

# High-Rely BARE METAL CHRONICLES

ADVENTURES IN THE CHALLENGES OF OBTAINING PEACE OF MIND WHILE COMPUTING

DATE: August 16<sup>th</sup>, 2007

ISSUE: V

## Bare Metal Restore using High-Rely SlimLine and Ubuntu Linux Edgy server version 6.10.

By Thomas Hoops

*Summary: One way to perform a bare-metal restore from a High-Rely SlimLine backup drive to an Ubuntu Linux version 6.10 server.*

While we don't offer support at the present time for LINUX on our products, I personally have had good luck with our products working on it thus far – at least Ubuntu 6.10. First let me state that I am not a LINUX GURU. I like LINUX a lot and have been a strong promoter and cheerleader for it for the right purposes. But in the past, I've let employees more savvy than I tackle the details. So, please forgive any mistakes you may encounter in my presentation here. Also, under the circumstances of this event, no screen shots were available. This document assumes you have a fair understanding of LINUX.

We have a rather important machine which I recently converted over to Ubuntu Linux Server 6.10 (Edgy-EFT). Once operating, the question arose about its disaster recovery plan. Unlike other OS's, one of LINUX's shining points is that at a certain scale, no expensive software – is necessary to perform a bare-metal restore. As a simple webserver, that happened to be the scale this machine was running at.

### BACKUP Hardware:

An older Intel Pentium D 2.8GHz with 1GB of RAM running Ubuntu Edgy-EFT server LAMP version 6.10 constitutes the server. Two High-Rely SlimLine drives sit outside the server box. Each drive is connected directly to the SATA ports on the motherboard via an iSATA to eSATA connector on

a slot cover. One is the main drive, the other is backup drive. Thus, both drives are easily removed. The lever for the system drive has a label warning of dire consequences for removing the drive while the system is running.

### BACKUP Software:

Because this machine deals with a rather small volume of data (about 20 GB), we decided that each night, it would do a backup simply using TAR. Because it runs a MYSQL database, the mysql database is dumped before the tar occurs and thus is part of the tar file. Below is an excerpt from the backup script run each night by a CRON task.

```
echo "Last date run:" >> backup.log
date >> backup.log
umount /media/PUE >> backup.log
mount /dev/sda1 /media/PUE >> backup.log
echo "Mounted PUE" >> backup.log
mysqldump -ubcn -pmentos COATData > mysql.bak
echo "Mysqldump completed." >> backup.log
tar cvpzf backup.tgz --exclude=/proc --
exclude=/lost+found --exclude=/var/www/b$
echo "TAR completed." >> backup.log
cd /media/PUE
rm backup7.tgz
mv backup6.tgz backup7.tgz
mv backup5.tgz backup6.tgz
mv backup4.tgz backup5.tgz
mv backup3.tgz backup4.tgz
mv backup2.tgz backup3.tgz
mv backup1.tgz backup2.tgz
mv backup.tgz backup1.tgz
echo "Sliding of backup copies complete." >>
/var/nolt/backup.log
mv /var/nolt/backup.tgz /media/PUE/backup.tgz
echo "Completed copy of backup.tgz to PUE" >>
/var/nolt/backup.log
cd /var/nolt
umount /media/PUE
echo "Backup Complete" >> backup.log
exit 0;
```

Hence, the backup drive has 7 tar files rotated through it. Once a week, the drive is swapped with another and taken offsite.

**Preparation:**

Because restoration involves the LINUX OS, it is best that the target machine has hardware capable of accepting at least one High-Rely SlimeLine for pulling the backup data off of. Alternatively, the High-Rely media can also be opened and the raw drive removed and installed as a secondary on the target machine.

**The Restoration:**

Starting with a functioning machine, use your original Ubuntu Server 6.10 disk, and perform an install on the target machine. The only important step here is that when the disk partitioner comes up, be sure to select the same partition type (LVM or non LVM) as the type your server was built with or GRUB will not be able to boot your new machine after you restore your backup. This is not to say that you can not do this but the details of how to perform a restore that way are not covered in this document.

Next, mount the backup drive so that your backup files can be accessed. Be sure you are logged in as root.

Next untar the particular backup you wish to restore – for example type:

```
tar xvpfz /media/PUE/backup3.tgz -C /
```

There maybe an error message at the very end after the last file was restored. This message can be ignored. Now, reboot your machine. The machine should now be a clone of your former machine. If you're using a different vendor of network card or other devices from your original server, those drivers may require adjusting.

Additionally, you may wish to delete your MYSQL database and reload it from mysql.bak. However, my experience was that this was not necessary.

That's it. No \$2,000 software, no registrations, no endless configurations or disk building, it's just done - fast and simple.

Overall, this might be considered more of a case study because it is rather specific in its purpose, components and methods. But, I hope the information here will help guide someone to a workable disaster recovery plan of their own.