Testing Tape

The problem

☑ You've got your production NAS running a full backup to tape every day/week/month as per your RPO requirements for archival, so your backup is covered.
☑ You've got a system to test the data integrity of the tape, so your data is covered.

Now, how do test your recovery?

What will be the effect of a new backup client version? …an OS patch? …and how long is it going to take to figure that out? …and do you have the space to restore?

A Strategy

- Set up two backup locations:
  - a fast one for OS recoverability testing
  - a slow one for data backup and integrity
- Backup just the OS to the fast store
- Test the recovery of the OS frequently
- Test the integrity of the data less frequently
- Test the recovery of the data less frequently still

…but how fast is fast, and how similar is similar?

We have a trade-off with this strategy. We want our fast backup/restore test to accurately mirror our slow backup/restore but if we use actual tape we're limited by tape speed and, arguably worse, by tape lifespan – not to mention drive/library/robotics logistics, expenses and additional backup system licences.

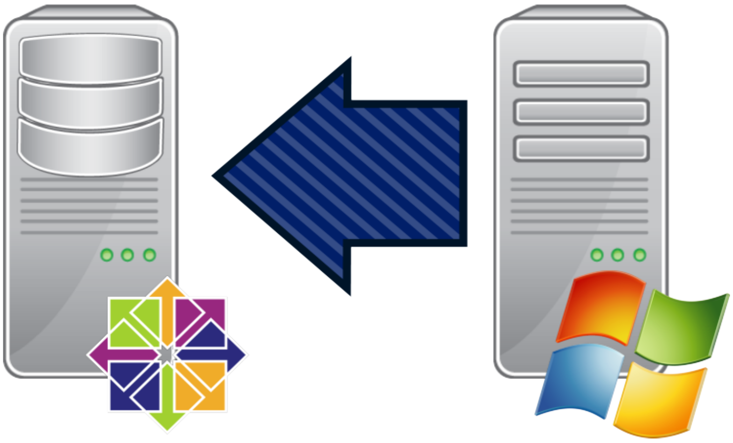
So we can backup to disk instead right? Well, yes, but writing to file(s) on disk and writing to tape are two very different kettles of fish. Tape support in various backup products has been broken by kernel updates, scsi module updates, software updates, you name it. In theory all tapes support a sensible generic subset of SCSI commands… but they all have their quirks and it would be really nice to test them.

Enter Virtual Tape

mhvtl (https://sites.google.com/site/linuxvtl2/) is a piece of GPLv2 software by Mark Harvey to create tape drives and libraries that look and act like the real thing. However, as they are disk based they are significantly faster.

Set up

Create a separate machine that exports the tape drive as iSCSI such that it can be imported to the backup machine:



Creating the virtual tape library

ElRepo used to produce a pre-packaged version of mhvtl, but (at time of writing) this no longer seems to be available. Cristie have made a packaged version using DKMS which has been tested on CentOS 7 and is available here:

https://github.com/JamesReynolds/mhvtl-rpm/releases

You can install this on a minimal CentOS 7 machine using:

```
$ sudo yum install -y epel-release
$ sudo yum install -y mhvtl-1.5-3.x86_64.rpm \
                      mhvtl-dkms-1.5-3.noarch.rpm
```

By default this will create a sample configuration file with a large number of devices, as running lsscsi shows:

```
$ lsscsi
[2:0:0:0]    mediumx STK      L700             0105   /dev/sch0
[2:0:1:0]    tape    IBM      ULT3580-TD5      0105   /dev/st0
[2:0:2:0]    tape    IBM      ULT3580-TD5      0105   /dev/st1
[2:0:3:0]    tape    IBM      ULT3580-TD4      0105   /dev/st2
[2:0:4:0]    tape    IBM      ULT3580-TD4      0105   /dev/st3
[2:0:8:0]    mediumx STK      L80              0105   /dev/sch1
[2:0:9:0]    tape    STK      T10000B          0105   /dev/st4
[2:0:10:0]   tape    STK      T10000B          0105   /dev/st5
[2:0:11:0]   tape    STK      T10000B          0105   /dev/st6
[2:0:12:0]   tape    STK      T10000B          0105   /dev/st7
```

By default all of the tape devices are empty but can be loaded as follows:

```
$ vtlcmd 11 load E01001L4
```

The bar-codes for the tapes are defined in /etc/mhvtl/library_contents.N and have a special form that is detailed in that file. By default the tapes are created as 500MB but they can easily be increased using edit_tape:

```
$ dump_tape -l 11 -f E01002L4
Looking for PCL: E01002L4 in library 11
Media density code: 0x46
Media type code    : 0x08
Media description : Ultrium 4/16T
Tape Capacity      : 524288000 (500 MBytes)
Media type         : Normal data
Media              : read-write
Remaining Tape Capacity : 524288000 (500 MBytes)
Total num of filemarks: 0
Hdr:      End of Data(05), sz            0, Blk No.: 0, data 0

$ edit_tape -l 11 -m E01002L4 -s 50000 -t data -d LTO4 -w off
Looking for PCL: E01002L4 in library 11
New capacity for E01002L4: 50000MB
Setting density to LTO4
Turning off write-protect for E01002L4
```

All of these operations can also be done via SCSI control of the tape library itself – which can be viewed via your backup software.

Exporting via iSCSI

Exporting devices via iSCSI is covered in a lot of places, so we'll leave detail out here, but the following script will export all of the tape and library devices over iSCSI using the initiator defined in `/etc/scsi`:

```
$ yum install -y scsi-target-utils iscsi-initiator-utils
$ systemctl start tgtd

$ . /etc/iscsi/initiatorname.iscsi

$ tgtadm --lld iscsi --op new --mode target --tid 1 -T ${InitiatorName}

$ LUN=0
$ for device in `lsscsi -g | grep -E 'tape|mediumx' | awk '{print $NF}'`;
  do
    LUN=$(( LUN + 1 ))
    tgtadm --lld iscsi --mode logicalunit \
           --op new -Y pt -E sg --tid 1 --lun ${LUN} -b ${device}
  done

$ firewall-cmd --permanent --add-port=3260/tcp
$ firewall-cmd --reload
$ tgtadm --lld iscsi --op bind --mode target --tid 1 -I ALL
```

Testing a recovery

We can now test the operating-system level backup and recovery of a system by:

1. Add the iSCSI initiator
2. Configure the backup software to use this device
3. Backup the OS, excluding the data drives
4. Configure the recovery software to use the iSCSI initiator
5. Restore the OS
6. Test

Cristie CBMR Linux

As a full example, we can use Cristie CBMR to perform the backup and recovery of the system.

1. Add the iSCSI initiator

```
$ iscsiadm --mode discovery --type sendtargets \
           --portal <IP of tape machine>
$ iscsiadm --mode node -p <IP of tape machine> --login
```

2. Import the devices into CBMR

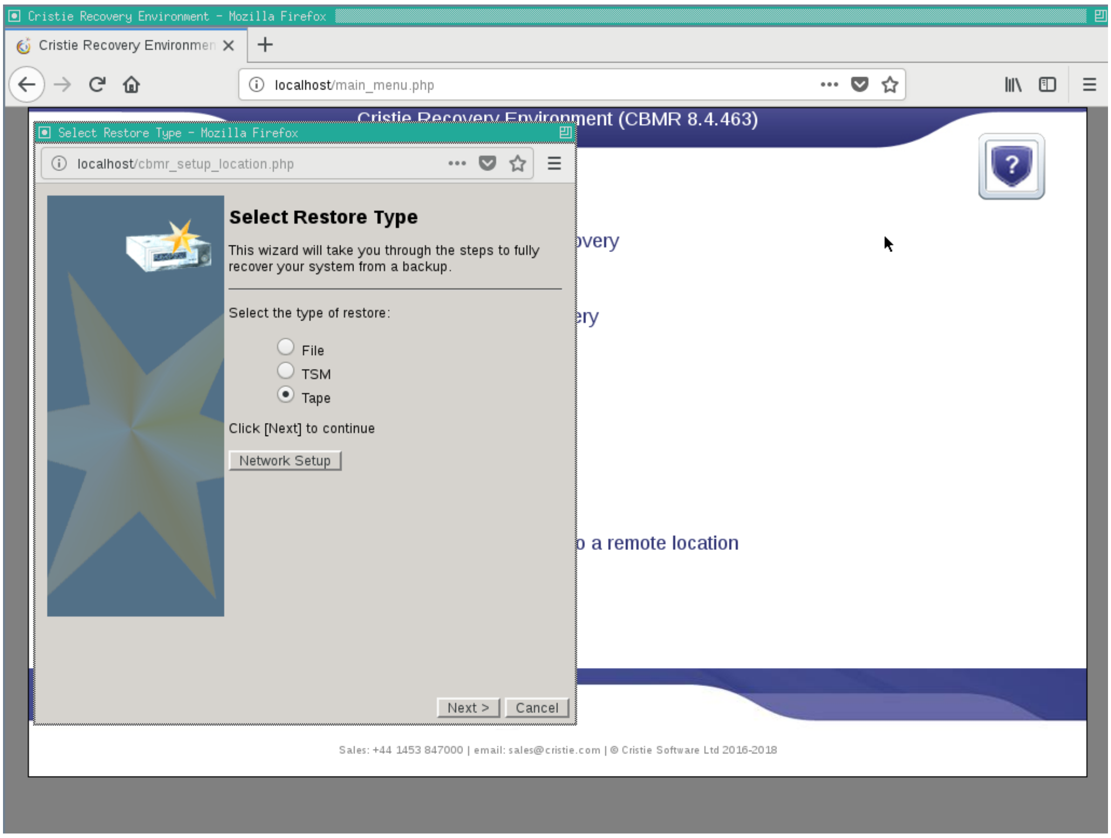
```
$ ubax –scan_devices
```

3. Run the backup

```
$ ubax -d Tape0 -b /etc/cbmr/scripts/cbmr.scp
```

4. Boot the recovery ISO then connect the iSCSI tools in a shell as step 1

5. Add the tape device to restore


Cristie Recovery Environment - Mozilla Firefox
Cristie Recovery Environmen
localhost/main_menu.php
Cristie Recovery Environment (CBMR 8.4.463)
Select Restore Type - Mozilla Firefox
localhost/cbmr_setup_location.php
?
overy
ery
o a remote location
Network Setup
Next >
Cancel

Select Restore Type

This wizard will take you through the steps to fully recover your system from a backup.

Select the type of restore:

○ File
○ TSM
● Tape

Click [Next] to continue

Sales: +44 1453 847000 | email: sales@cristie.com | © Cristie Software Ltd 2016-2018

6. Run the recovery

Summary

Depending on the complexity of the operating system and the use of excluding directories it should be possible to test a backup / restore to tape in under 15 minutes. At Cristie we run the creation of the tape machine via Vagrant and automated the recovery process using the Cristie VA so we can run a backup/restore test of an operating system, start to finish, at the push of a button.

This gives us a round-trip of under 30 minutes to test new versions of Linux and their interoperability with different kinds of tape.

CBMR

CBMR was used for both the tape backup and restore of the operating systems. It has been designed to work with tape for Windows, Linux, Solaris and AIX.