

Interacting with

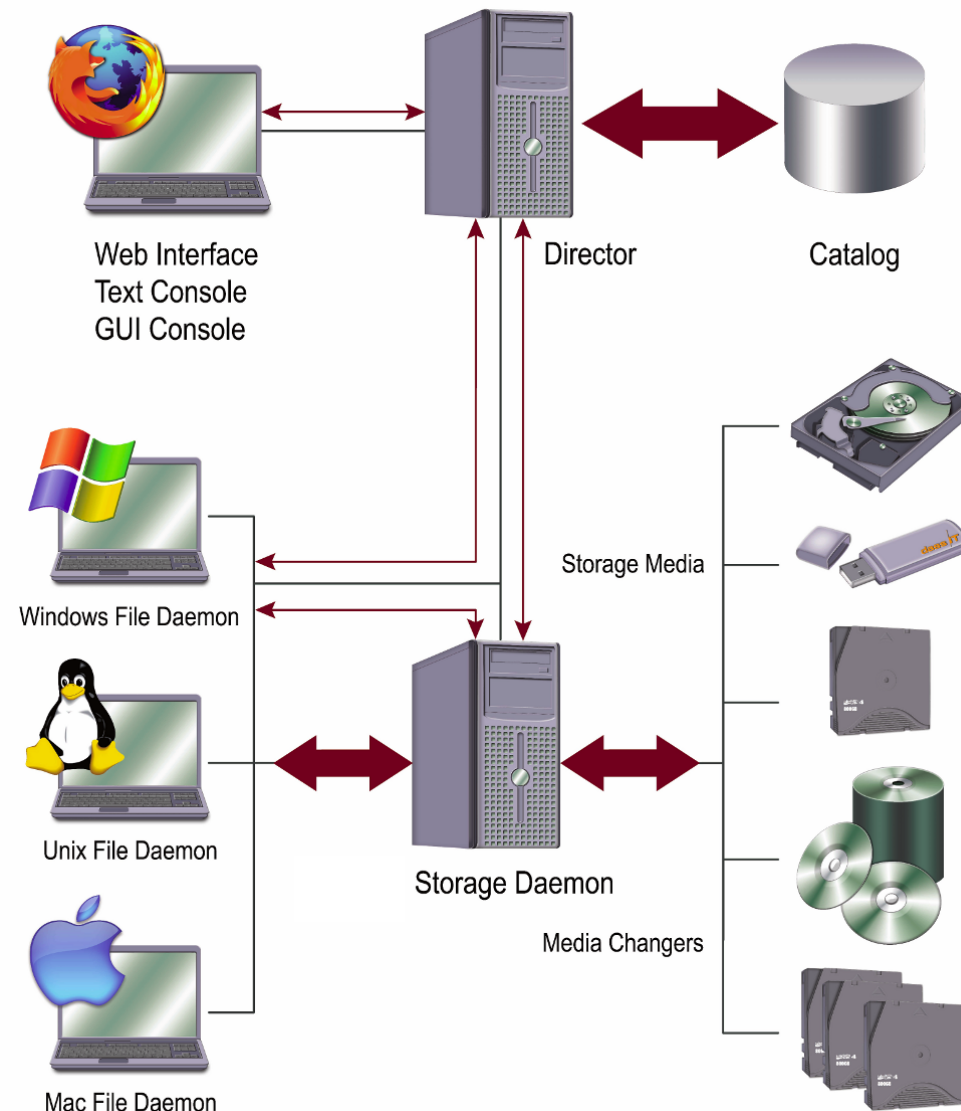


Jörg Steffens, Bareos GmbH & Co. KG

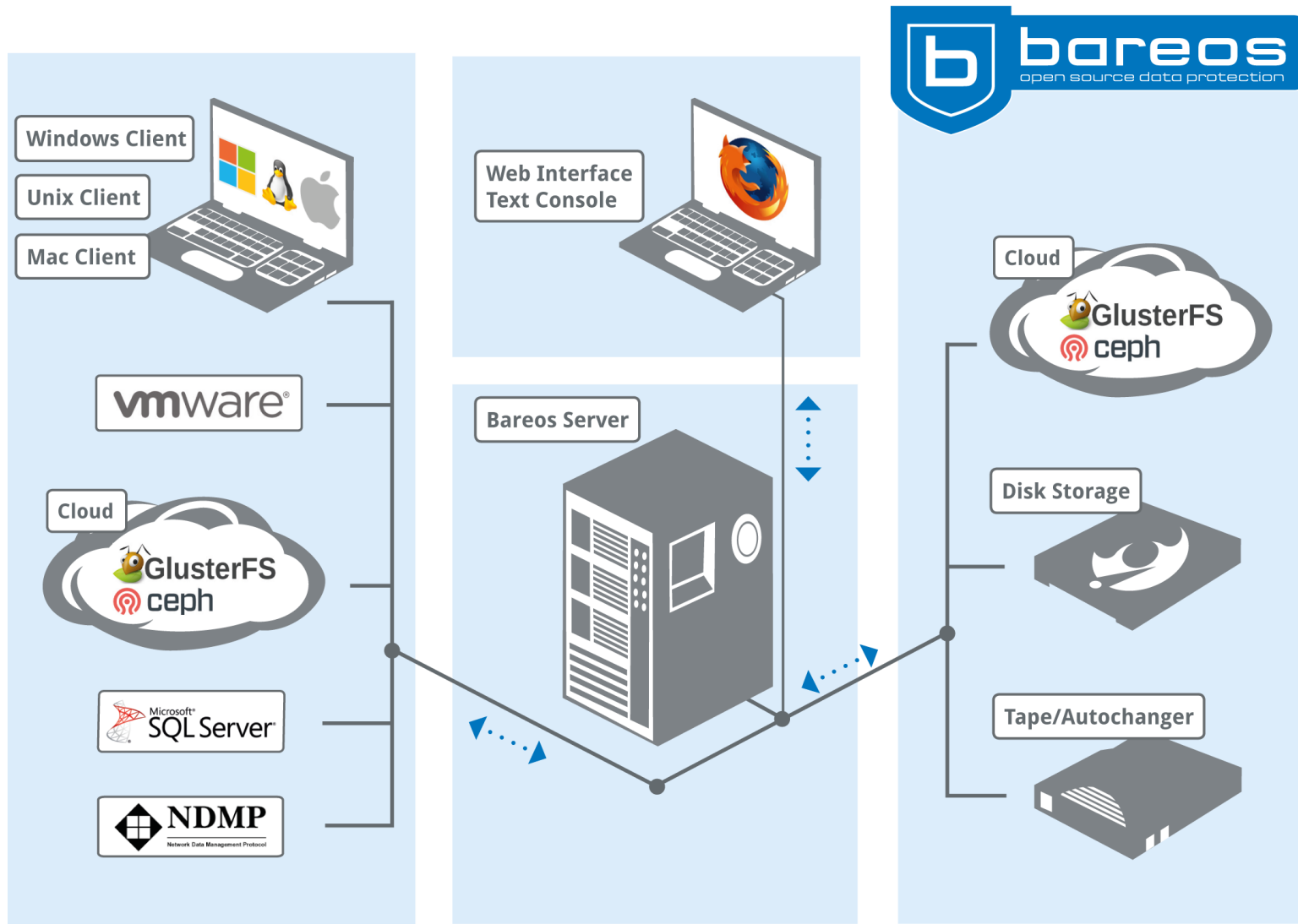
Agenda

- Bareos Overview
- Interaction Methods
 - Configuration Files
 - Run-Time Control of Bareos
 - Triggered by Bareos
 - Run Scripts
 - Plugins
- Roadmap

Bareos Architecture



Bareos Overview



Configuration Files

- Resources: director, catalog, client, fileset, job, pool, schedule, storage, ...
- Job Resource Example:

```
Job {  
  Name = "backup-bareos-fd"  
  Type = "Backup"  
  Level = "Incremental"  
  Client = "bareos-fd"  
  FileSet = "LinuxAll"  
  Schedule = "WeeklyCycle"  
  Storage = "File"  
  Pool = "Incremental"  
  Full Backup Pool = "Full"  
  Differential Backup Pool = "Differential"  
  Incremental Backup Pool = "Incremental"  
  Enable = "yes"  
}
```

- `bareos-dir -xs`: list all available directives
- Director: reload
- Storage-/File-Daemon: restart

Integration into Configuration Management Tools

- Ansible
- Chef
- Puppet
- Salt

Run-time Control of Bareos

```
linux# bconsole
Connecting to Director bareos:9101
1000 OK: bareos-dir Version: 16.2.4 (01 July 2016)
Enter a period to cancel a command.
*
```

- Interactive Console to a Bareos Director
- TCP connection to the Director
- `help` will list the available commands

bconsole: stdin and stdout

```
linux# echo "status client=bareos-fd" | bconsole
```

```
Connecting to Director bareos-dir:9101
```

```
1000 OK: bareos-dir Version: 16.2.4 (01 July 2016)
```

```
Enter a period to cancel a command.
```

```
status client=bareos-fd
```

```
Connecting to Client bareos-fd at bareos-fd:9102
```

```
bareos-fd Version: 16.2.4 (01 July 2016) x86_64-pc-linux-gnu ubuntu Ubuntu 1
```

```
Daemon started 16-Jan-17 16:12. Jobs: run=9 running=0.
```

```
Heap: heap=73,728 smbytes=57,497 max_bytes=2,427,435 bufs=325 max_bufs=722
```

```
Sizeof: boffset_t=8 size_t=8 debug=0 trace=0 bwlimit=0kB/s
```

```
Running Jobs:
```

```
bareos-mon (director) connected at: 16-Jan-17 16:30
```

```
bareos-dir (director) connected at: 20-Jan-17 16:51
```

```
No Jobs running.
```

```
=====
```

bconsole batch

run a backup and restore it

```
bconsole <<END_OF_DATA
@output /tmp/log1.out
run job=backup-client1 yes
wait
@#
@# now do a restore
@#
restore current all yes
wait
quit
END_OF_DATA
```

- @ commands:
 - @input, @output, @tee, @sleep, @time, ...

Interface Programs or Libraries

- bconsole
- python-bareos
- bareos-webui (PHP)
- bat (QT-GUI, deprecated)

The image displays two screenshots of Bareos software interfaces. The left screenshot shows the Bareos Director web UI, which includes a dashboard with a pie chart for job status, a table of recent jobs, and a log of current messages. The right screenshot shows the Bareos Administration Tool (BAT) desktop application, which features a sidebar for navigation and a main window displaying a detailed table of tape volumes.

Bareos Director Web UI Screenshot:

Jobs started during the past 24 hours

Running: 1, Waiting: 0, Terminated successfully: 0, Terminated unsuccessfully: 0

Most recent job status per job name

Job name	Status	Job ID	Client	Level	Start	End	Bytes
BackupCatalog	Successful	365	sheldon-fd	Full	2017-01-31 12:39:53	2017-01-31 12:39:53	3.19 MB
backup-amy	Successful	363	amy-fd	Incremental	2017-01-30 21:00:04	2017-01-30 21:00:04	29.4 KB
backup-sheldon	Successful	362	sheldon-fd	Incremental	2017-01-30 21:00:00	2017-01-30 21:00:03	87.13 KB
copy2tape-sheldon	Successful	360	sheldon-fd	Full	2017-01-29 23:00:06	2017-01-29 23:00:06	0.00 B
RestoreFiles	Successful	344	amy-fd	Full	2017-01-26 13:42:46	2017-01-26 13:42:46	136.84 KB

Showing 1 to 5 of 5 entries

Current Director messages

```
2017-01-31 12:39:53 sheldon-fd JobId 365: shell command: run After
2017-01-31 12:39:53 sheldon-fd JobId 365: End auto grow.
2017-01-31 12:39:53 sheldon-fd JobId 365: No files found to grow.
2017-01-31 12:39:53 sheldon-fd JobId 365: Begin growing files.
2017-01-31 12:39:53 sheldon-fd JobId 365: No files found to grow.
2017-01-31 12:39:53 sheldon-fd JobId 365: Begin printing data volume
2017-01-31 12:39:53 sheldon-fd JobId 365: Remove sheldon-fd 16.2.
2017-01-31 12:39:53 sheldon-fd JobId 365: BackupCatalog:2017-01-31_12_39_53_35
2017-01-31 12:39:53 sheldon-fd JobId 365: BackupCatalog:2017-01-31_12_39_53_35
2017-01-31 12:39:53 sheldon-fd JobId 365: BackupCatalog:2017-01-31_12_39_53_35
```

Bareos Administration Tool Screenshot:

Media Type: Status: Limit: 100 Name: Filter: Apply

Volume Name	Online	Vol Bytes	Vol Usage	Vol Status	Pool	Media Type	Last Written	When expires
vol001	0	0 B	0%	Cleaning	Scratch			
vol002	0	10.08 MB	100%	Full	DefWk	LT01	2009-10-27 17:48:43	2010-10-27 18:48:43
vol003	0	4.06 MB	100%	Full	DefWk	LT01	2009-10-27 17:48:32	2010-10-27 18:48:32
vol004	0	14.26 MB	100%	Full	DefWk	LT01	2009-10-27 17:48:32	2010-10-27 18:48:32
vol005	0	51.16 MB	100%	Full	DefWk	LT01	2009-10-27 17:48:43	2010-10-27 18:48:43
vol006	0	30.88 MB	100%	Full	DefWk	LT01	2009-10-27 17:48:34	2010-10-27 18:48:34
vol007	0	30.88 MB	100%	Full	DefWk	LT01	2009-10-27 17:48:36	2010-10-27 18:48:36
vol008	0	30.88 MB	100%	Full	DefWk	LT01	2009-10-27 17:48:42	2010-10-27 18:48:42
vol009	0	14.26 MB	100%	Full	DefWk	LT01	2009-10-27 17:48:43	2010-10-27 18:48:43
vol010	0	51.16 MB	100%	Full	DefWk	LT01	2009-10-27 17:48:44	2010-10-27 18:48:44
vol011	0	4.06 MB	100%	Full	DefWk	LT01	2009-10-27 17:48:43	2010-10-27 18:48:43
vol012	0	64.51 KB	100%	Full	Scratch	LT01		
vol013	0	64.51 KB	100%	Full	Scratch	LT01		
vol014	0	64.51 KB	100%	Full	Scratch	LT01		
vol015	0	64.51 KB	100%	Full	Scratch	LT01		
vol016	0	64.51 KB	100%	Full	Scratch	LT01		
vol017	0	64.51 KB	100%	Full	Scratch	LT01		
vol018	0	64.51 KB	100%	Full	Scratch	LT01		
vol019	0	64.51 KB	100%	Full	Scratch	LT01		
vol020	0	64.51 KB	100%	Full	Scratch	LT01		
vol021	0	64.51 KB	100%	Full	Scratch	LT01		
vol022	0	64.51 KB	100%	Full	Scratch	LT01		
vol023	0	64.51 KB	100%	Full	Scratch	LT01		
vol024	0	64.51 KB	100%	Full	Scratch	LT01		
vol025	0	64.51 KB	100%	Full	Scratch	LT01		
vol026	0	64.51 KB	100%	Full	Scratch	LT01		
vol027	0	64.51 KB	100%	Full	Scratch	LT01		
vol028	0	64.51 KB	100%	Full	Scratch	LT01		
vol029	0	64.51 KB	100%	Full	Scratch	LT01		

Console Types

- Default Console (Root Console)
 - Access to all Resources
- Named Console (Restricted Console)

```
Console {  
  Name = user1  
  Password = secret  
  Command ACL = !delete, *all*  
  Catalog ACL = MyCatalog  
  Client ACL = client1-fd, client2-fd  
  FileSet ACL = Linux.*  
  Job ACL = backup-client1, restore-client1, backup-cl  
  Plugin Options ACL = *all*  
  Pool ACL = *all*  
  Schedule ACL = *all*  
  Storage ACL = *all*  
  Where ACL = *all*  
}
```

Important Console commands

```
*help
  Command      Description
  =====
  add           Add media to a pool
  autodisplay   Autodisplay console messages
  automount     Automount after label
  cancel        Cancel a job
  configure     Configure director resources
  ...
*help list
  Command      Description
  =====
  list         List objects from catalog

Arguments:
  basefiles jobid=jobid |
  basefiles ujobid=complete_name |
  backups client=client [fileset=fileset]
```

Command: status

Run-Time information about components

```
*status director
*status storage=File
*status storage=File slots
*status client=client1-fd
```

```
*status scheduler days=360
...
Fr 03-Feb-2017 21:00 WeeklyCycle Level=Incremental
Fr 03-Feb-2017 21:10 WeeklyCycleAfterBackup Level=Full
Sa 04-Feb-2017 21:00 WeeklyCycle Level=Full
Mo 06-Feb-2017 21:00 WeeklyCycle Level=Incremental
Mo 06-Feb-2017 21:10 WeeklyCycleAfterBackup Level=Full
...
```

Command: list

- Lists database (Catalog) entries
- Two forms:
 - `list` and more verbose `llist`
- Options:
 - backups, clients, copies, files, filesets, jobs, joblog, jobmedia, pools, storages, volumes
- Examples:
 - `list jobs volume=Full-0001`
 - `list jobmedia jobid=123`
 - `list joblog jobid=123`
 - `list jobs last`
- `show`: show current configuration

Run jobs

- run
- restore
- wait jobid=jobid

Adding a Client

```
*configure add client name=client2-fd address=192.168.0.2 password=secret
Created resource config file "/etc/bareos/bareos-dir.d/client/client2-fd.conf"

*status client=client2-fd
Connecting to Client client2-fd at 192.168.0.2:9102
...

*configure add job name=client2-job client=client2-fd jobdefs=DefaultJob
Created resource config file "/etc/bareos/bareos-dir.d/job/client2-job.conf"

*run job=client2-job
Job queued. JobId=256

*wait jobid=256
JobId=256
JobStatus=OK (T)

*list joblog jobid=256
```

Enable debugging during run-time

```
*setdebug level=100 trace=1 timestamp=1 director  
level=100 trace=1 hangup=0 timestamp=1 tracefilename=/var/lib/bareos/bareos-d
```

- storage=STORAGE
- client=CLIENT

Console Commands

- Normal Commands
 - help
 - status
 - list
 - run
 - ...
- dot (.) Commands
 - Special commands for non-interactive use.
 - Therefore not shown by `help`, but by `.help`

.sql

- allow to execute arbitrary sql commands
- only if it is really required
- `.sql query="select * from job;"`
- or `sqlquery` for interactive use
- or `query` for customized predefined queries

API modes

- .api 0
 - normal output
 - human readable, difficult to parse
- .api 1
 - modified output
 - used by BAT
 - inconsistent, difficult to parse
- .api json
 - JSON format
 - introduced for bareos-webui
 - bareos-webui: no direct database access

API modes example

```
*.api 0
*list jobid=1
+-----+-----+-----+-----+-----+-----+
| JobId | Name                | Client   | StartTime                | Type | Level |
+-----+-----+-----+-----+-----+-----+
| 1      | backup-bareos-fd    | bareos-fd | 2017-01-16 16:16:18    | B    | F      |
+-----+-----+-----+-----+-----+-----+

*.api 1
*list jobid=1
  1      | backup-bareos-fd    | bareos-fd | 2017-01-16 16:16:18    | B    | F      |

*.api json
*list jobid=1
{
  "jsonrpc": "2.0",
  "result": {
    "jobs": [
```

python-bareos

- Python module to connect to a Bareos Director (Console)
- optional: direct connection to Bareos Storage- or File-Daemon
- optional: support for JSON API mode

python-bareos json

```
import bareos.bsock

password=bareos.bsock.Password('secret')
directorconsole=bareos.bsock.DirectorConsoleJson(address='bareos-dir.example.',
                                                    name='user1',
                                                    password=password)

result=directorconsole.call('list jobid=1')
print(result)
{'jobs': [{u'name': u'backup-bareos-fd', u'level': u'F', u'jobbytes': u'2840
```


Retrieving File Information

List of files and directories

Before running a backup job

```
*estimate listing job=backup-bareos-fd
Using Catalog "MyCatalog"
Connecting to Client bareos-fd at bareos.example.com:9102
-rwxr-xr-x  1 root    root          3358 2016-10-17 00:13:20 /usr/sbin/b
-rwxr-xr-x  1 root    root        31544 2016-09-28 23:17:24 /usr/sbin/s
-rwxr-xr-x  1 root    root        57032 2016-09-28 22:53:19 /usr/sbin/v
-rwxr-xr-x  1 root    root         1395 2016-09-28 22:51:11 /usr/sbin/a
...
```

Stored by a backup job

```
*list files jobid=1
/usr/sbin/
/usr/sbin/a2disconf
/usr/sbin/a2dismod
...
```

Retrieving File Information

restore (interactive)

```
*restore client=bareos-fd fileset=SelfTest select current
Building directory tree for JobId(s) 10 ... ++++++
cwd is: /
$ dir
----- 0 root      root              0  1970-01-01 01:00:00  /usr/
$ cd /usr/sbin
cwd is: /usr/sbin/
$ dir
lrwxrwxrwx    1 root      root              7  2016-09-28 23:14:12  /usr/sbi
lrwxrwxrwx    1 root      root              7  2016-09-28 23:14:12  /usr/sbi
lrwxrwxrwx    1 root      root              7  2016-09-28 23:14:12  /usr/sbi
lrwxrwxrwx    1 root      root              7  2016-09-28 23:14:12  /usr/sbi
-rwxr-xr-x    1 root      root            15424  2016-09-28 23:14:12  /usr/sbi
...
$ mark a2enmod
1 file marked.
$ done
```

Retrieving File Details

bvfs api

<http://doc.bareos.org/master/html/bareos-developer-guide.html#sec:bvfs>

```
*@# update cache
*.bvfs_update jobid=10
*@# get root directory of job
*.bvfs_lsdirs jobid=10 path=
4      0      0      0      A A A A A A A A A A A A A A A A .
3      0      0      0      A A A A A A A A A A A A A A A A /
*@# get directories of path /
*.bvfs_lsdirs jobid=10 pathid=3
3      0      0      0      A A A A A A A A A A A A A A A A .
4      0      0      0      A A A A A A A A A A A A A A A A ..
2      0      0      0      A A A A A A A A A A A A A A A A usr/
*@# get directories of path /usr/
*.bvfs_lsdirs jobid=10 pathid=2
2      0      0      0      A A A A A A A A A A A A A A A A .
3      0      0      0      A A A A A A A A A A A A A A A A ..
1      0      366    10      gD IBAH EHt C A A A FAA BAA o BWA5Fq BYfI8X
*@# get directories of path /usr/sbin/
*.bvfs_lsdirs jobid=10 pathid=1
```

bvfs api json output

```
{
  "jobid": 10,
  "pathid": 1,
  "fileid": 365,
  "filenameid": 309,
  "name": "update-passwd",
  "type": "F",
  "lstat": "gD IC/n Iht B A A A Hmg BAA BA BX7C7h BWiFmV BX7C7i A A U",
  "stat": {
    "dev": 2051,
    "group": "root",
    "ctime": 1475096290,
    "rdev": 0,
    "ino": 2109415,
    "mode": 33261,
    "user": "root",
    "nlink": 1,
    "size": 31136
  }
}
```

bareos-fuse

- <https://www.bareos.org/en/bareos-fuse.html>
- uses python-bareos (JSON)
- prototype but useful
- access to jobs, job logs, volumes, pools, meta data of backed up files

```
$ cd /srv/bareosfs/
$ ls -la
drwxr-xr-x 81 root root 4096 Jan  1 1970 clients
drwxr-xr-x  2 root root 4096 Jan  1 1970 jobs
drwxr-xr-x  2 root root 4096 Jan  1 1970 pools
drwxr-xr-x  2 root root 4096 Jan  1 1970 volumes
$ ls -la jobs/running/
drwxr-xr-x  5 root root 4096 Jan 23 00:00 jobid=186058_name=gonzo_client=gonzo
```

bareos-fuse

jobs of client ting-fd

```
$ ls -la clients/ting-fd/jobs
...
drwxr-xr-x 5 root root 4096 Jan 17 12:43 jobid=185743_name=ting_client=ting-f
drwxr-xr-x 5 root root 4096 Jan 18 11:40 jobid=185785_name=ting_client=ting-f
drwxr-xr-x 5 root root 4096 Jan 19 11:38 jobid=185826_name=ting_client=ting-f
drwxr-xr-x 5 root root 4096 Jan 20 13:47 jobid=185868_name=ting_client=ting-f

$ ls -la clients/ting-fd/jobs/jobid=185868_name=ting_client=ting-fd_level=F_s
drwxr-xr-x 12 root root 0 Jan 1 1970 data
-r--r--r-- 1 root root 679 Jan 1 1970 info.txt
-r--r--r-- 1 root root 19262 Jan 1 1970 job.log

$ ls -la clients/ting-fd/jobs/jobid=185868_name=ting_client=ting-fd_level=F_s
total 72
drwxr-xr-x 2 root root 12288 Dec 16 07:15 bin
drwxr-xr-x 3 root root 12288 Jan 16 10:15 boot
drwxr-xr-x 199 root root 16384 Jan 20 11:22 etc
drwxr-xr-x 5 root root 4096 Oct 27 20:42 home
```

bareos-fuse

volumes and their jobs

```
$ ls -la pools/AI-Full/volumes/
drwxr-xr-x 5 root root      4096 Jan  1  1970 AI-Full-11160
-r--r----- 1 root root 53687079127 Jan 17 13:09 AI-Full-11160=Full
drwxr-xr-x 5 root root      4096 Jan  1  1970 AI-Full-11369
-rw-rw---- 1 root root 23146759602 Jan 20 14:05 AI-Full-11369=Append
...

$ ls -la pools/AI-Full/volumes/AI-Full-11369
-r--r--r-- 1 root root      931 Jan  1  1970 info.txt
drwxr-xr-x 5 root root      4096 Jan  1  1970 jobs
-rw-rw---- 1 root root 23146759602 Jan 20 14:05 status=Append

$ ls -la pools/AI-Full/volumes/AI-Full-11369/jobs/
drwxr-xr-x 5 root root 4096 Jan 20 14:04 jobid=185876_name=obs_client=obs-fd_
drwxr-xr-x 5 root root 4096 Jan 20 14:05 jobid=185877_name=mail.bareos.com_cl
drwxr-xr-x 5 root root 4096 Jan 20 14:06 jobid=185878_name=jenkins_client=dir
```

When Client is present, trigger backup

- common problem: backup of laptop
- not always preset in the network
- (changing IP addresses)

Get connected clients, trigger backup if no backup is younger than 24h

- Uses "Client Initiated Connection" to get list of available clients.
- Bareos Director does not require to know the client IP address.
- uses python-bareos (JSON)

```
def get_connected_clients(director):
    result=director.call('status director')['client-connection']
    clients = [ client['name'] for client in result ]
    return clients

def trigger(director, jobnames):
    for client in get_connected_clients(director):
        jobname = 'backup-{}'.format(client)
        if jobname in jobnames:
            jobs = director.call('list jobs client={} hours=24'.format(client))
            if not jobs:
                jobid = director.call('run {} yes'.format(jobname))['run']['j']
                print('{}: backup triggered, jobid={}'.format(jobname, jobid))
```

Prune and Truncate all Volumes

- Bareos recycles volumes automatically.
 - However, it does so only when space is required for a new backup job.
 - By this, data is kept as long as possible.
- script
 - prune: checks if retention time is expired. If yes, volume will be purged.
 - prune can only be used for single volumes, not all volumes in a pool.
 - truncate: frees space used by (disk) volumes.

```
def get_volumes(director):  
    result = director.call('list volumes all')['volumes']  
    volumes = [ volume['volumename'] for volume in result ]  
    return volumes  
  
def prune_and_truncate_volumes(director):  
    for volume in get_volumes(director):  
        director.call('prune volume={} yes'.format(volume))  
    director.call('truncate volstatus=Purged yes')
```

Triggering by Bareos

Run Scripts in Backup Jobs

- Console
 - Bareos console command to execute
 - Options
 - Runs On Success: yes | no
 - Runs On Failure: yes | no
 - Runs When
 - Never|Before|After|Always|AfterVSS
- Command
 - system command to execute
 - Additional Options:
 - Runs On Client: yes | no
 - Fail Job On Error: yes | no

Set Job To Archive

- Prevents that a generated VirtualFull is used as a Full on normal operation.

```
Job {  
  Name = "VirtualLongtermFull"  
  Schedule = ArchiveJob  
  Type = Backup  
  Client = bareos-fd  
  FileSet = LinuxAll  
  Level = VirtualFull  
  Pool = ArchivePool  
  Run Script {  
    Runs When = After  
    Console = "update jobid=%i jobtype=A"  
    Runs On Client = No  
    Runs On Failure = No  
  }  
}
```

Admin Job

```
Job {  
  Name = FileTableMaintJob  
  JobDefs = DefaultJob  
  Schedule = "WeeklyCycleAfterBackup"  
  Type = Admin  
  Priority = 200  
  
  Run Script {  
    Runs When = Before  
    Runs On Client = no  
    Fail Job On Error = yes  
    Command = "sudo -u postgres /usr/local/lib/bareos/scripts/postgresql_file  
  }  
}
```

Relax and Recover integration

- create Rear Boot Image on Full backup

```
Job {  
  ...  
  Run Script {  
    Runs When = Before  
    Runs On Client = yes  
    Command = "/usr/local/bin/rear-mkrescue-on-Full.sh %l"  
  }  
}
```

```
#!/bin/bash  
if [ "$1" != "Full" ]; then  
  echo "SKIPPED: creating rear boot image only when level=Full (not level=$1)"  
  exit 0  
fi  
rear -v mkrescue  
exit $?
```

```
Job {  
  Name = "restore-rear-media"  
  Type = Restore  
  ...  
  # restore boot media to /tmp  
  Regex Where = !/var/lib/rear/output/!/tmp/!  
}
```

```
*restore current file=/var/lib/rear/output/rear-bareos.iso restorejob=restore
```

Plugins

- Director, Storage Daemon and File Daemon plugins
- register to events
- File Daemon plugins:
 - backup specific data (database, virtual machine, ...)
- Storage Daemon
 - rewrite data (e.g. autoxflate-sd, scsicrypto-sd)
 - status information
- Director Daemon
 - status information
 - Icinga, Graphite, ...

Director Plugin in Python for Status

```
import bareosdir

def __init__(self, context, plugindef):
    events = [ bDirEventType['bDirEventJobEnd'] ]
    bareosdir.RegisterEvents(context, events)

def handle_plugin_event(self, context, event):
    if event == bDirEventType['bDirEventJobEnd']:
        data = getFormattedOutput(self.jobClient, self.jobErrors, self.jobBytes, s
        self.send_to_graphite(data)
```

Roadmap for Bareos-17.2

- integrate python-bareos into bareos core (currently bareos-contrib)
- bareos-webui enhancements
- optimize volume handling for disk storages