Bacula

STLLUG
15 May 2014

Bacula is a networked client/server backup solution that creates cataloged backups of Unix, Linux, Windows, and MacOS systems on a wide variety of media.

About Me

I have used Bacula for six years on a server running Debian over two generations of system hardware and LTO tape drives.

I work as an independent consultant performing system and small network administration, and writing specialized technical documentation.

Acknowledgments

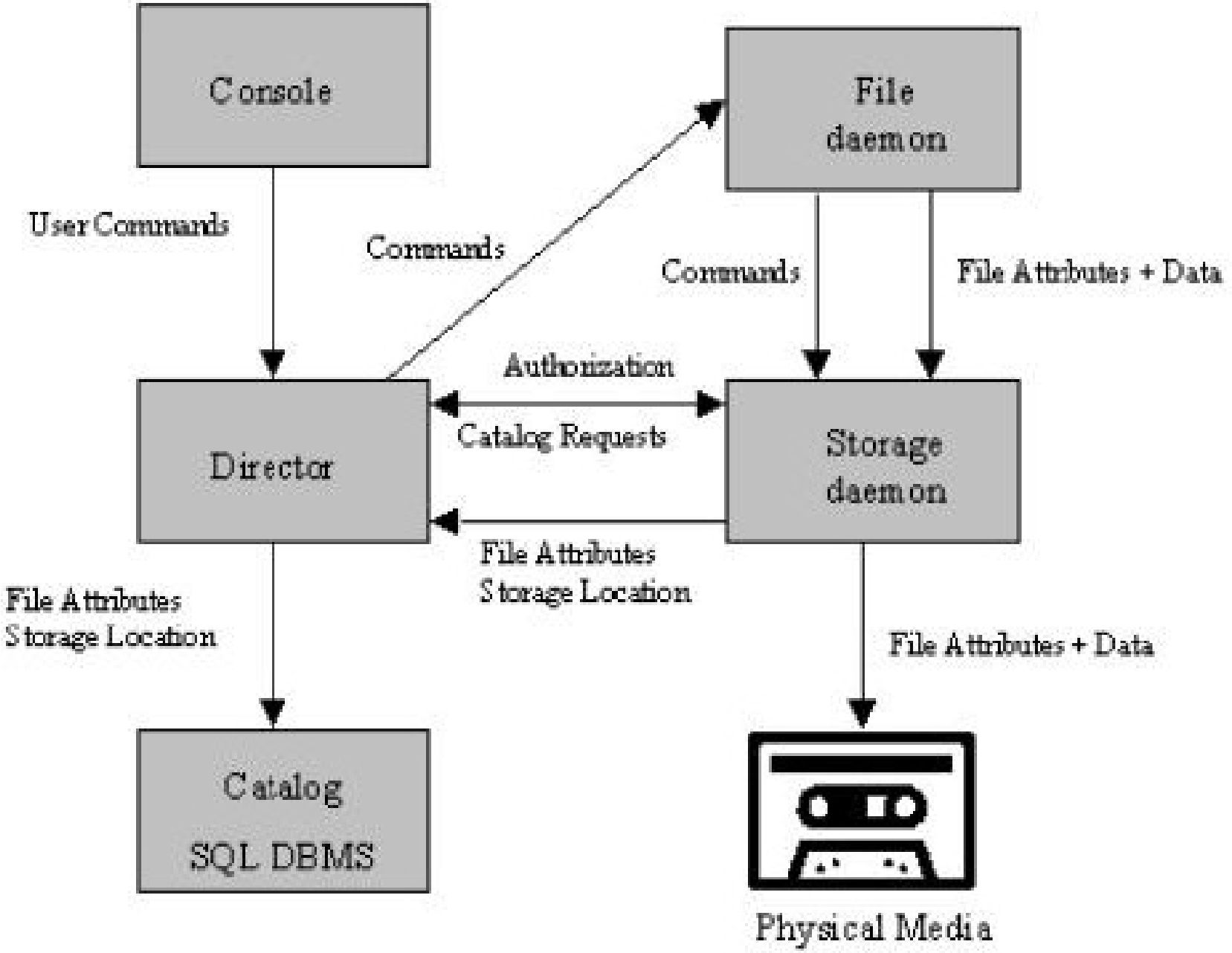
- 'Bacula' is a registered trademark of Kern Sibbald.
- Excerpts from the Bacula documentation used under the Creative Commons Attribution-ShareAlike 4.0 International License.
- Excerpts from the Bacula source code (including template configuration files) used under Affero GNU General Public License version 3.

Outline

- Bacula Basics
- How I use Bacula
- How you can use Bacula
- Where to go next

Bacula Components

- Catalog
- Client File Daemon
- Console
- Director Daemon
- Storage Daemon
- Configuration files


Console
File daemon
User Commands
Commands
Commands
File Attributes + Data
Authorization
Director
Catalog Requests
Storage daemon
File Attributes
Storage Location
File Attributes
Storage Location
File Attributes + Data
Catalog
SQL DBMS
Physical Media

Catalog

- Record of files and filesystems stored
- Record of media usage
- Relational Database – MySQL, PostgreSQL, or SQLite

Client File Daemon

- Runs on the system whose files you are backing up.
- Configured by bacula-fd.conf
- Needs to know: own name, keys, how to talk to the Director daemon, where to send messages.

Console

- The human interface
- Command-line console adequate
- Gui consoles available
- bconsole.conf
- Needs to know: own name, how to talk to the director

Director Daemon

- Runs the show
- Hardest to configure, because...
- Needs to know everything
- bacula-dir.conf

Storage Daemon

- Writes to storage media
- Needs to know: own name, how to talk to Director, what to do with messages, what/how for the storage devices or media
- Bacula-sd.conf

Configuring the Director

- Jobs
- Pools and Media
- Filesets
- Messages
- Schedules

How I use Bacula

- Why I chose Bacula
- Admin Jobs
- Backup Jobs
- Restore Job
- Hints and tricks

Why I chose Bacula

- A timeline –
- No backups
- Ken arrives
- CDs – 700 MB
- DVDs – 4.7 GB
- DL-DVDs – 8 GB
- Now what?

Why I chose Bacula

Off-Site and offline requirement.

Full backups requirement (on the restore end).

Bacula 2.x was well-developed, well documented, open-source backup tool that supported LTO (Ultrium) tape drives.

Why I chose Bacula

Bacula was more complex and more capable than any backup tool I had used previously.

Once Bacula was set up and running, it just worked.

2.x to 5.x per Debian; LTO-2 to LTO-4.

Jobs

- Run according to Schedules
- Have Run before, Run after scripts
- Holidays via run before script
- Jobs run at the same time run in priority order

Admin Jobs

- Execute a shell script, to...
- Mount/dismount devices or media
- Extract database contents
- Clean up after a backup

Backup Jobs

- Once for the data, once for the catalog
- 5 times a week (tape changes)
- Pools: 10 'daily', 12 Monthly, 12 Offsite, yearly
- Cached svn extracts

Backup Jobs – Main Job

```
Job {
  Name = "ServerBackup"
  FileSet = "ServerBackup"
  Priority = 12
  RunBeforeJob = "/etc/bacula/holiday_check.py"
  RunAfterJob  = "/etc/bacula/cleanup.pl"
  Type = Backup
  Schedule = "NightlySave"
  Client = linux2-fd
  Storage = QuantumUltriumLTO-4
}
```

Backup Jobs - Fileset

```
FileSet {
  Name = "ServerBackup"
  Include {...}
    File = /home
}
```

- Others: /root, /etc, /var/lib, /usr/local, /opt
  and database extracts. And Excludes, of course

Backup Jobs - Schedule

```
Schedule {
  Name = "NightlySave"
  Run = Pool=Yearly     mar     3rd Wed at 02:00
  Run = Pool=Monthly    monthly 2nd Wed at 02:00
  Run = Pool=OffSite    monthly 1st wed at 02:00
  Run = Pool=Daily            tue     at 02:00
 ...
```

Backup Jobs - Schedule

```
   ...
  Run = Pool=Daily          thu-sat at 02:00
  Run = Pool=Daily   monthly 4th Wed at 02:00
  Run = Pool=Daily   monthly 5th Wed at 02:00
  Run = Pool=Daily   jan-feb 3rd Wed at 02:00
  Run = Pool=Daily   apr-dec 3rd Wed at 02:00
}
```

Five-Nightly Jobs

- Mount Tape
- Extract special data
- Backup and cleanup
- Extract Catalog
- Backup and cleanup
- UnMount Tape

Restore Job

- One defined restore job, 'fill in the blanks'
- Select a job ID via menu of options
- Select files to be restored
- Five minutes typical to define the restore.

Restore file selection

- **cd** change current directory
- **done** leave file selection mode
- **find** find files, wildcards allowed
- **ls** list current directory
- **mark** mark dir/file to be restored
- **pwd** print current working directory
- **quit** quit and do not do restore

Hints and Tricks

- Be wary of reusing names in different categories
- Schedule Indefinite hold – February 31
- Leave a gap in job priorities – I use evens
- Cache svn dump files

How you can use Bacula

- Backup >1 system
- Multiple File Daemons and bacula-fd.conf
- Handle catalog differently
- Might handle pools differently

How you can use Bacula

- Back up Windows systems
- Back up MacOS systems
- And Solaris, BSD,

How you can Use Bacula

- Get an LTO-6 changer and back up 3 TB per cartridge!

Where to go next

- blog.bacula.org/documentation/documentation/
- wiki.bacula.org/doku.php
- bugs.bacula.org
- bacula-users@lists.sourceforge.net
- ULSAH, 4th Ed. (Nemeth, et al) Chapter 10, section 8, pages 318-335

Bacula

STLLUG
15 May 2014

Bacula

STLLUG
15 May 2014

“It comes in the night and sucks the essence from your computers.”

Bacula is a networked client/server backup solution that creates cataloged backups of Unix, Linux, Windows, and MacOS systems on a wide variety of media.

Bacula: STLLUG meeting, 15 May 2014; Ken Johnson (contact-point@pobox.com)

One sentence summary –

Networked client/server – pieces on different systems, each doing what they do best.

Catalogs – know where to retrieve files without reading media

Supports many O/S

Supports many devices

About Me

I have used Bacula for six years on a server running Debian over two generations of system hardware and LTO tape drives.

I work as an independent consultant performing system and small network administration, and writing specialized technical documentation.

I have used Bacula once to restore the /home filesystem of this server (including the svn repositories) after a disk failure.

XML or SGML based mil-spec documents

Run the Visual C++ debugger or read a switchbox schematic.

Generate 'picture books' for custom test program sets using Python and MS Word.

Acknowledgments

- 'Bacula' is a registered trademark of Kern Sibbald.
- Excerpts from the Bacula documentation used under the Creative Commons Attribution-ShareAlike 4.0 International License.
- Excerpts from the Bacula source code (including template configuration files) used under Affero GNU General Public License version 3.

Outline

- Bacula Basics
- How I use Bacula
- How you can use Bacula
- Where to go next

Here's what we'll cover...

Bacula Components

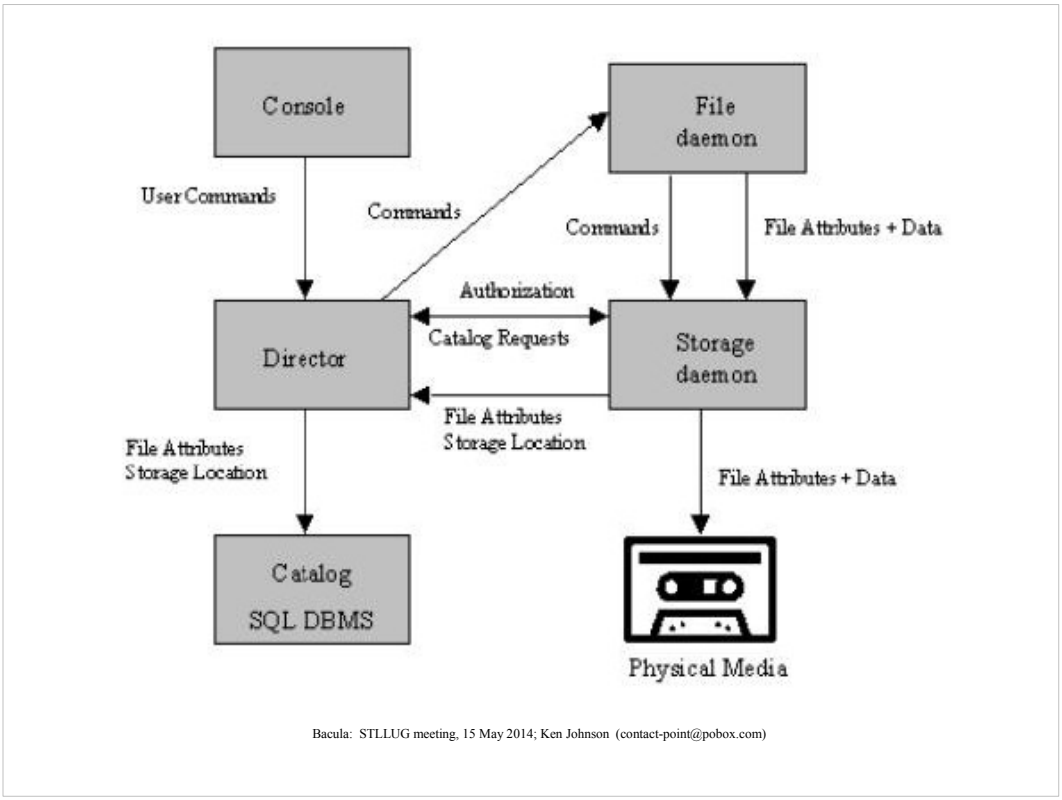
- Catalog
- Client File Daemon
- Console
- Director Daemon
- Storage Daemon
- Configuration files

In alphabetic order

These are the components you need to understand to set up Bacula.

(one-sentence summary of each)

Fewer than seven!



How the components interact. The lines are TCP/IP connections, except the Storage Daemon to Physical Media.

Catalog

- Record of files and filesystems stored
- Record of media usage
- Relational Database – MySQL, PostgreSQL, or SQLite

Bacula knows where to find a file or filesystem without reading backup media.

Bacula can track and limit media usage – most tapes have limits on passes; you can easily see how close you are to those limits.

Client File Daemon

- Runs on the system whose files you are backing up.
- Configured by bacula-fd.conf
- Needs to know: own name, keys, how to talk to the Director daemon, where to send messages.

Console

- The human interface
- Command-line console adequate
- Gui consoles available
- bconsole.conf
- Needs to know: own name, how to talk to the director

/etc/bacula on Debian.

Director Daemon

- Runs the show
- Hardest to configure, because...
- Needs to know everything
- bacula-dir.conf

Storage Daemon

- Writes to storage media
- Needs to know: own name, how to talk to Director, what to do with messages, what/how for the storage devices or media
- Bacula-sd.conf

Configuring the Director

- Jobs
- Pools and Media
- Filesets
- Messages
- Schedules

Jobs: Backup, Restore, and Admin. Defined in .conf file. Backup jobs ref. Filesets and Schedules.

Pools: collections of (labeled) media. Tapes or directories on filesystems.

Filesets: the filesystems or directory trees you want to back up. Excludes are possible

Messages: Where to send messages – typically email.

Schedules: When jobs are run and which pool to use. M-F except first Tuesday is straightforward.

How I use Bacula

- Why I chose Bacula
- Admin Jobs
- Backup Jobs
- Restore Job
- Hints and tricks

Additional details on my particular experience, with tips and tricks I found helpful.

Keep in mind as we go through this that this is the 'simplest case'. Bacula can handle much more complex environments.

Why I chose Bacula

- A timeline –
- No backups
- Ken arrives
- CDs – 700 MB
- DVDs – 4.7 GB
- DL-DVDs – 8 GB
- Now what?

Off-Site and offline backups were/are a requirement.

Full backups were/are a requirement (on the restore end).

Cron jobs, perl...

Why I chose Bacula

Off-Site and offline requirement.

Full backups requirement (on the restore end).

Bacula 2.x was well-developed, well documented, open-source backup tool that supported LTO (Ultrium) tape drives.

Off-Site protects against disasters.

Offline protects against malicious intruders...

Except possibly the GoodTimes virus.

Why I chose Bacula

Bacula was more complex and more capable than any backup tool I had used previously.

Once Bacula was set up and running, it just worked.

2.x to 5.x per Debian; LTO-2 to LTO-4.

Jobs

- Run according to Schedules
- Have Run before, Run after scripts
- Holidays via run before script
- Jobs run at the same time run in priority order

The jobs which write to tape have a separate schedule which selects a pool.

Jobs scheduled at the same time run in priority order.

Smaller priority numbers run first.

Period.py University of Madison

Admin Jobs

- Execute a shell script, to...
- Mount/dismount devices or media
- Extract database contents
- Clean up after a backup

MySQL databases extracted to .sql files, that are backed up, then deleted.

Svnadmin dump

Mount and dismount tapes

Backup Jobs

- Once for the data, once for the catalog
- 5 times a week (tape changes)
- Pools: 10 'daily', 12 Monthly, 12 Offsite, yearly
- Cached svn extracts

Period.py University of Madison

If you were backing up >1 system, you would handle catalog backups differently.

Backup Jobs – Main Job

```
Job {
  Name = "ServerBackup"
  FileSet = "ServerBackup"
  Priority = 12
  RunBeforeJob = "/etc/bacula/holiday_check.py"
  RunAfterJob  = "/etc/bacula/cleanup.pl"
  Type = Backup
  Schedule = "NightlySave"
  Client = linux2-fd
  Storage = QuantumUltriumLTO-4
}
```

Backup Jobs - Fileset

```
FileSet {
  Name = "ServerBackup"
  Include {...}
    File = /home
}
```

- Others: /root, /etc, /var/lib, /usr/local, /opt
  and database extracts. And Excludes, of course

Here's a simple fileset definition.

Backup Jobs - Schedule

```
Schedule {
  Name = "NightlySave"
  Run = Pool=Yearly     mar     3rd Wed at 02:00
  Run = Pool=Monthly    monthly 2nd Wed at 02:00
  Run = Pool=OffSite    monthly 1st wed at 02:00
  Run = Pool=Daily            tue     at 02:00
 ...
```


Here's a schedule definition.

Backup Jobs - Schedule

```
  ...
  Run = Pool=Daily          thu-sat at 02:00
  Run = Pool=Daily   monthly 4th Wed at 02:00
  Run = Pool=Daily   monthly 5th Wed at 02:00
  Run = Pool=Daily   jan-feb 3rd Wed at 02:00
  Run = Pool=Daily   apr-dec 3rd Wed at 02:00
}
```


Here's a schedule definition.

Five-Nightly Jobs

- Mount Tape
- Extract special data
- Backup and cleanup
- Extract Catalog
- Backup and cleanup
- UnMount Tape

Restore Job

- One defined restore job, 'fill in the blanks'
- Select a job ID via menu of options
- Select files to be restored
- Five minutes typical to define the restore.

I typically do this twice a year.

I keep a how-to file in /etc/bacula.

Restore file selection

- **cd** **change current directory**
- **done** **leave file selection mode**
- **find** **find files, wildcards allowed**
- **ls** **list current directory**
- **mark** **mark dir/file to be restored**
- **pwd** **print current working directory**
- **quit** **quit and do not do restore**


```
cd          count       dir         done
estimate    exit        find        help
ls          lsmark      mark        markdir
pwd         unmark      unmarkdir   quit
```

Hints and Tricks

- Be wary of reusing names in different categories
- Schedule Indefinite hold – February 31
- Leave a gap in job priorities – I use evens
- Cache svn dump files

How you can use Bacula

- Backup >1 system
- Multiple File Daemons and bacula-fd.conf
- Handle catalog differently
- Might handle pools differently

How you can use Bacula

- Back up Windows systems
- Back up MacOS systems
- And Solaris, BSD,

How you can Use Bacula

- Get an LTO-6 changer and back up 3 TB per cartridge!

Where to go next

- blog.bacula.org/documentation/documentation/
- wiki.bacula.org/doku.php
- bugs.bacula.org
- bacula-users@lists.sourceforge.net
- ULSAH, 4th Ed. (Nemeth, et al) Chapter 10, section 8, pages 318-335

Hold up Nemeth, et al.

Bacula

STLLUG
15 May 2014

“It comes in the night and sucks the essence from your computers.”