

Bacula Bextract

Bextract

1. Get list of volumes from TechDB

To get an idea of what files you'll need, you need to examine the TechDB Backups tab. For each backup, a line will be printed which starts "Volume name(s):" - this should be in the following format:

```
srv-7813711322-3889
srv-7813711322-3882
srv-7813711322-3875
srv-7813711322-3868
srv-7813711322-3861
srv-7813711322-3854
srv-7813711322-3847
srv-7813711322-3891
```

2. Create bootstrap file

This is a text file containing a list of volumes for Bacula to read from, in the correct order, it's format should be as follows (oldest to newest):

```
Volume=srv-78_137_113_22-3847
Volume=srv-78_137_113_22-3854
Volume=srv-78_137_113_22-3861
Volume=srv-78_137_113_22-3868
Volume=srv-78_137_113_22-3875
Volume=srv-78_137_113_22-3882
Volume=srv-78_137_113_22-3889
Volume=srv-78_137_113_22-3891
```

There's a little cheat to make this step easier which is the below command that should be run from within the backups directory on backups server (typically /home/bacula/clientIP_IP_IP_IP):

```
ls -latr srv* | awk '{print $9}' | sed 's/\(.*\) /Volume=\1/' > bootstrap.bsr
```

3. Create includes file

The next thing to create is a file that describes which files you want to extract from the backups, this is just another text file, containing the paths to the files you want to restore, separated by linebreaks, e.g:

```
/path/to/required/files  
/path2/to/required/files
```

Note: This file is not needed if you need to restore everything within a backup file.

4. Create the output directory (on backup server)

Create a directory for the extract command to output the backup contents to. This needs to be on the backup server as we'll look to transfer the files to the client later on.

5. Running the command

There are 2 sets of commands here, differing for PyBaculaV2 and none-pybac backup server. (Most we deal with now have been upgraded to PyBacV2).

The only real difference between the 2 options is the storage type that's specified. For PyBacV2, you'll need to specify the device prefix, whereas for none-pybac, you'll need to specify the storage prefix.

PyBaculaV2:

View files included in backup (with includes.txt taken into account):

```
bls -b bootstrap.bsr -i includes.txt -pv device-IP_IP_IP_IP
```

Initiate bextract:

```
bextract -b bootstrap.bsr -i includes.txt -pv storage-IP_IP_IP_IP /path/to/restore/to
```

None-PyBac

View files included in backup (with includes.txt taken into account):

```
bls -b bootstrap.bsr -i includes.txt -pv storage-IP_IP_IP_IP
```

Initiate bextract:

```
bextract -b bootstrap.bsr -i includes.txt -pv storage-IP_IP_IP_IP /path/to/output/
```

6. Transfer files to client-server

Feel free to use your preferred file transfer method for this step, for the below I've detailed how to achieve this using rsync.

A. Compress the file before transfer

[tar guide](#)

```
tar -czf zipfilename.tar.gz filetoarchive
```

B. Transfer the files

```
rsync -r -e "ssh -p2020" /path/to/files/on/backup/server  
'root@[client:server:BAC:address]':/client/server/restore/path
```

C. Uncompress files on client-server

(extracts into current working directory)

```
tar -xzf archivename.tar.gz
```

Revision #5

Created 2024-06-04 14:48:06 UTC by Daniel

Updated 2024-06-05 01:10:13 UTC by Daniel