Bacula and ZFS

Great tools for use with PostgreSQL

Chapter 1: Bacula

Dan Langille
PGCon 2018

What we won't cover

- Installation.
- Concurrent jobs.
- lots of other things.

Disclaimer

- Everyone is biased
- Personal experiences
- Personal preferences

What is Bacula?

- Set of programs.
- client-server model.
- Backup, recovery, and verification of data.
- Network of computers of different kinds.
- Backup to disk/tape.

http://www.bacula.org/5.2.x-manuals/en/main/main/What_is_Bacula.html

HOT TIP

- Bacula does not use tar. For disaster recovery, use **bextract** or **bls**
- Best practice: copy .conf and .sql files to multiple accessible locations so you never have to use **bextract**.

Overland & Digital

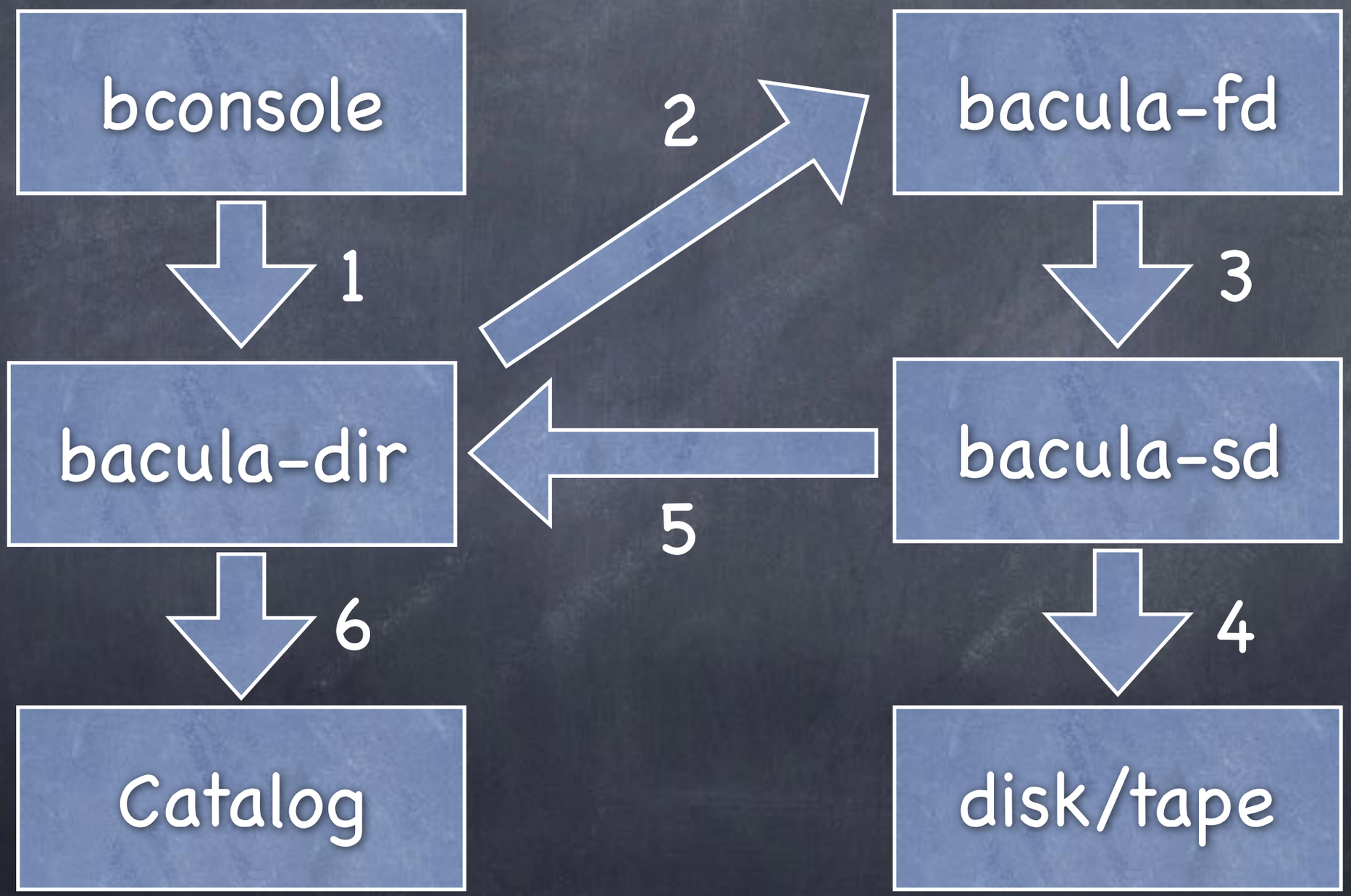
My backup strategy

- back up to local disk, then copy to tape
- was DLT with DLT-7000 drives, then SDLT, & now LTO-4
- keep full backups for three years (on both disk and tape)
- take most recent full backups off-site for 3 months

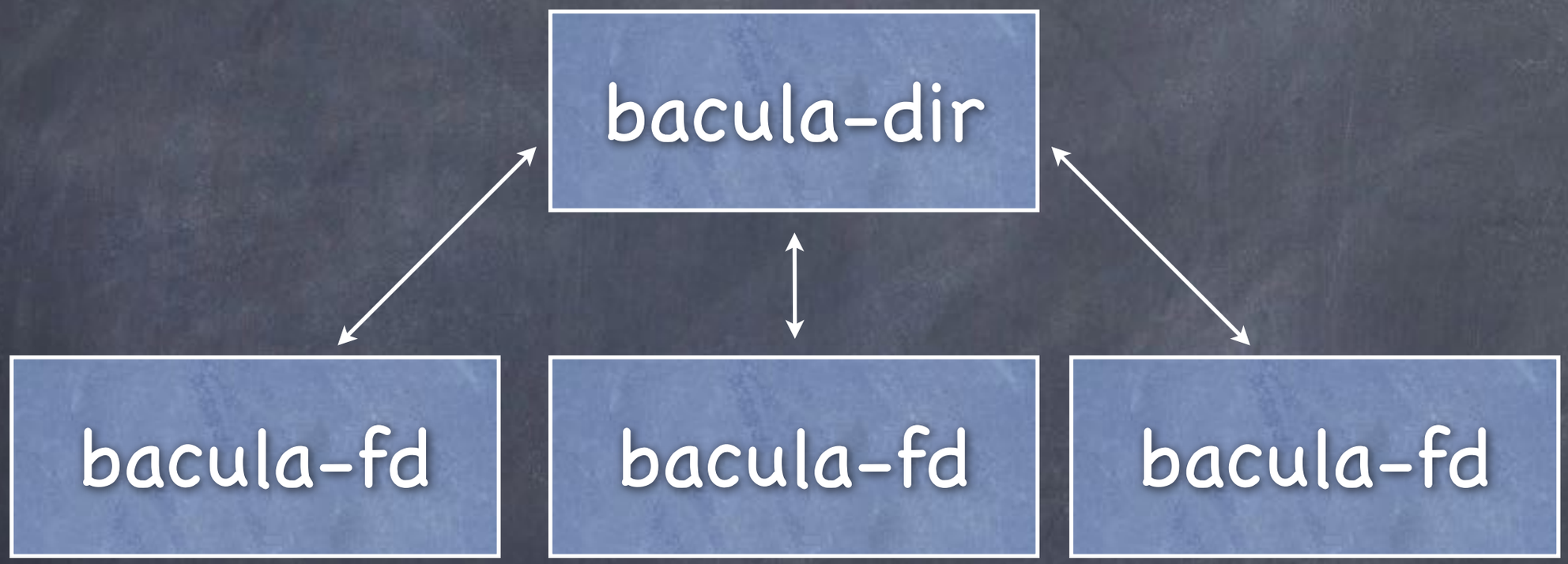
Abbreviations & Terms

- **DIR** = bacula-dir = Director
  - knows & starts EVERYTHING
- **SD** = bacula-sd = Storage Daemon
  - stores everything but knows nothing
- **DIR** & **SD** often referred to as **server**
- **FD** = bacula-fd = File Daemon = Client
  - often a server, but referred to as a **Client**

Steps in a backup

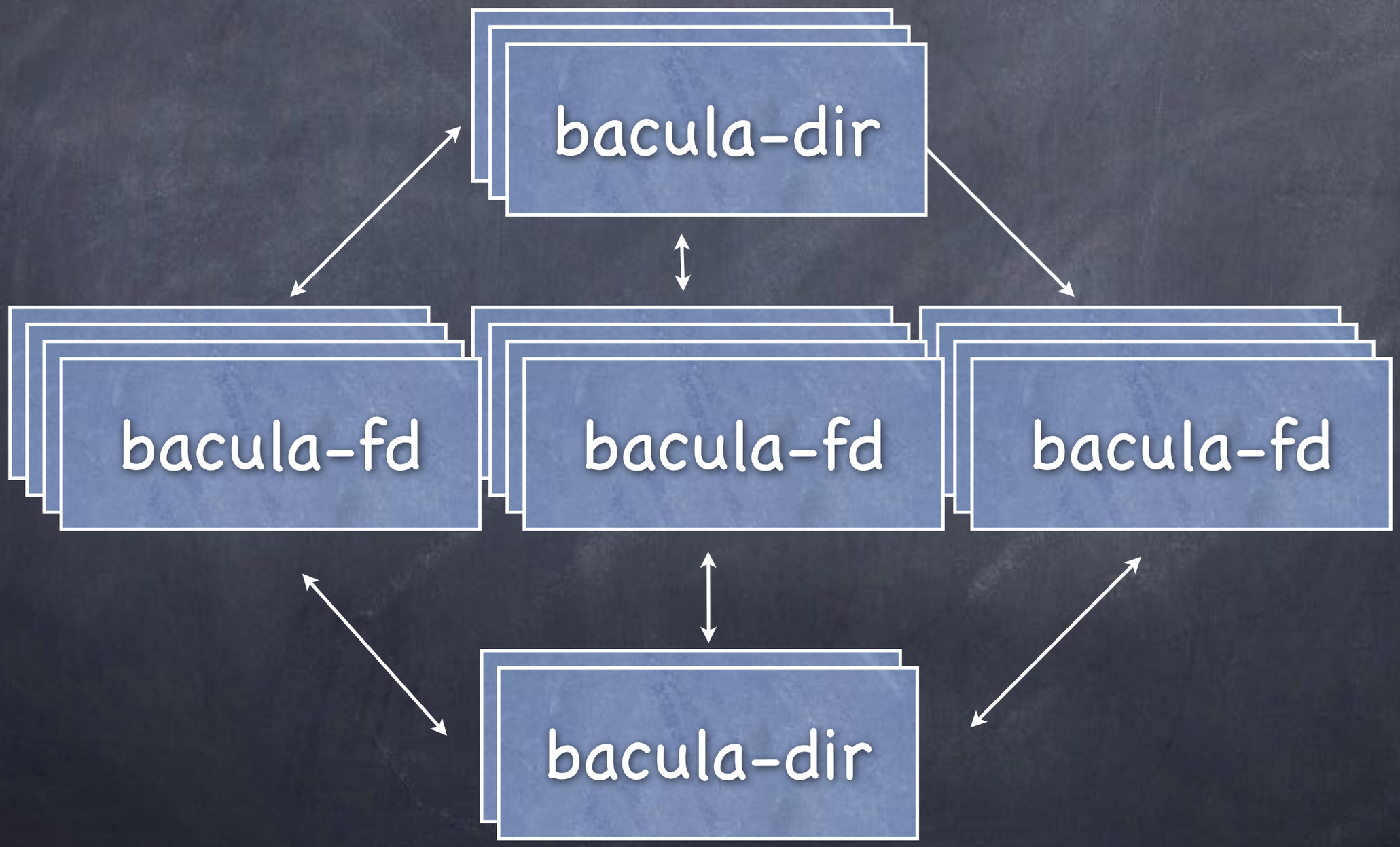


The usual starting point

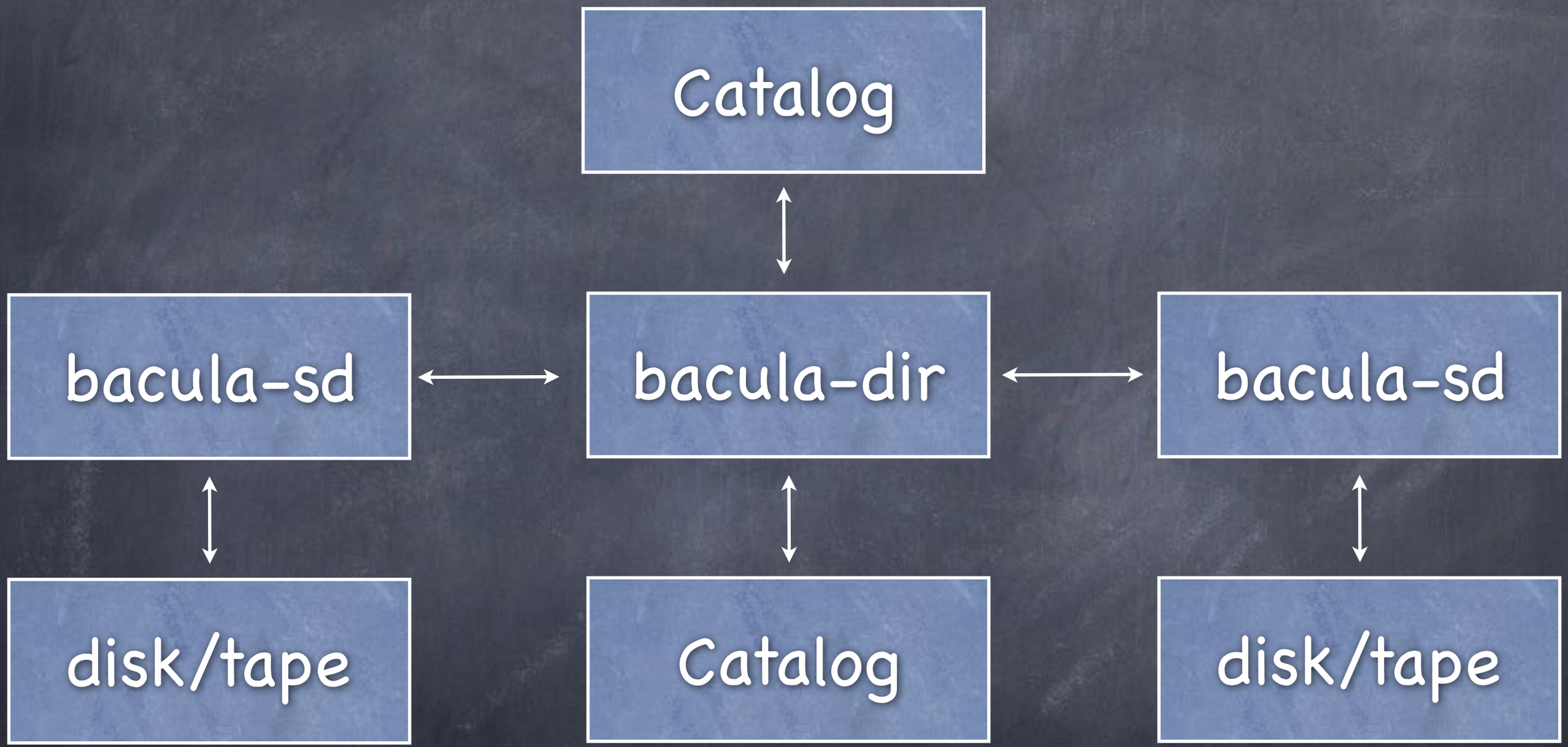


The usual starting point

Advanced



Advanced


Catalog
bacula-sd
bacula-dir
bacula-sd
disk/tape
Catalog
disk/tape

running a backup

- automatic - not based on cron(8)
- manual - from the command line (more or less)
- many configuration options when run manually

restore

- cannot be scheduled
- but can be automated
- usually run from bconsole using restore command

HOT TIP!

- echo 'run job=dent yes' | bconsole

```
Connecting to Director
bacula.example.org:9101
1000 OK: bacula-dir Version: 5.2.12 (12
September 2012)
Enter a period to cancel a command.
run job=dent yes
Using Catalog "MyCatalog"
Job queued. JobId=123679
```

Bacula tools

- bconsole
- btape
- bat
- bsmtp
- bwild
- bextract
- bcopy
- bscan
- btraceback
- dbcheck
- bregex
- chio-bacula

bconsole(8)

- the best user interface
- works
- heavily tested
- used to conduct regression tests
- status, run, restore, maintenance

bconsole commands

- . <= escape character; use it to get out of a command
- status – what's happening on a client, storage, or director
- run
- restore
- m (short for messages)

btape(8)

- If not using tape, ignore this.
- Use to test your tape drive with respect to Bacula.
- You must do this if using tape.
- You will regret it if you do not.

HOT TIP!

- in status output, do not worry about old jobs or clients.
- these are temp logs.
- don't waste time trying to clear them out.
- They will rotate out eventually.

daemons run as what users?

- bacula-dir runs as **bacula:bacula**
- bacula-sd runs as **bacula:bacula**
- bacula-fd runs as **root:wheel**
- on systems with bacula-sd, I put bacula in the operator group to access tapes

passwords = shared

- Thus, every password is stored in two locations:
  - In the bacula-dir.conf file.
  - In the FD/SD/bconsole configuration file.
- Thus, it is a shared secret.
- THIS IS VERY IMPORTANT

bconsole configuration

- bconsole.conf
- What DIR do you want to contact:

```
Director {
   Name       = “dirName”
   DIRport    = 9101
   address    = bacula.example.org
   Password = "passwd for dirName"
}
```

DIR configuration

- bacula-dir.conf
- defines what DIR am I?

```
Director {
Name              = dirName
DIRport           = 9101
Password          = "passwd for dirName"
Messages          = Standard
WorkingDirectory = "/home/bacula/working"
PidDirectory       = "/var/run"
...
```

Name/Password is wrong

```
$ bconsole
Connecting to Director bacula.example.org:9101
Director authorization problem.
Most likely the passwords do not agree.
If you are using TLS, there may have been a
certificate validation error during the TLS
handshake.
Please see http://www.bacula.org/en/rel-manual/
Bacula_Freque_Asked_Questi.html#SECTION00376000
0000000000000 for help.
```

HOT TIP!

- When it says the name and password do not match....
- check to see if the name and passwords match.
- right client? right hostname/address?

SD & FD configuration

- bacula-dir.conf / bacula-sd.conf
- Every SD and FD needs at least one entry like this:

```
Director {
   Name       = dirName
   Password = "passwdForThisSD/FD"
}
```

FileSet

- A FileSet is a list of files / directories to backup.
- A FileSet can be used by zero or more jobs.
- Exactly one FileSet per job.
- Can specify files / directories to exclude.

defining a Client resource

- in bacula-dir.conf:

```
Client {
  Name        = nyi-fd
  Address     = nyi.example.org
  FDPort      = 9102
  Catalog     = MyCatalog
  Password = "passwd for NYI"

  File Retention = 3 years
  Job Retention  = 3 years
}
```

defining a client

- in bacula-fd.conf:

```
Director {
   Name       = bacula-dir
   Password = "passwd for NYI"
}

FileDaemon {
   Name                  = nyi-fd
   FDport                = 9102
   WorkingDirectory = /home/bacula/working
   Pid Directory     = /var/run
}
```

Schedule Resource

```
Schedule {
  Name = "WeeklyCycle"

  Run = Level=Full                1st sun      at 5:55
  Run = Level=Differential 2nd-5th sun at 5:55
  Run = Level=Incremental  mon-sat      at 5:55
}

Schedule {
  Name = "Never"
}
```

defining a Job resource

- in bacula-dir.conf:

```
Job {
  Name     = "nyi basic"
  JobDefs = "DefaultJobRemote"
  Client   = "nyi-fd"
  FileSet = "basic backup"
}
```

Job basics

- A job runs on exactly one client.
- A job consists of exactly one FileSet.
- A job backs up to exactly one SD.
- A job has just one schedule, which can be simple or complex.
- You can have multiple jobs per client.

JobDefs

```
JobDefs {
   Name        = "DefaultJobRemote"
   Type        = Backup
   Level       = Incremental
   Client      = ngaio-fd
   FileSet     = "Full Set"
   Schedule    = "WeeklyCycle"
   Storage     = MegaFile
   Messages    = Standard

...
```

JobDefs II

```
Pool = FullFile

Full            Backup Pool = FullFile
  Differential Backup Pool = DiffFile
  Incremental  Backup Pool = IncrFile

  Priority              = 20

  Spool Data            = no
  Spool Attributes = yes
}
```

Job Level

- Full – backup **everything** in the FileSet.
- Incremental - all files in the FileSet that have changed since the **last successful backup***.
- Differential - all files specified in the FileSet that have changed since the **last successful Full backup***.
  - * of the the same Job using the same FileSet and Client

What to backup?

- Full = everything
- Incremental / Differential: only changes
- look at st_ctime & st_mtime
- Moving files messes with this
- new location, same times

Accurate Backup

- Accurate = yes
- list of files sent to FD
- directories and paths
- needs more CPU/RAM

Virtual Backups

- Like doing a full backup every time!
- But without copying data from client.

```
run job=MyBackup level=VirtualFull
```

Schedule

- Jobs are run automatically according to the schedule assigned to that job.
- A Schedule can be used by zero or more jobs.
- A Schedule can indicate that a job is never run automatically (i.e. manually only).

HOT TIP!

- If you make a change to your FileSet, the next run of any Job involving that FileSet will be promoted to a Full.
- This FileSet directive avoids that upgrade (at a price):
  - Ignore FileSet Changes = yes

Volumes

- A Volume is a place to put a backup.
- Not to be confused with filesystem volumes.
- It may be disk, tape (DVD – not really supported any more).
- Bacula treats disk and tape the same (more or less).
- A backup resides may span Volumes.

Pool

- A Pool is a collection of Volumes with similar attributes.
- A Volume is created based upon a Pool definition.
- You can have multiple Pools.
- A Volume must belong to exactly one Pool.

Pool (II)

- The common Pool attributes are:
  - Name
  - Pool Type (usually Backup)
  - Recycle (yes/no)
  - Volume Retention
  - Storage (what SD is this Pool located at?)
  - LabelFormat (not recommended for bar code enabled tape libraries)

Pool FullFile

```
Pool {
  Name                        = FullFile
  Pool Type                   = Backup
  Recycle                     = yes
  AutoPrune                   = yes
  Volume Retention            = 3 years
  Storage                     = MegaFile
  Next Pool                   = Fulls
  Maximum Volume Bytes = 5G
  LabelFormat                 = "FullAuto-"
}
```

Pool DiffFile

```
Pool {
  Name                            = DiffFile
  Pool Type                       = Backup
  Recycle                         = yes
  AutoPrune                       = yes
  Volume Retention                = 6 weeks
  Storage                         = MegaFile
  Next Pool                       = Differentials
  Maximum Volume Bytes = 5G
  LabelFormat                     = "DiffAuto-"
}
```

Pool IncrFile

```
Pool {
  Name                        = IncrFile
  Pool Type                   = Backup
  Recycle                     = yes
  AutoPrune                   = yes
  Volume Retention            = 3 weeks
  Storage                     = MegaFile
  Next Pool                   = Incrementals
  Maximum Volume Bytes = 5G
  LabelFormat                 = "IncrAuto-"
}
```

HOT TIP!

- Bacula will not label a volume which is already labeled (i.e. a tape)

```
mt -f /dev/nsa0 rewind
mt -f /dev/nsa0 weof
```

Defining Storage Resources

- Much like client, you have a Name, Address, and Password
- Passwords appear twice; bacula-sd.conf and in bacula-dir.conf

the storage resource

- in bacula-dir.conf:

```
Storage {
   Name          = MySD
   Address       = storage1.example.org
   SDPort        = 9103
   Password      = "MySDPasswordFOO"
   Device        = FileStorage
   Media Type = File
}
```

the storage daemon

- in bacula-sd.conf:

```
Storage {
   Name                = kraken-sd
   SDPort              = 9103
   SDAddress           = 10.0.0.12
   WorkingDirectory = "/bacula/working"
   Pid Directory       = "/var/run"
}
```

Who can contact me?

- in bacula-sd.conf:

```
Director {
   Name       = bacula-dir
   Password = "MySDPasswordFOO"
}
```

backup Device

- in bacula-sd.conf:

```
Device   {
   Name                 = MegaFile
   Media Type           = File
   Archive Device = /bacula/volumes
   LabelMedia           = yes
   Random Access  = yes
   AutomaticMount = yes
   RemovableMedia = no
   AlwaysOpen           = no
}
```

Catalog

- The Catalog is a list of what was backed up, when, and from what client.
- The Catalog is stored in a Database.

```
Catalog {
   Name = MyCatalog
   dbname = bacula; dbaddress =
localhost; user = bacula; password = ""
}
```

What's in a Catalog?

- Data within the Catalog includes:
  - What Jobs were run.
  - The FileSet used.
  - The list of files that were backed up.
  - Optional checksum of each file.
  - Where that backup is located.
  - What client it was run on.
  - List of Pools.
  - List of Volumes in that Pool.

With a Catalog, you can:

- Think about what you just read...
- What you can do with it...
- You can restore anything...
- from anywhere...
- to anywhere...
- on any client...
- from bconsole.

Retention means Catalog

- Retention determines how long entries are retained in the Catalog.
- Retention is only indirectly related to how long backups will remain within a Volume.
- Backups might still be available after Retention expires, but don't count on it.
- More on Retention later.

Catalogs grow/shrink

- Catalogs grow. Disk space is cheap. Use it.
- Data is manually removed from the Catalog via the prune and purge commands:
  - Pruning – removes data from the Catalog based upon Retention times
  - Purging – removes data from the Catalog, completely ignoring Retention times (e.g. rm -rf)
- Pruning can done via admin job or after every job.

Lost Catalog?

What to do?

- What if you lose your Catalog?
- What? No backup?
- daily cron job to copy *.conf and *.sql
- bextract is your best tool for backup retrieval after Retention expires; I have never used it and I wish I never have to.
- I hope you never had to use it either.

Catalog rules!

- Your Catalog is your best tool.
- Your Catalog is more important than your backups.
- Heavily used for restores.
- Without your Catalog, what you have it about the same as a tarball, more or less.
- The Catalog knows where everything is and constructs the right procedure to restore it properly.

Recycling

- Bacula will do everything it can avoid overwriting a Volume
  - EVERYTHING!
  - Overwriting is known as Recycling
  - Learn the Bacula Recycling Algorithm (it is in the documentation)

HOT TIP!

- For my tapes, I initially put no limits on my pools.
- I wait to see how long it takes to run out of tapes.
- Then prune until I have enough free tapes.
- Then set max num volumes.
- Could do similar with disk pools.

Retention

- Three types:
  - Volume
  - File
  - Job
- Retention refers to Catalog, not Volumes.

My retention

- Job Retention = 3 Years
- File Retention = 3 Years
- Volume Retention = variable depending on goal of Pool
- I suggest always having File = Job Retention

Passwords

- plain text
- not encrypted
- relies on filesystem security
- never passed in plain text

Databases

- Pick your religion.
- As the author of the PostgreSQL backend, I always prefer PostgreSQL.

disk versus tape

- Some people love tape.
- Some people loathe tape.
- Why have tape when you can have disk?
- I love tape.
- I also use disk. Lots of disk.
- On ZFS.

What's the diff?

- Not much.
- Bacula treats them the same, more or less.
- For file Volumes, Bacula creates a file with the same name as the label.
- Newbies run into disk space problems because they haven't monitored the free disk space and fail to implement a strategy.

Running a Job

- start bconsole

```
$ bconsole
Connecting to Director
bacula.example.org:9101
1000 OK: bacula-dir Version: 5.2.12
(12 September 2012)
Enter a period to cancel a command.
```

Running a Job

```
*run job=dent
Run Backup job
JobName: dent
Level: Incremental
Client: dent-fd
FileSet: dent files
Pool: FullFile (From Job resource)
Storage: MegaFile (From Pool
resource)
When: 2013-01-27 17:41:32
Priority: 10
OK to run? (yes/mod/no): yes
Job queued. JobId=118611
*
```

Restoring a Job

- You need just one restore Job
- You can override all Job attributes at run time
- Lots of restore options
- Mark files you want
- Restore to a different client

Storing a Job

```
*restore client=dent-fd
First you select one or more JobIds that contain files
to be restored. You will be presented several methods
of specifying the JobIds. Then you will be allowed to
select which files from those JobIds are to be restored.
To select the JobIds, you have the following choices:
1: List last 20 Jobs run
2: List Jobs where a given File is saved
3: Enter list of comma separated JobIds to select
4: Enter SQL list command
5: Select the most recent backup for a client
6: Select backup for a client before a specified time
7: Enter a list of files to restore
8: Enter a list of files to restore before a specified time
9: Find the JobIds of the most recent backup for a client
10: Find the JobIds for a backup for a client before a specified
time
11: Enter a list of directories to restore for found JobIds
12: Select full restore to a specified Job date
13: Cancel
Select item: (1-13): 5
```

Insert demo here

Tape Libraries

- No Bacula drivers required.
- If your OS can talk to the tape library, then Bacula can.
  - use mtx-changer script supplied with Bacula
  - bacula user needs access to devices & scripts
  - alter permissions on devices if required
  - or add bacula to the appropriate groups if appropriate

Tape Libraries (II)

- run btape tests
- test spanning tape backups
- patience
- My experiences with tape libraries:
- http://www.freebsddiary.org/tape-library-integration.php
- http://www.freebsddiary.org/tape-library.php

HOT TIP!

- use sudo to test bacula commands

```
su -m bacula -c mtx-changer ...
```

Tips

- FileSet changes cause Full
- onefs will not descend
- When a disk Volume is recycled, it is first truncated before writing
- On DragonflyBSD, if backing up to disk, set your history off / small to avoid soaking up disk space with daily versions of each Volume you write to.

Spooling

- spool backup to HDD before writing to tape
- avoid shoeshine (start, stop, start, stop) of tape
- can increase throughput

```
set Spool Data = yes
```

HOT TIP!

- When spooling attributes, do not worry about status dir != status client
- The backup Job will finish; Client done.
- Director then updates the database.

Don't waste your time!

- Labels / Volume names.
- e.g. laptop-2013-01-13.from.Paris
- Just keep it simple like INC-50023
- Don't worry about counters


And we're done!