Interacting with

bareos

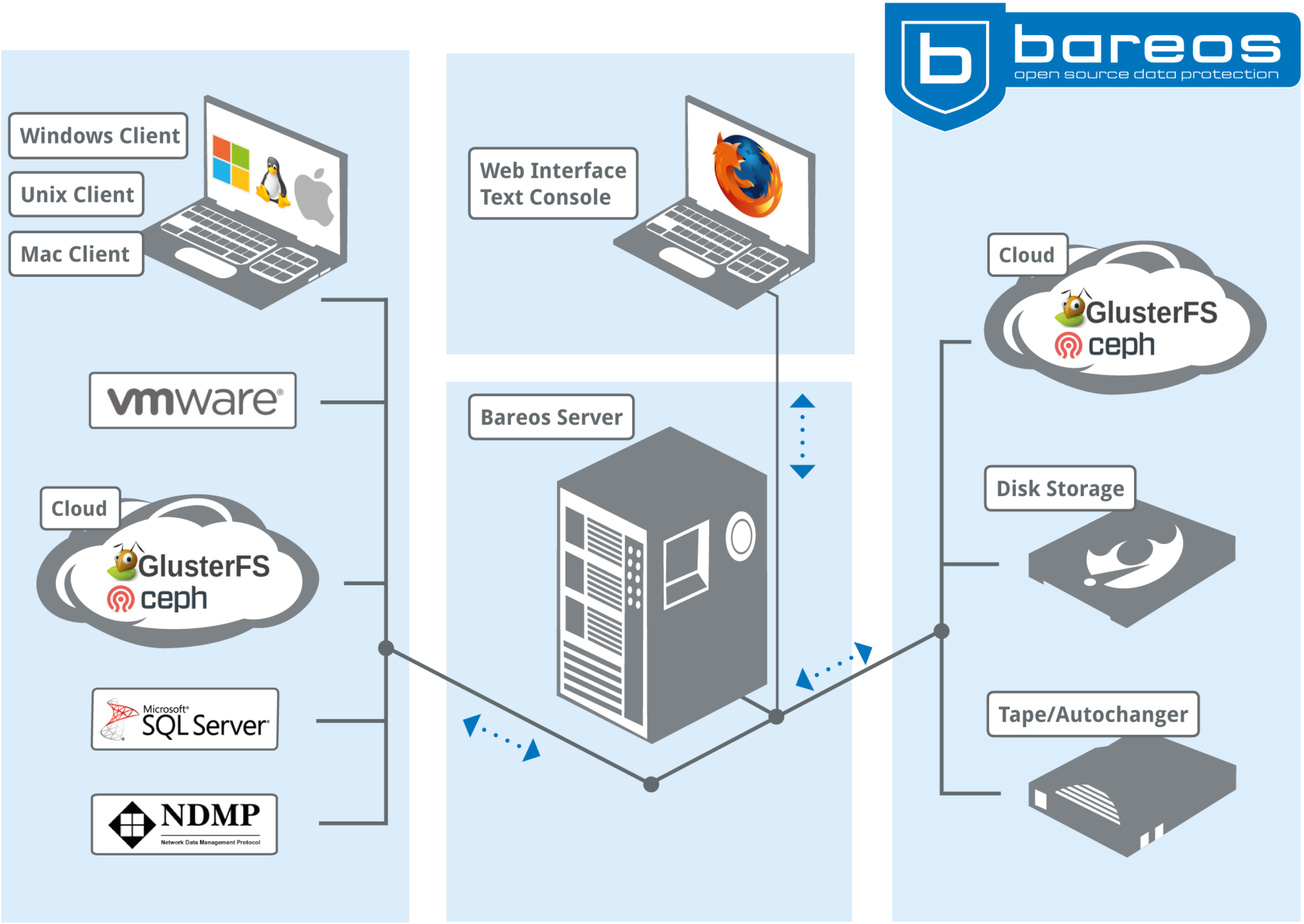
open source data protection

Jörg Steffens, Maik Außendorf, Bareos GmbH & Co. KG

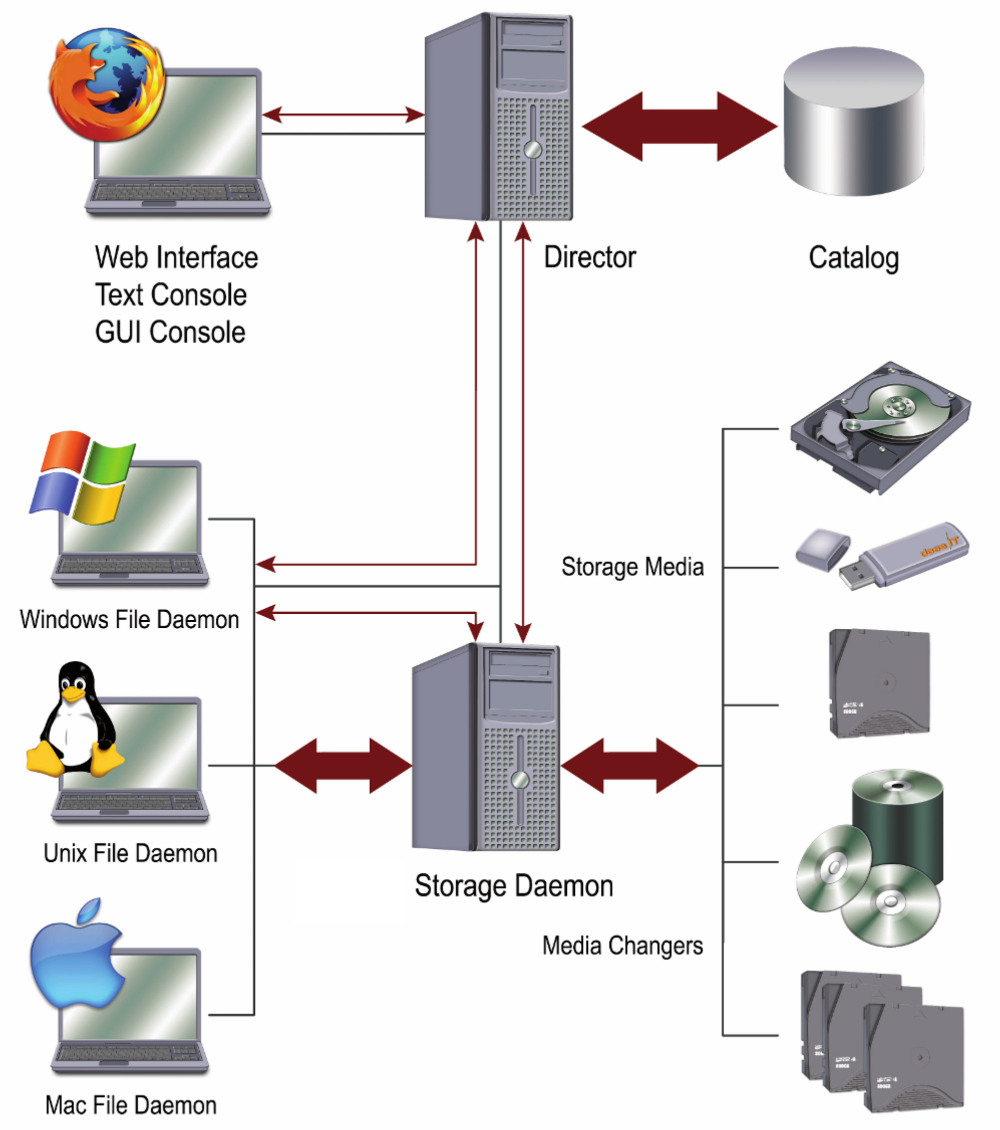
Agenda

- Bareos Overview
- Interaction Methods Overview
- Get Demo / Development System ready
- Hands on with Focus on Scripting using Director Interface

Bareos Overview



Bareos Architecture



Interacting Methods Overview

Configuration Files

Integration into Configuration Management Tools

- Include-Friendly directory hierarchy for configuration files since 16.2
- Ansible
- Chef
- Puppet
- Salt

Configuration file hierarchy *tree /etc/bareos*

```
├── bareos-dir.d
│   ├── catalog
│   │   └── MyCatalog.conf
│   ├── client
│   │   └── bareos-fd.conf
│   ├── console
│   │   └── bareos-mon.conf
│   ├── counter
│   ├── director
│   │   └── bareos-dir.conf
│   ├── fileset
│   │   ├── Catalog.conf
│   │   ├── LinuxAll.conf
│   │   ├── SelfTest.conf
│   │   └── Windows\ All\ Drives.conf
│   ├── job
│   │   ├── archivejob.conf
│   │   ├── backup-bareos-fd.conf
```

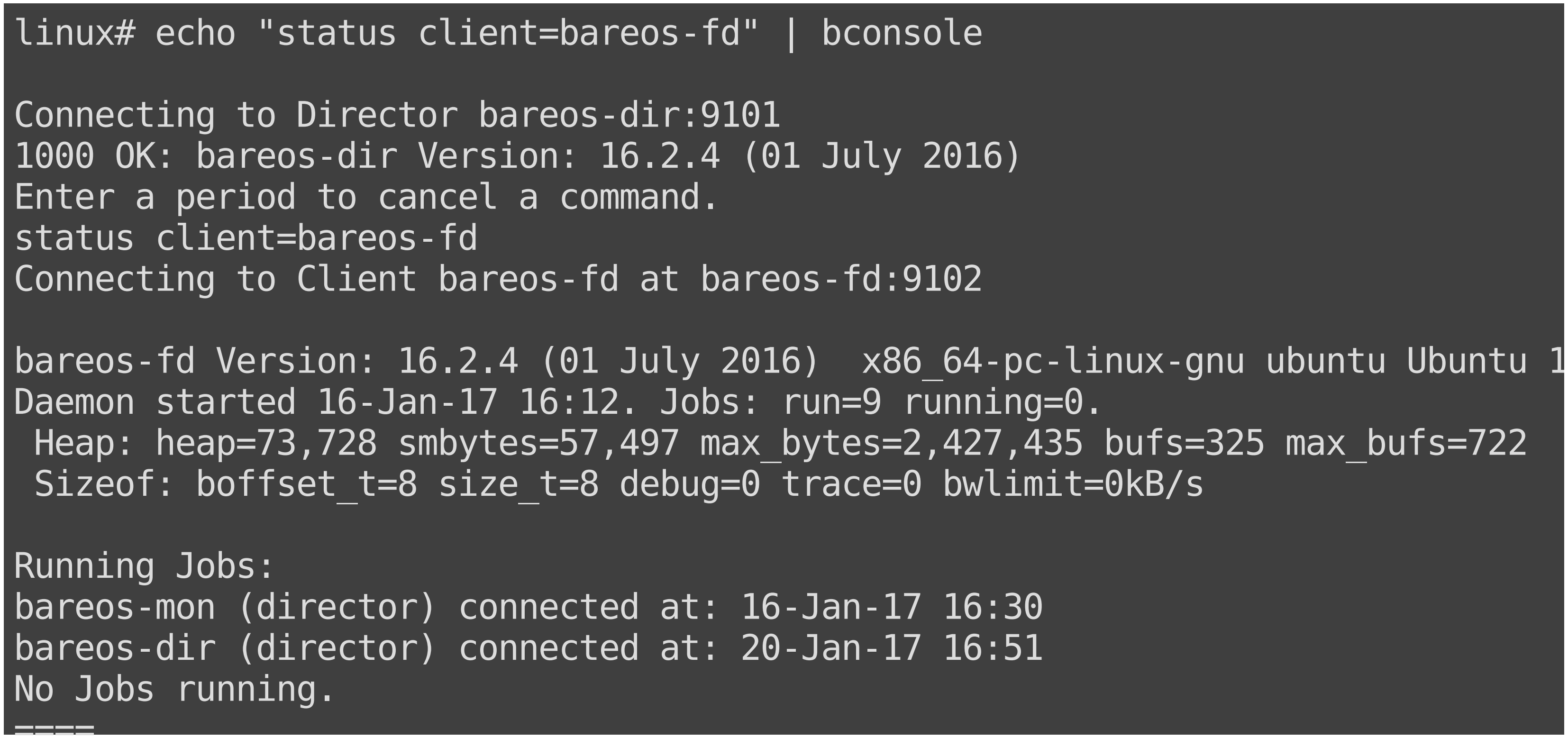
Run-time Control of Bareos

```
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

  Excercise: get bconsole running

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, ...
- Alternative: use *echo -e* and *\n* for newline:

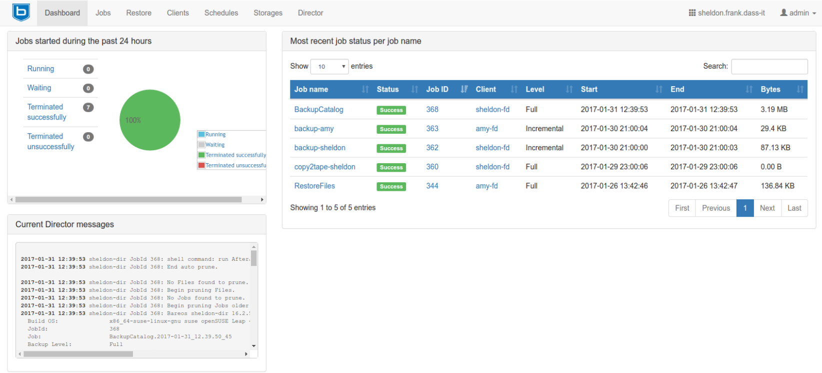
```
echo -e "run job=backup-client1 yes\nwait\n" | bconsole
```

Excercise

- Use bconsole and pipe or redirection to run jobs without any interaction
  - Run an incremental backup job of your client
  - Run a restore of all files or only one particular file
- Use *help restore* to see available options.
- Consider *client*= *fileset*= *select current all*

Interface Programs or Libraries

- bconsole
- python-bareos
- bareos-webui (PHP)
- bat (QT-GUI, dropped in 17.2)



Jobs started during the past 24 hours

Running	0
Waiting	0
Terminated successfully	7
Terminated unsuccessfully	0

Most recent job status per job name

Show 10 entries Search:

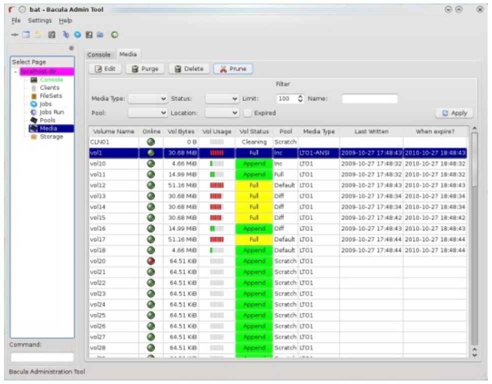
Job name	Status	Job ID	Client	Level	Start	End	Bytes
BackupCatalog	Success	368	sheldon-fd	Full	2017-01-31 12:39:53	2017-01-31 12:39:53	3.19 MB
backup-amy	Success	363	amy-fd	Incremental	2017-01-30 21:00:04	2017-01-30 21:00:04	29.4 KB
backup-sheldon	Success	362	sheldon-fd	Incremental	2017-01-30 21:00:00	2017-01-30 21:00:03	87.13 KB
copy2tape-sheldon	Success	360	sheldon-fd	Full	2017-01-29 23:00:06	2017-01-29 23:00:06	0.00 B
RestoreFiles	Success	344	amy-fd	Full	2017-01-26 13:42:46	2017-01-26 13:42:47	136.84 KB

Showing 1 to 5 of 5 entries

First Previous 1 Next Last

Current Director messages

```
2017-01-31 12:39:53 sheldon-dir JobId 368: shell command: run After
2017-01-31 12:39:53 sheldon-dir JobId 368: End auto prune.

2017-01-31 12:39:53 sheldon-dir JobId 368: No Files found to prune.
2017-01-31 12:39:53 sheldon-dir JobId 368: Begin pruning Files.
2017-01-31 12:39:53 sheldon-dir JobId 368: No Jobs found to prune.
2017-01-31 12:39:53 sheldon-dir JobId 368: Begin pruning Jobs older
2017-01-31 12:39:53 sheldon-dir JobId 368: Bareos sheldon-dir 16.2.
  Build OS:               x86_64-suse-linux-gnu suse openSUSE Leap
  JobId:                  368
  Job:                    BackupCatalog.2017-01-31_12.39.50_45
  Backup Level:           Full
```



Filter

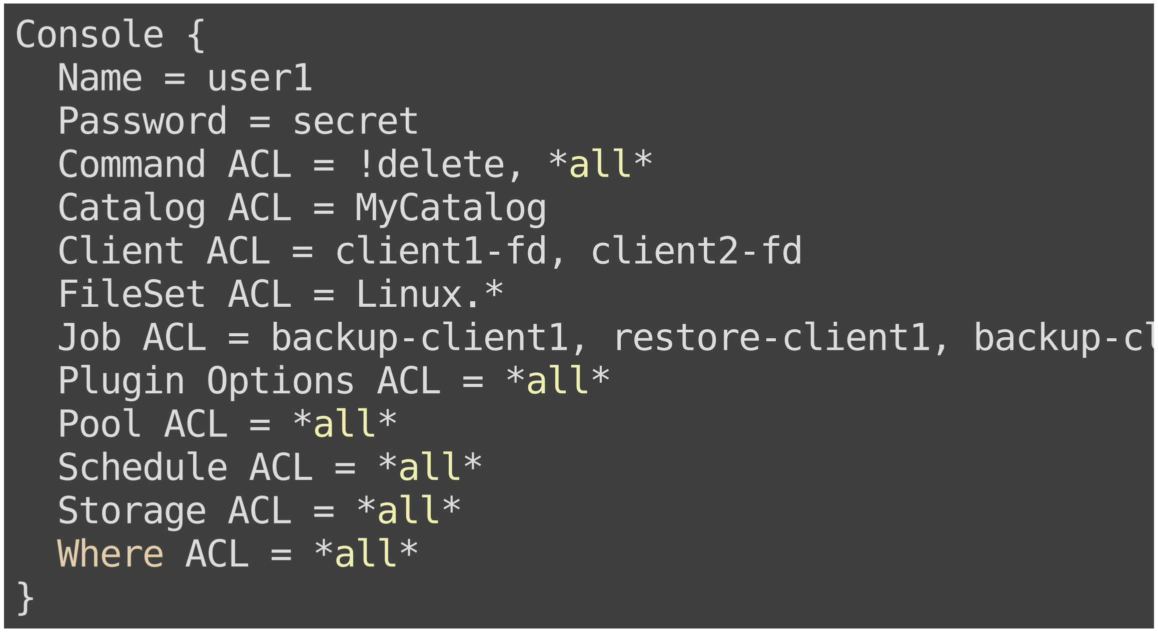
Media Type: Status: Limit: 100 Name:

Pool: Location: □ Expired

Volume Name	Online	Vol Bytes	Vol Usage	Vol Status	Pool	Media Type	Last Written	When expire?
CLN01	●	0 B		Cleaning	Scratch			
vol1	●	30.68 MiB		Full	Inc	LTO1-ANSI	2009-10-27 17:48:43	2010-10-27 18:48:43
vol10	●	4.66 MiB		Append	Inc	LTO1	2009-10-27 17:48:32	2010-10-27 18:48:32
vol11	●	14.99 MiB		Append	Full	LTO1	2009-10-27 17:48:32	2010-10-27 18:48:32
vol12	●	51.16 MiB		Full	Default	LTO1	2009-10-27 17:48:43	2010-10-27 18:48:43
vol13	●	30.68 MiB		Full	Diff	LTO1	2009-10-27 17:48:34	2010-10-27 18:48:34
vol14	●	30.68 MiB		Full	Diff	LTO1	2009-10-27 17:48:34	2010-10-27 18:48:34
vol15	●	30.68 MiB		Full	Diff	LTO1	2009-10-27 17:48:42	2010-10-27 18:48:42
vol16	●	14.99 MiB		Append	Diff	LTO1	2009-10-27 17:48:43	2010-10-27 18:48:43
vol17	●	51.16 MiB		Full	Default	LTO1	2009-10-27 17:48:44	2010-10-27 18:48:44
vol18	●	4.66 MiB		Append	Default	LTO1	2009-10-27 17:48:44	2010-10-27 18:48:44
vol20	●	64.51 KiB		Append	Scratch	LTO1		
vol21	●	64.51 KiB		Append	Scratch	LTO1		
vol22	●	64.51 KiB		Append	Scratch	LTO1		
vol23	●	64.51 KiB		Append	Scratch	LTO1		
vol24	●	64.51 KiB		Append	Scratch	LTO1		
vol25	●	64.51 KiB		Append	Scratch	LTO1		
vol26	●	64.51 KiB		Append	Scratch	LTO1		
vol27	●	64.51 KiB		Append	Scratch	LTO1		
vol28	●	64.51 KiB		Append	Scratch	LTO1		

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`

Example: Cleanup Disk

- Delete File Storage Volumes older than 270 days
  - from catalog via *bconsole*
  - from disk by *rm*

```
cd /your/filestorage/dir
for i in `find . -mtime +270 | sed 's/\.\///g'`
do echo $i
echo "delete volume=$i yes" | bconsole
rm -f $i
done
```

.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 predefinend 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

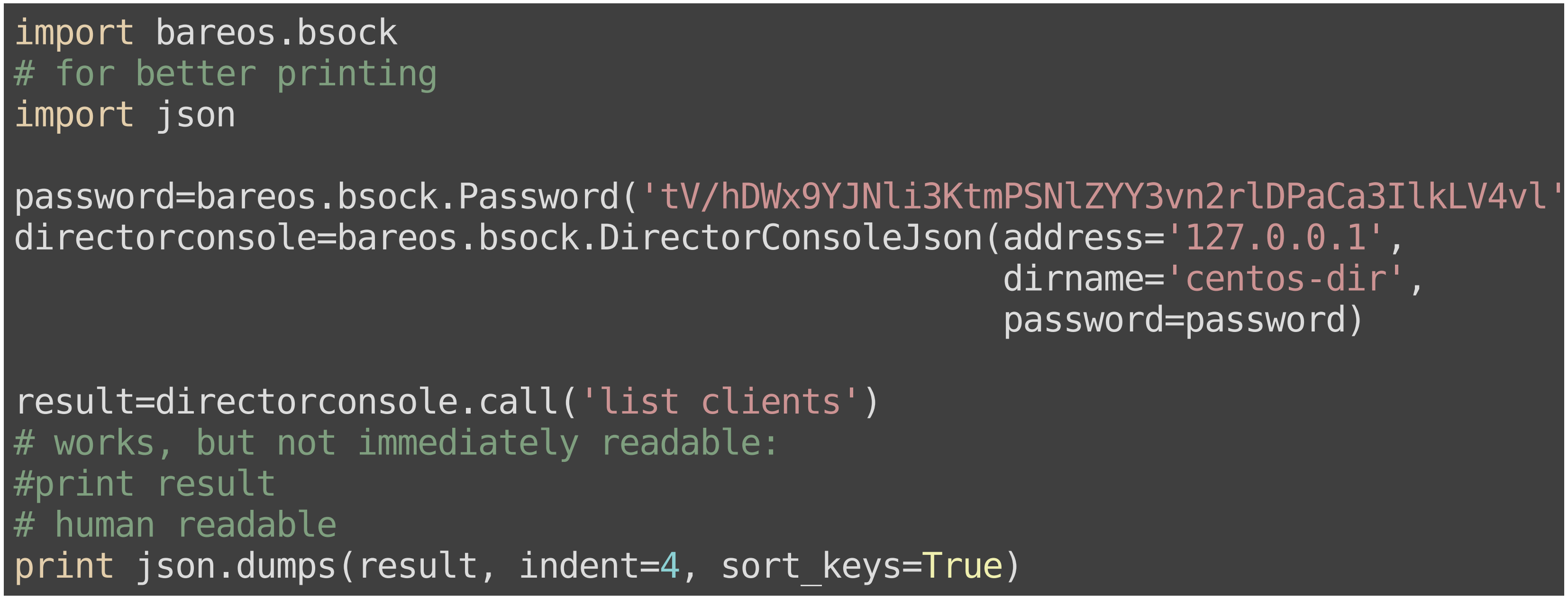
API modes example

```
*.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
- Source: GitHub
- Packages: Included in Bareos repositories *python-bareos*

python-bareos json

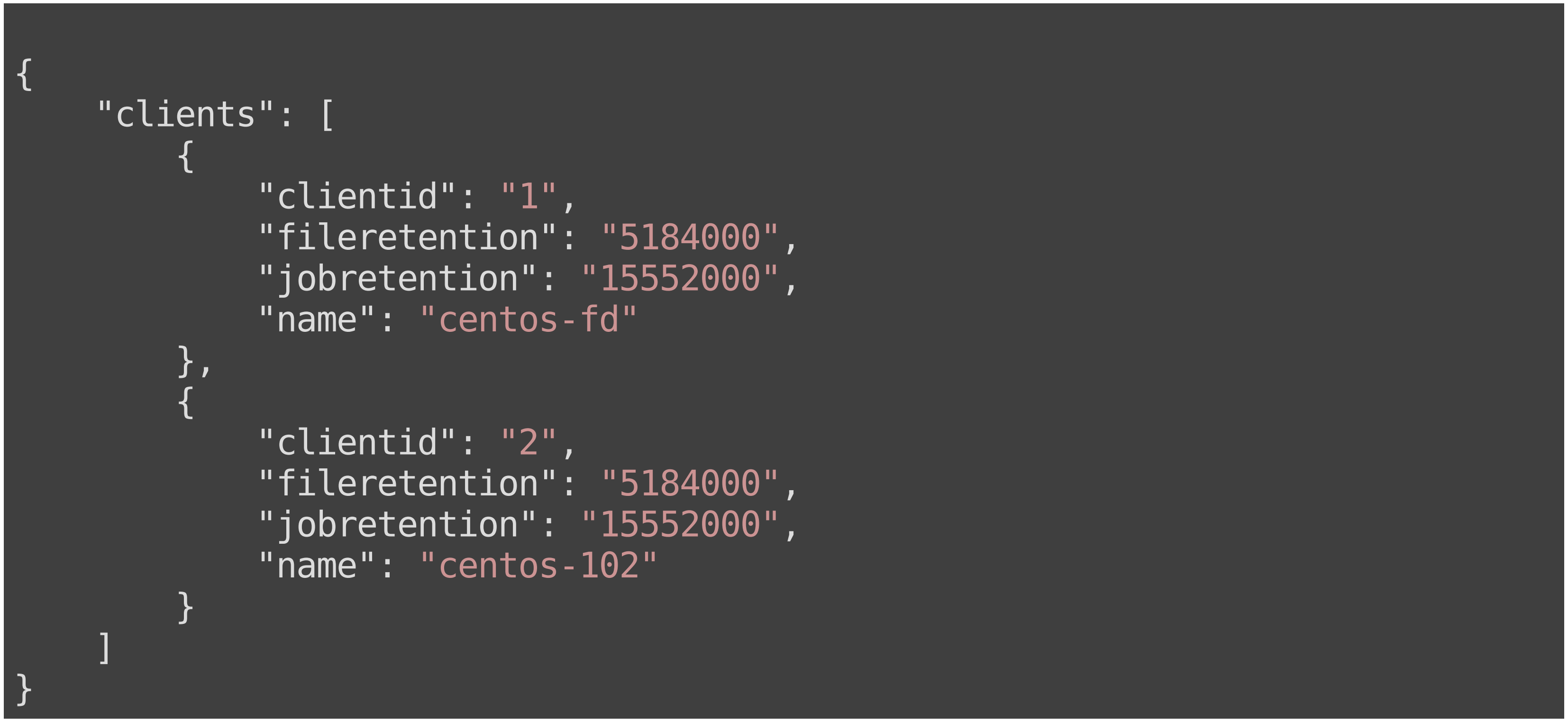


```
import bareos.bsock
# for better printing
import json

password=bareos.bsock.Password('tV/hDWx9YJNli3KtmPSNlZYY3vn2rlDPaCa3IlkLV4vl'
directorconsole=bareos.bsock.DirectorConsoleJson(address='127.0.0.1',
                                                 dirname='centos-dir',
                                                 password=password)

result=directorconsole.call('list clients')
# works, but not immediately readable:
#print result
# human readable
print json.dumps(result, indent=4, sort_keys=True)
```

Excercise

Get the above scriptlet running on your installation

Sample Script Output:



```
{
    "clients": [
        {
            "clientid": "1",
            "fileretention": "5184000",
            "jobretention": "15552000",
            "name": "centos-fd"
        },
        {
            "clientid": "2",
            "fileretention": "5184000",
            "jobretention": "15552000",
            "name": "centos-102"
        }
    ]
}
```

Retrieving File Information

List of files and directories

Before running a backup job

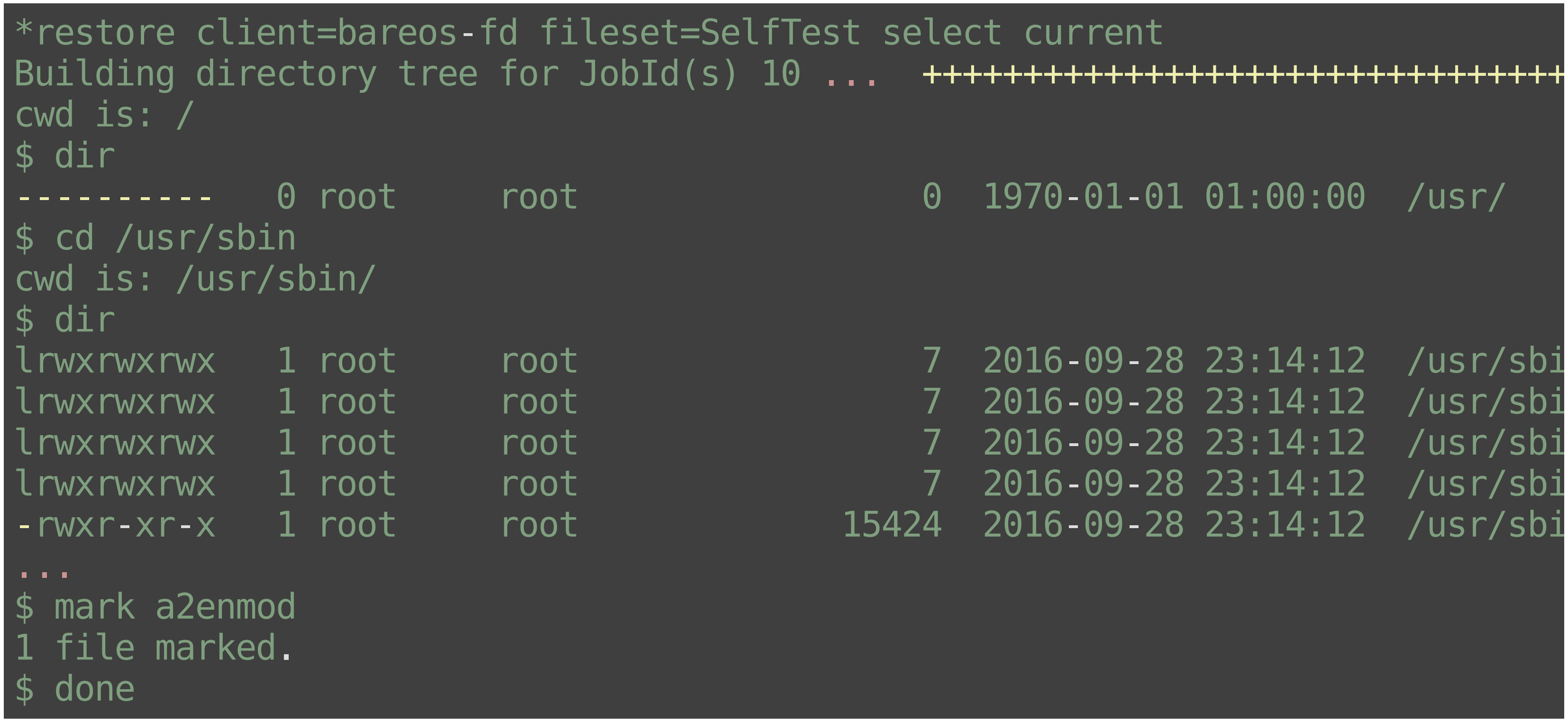
```
*estimate listing job=backup-bareos-fd
Using Catalog "MyCatalog"
Connecting to Client bareos-fd at bareos.example.com:9102
-rwxr-xr-x   1 root     root             3358 2016-10-17 00:13:20  /usr/sbin/b
-rwxr-xr-x   1 root     root            31544 2016-09-28 23:17:24  /usr/sbin/s
-rwxr-xr-x   1 root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

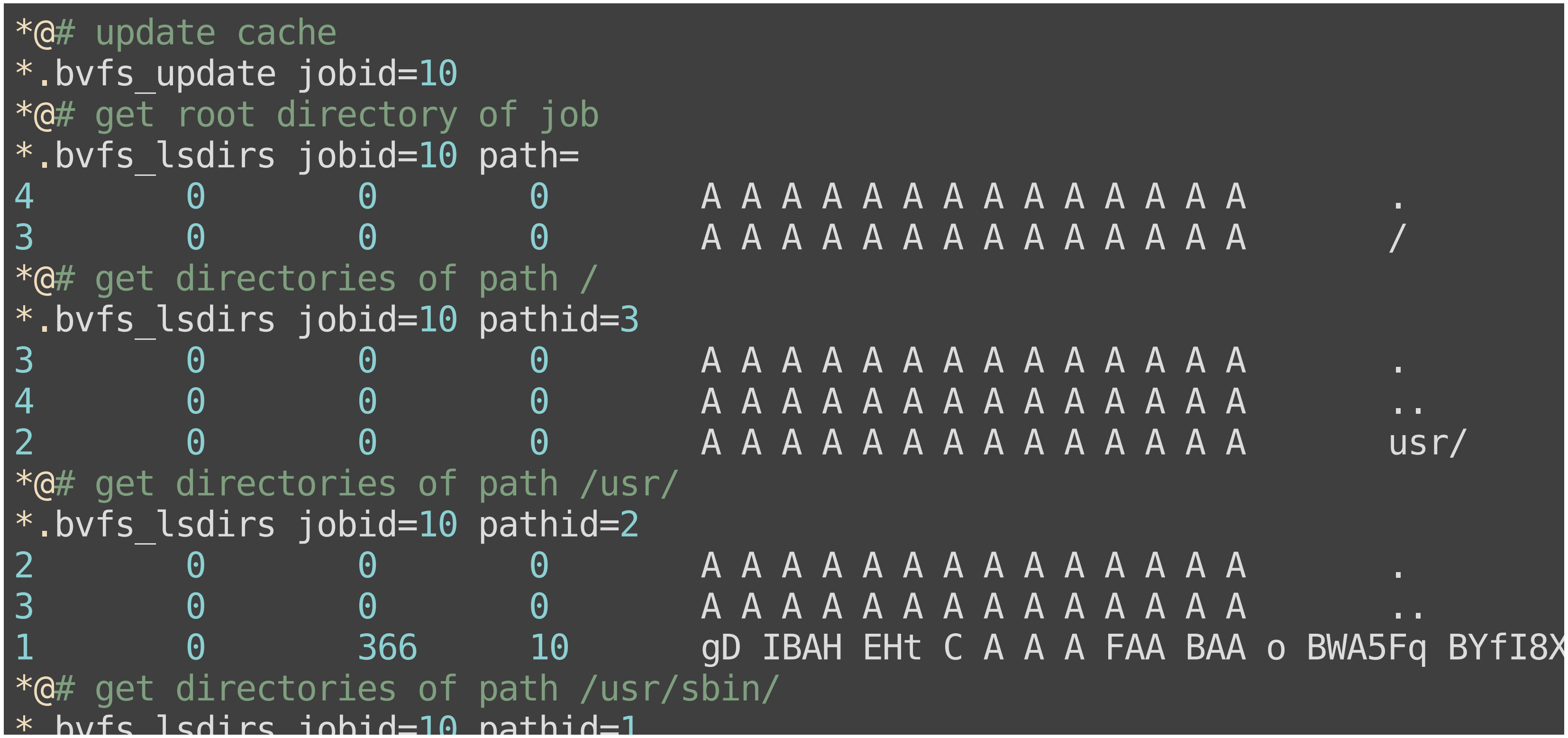


```
*restore client=bareos-fd fileset=SelfTest select current
Building directory tree for JobId(s) 10 ...  ++++++++++++++++++++++++++++++++
cwd is: /
$ dir
----------   0 root     root                    0  1970-01-01 01:00:00  /usr/
$ cd /usr/sbin
cwd is: /usr/sbin/
$ dir
lrwxrwxrwx   1 root     root                    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

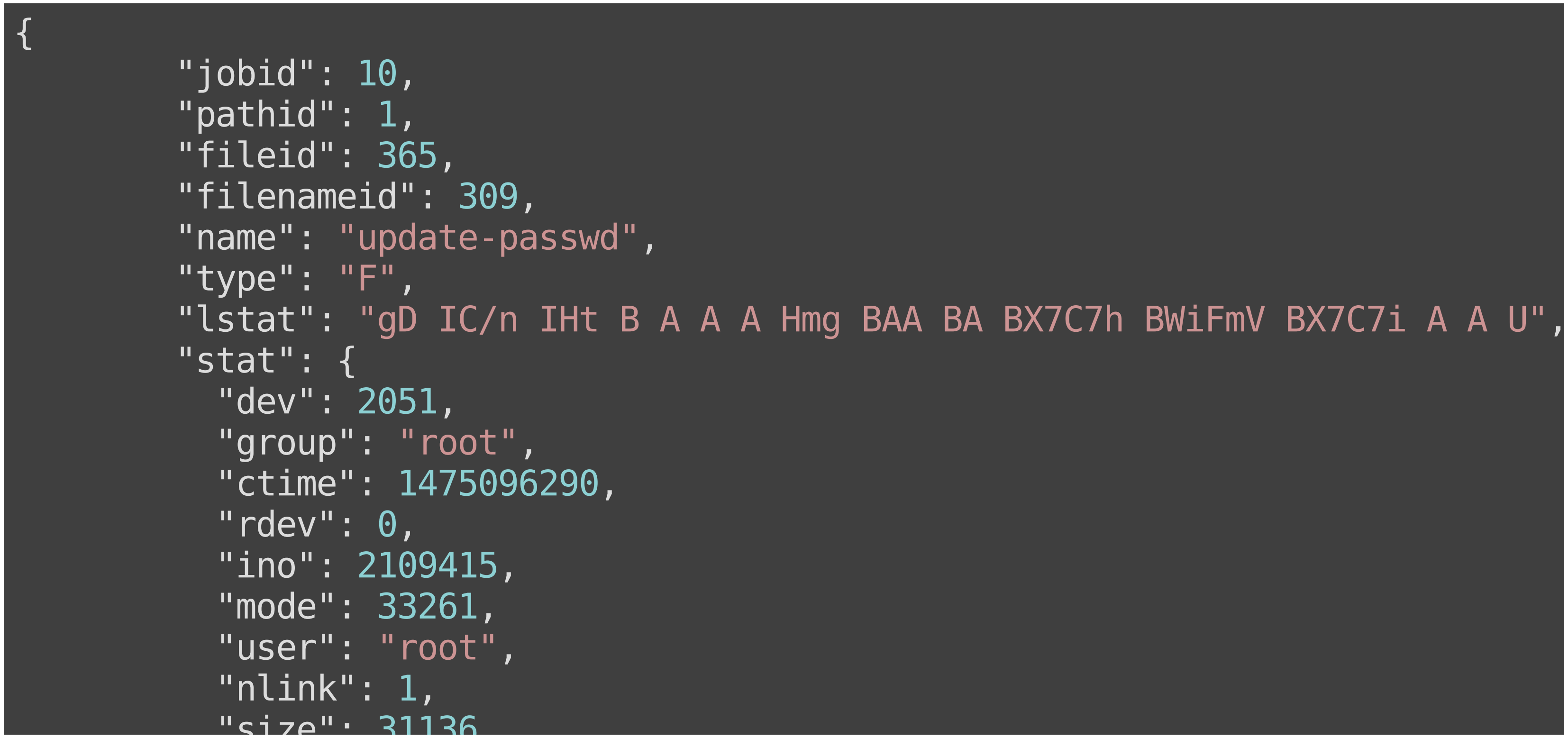
bvfs api

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     .
3       0       0       0       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     .
4       0       0       0       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     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     .
3       0       0       0       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
```

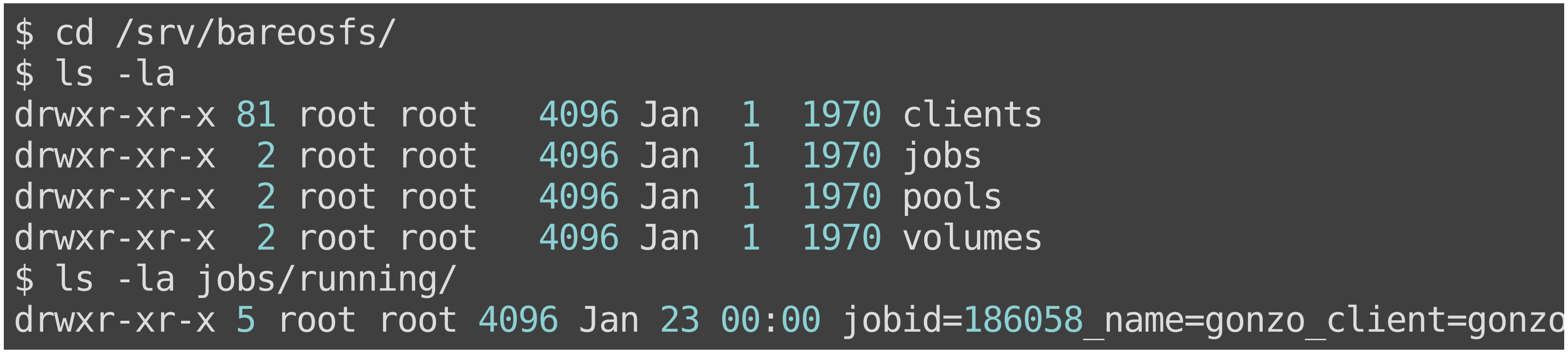
bvfs api json output



```
{
        "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

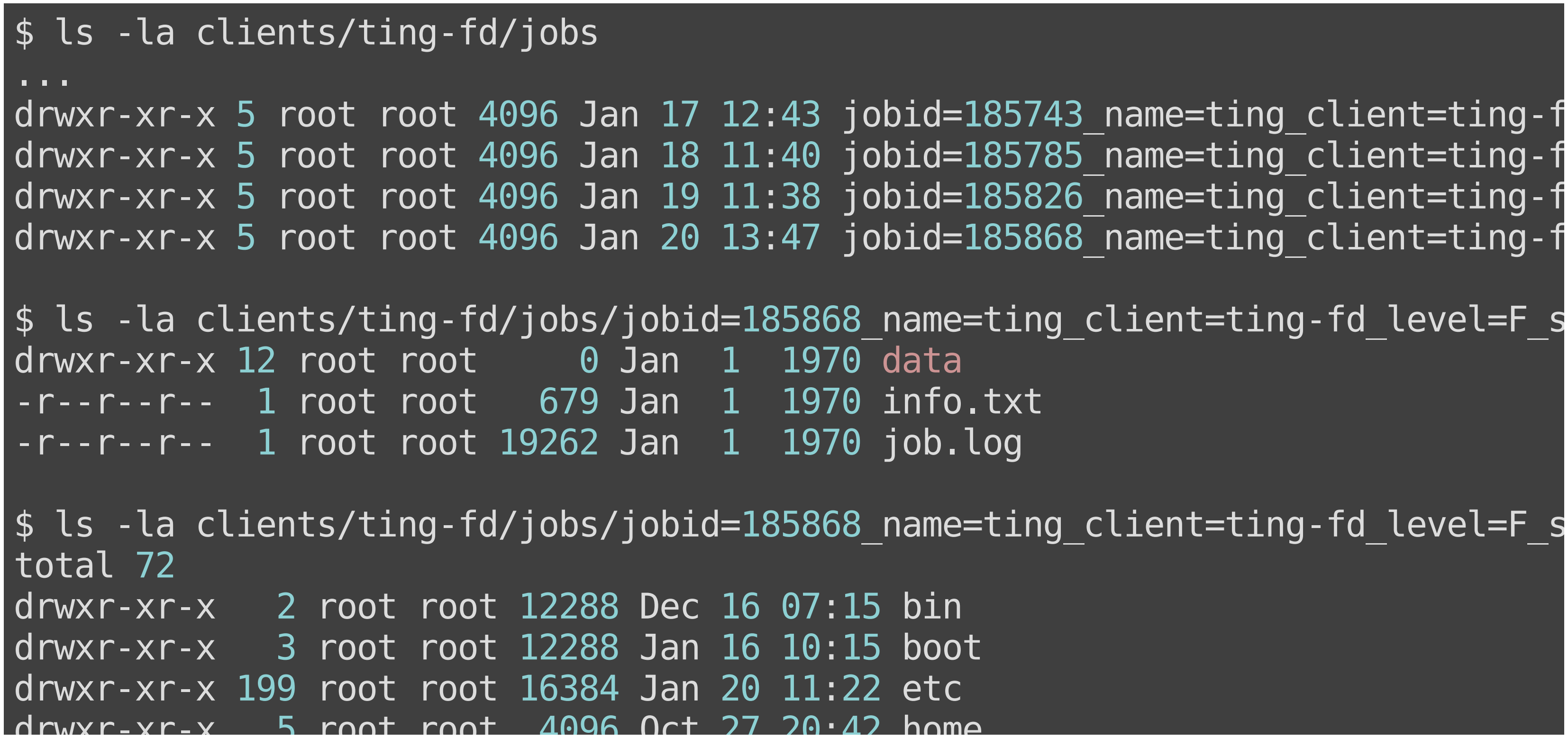
- https://www.bareos.org/en/bareos-fuse.html
- uses python-bareos (JSON)
- prototype but useful
- access to jobs, job logs, volumes, pools, meta data of backed up files



```
$ 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

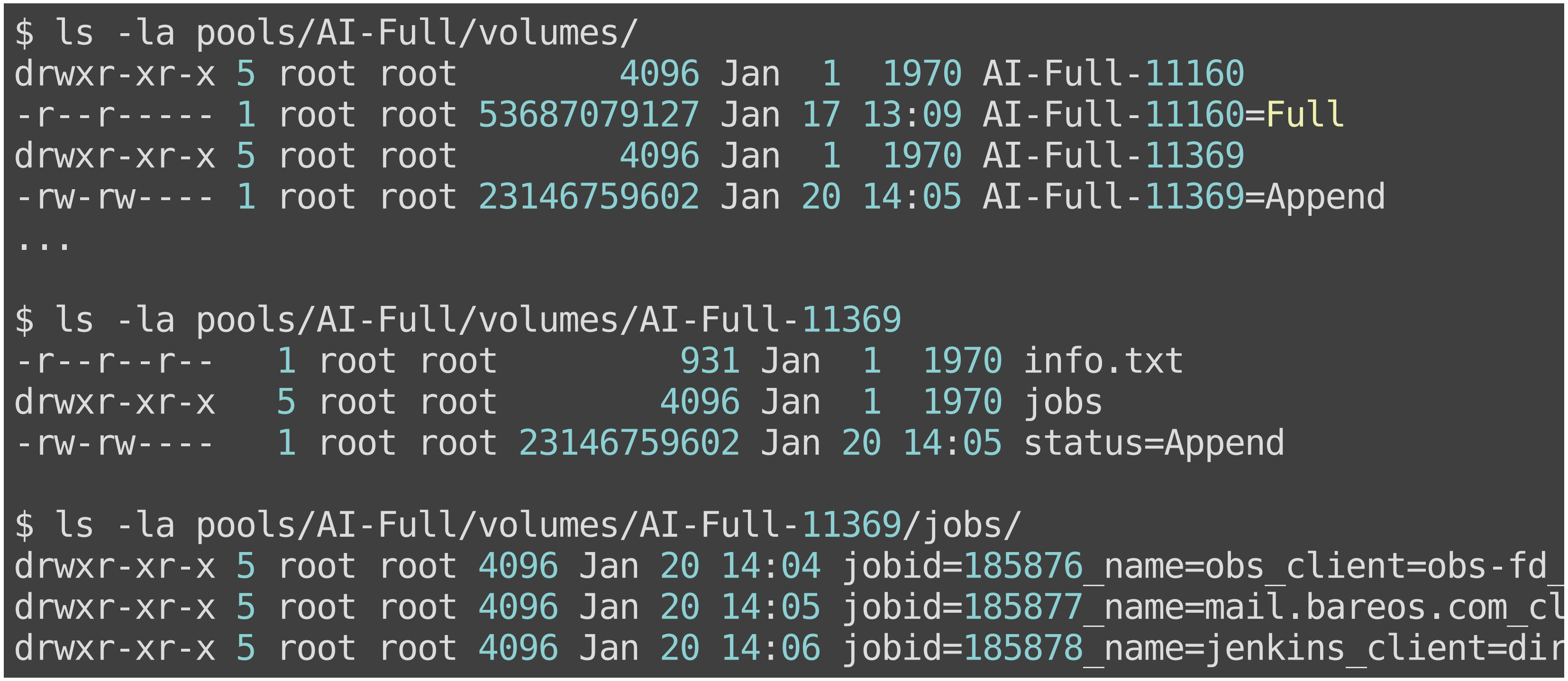
jobs of client ting-fd



```
$ 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
```

bareos-fuse

installation

```
DIST=Debian_9.0
URL=http://download.bareos.org/bareos/contrib/$DIST/
wget -q $URL/Release.key -O- | apt-key add -
printf "deb $URL /\n" > /etc/apt/sources.list.d/bareos-contrib.list
apt-get update
aptitude install bareos-fuse
```

- /etc/bareos/bareos-dir.d/profile/bareosfs-all.conf
- /etc/bareos/bareos-dir.d/console/bareosfs.conf.example

```
*configure add console=bareosfs password=secret profile=bareosfs-all
```

bareos-fuse

```
mkdir /tmp/mnt

# minimal
bareos-fuse.py /tmp/mnt/ -o name=bareosfs,password=secret

# useful for testing, non root
bareos-fuse.py /tmp/mnt/ -d -o name=bareosfs,password=secret,logfile=/tmp/bar

# stop bareos-fuse
killall bareos-fuse.py

# restore
getfattr -d a2enmod
setfattr -n user.bareos.do -v restore a2enmod
getfattr -d a2enmod
```

- /var/log/bareos/bareos-audit.log

Excercise

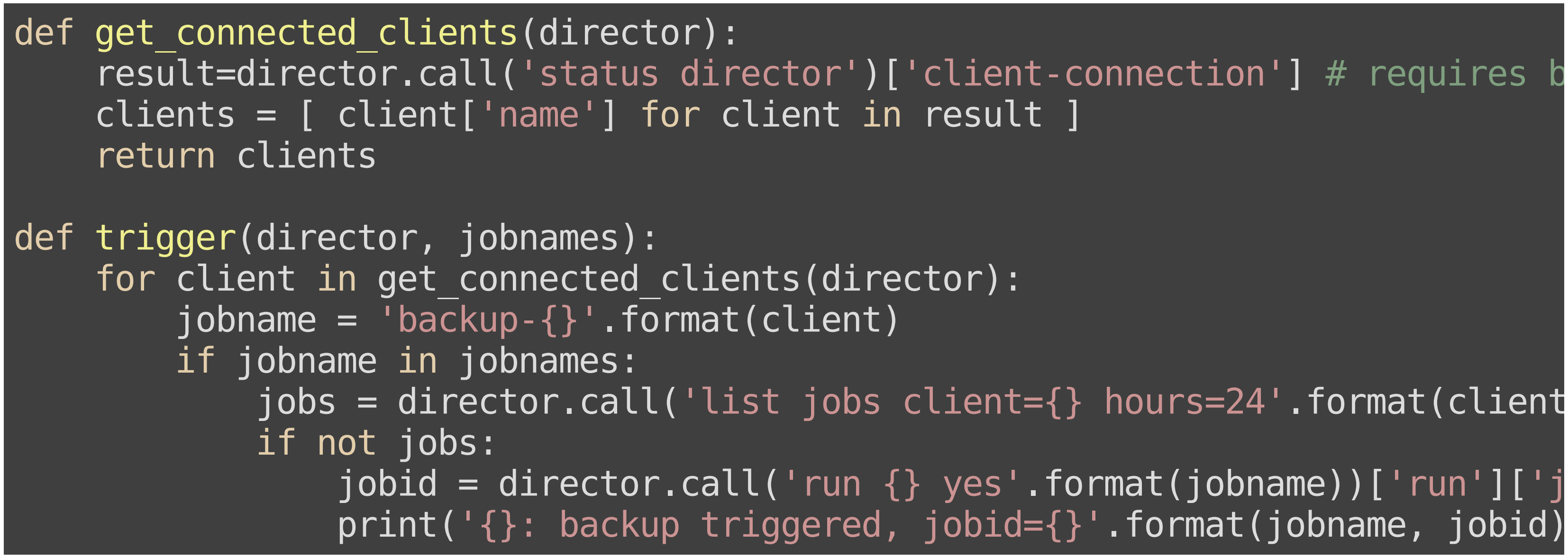
- mount bareos-fuse
- check on jobs
- find last volume
- restore a file

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'] # requires b
    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 = getFormatedOutput(self.jobClient, self.jobErrors, self.jobBytes, s
    self.send_to_graphite(data)
```

bsmc : bareos simple management client

- Python script using bareos-bsock module
- Showcase for python-bareos implementing some useful routine tasks
- Source on GitHub
- Package in Bareos-contrib

```
[root@centos ~]# bsmc -h
usage: bsmc [-h] [-d] [-p PASSWORD] {query,incr,rest,archive} ...

CLI to Bareos director.

positional arguments:
  {query,incr,rest,archive}


Run 'bsmc subcommand help'
    query
       query command is used to query information from


director
    incr
```

bsmc.conf in */etc/bareos*

```
[director]
# director network FQHN / IP
server=centos
# director name
name=centos-dir
password=secretsecret

[client]
name=centos-fd
# comma separated list of backup jobs relevant for this client
jobs=BackupClient1
# dedicated job for longterm archiving files
archivejob=archive
# temporary file for archive job
archivefilelist=/tmp/bsmc.archive
```

Example Archive Job

```
JobDefs {
  Name = "ArchiveDefaults"
  Type = Backup
  Level = Full
  Messages = "Standard"
  Storage = "File"
  Pool = "Archive"
  FileSet = "archive"
  Schedule = "Never"
  SpoolData = yes
}

Job {
  Name = "archive"
  Client = "centos-fd"
  JobDefs = "ArchiveDefaults"
}
```

Excercise

- Get *bsmc* configured (without archive job for now)
- Run incr job and restore a single file

Hands on (1/4)

- Extend *bsmc*, use *bareos.bsock* or bash / *bconsole*
  - Retrieve Media Information
    - Print a media list with useful information like last-written, number of jobs, amount of data
    - Create paper-lables for media with information like before, QR-Code with link to media-page in WebUI.

Solution Roadmap

- General hints
  - Try manually bconsole commands before scripting
  - To debug: dump commands and execute manually, if in doubt

Solution Roadmap

- Show **media list** with last-written, number of jobs, amount of data, Pool, Expire date
  - use *llist media*
  - Get pool name by pool id using *list pools*
  - Expire date: for 'Full' media, LastWritten + VolRetention

Hands on (2/4)

- Extend *bsmc*, use *bareos.bsock* or bash / *bconsole*
  - Show *lost* files in current directory (files that do not exist anymore on system but are available in backup)
    - Show newest available version
    - Show list of available versions with timestamp, size and jobid
  - Show available versions from backup of a specific file with timestamp, size and jobid

Solution Roadmap

- Show **lost files** in current directory (files that do not exist anymore on system but are available in backup)
    - Get jobids for client: *list jobs client=clientname*
    - Before using *.bvfs* commands, run **.bvfs_update**
    - Use *.bvfs_lsdirs jobid=i,k path=/* to get list of directories with pathids
    - Get filelist with *.bvfs_lsfiles pathid=... jobid=i,k*
        - Stat-information readable in *.api 2* mode only
    - use *os.listdir('./')* to get list of files / subdirectories for current path
    - use *os.path.isfile()* and *os.path.isdir()* to distinguish between files / directories

Solution Roadmap

- Show **available versions** of a given file with timestamp, size and jobid
  - Get jobids for client: *list jobs client=clientname*
  - Use *.bvfs_lsdirs jobid=i,k path=/* to get list of directories with pathids
  - Get filelist with *.bvfs_lsfiles pathid=... jobid=i,k* note filenameid (fnid)
  - Get versions with _.bvfs_versions pathid=.. jobid=i,j,k fnid=.. client=..
  - To resolve stat-string, use .bvfs_lsfiles
- **WARNING**: with 17.2 filenameid will be replaced by filename (denormalization of filename-table)

Hands on (3/4)

- Extend *bsmc*, use *bareos.bsock* or bash / *bconsole*
  - Restore a given file with optional jobid
  - Script to run a full backup for a client
  - Allow bsmc authentication with named consoles

Solution Roadmap

- Restore file with given **jobid**
  - See method *restoreFile* in *bsmc*
  - add option *jobid*=..

Solution Roadmap

- Script to run **full** backup for a client
    - See method *incremental* in *bsmc*
    - add option *level={Full, Incremental, ..}*
    - rename method *incremental* to *runBackup* and add method option *level*

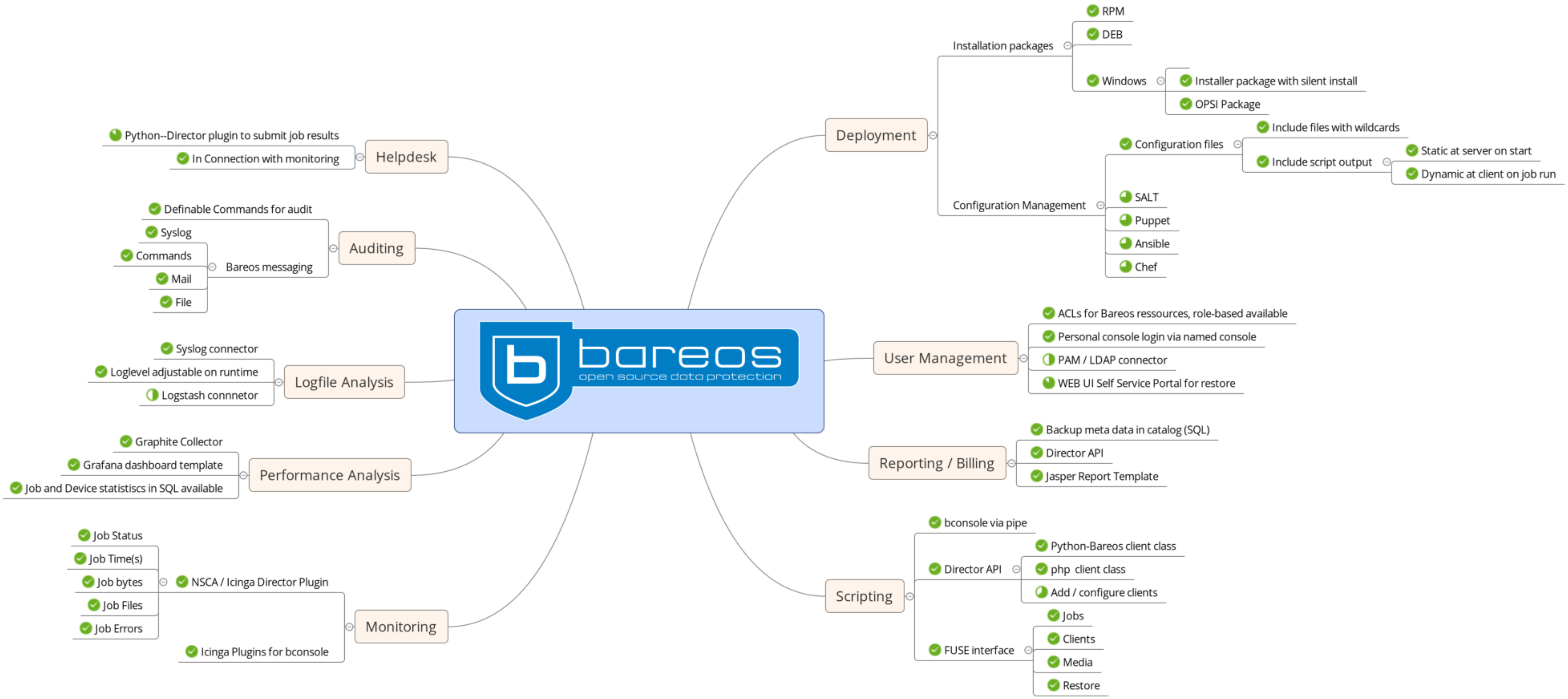
Solution Roadmap

- Allow bsmc to authenticate with **named console**
  - See options in constructor of *bsock.py* directorconsole.py
  - Add configuration file option in *bsmc.conf*
  - Adjust bsmc-call of *bareos.bsock.BSock*

Hands on (4/4)

- Other ideas
  - Write Windows-Explorer plugin that implements some of the before named options
    - On right-click on file / directory: show lost files, other versions of a file in backup, ..
- Your ideas ?
  -
  -
  -

Outlook: Toolchain Integration



bareos
open source data protection

- Deployment
  - Installation packages
    - ✓ RPM
    - ✓ DEB
    - ✓ Windows
      - ✓ Installer package with silent install
      - ✓ OPSI Package
  - Configuration Management
    - ✓ Configuration files
      - ✓ Include files with wildcards
      - ✓ Include script output
        - ✓ Static at server on start
        - ✓ Dynamic at client on job run
    - ◕ SALT
    - ◕ Puppet
    - ◕ Ansible
    - ◕ Chef
- User Management
  - ✓ ACLs for Bareos ressources, role-based available
  - ✓ Personal console login via named console
  - ◑ PAM / LDAP connector
  - ◕ WEB UI Self Service Portal for restore
- Reporting / Billing
  - ✓ Backup meta data in catalog (SQL)
  - ✓ Director API
  - ✓ Jasper Report Template
- Scripting
  - ✓ bconsole via pipe
  - ✓ Director API
    - ✓ Python-Bareos client class
    - ✓ php client class
    - ◑ Add / configure clients
  - ✓ FUSE interface
    - ✓ Jobs
    - ✓ Clients
    - ✓ Media
    - ✓ Restore
- Helpdesk
  - ◕ Python--Director plugin to submit job results
  - ✓ In Connection with monitoring
- Auditing
  - ✓ Definable Commands for audit
  - Bareos messaging
    - ✓ Syslog
    - ✓ Commands
    - ✓ Mail
    - ✓ File
- Logfile Analysis
  - ✓ Syslog connector
  - ✓ Loglevel adjustable on runtime
  - ◑ Logstash connnetor
- Performance Analysis
  - ✓ Graphite Collector
  - ✓ Grafana dashboard template
  - ✓ Job and Device statistiscs in SQL available
- Monitoring
  - ✓ NSCA / Icinga Director Plugin
    - ✓ Job Status
    - ✓ Job Time(s)
    - ✓ Job bytes
    - ✓ Job Files
    - ✓ Job Errors
  - ✓ Icinga Plugins for bconsole