[bad/old_drive] /dev/[good/new_drive] /root/recovery.log
```

> -f Force ddrescue to run even if the destination file already exists (this is required when writing to a disk). It will overwrite.

> -n Short for'–no-scrape'. This option prevents ddrescue from running through the scraping phase, essentially preventing the utility from spending too much time attempting to recreate heavily damaged areas of a file.
>
> -r3 Tells ddrescue to keep retrying damaged areas until 3 passes have been completed. If you set 'r=-1', the utility will make infinite attempts. However, this can be destructive, and ddrescue will rarely restore anything new after three complete passes.

SSH tunneling

> -L = local, the 666 will be the port that will be opened on the localhost and the 8080 is the port listening on the remote host (192.168.1.100 example). -N = do nothing

```
ssh root@my-server.com -L 666:192.168.1.100:8080
```

AutoSSH

Autossh is a tool that sets up a tunnel and then checks on it every 10 seconds. If the tunnel stopped working autossh will simply restart it again. So instead of running the command above you could run

```
autossh -NL 8080:127.0.0.1:80 root@192.168.1.100
```

sshutle

```
sudo sshuttle -r root@sshserver.com:2222 0/0
sudo sshuttle --dns -r root@sshserver.com 0/0
```

Force reinstall all arch packages

```
pacman -Qqen > pkglist.txt
pacman --force -S $(< pkglist.txt)
```

Check Mobo info

```
dmidecode --string baseboard-product-name
```

More Details:

```
dmidecode | grep -A4 'Base Board'
```

Check BIOS version

```
dmidecode | grep Version | head -n1
```

Temp Python FTP WebServer

```
python -m SimpleHTTPServer 8000
```

Find what is taking all the space

List of the biggest directory's

```
du -Sh / | sort -rh | head -5
```

List of the biggest files

```
find /* -type f -exec du -Sh {} + | sort -rh | head -n 5
```

Put a +2TB drive in GPT

Start parted on the drive you want in gpt

```
parted /dev/sdd
mklabel gpt
unit TB
mkpart primary 0.00TB 16.00TB
print
```

Unable to mount Windows (NTFS) filesystem due to hibernation

Fix ntfs

```
ntfsfix /dev/sdXY
```

Mount read-only

```
mount -t ntfs-3g -o ro /dev/sdXY /mnt/windows
```

Repair rpm DB

```
rm -f /var/lib/rpm/__db*
db_verify /var/lib/rpm/Packages
rpm --rebuilddb
yum clean all
```

Stresstestapp

Install the app from source:

```
git clone https://github.com/stressapptest/stressapptest.git
cd stressapptest
./configure
make
sudo make install
```

```
stressapptest -s 10800 -W -v 9 --cc_test --random-threads --local_numa --remote_numa --
stop_on_errors >> /root/stresstest-test-01.txt
```

(10800 = 3 hours)

Create a ISO from a folder

```
mkisofs -o XYZ.iso XYZ/
```

Revision #26
Created 2017-07-18 22:27:28 UTC by Dave
Updated 2019-07-17 04:08:44 UTC by Dave