Nobody Ever Regretted Making a Backup

Dan Langille
BSDCan 2013

man 1 id

- Dan Langille, FreeBSD user since 1998
- Software Engineer with System Adminstrator and DBA tendencies
- main author, The FreeBSD Diary
- herder of cats for BSDCan and PGCon
- author of PostgreSQL module for Bacula
- main author of http://FreshPorts.org

My network

- home network of gateway, jail server (12 jails), three development servers, MacBook, and Windows box
- public facing servers: 2 in New York, 1 in Austin, 1 in Montreal, another in ... I don't know where that one is
- each backed up by bacula-fd

What will we cover

- The more misunderstood or under-appreciated aspects of Bacula.
- The trips and hurdles that I encountered and that I see others encounter.
- Won't be complete, but you will be ready.

What we won't cover

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

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

What is Bacula? (II)

- Modular.
- Scalable.
- Robust.
- Reliable.

HOT TIP

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

My bacula network

- bacula-dir - at home
- bacula-sd also on bacula-dir server
- another bacula-sd on jail server with 12TB on ZFS on 8x2TB Seagate drives
- attached are two tape libraries
- Building up 2 more ZFS servers with 27TB and 18TB respectively.

Overland & Digital

Bacula Enterprise

- provided by commercial enterprise
- commercial open source
- features often migrate to community edition

Bacula

- Sometimes referred to as community edition
- It's what's in FreeBSD, NetBSD, OpenBSD, etc
- Unless otherwise specified, these slides are about Bacula, not Bacula Enterprise.

What do you want from a backup system?

- state your terms!

file locations

- On DragonflyBSD: /usr/pkg/etc/bacula/
- On FreeBSD: /usr/local/etc/
  - but soon to be /usr/local/etc/bacula/
- On NetBSD: /usr/pkg
- On OpenBSD: /etc/
- On OS-X: /opt/local/etc/bacula/

Danger!

Danger!

- Spend a lot of time on boring stuff.

Danger!

- Spend a lot of time on boring stuff.
- You may fall asleep.

Danger!

- Spend a lot of time on boring stuff.
- You may fall asleep.
- But in the long term, you will appreciate it.

Danger!

- Spend a lot of time on boring stuff.
- You may fall asleep.
- But in the long term, you will appreciate it.
- Later.

Danger!

- Spend a lot of time on boring stuff.
- You may fall asleep.
- But in the long term, you will appreciate it.
- Later.
- We won't cover everything!

Incomplete examples

- This screen is small.

My backup strategy

- back up to local disk
- copy to tape (DLT with DLT-7000 drives)
- keep full backups for three years (on both disk and tape)
- take most recent full backups off-site for 3 months

Retention

- Full - 3 years
- Differential - 6 weeks
- Incremental - 3 weeks

What is important?

- Backups are not the most important thing you will do.
- You can do all the backups you want. They are useless....

Restore is everything

- ....useless if you cannot restore

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**

Abbreviations & Terms II

- Bacula resources will be capitalized
  - FileSet
  - Client
  - Storage
  - Volume
  - Retention
- etc

Steps in a backup

bconsole

bacula-fd

bacula-dir

bacula-sd

Catalog

disk/tape

Steps in a backup

bconsole

bacula-fd

1

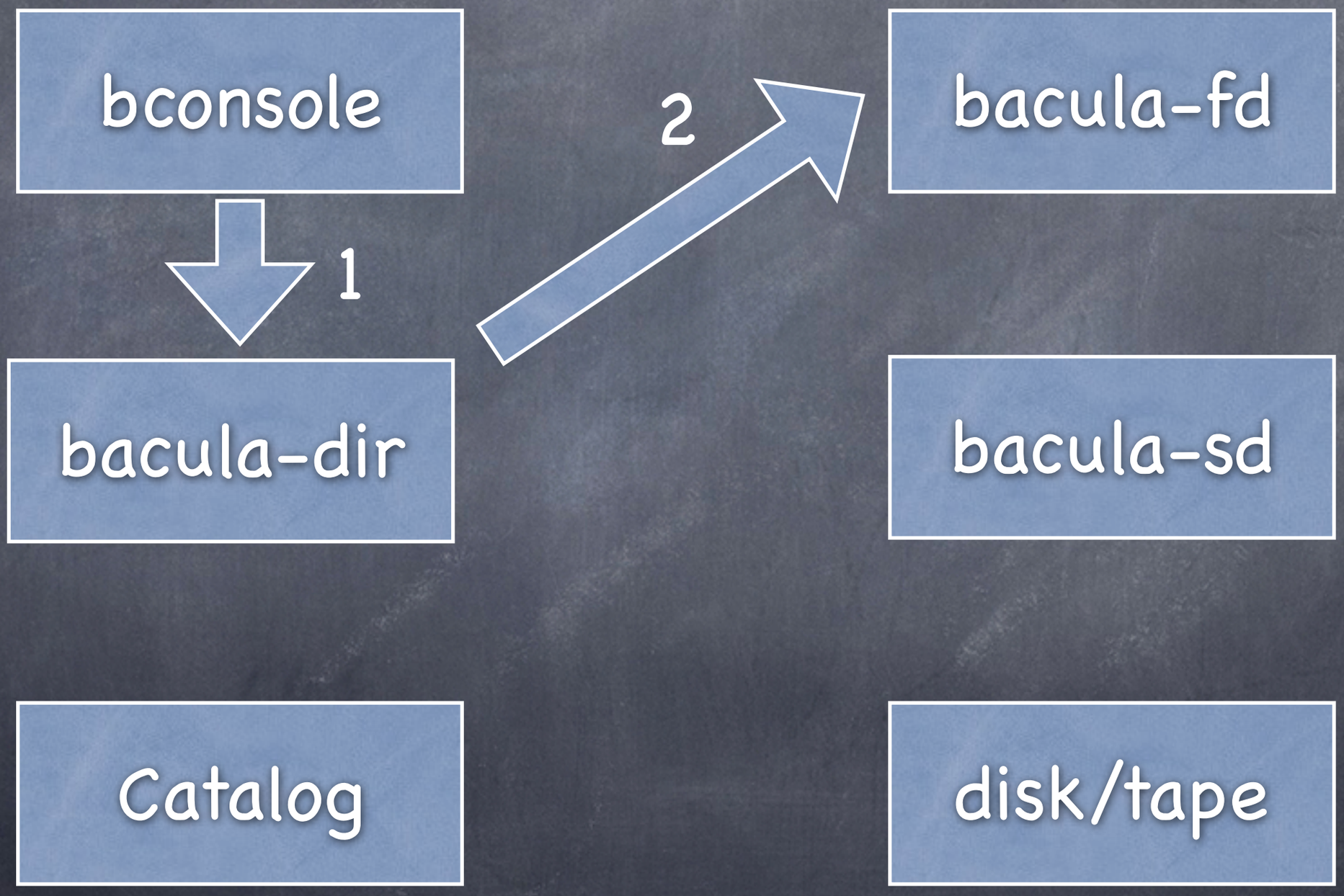
bacula-dir

bacula-sd

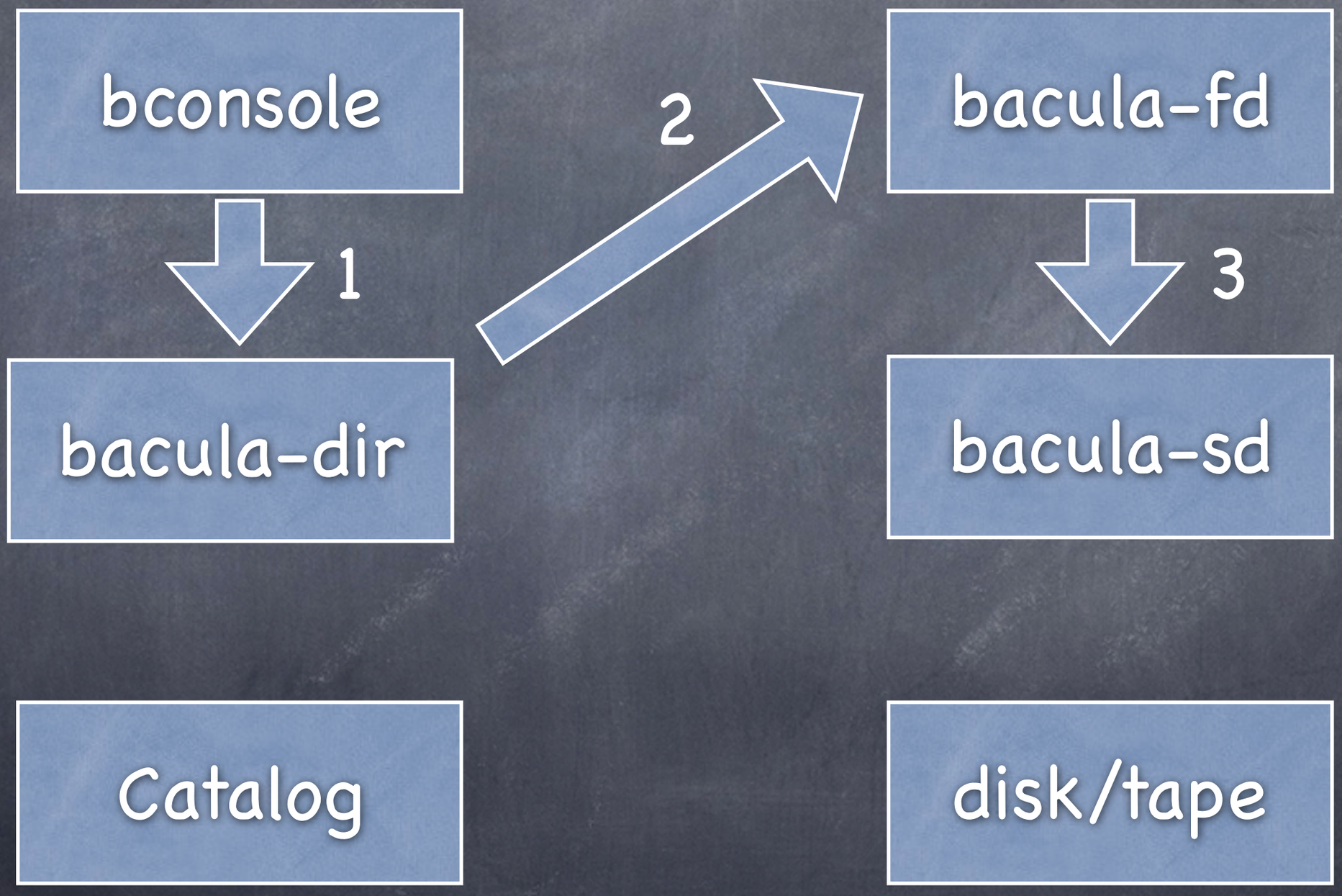
Catalog

disk/tape

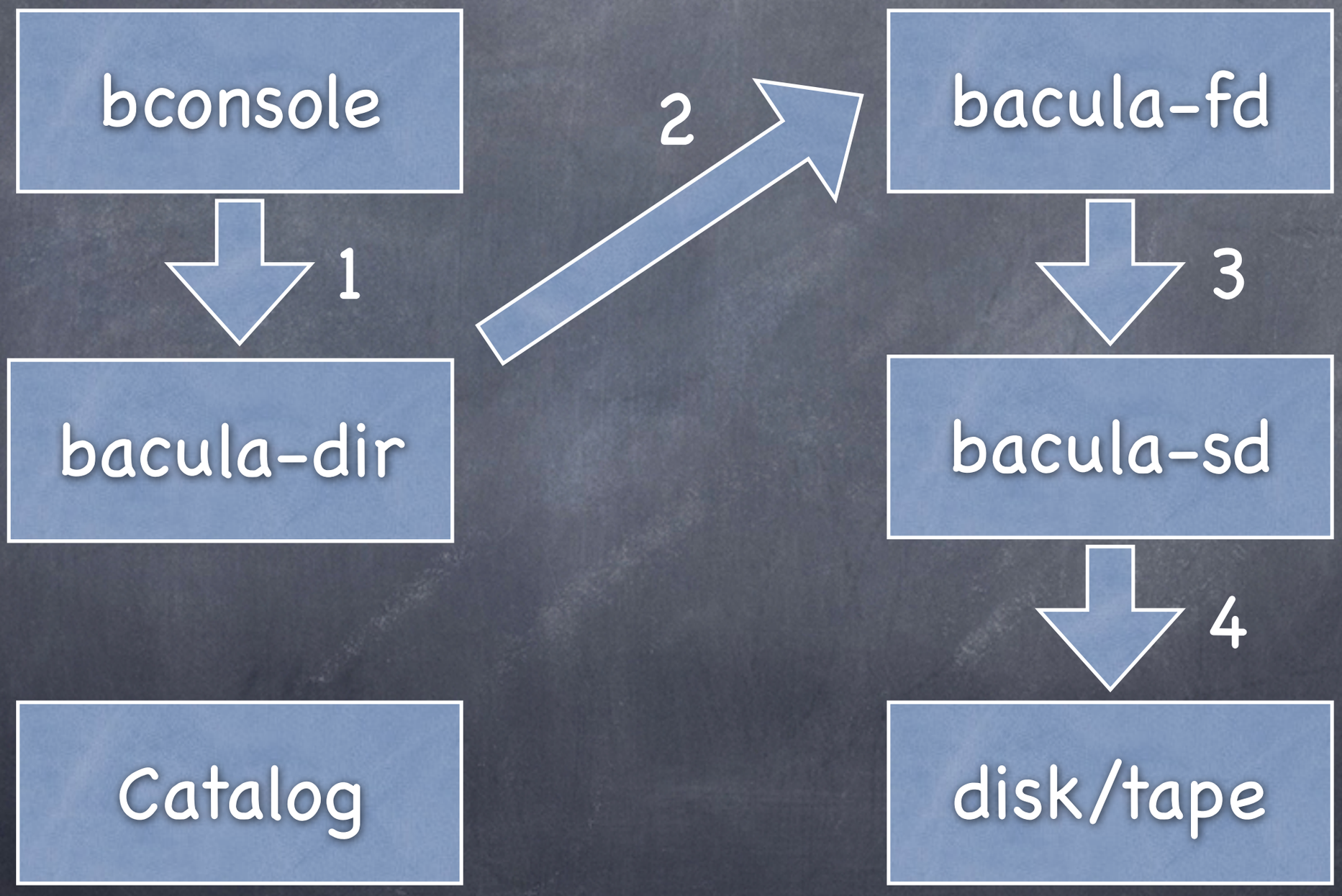
Steps in a backup



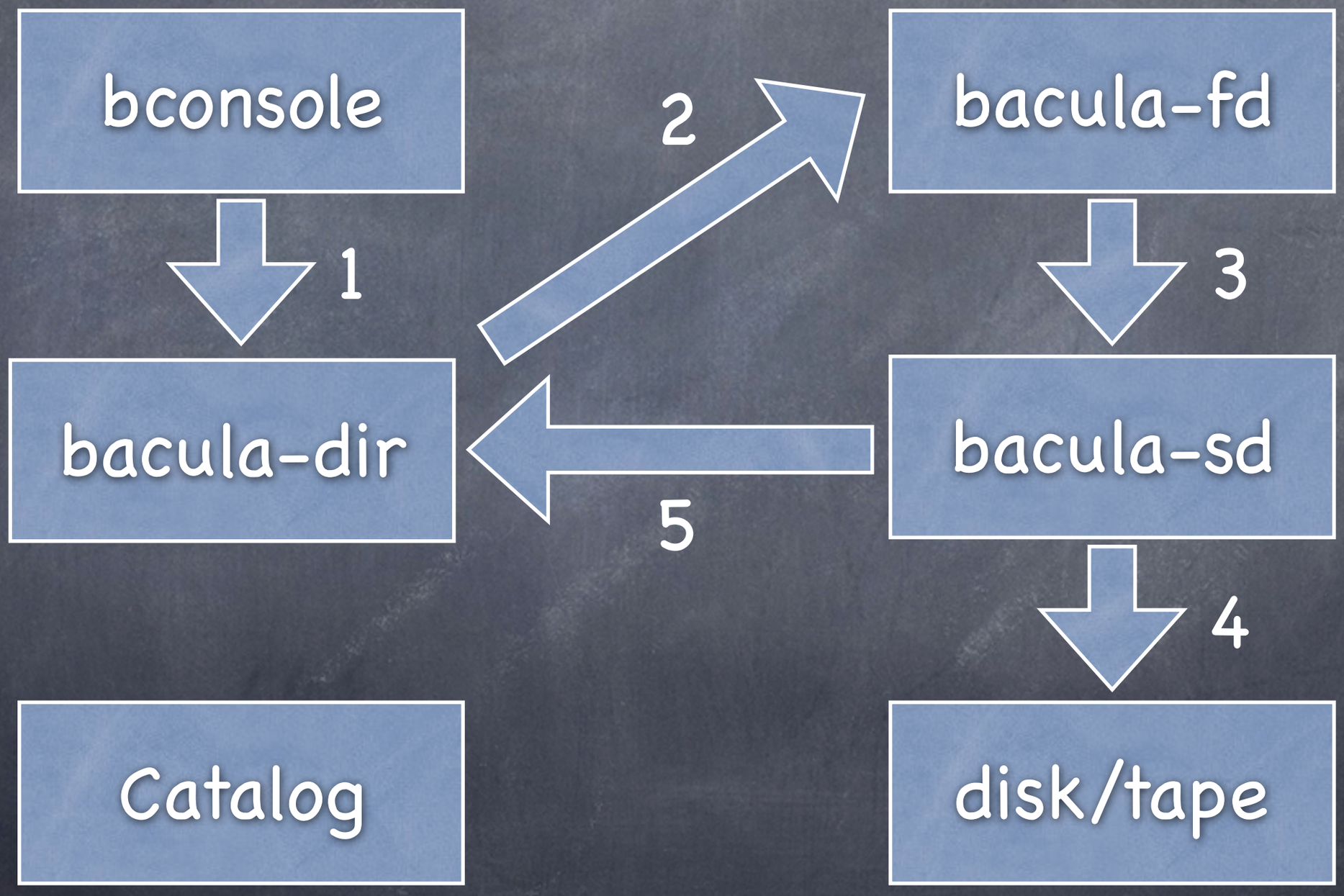
Steps in a backup



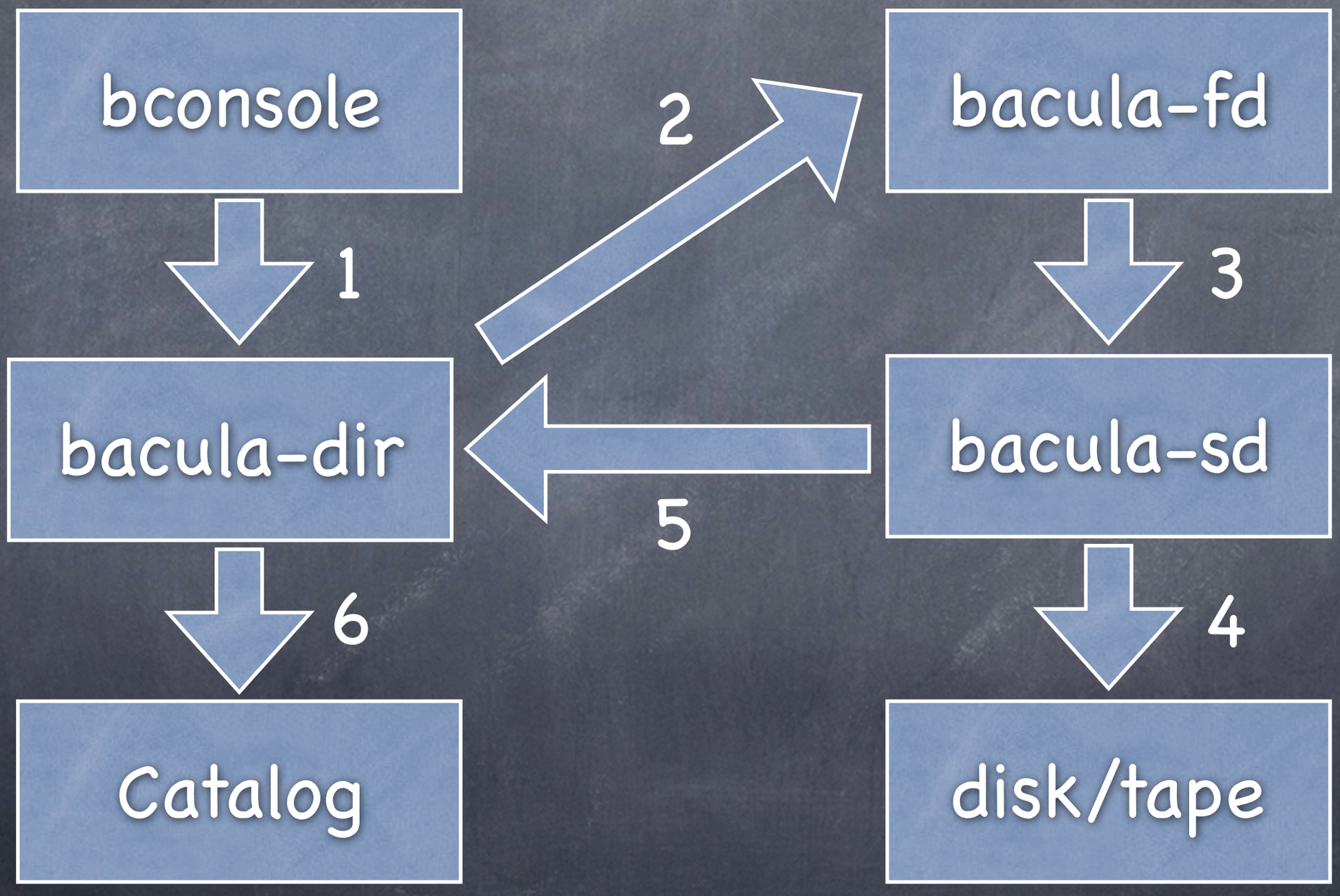
Steps in a backup



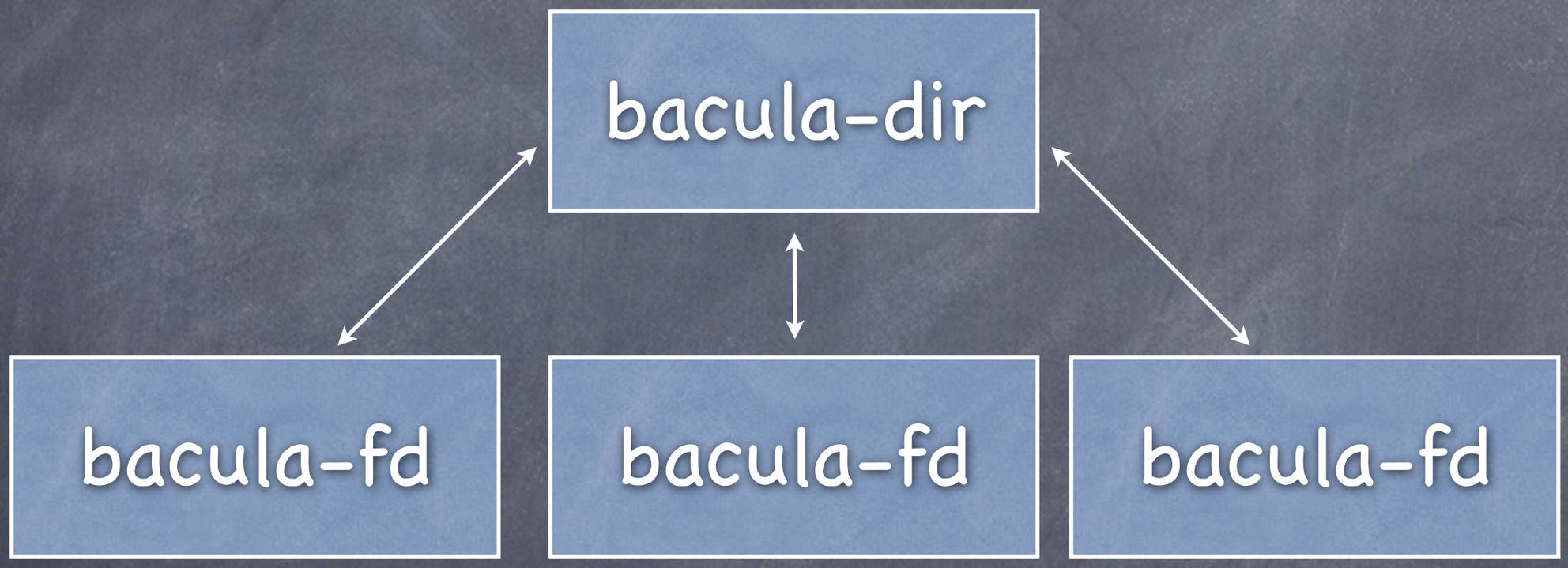
Steps in a backup



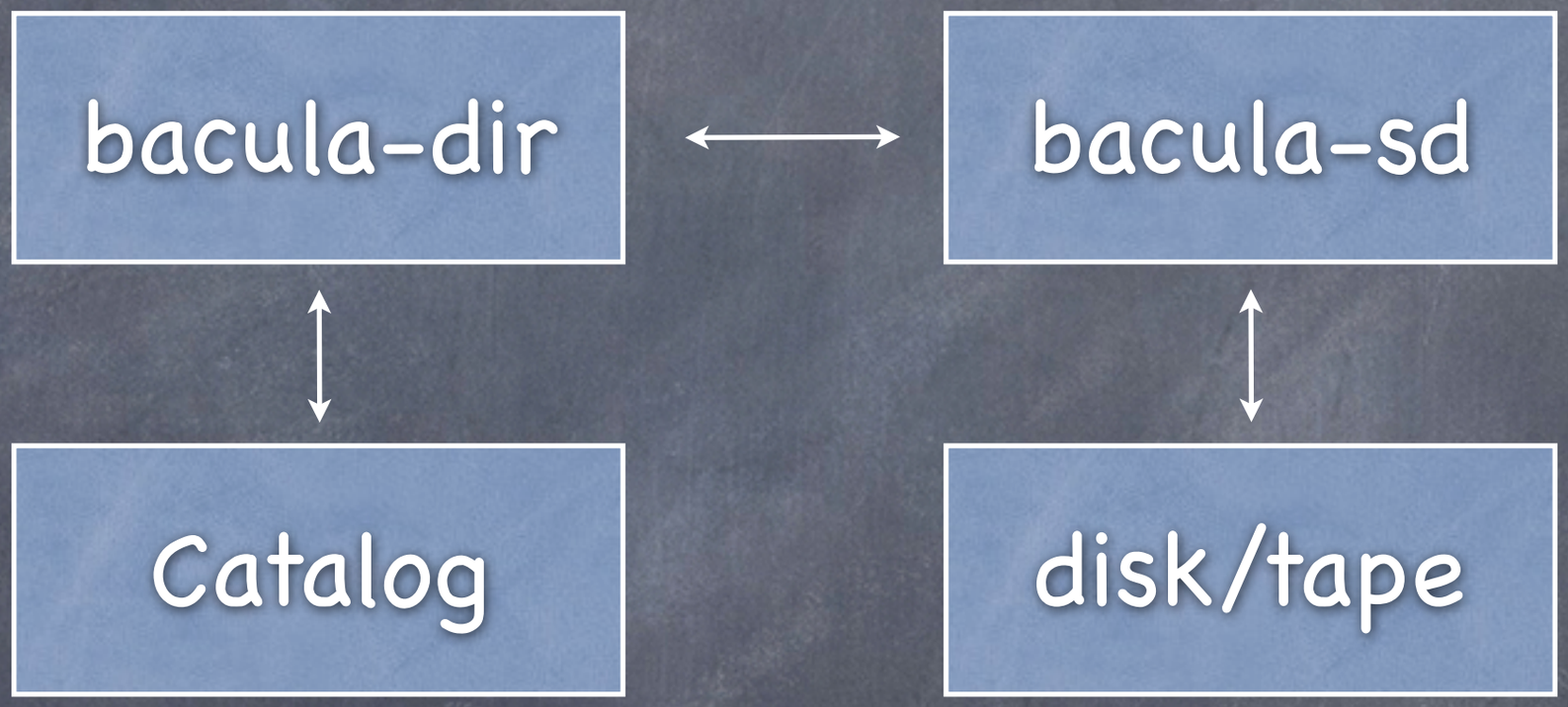
Steps in a backup



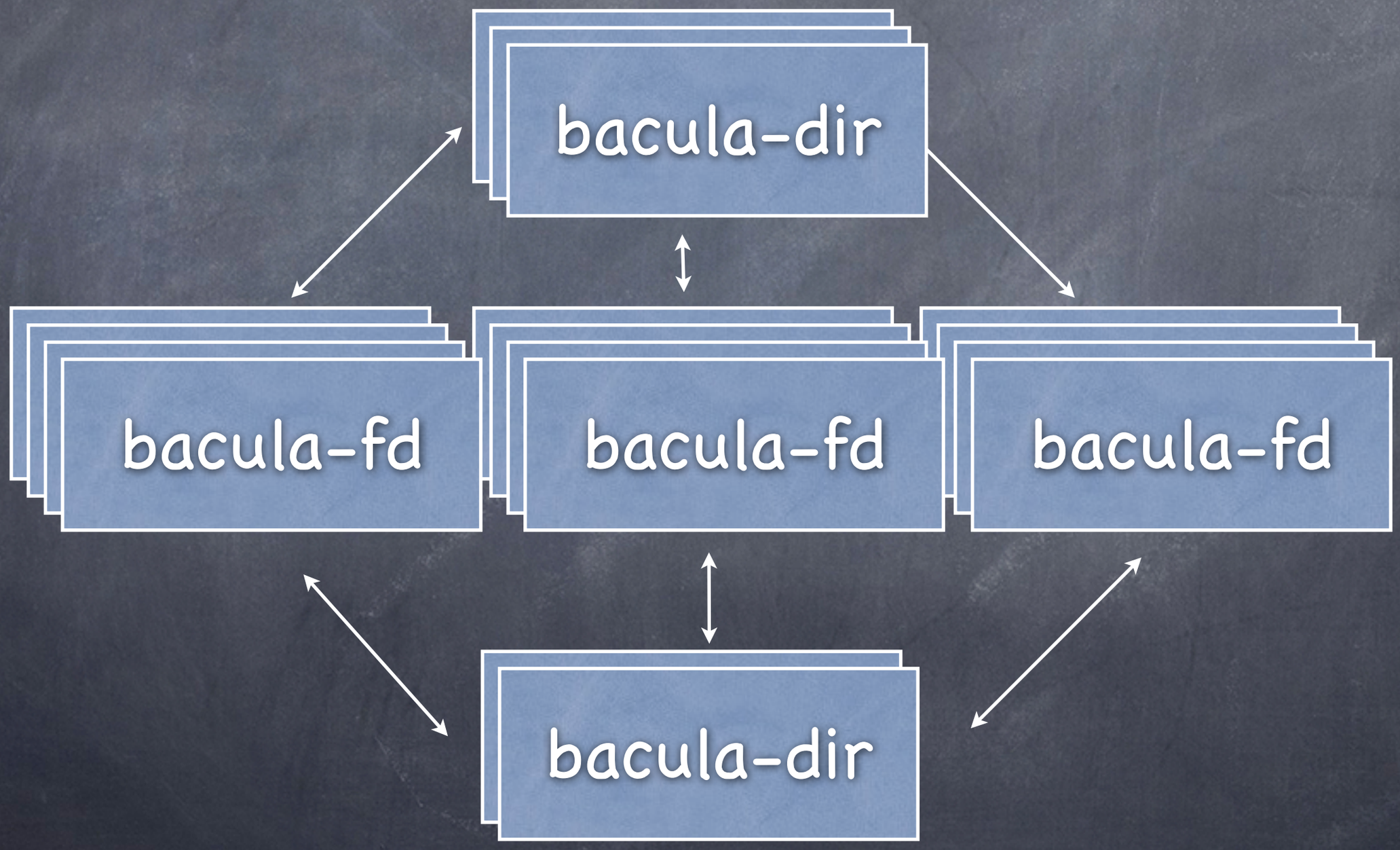
The usual starting point



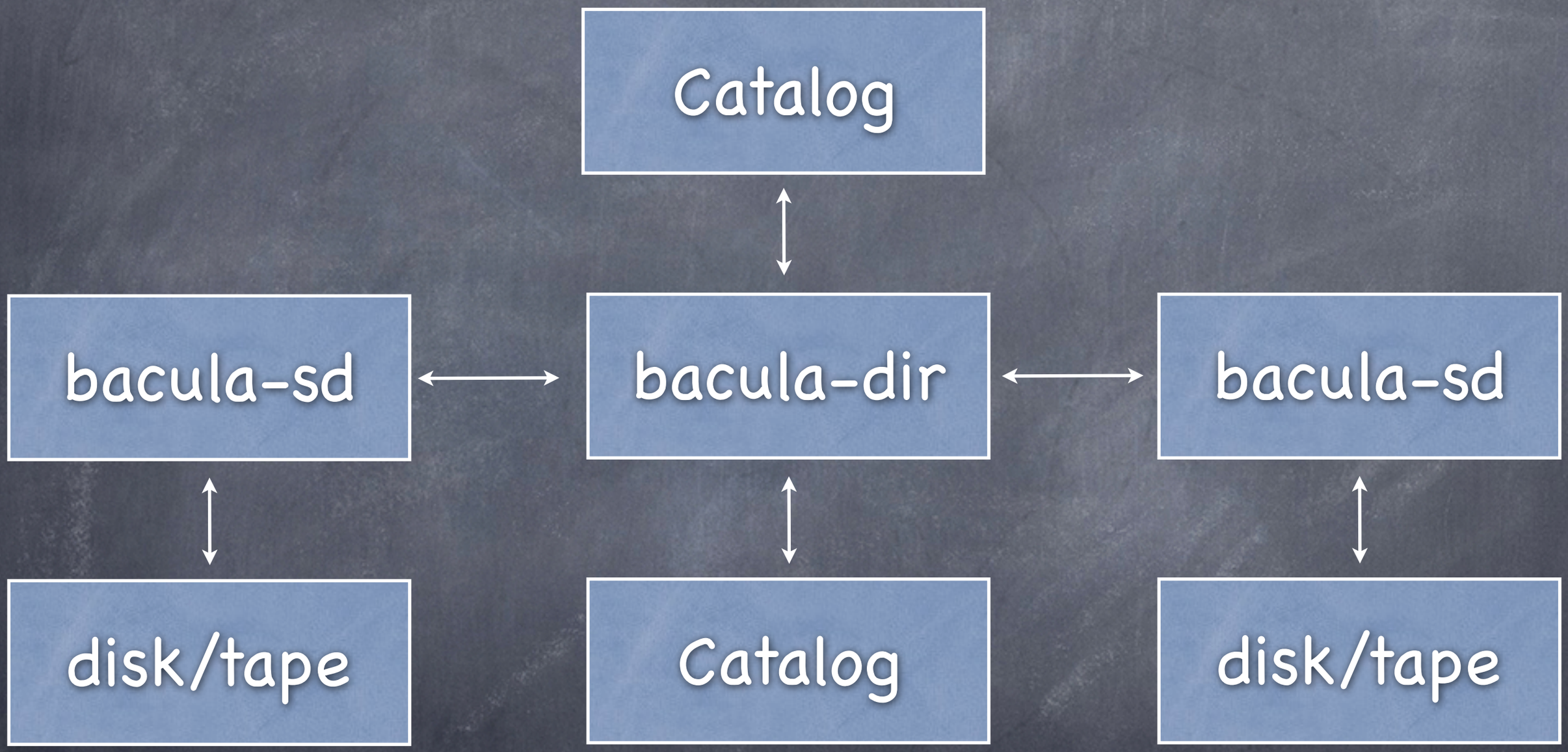
The usual starting point



Advanced



Advanced


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

HOT TIP!

- Use a VPN.
- Easier to secure.
- less NAT problems.
- bi-directional (you can ssh to the FD).

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

some restore options

- Select the most recent backup for a client
- Select backup for a client before a specified time
- Enter a list of files to restore
- Enter a list of files to restore before a specified time

some restore options (II)

- Find the JobIds of the most recent backup for a client
- Find the JobIds for a backup for a client before a specified time
- Enter a list of directories to restore for found JobIds
- Select full restore to a specified Job date

HOT TIP!

HOT TIP!

- echo `run job=dent yes' | bconsole

HOT TIP!

- echo `run job=dent yes' | bconsole

```
Connecting to Director
bacula.example.org:9101
```

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)
```

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.
```

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
```

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"
```

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 charater; 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.

bat

- GUI interface
- I have used it
- not often
- not recently
- other GUI: bweb, almir, and others

bsmtp(1)

- Bacula's SMTP client.
- used to submit email for notifications.
- I use it. You can use your own if it matches the parameters.
- You'll see it used in the Messages resource.

bwild(8)

- Bacula's 'wildcard' engine.
- Allows you to test wild card expressions against a set of data.
- Use when composing FileSet which contains wild cards.
- see also bregex(8)
- VERY USEFUL

Other commands

- bcopy(8) - copy parts of a Bacula Volume to another Volume.
- bextract(8) - extract a single file or a list of files from a Bacula Volume.
- bscan(8) - read (scan) a Bacula Volume and to recreate or update the database contents with the information found on the Volume.

Other commands #2

- btraceback(1) - exception handlers of the Bacula daemons during a crash.
- dbcheck(8) - search for logical inconsistencies in the Bacula tables in your database.

HOT TIP!

- If you migrate your Bacula server to a new machine, the clients do not care.
- New bacula-dir server, same name & password?
- No worries!

What daemons run as (in FreeBSD)

- 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

/etc/devfs.conf

```
# so bacula can access the DEC
# autochanger via the operator group
perm pass11    0660

# so bacula can access the overland
# autochanger via the operator group
perm pass14    0660
```

HOT TIP!

- You can run bacula-fd in read-only mode
- use the -k option

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"
}
```

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
}
```

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** according to the FileSet.
- Incremental - all files specified in the FileSet that have changed since the **last successful backup** of the same Job using the same FileSet and Client.
- Differential - all files specified in the FileSet that have changed since the **last successful Full backup** of the same Job and client will be backed up.

Job Level (II)

- From the Bacula FAQ:

  Before doing an **Incremental** or a **Differential** backup, Bacula checks to see if there was a **prior Full** backup of the same Job that terminated **successfully**. If so, it uses the date that full backup **started** as the time for comparing if files have **changed**. If Bacula does not find a successful full backup, it proceeds to do one.

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).

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"
}
```

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.

FileSet (II)

- By default, does not span mount points (can be overridden but use with care; e.g. NFS).
- Be sure to enclose path names in "quotes" for Windows clients.
- Test your FileSet using the estimate command in bconsole. You may be surprised.
- Use the list command to see what was backed up.

FileSet (III)

```
FileSet {
  Name = "basic backup"
  Include {
    Options {
    signature=MD5
    }
    Exclude Dir Containing = .NOBACKUP
    File = /usr/src/sys/i386/conf
    File = /etc
    File = /usr/local/etc
    File = /var/log
  }
}
```

FileSet (IV)

```
FileSet {
   Name = "ngaio files"
   Include {
     Options {
     signature =MD5
     verify    = pnugsmcs5
     }
     File = /home/dan
     File = /usr/local/libexec/nagios
     }
   Exclude {
     File = *~
     File = /home/dan/tmp
   }
}
```

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.

HOT TIP!

- 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.

Volumes (II)

- Each Volume has a distinct label.
- May be automatically or manually labeled.
- Automatic labelling makes great sense for disk Volumes
  - To implement: LabelMedia = yes in Device resource in bacula-sd.conf
  - For disk, you need LabelFormat in Pool resource

Volumes (III)

- Tape libraries often have barcodes; use them for automatic labeling.
- If you have barcodes, use the 'label barcodes' command.
- If you don't, you can still use it and fake it. See 'Simulating Barcodes in your Autochanger' in the documentation.

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 (III)

- The Pool definition is the template for a new Volume. It defines the attributes of each new Volume in that Pool.
- If you edit the Pool resource in bacula-dir.conf, the 'update' command in bconsole will update the Pool definition. This does not affect existing Volumes. The 'update' command can also update the Volumes from the Pool definition.

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...

With a Catalog, you can:

- Think about what you just read...
- What you can do with it...

With a Catalog, you can:

- Think about what you just read...
- What you can do with it...
- You can restore anything...

With a Catalog, you can:

- Think about what you just read...
- What you can do with it...
- You can restore anything...
- from anywhere...

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...

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...

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 means Catalog

- Retention determines how long entries are retained in the Catalog.

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.

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.

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/shrink

- Catalogs grow. Disk space is cheap. Use it.

Catalogs grow/shrink

- Catalogs grow. Disk space is cheap. Use it.
- Data is manually removed from the Catalog via the prune and purge commands:

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

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)

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?

Lost Catalog?

What to do?

What to do?

- What if you lose your Catalog?

What to do?

- What if you lose your Catalog?
- What? No backup?

What to do?

- What if you lose your Catalog?
- What? No backup?
- daily cron job to copy *.conf and *.sql

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.

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.

Catalog rules!

- Your Catalog is your best tool.
- Your Catalog is more important than your backups.

Catalog rules!

- Your Catalog is your best tool.
- Your Catalog is more important than your backups.
- Heavily used for restores.

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.

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.

Backup, backup, backup

- Backup your Catalog. To multiple locations.

Backup, backup, backup

- Backup your Catalog. To multiple locations.
- Catalogs can be reconstructed by using bscan(8).

Backup, backup, backup

- Backup your Catalog. To multiple locations.
- Catalogs can be reconstructed by using bscan(8).
- Reads all your Volumes.

Backup, backup, backup

- Backup your Catalog. To multiple locations.
- Catalogs can be reconstructed by using bscan(8).
- Reads all your Volumes.
- Lengthy and troublesome process.

Backup, backup, backup

- Backup your Catalog. To multiple locations.
- Catalogs can be reconstructed by using bscan(8).
- Reads all your Volumes.
- Lengthy and troublesome process.
- Avoid it at all costs.

Backing up the Catalog

- Job like any other

Backing up the Catalog

- Job like any other
- I recommend against backing up database files

Backing up the Catalog

- Job like any other
- I recommend against backing up database files
- I recommend backing up text dump of the database (e.g. pg_dump/mysqldump)

Backing up the Catalog

- Job like any other
- I recommend against backing up database files
- I recommend backing up text dump of the database (e.g. pg_dump/mysqldump)
- Run before script to dump the file[s].

Backing up the Catalog

- Job like any other
- I recommend against backing up database files
- I recommend backing up text dump of the database (e.g. pg_dump/mysqldump)
- Run before script to dump the file[s].
- Don't delete the file[s] after the backup.

Catalog backup job

```
Job {
  Name        = "BackupCatalog"
  JobDefs     = "DefaultJob"
  Client      = ngaio-fd
  Level       = Full
  FileSet     = "Catalog"
  Schedule    = "DailyAfterBackup"
  RunBeforeJob = "/a/b/catalog_dump MyCatalog"
  # or ClientRunBeforeJob
}
```

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)

Bacula Recycling Algorithm

- Automatic recycling of Volumes is controlled by four records in the Pool resource.
  - AutoPrune = yes
  - VolumeRetention = time
  - Recycle = yes
  - RecyclePool = UsuallyScratch

That's all you need!

- Those four directives are all you need.
- Assuming that you write to each Volume until it is full.
- Then wait the Volume Retention period before reusing them.

But wait! There's more!

- If you want Bacula to stop using a Volume before it is full, you will need to use one or more additional directives such as:
  - Use Volume Once = yes
  - Volume Use Duration = ttt
  - Maximum Volume Jobs = nnn
  - Maximum Volume Bytes = mmm

When to recycle?

- Automatic recycling of Volumes is performed by Bacula only when it wants a new Volume and no appendable Volumes are available in the Pool in question.
- Bacula will then search the Pool for any Volumes with the Recycle flag set and with Volume Status = Purged.
- At that point, Bacula will choose the oldest purged volume, and recycle it.

No Purged Volumes?

- Bacula will search the Catalog for Volumes which have been pruned of all Jobs.
- Only Volumes marked Full or Used will be searched.
- The Volume then gets purged if the Volume Retention period has expired.
- If something found, go back to the previous slide and continue.

It's not over til it's over

- Data on Volume remains intact until Recycling occurs.
- That is, the Volume may be pruned and/or purged, and the backup is still there, you can just can't easily get the data off with a standard restore command (see bextract if you need that data).

My main tape pools

```
Pool {
   Name             = Scratch
   Pool Type = Backup
}

Pool {
   Name                    = Fulls
   Pool Type               = Backup
   Recycle                 = yes
   AutoPrune               = yes
   Volume Retention = 3 years
   Storage                 = DigitalTapeLibrary
   RecyclePool             = Scratch
}
```

My main tape pools (II)

```
Pool {
  Name                = Differentials
  Pool Type           = Backup
  Recycle             = yes
  AutoPrune           = yes
  Volume Retention    = 2 months
  Storage             = DigitalTapeLibrary
  RecyclePool         = Scratch
}
```

My main tape pools (III)

```
Pool {
  Name                = Incrementals
  Pool Type           = Backup
  Recycle             = yes
  AutoPrune           = yes
  Volume Retention    = 4 weeks
  Storage             = DigitalTapeLibrary
  RecyclePool         = Scratch
}
```

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.

Volume Retention

- This is defined in the Pool resource

File Retention

- defined in Client resource
- relative to the Job End time
- default is 60 days

Job Retention

- defined in Client resource
- relative to Job End time
- default is 180 days

Pool Retention Directives

- Two new Pool directives, **File Retention** and **Job Retention**, that take precedence over Client directives of the same name.
- They allows you to control the Catalog pruning algorithm Pool by Pool. For example, you can decide to increase Retention times for Archive or OffSite Pool.

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

Defining clients

- look at bacula-fd.conf
- the corresponding entries in bacula-dir.conf

bacula-fd.conf

- A given client (bacula-fd) can be used by multiple bacula-dir
- That is: a client, say Client1, can take instructions from zero or more Directors

bacula-fd.conf

- Each director is listed separately:

```
Director {
  Name = Foo
  Password = "FooPassword"
}

Director {
  Name = bar
  Password = "BarPassword"
}
```

bacula-dir.conf

```
# from foo's bacula-dir.conf
Client {
  Name       = Client1
  Password = "FooPassword"
  Address   = client1.example.org
}
```

```
# from bar's bacula-dir.conf
Client {
  Name       = Client1
  Password = "BarPassword"
  Address   = client1.example.org
}
```

Databases

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

Database scripts

```
$ cd /usr/local/share/
$ grep -l psql *
create_postgresql_database
drop_postgresql_tables
grant_postgresql_privileges
make_catalog_backup
make_postgresql_tables
update_postgresql_tables
```

The parent scripts

```
$ cd /usr/local/share/
$ ls *bacula*
create_bacula_database
drop_bacula_tables
grant_bacula_privileges
make_bacula_tables
drop_bacula_database
update_bacula_tables
```

disk versus tape

- Some people love tape.

disk versus tape

- Some people love tape.
- Some people loathe tape.

disk versus tape

- Some people love tape.
- Some people loathe tape.
- Why have tape when you can have disk?

disk versus tape

- Some people love tape.
- Some people loathe tape.
- Why have tape when you can have disk?
- I love tape.

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.

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
*
```

The output

prune.
27-Jan 17:41 bacula-dir JobId 118611: End auto prune.
*

The output

Once again…

Termination: Backup OK
27-Jan 17:41 bacula-dir JobId 118611: Begin pruning Jobs older than 3 years.
27-Jan 17:41 bacula-dir JobId 118611: No Jobs found to prune.
27-Jan 17:41 bacula-dir JobId 118611: Begin pruning Files.
27-Jan 17:41 bacula-dir JobId 118611: No Files found to prune.
27-Jan 17:41 bacula-dir JobId 118611: End auto prune.

Once again...

*

Appending...

- 27-Jan 17:41 kraken-sd JobId 118611: Volume "Incr-2341" previously written, moving to end of data.
- 27-Jan 17:41 kraken-sd JobId 118611: Ready to append to end of Volume "IncrAuto-2341" size=2922249218

Spooling Attributes

- 27-Jan 17:41 kraken-sd JobId 118611: Elapsed time=00:00:01, Transfer rate=95.47 K Bytes/second
- 27-Jan 17:41 kraken-sd JobId 118611: Sending spooled attrs to the Director.
- Despooling 999 bytes ...

Times

- Scheduled time: 27-Jan-2013 17:41:32
- Start time: 27-Jan-2013 17:41:39
- End time: 27-Jan-2013 17:41:43
- Elapsed time: 4 secs
- Priority: 10

How much?

- FD Files Written: 4
- SD Files Written: 4
- FD Bytes Written: 95,017 (95.01 KB)
- SD Bytes Written: 95,470 (95.47 KB)
- Rate: 23.8 KB/s

Amount backed up

- Software Compression: None
- VSS: no
- Encryption: no
- Accurate: no

Amount backed up

- 27-Jan 17:41 bacula-dir JobId 118611: Begin pruning Jobs older than 3 years .
- 27-Jan 17:41 bacula-dir JobId 118611: No Jobs found to prune.
- 27-Jan 17:41 bacula-dir JobId 118611: Begin pruning Files.
- 27-Jan 17:41 bacula-dir JobId 118611: No Files found to prune.

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

Pruning

- Previous example during run
- Manual pruning is useful if you need to free up Volumes for recycling.

Manual pruning (II)

***prune volume=ETU135**
The current Volume retention period is: 3 years
Continue? (yes/mod/no): **yes**
*

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.

Limiting disk space used

- Bacula does not care about disk space used
- Bacula will use all your disk if allowed
- Disk space monitoring is out of scope
- Use Nagios etc instead

Set Limits

- Pool resource:
  - `Maximum Volumes = N`
  - `Maximum Volume Bytes = S`
- Maximum space used by Pool = (NxS) bytes
- Do this for every Pool

Other moderating things

- `UseVolumeOnce = yes`
- `Maximum Volume Jobs = nnn`
- `Volume Use Duration = ttt`

copy/migrate Job

- Duplicate a **Job** to another **Pool** (**copy**)
- Move a **Job** to another **Pool** (**migrate**)
- Can **only** be done within a **single** SD

copy/migrate difference

- **copy** does not affect the original job.
- Original job is still available for use.
- Documentation treats them the same.

What copy/migrate?

- Only Jobs in **Volumes** which are marked **Full**, **Used**, or **Error**.
- **Volumes** marked as **Append** will not be considered.

http://www.bacula.org/manuals/en/concepts/concepts/Migration_Copy.html

copy/migrate selection

```
Job {
  Name         = "CopyToTape-Inc"
  Type         = Copy
  Level        = Incremental
  Pool         = IncrFile
  FileSet      = "EmptyCopyToTape"
  Client       = kraken-fd
  Schedule     = "CopyToTape-Inc"
  Storage      = DigitalTapeLibrary
  Messages     = Standard

  Spool Data           = no
  Spool Attributes     = yes
  Selection Type       = SQL Query
...
```

copy/migrate selection

```
Selection Pattern = "
SELECT DISTINCT J.JobId, J.StartTime
  FROM Job J, Pool P
 WHERE P.Name    = 'IncrFile'
   AND P.PoolId = J.PoolId
   AND J.Type    = 'B'
   AND J.JobStatus IN ('T','W')
   AND J.jobBytes > 0
   AND J.JobId NOT IN
    (SELECT PriorJobId
       FROM Job
      WHERE Type IN ('B','C')
        AND Job.JobStatus IN ('T','W')
        AND PriorJobId != 0)
 ORDER BY J.StartTime
"}
```

Spooling

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

```
set Spool Data = yes
```

Spooling (II)

- Backing up to disk?
- Does spooling make sense there?

```
set Spool Data = no
```

Spooling (III)

- By default, database is updated for each file backed up.
- With small files, this can add up.
- Defer the database update to the end of job.

```
Spool Attributes = 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.

name/password error

- How to debug the name/password errors
- I really want to create a how-to for that
  - network
  - firewall
  - listening on right IP, right port

Deduplication

- This code will be released in the Bacula Enterprise version at the end of March 2013.
- For now, I recommend ZFS if you want dedup.

Base Jobs

- Base jobs good for things that rarely change
- e.g. 100 clients, all the same

Delta Backups

- Available in Bacula Enterprise

Compression

- FileSet option.
- Can be done at FD (gzip, lzo).
- Can be done by tape drive.
- Can be done by filesystem (e.g. ZFS).

Signature

- FileSet option.
- for each file.
- SHA1.
- MD5.

Don't waste your time!

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


And we're done!