
Bacula in the ISDS

It comes by night and sucks the vital essence from your computers.

Lance Brown

ISDS

December 2, 2005



ISDS Environment

- 6 Servers
- 1 Beowulf Cluster
- 53 Linux Desktops
 - NFS mount home and data directories from servers.
- 10 Windows Desktops
 - Roaming Profiles on Samba Server
- 1 SCSI-attached 11-slot LTO-1 Tape Library
- **Only** servers and beowulf head node house data that must be backed up. No data on desktop systems is backed up.



ISDS Backup Requirements

- Selected Filesystems and directory trees are backed up. Complete server backups are not done.
- Monthly full backups with nightly incremental backups. Monthly backups are kept for 3 months.
- Archive backups are taken quarterly and kept indefinitely.
- ISDS has backup tapes of various types going back to the mid-1990s.



ISDS Backup History

- Custom scripts in place when I started in June, 2005
- Switched to AMANDA after discovering scripts were broken due to growth of data in August, 2005.
 - 9 years previous experience with AMANDA in a very different environment.
- Switched to Bacula after AMANDA proved to be a poor fit to ISDS needs.



What is Bacula?

- Backup, restore, and verification program
- Client-Server based
- Catalogs stored in database
- Licensed under modified version of GPLv2
- Multi-Platform

Linux (most flavors), Solaris, FreeBSD, MacOS X

File Client Only: MS Windows, OpenBSD, Irix



What is Bacula not?

- Not a complete disaster recovery system in itself, but can be an integral part of one.
- Not a “Fire and Forget” backup system, but what is?



Bacula Pros and Cons

Pros

GPL Open Source

Tape Spanning

Encrypted network channels

Flexible scripting capability

Active development community

Excellent documentation

CD and DVD writer support

Backup-to-Disk support

Professional support available

Cons

Multiple library support not fully mature

No encryption of backups

Configuration data in many files

Baroque interaction between configuration directives



Why Bacula for ISDS?

- **Tape Spanning!!!**
 - Only a few servers
 - Filesystems and directory trees much larger than tape size (200GB compressed)
- GPL Open Source – The price was right
- Relatively easy to configure and get working quickly.



System Requirements

- Gnu C++ 2.95 or better
- SQLite, MySQL, or PostgreSQL
- mtx
- A “good” pthreads implementation, some older versions of FreeBSD has problems with their pthreads lib, for example.
- Ncurses or termcap for command line console program



Installation

- CentOS 3 servers
 - Installed RPMs for RHEL3 from Bacula site on SourceForge
- FreeBSD servers
 - Installed bacula 1.36.1 from ports collection



Bacula Components

- Director
- File Daemons
- Storage Daemons
- Database
- Consoles
- Monitors



Configuring Bacula

- You are in a twisty maze of text files, all alike!
- Each daemon on each machine has its own config file.
- The bconsole program has its own config file.
- Example config files available at:
<http://www.isds.duke.edu/~brown/bacula-talk>



ISDS Example Configurations

- Job definitions
- Jobs
- FileSets
- Pools
- Scheduling
- Reporting



Job Definitions

Job definitions let you set up common configuration data for many jobs at once.

```
JobDefs {  
  Name = "DefaultJob"  
  Type = Backup  
  Level = Incremental  
  Schedule = "MonthlyCycle"  
  Storage = LTO-1  
  Messages = Standard  
  Pool = Monthly  
  Priority = 20  
}
```



Jobs

Define the specific backups you want to make. Each job is client-specific.

```
Job {  
    Name = "What"  
    JobDefs = "DefaultJob"  
    Client = what-fd  
    FileSet = WhatFileSet  
    Write Bootstrap = "/var/bacula/what.bsr"  
}
```



FileSets

Defines a set of directories and files to back up.

```
FileSet {  
  Name = "WhatFileSet"  
  Include {  
    Options {  
      signature = MD5  
    }  
    File = /etc  
    File = /var/bacula  
  }  
}
```



Pools

Defines a set of backup volumes.

```
Pool {  
    Name = Monthly  
    Pool Type = Backup  
    Recycle = yes # Bacula can automatically recycle Volumes  
        Recycle Oldest  
    Volume = yes # Recycle oldest Volumes first  
    AutoPrune = yes # Prune expired volumes  
    Volume Retention = 128 days # Approx 4 months  
    Accept Any Volume = yes # write on any volume in the pool  
    Volume Use Duration = 30 days # Allow writing only for 30 days  
    Label Format = Monthly-  
}
```



Schedules

Defines when a job runs

```
Schedule {  
  Name = "MonthlyCycle"  
  # Run Full backup on first day of each month  
  Run = Level=Full on 1 at 1:05  
  # Run incremental on other days  
  Run = Level=Incremental on 2-31 at 1:05  
  # Run Archive full backup on 1st day of each quarter.  
  Run = Level=Full Pool=Archive on jan 1 at 1:00  
  Run = Level=Full Pool=Archive on apr 1 at 1:00  
  Run = Level=Full Pool=Archive on jul 1 at 1:00  
  Run = Level=Full Pool=Archive on oct 1 at 1:00  
}
```



Reporting

Bacula reporting controlled by Messages stanzas in config files.

From bacula-dir.conf:

```
Messages {
  Name = Standard

  mailcommand = "/usr/sbin/bsmtp -h localhost -f \"\\(Bacula\\) root@stat.duke.edu\" -s
    \"Bacula: %t %e of %c %l\" %r"

  operatorcommand = "/usr/sbin/bsmtp -h localhost -f \"\\(Bacula\\)
    root@stat.duke.edu\" -s \"Bacula: Intervention needed for %j\" %r"

  mail = brown@stat.duke.edu = all, !skipped
  operator = brown@stat.duke.edu = mount
  console = all, !skipped, !saved
  append = "/var/bacula/log" = all, !skipped
}
```



Reporting 2

I have all other components report their messages back to the
Director:

```
# Send all messages except skipped files back to Director
Messages {
  Name = Standard
  director = what-dir = all, !skipped
}
```



Handling Archives

- Schedule automatically generates extra set of full backups
- Create CD ROM archive of bacula source code, RPMs, config files, and catalog databases to store with tapes.



Gotchas

- Passwords
- Flexibility of scheduling jobs
- Director will initiate pending jobs if something changes to enable them.
- Restarting the Director loses current pending job state.



Passwords

In **when**'s bacula-fd.conf this Password field:

```
Director {  
    Name = what-dir  
    Password = "XXXYYYYZZZZ"  
}
```

must match the corresponding Client definition in bacula-dir.conf
on the director, **what**:

```
Client {  
    Name = when-fd  
    # password for FileDaemon  
    Password = "XXXYYYYZZZ"  
    ...  
}
```



Flexibility in Scheduling Jobs

Two primary ways to sequence jobs, start time and priority.

Start times overrule priority for queued jobs.



Pending Jobs

Director will immediately start a new job if you correct the situation that blocked the job from running.

Example: I forgot to put new tapes in the changer to handle the Full backups at the beginning of the month. After loading the tapes and labelling the first one, the pending backups started, preventing me from labelling the other tapes until the new tape filled.



Restarting the Director

Stopping and restarting the director process
looses the state of any pending jobs.

This is **bad** when you drop the pending jobs on
the 1st day of the month when you forgot to
put new tapes in the changer the night
before.



Things I've not covered

- Bare-metal recovery of systems
- Using bootstrap files to recover data
- Disk-to-Disk backups



What's next for Bacula?

- New version: 1.38.2
 - Python event scripting
 - Better DVD writer support
 - Better multi-library support
 - Better concurrent job support



What's next for Bacula 2

- Future Projects
 - Expand Python events framework
 - Merge multiple backups (Consolidation)
 - Copy Pools



Where to find more information

- Bacula website: <http://www.bacula.org/>
- Manual: <http://bacula.org/rel-bacula.pdf>
- Mailing lists
 - Bacula Users
<http://lists.sourceforge.net/lists/listinfo/bacula-users>
 - Bacula Devel
<http://lists.sourceforge.net/lists/listinfo/bacula-devel>

