

## Guide to setting up a probbox RAID-Z (RAID 5) system in Ubuntu 14.04

Luke Sjulson, June 2015

### 1) install zfs on your machine (<http://serverascode.com/2014/07/01/zfs-ubuntu-trusty.html>)

install zfs and run the module:

```
sudo apt-get install software-properties-common
sudo apt-add-repository ppa:zfs-native/stable
sudo apt-get update
sudo apt-get install -y ubuntu-zfs
sudo modprobe zfs
```

tell zfs not to use more than 1 GB of RAM for cache, and also to automatically mount your drives upon boot:

```
sudo bash
echo "options zfs zfs_arc_max=1073741824" >> /etc/modprobe.d/zfs.conf
echo "zfs mount -a" >> /etc/rc.local
exit
```

### 2) assemble the probbox

Put drives in Probox, use gparted to make partitions. Use "gpt" as the type of partition table. Make the primary partition the full size of the drive minus 1% of the total capacity (e.g. 30 GB for a 3 TB drive). Make only one partition per drive, and leave it unformatted. By discarding a small percentage of the drive's capacity, you can replace a broken 2.999 TB HD with a new 2.998 TB HD (e.g. two drives from different manufacturers with nominal 3 TB capacity). Before closing gparted, write down which partitions you are going to use (e.g. /dev/hdb1, /dev/hdc1, etc.)

Find your partitions by UUID, the "universal unique identifier" of each partition:

```
ls -l /dev/disk/by-uuid
```

then create the RAID array (a "zpool") by entering this very long command, all in one line:

```
sudo zpool create -o ashift=12 probbox_zpool raidz1
/dev/disk/by-uuid/117097d4-96f8-4600-9b20-ef0d8952817e
/dev/disk/by-uuid/3f980c2c-d5be-4797-a38a-a1f50b60bb4f
```

etc. where you're cutting and pasting the UUID of each partition you want to be part of your RAID array. It's a good idea to enter this into a text editor first, making sure this is all one long line, then cutting and pasting into the terminal.

This creates a zpool called `probox_zpool`, which is mounted in your root directory as `/probox_zpool`. You will want to make it user writable

```
sudo chmod a+rwX /probox_zpool
```

so you can put your data there. However, before you copy data onto the RAID array you should:

```
sudo zfs set compression=gzip-1 probox_zpool
```

this enables on-the-fly gzip1 compression, which is the lowest level of gzip. I tested different compression settings and found that this one provides decent compression of .dat files with minimal overhead. Compression supposedly makes reading the data slightly faster but at the cost of increased CPU usage, which is usually a good tradeoff. The data is compressed when you copy it onto the zpool - if the data is already there and you enable compression, it stays uncompressed.

### 3) install maintenance scripts

make a directory to hold the scripts

```
mkdir ~/scripts
```

then copy the scripts **run\_once** and **zfscheck** to the scripts folder and make them executable

```
chmod u+rwX zfscheck run_once
```

edit the top two lines of **zfscheck** to insert your username and home directory.

Then make sure **run\_once** is run every time a shell is opened. This just saves your screen's identifier to disk so that the script can make on-screen notifications if it detects errors

```
echo ~/scripts/run_once >> ~/.bashrc
```

then add some lines to root's crontab so the disks get checked regularly. start with

```
sudo crontab -e
```

then paste these lines into the crontab:

```
SHELL=/bin/bash
0 3 * * * /home/luke/scripts/zfscheck # check for errors at 3AM daily
* 0 * * 6 /sbin/zpool scrub probox_zpool # do full drive scans weekly
```

### 4) enabling snapshots

This is more advanced and is a work in progress...

```
sudo apt-get install zfs-auto-snapshot
```

making datasets

```
sudo zfs create probbox_zpool/preprocessing
```

```
sudo zfs create probbox_zpool/processed
```

```
sudo zfs create probbox_zpool/rawdata
```

```
zfs set com.sun:auto-snapshot=false probbox_zpool
```

```
zfs set com.sun:auto-snapshot=true probbox_zpool/preprocessing
```

```
zfs set com.sun:auto-snapshot=true probbox_zpool/processed
```

```
zfs set com.sun:auto-snapshot=true probbox_zpool/rawdata
```

to list all existing snapshots:

```
zfs list -t snapshot
```