Bareos Overview

www.bareos.org
Agenda

1. What is Bareos / Bareos Features
2. Bareos Architecture
3. Installation
4. Workflow (run jobs, restore)
5. Configuration
6. Plugins
7. Roadmap
What is Bareos?
Bareos

- Network based multi-platform backup solution
- License: AGPL, 100% open source
- [https://github.com/bareos/](https://github.com/bareos/)
- Core written in C/C++
- Forked from Bacula in 2010
- First release in 2013 (bareos-12.4.3)
- One major release every year
  - current: bareos-16.2.4 (16.2.5 soon)
Multi-platform
Installation packages for

- bareos.org/bareos.com
  - CentOS, Debian, Fedora, openSUSE, RHEL, SLES, Ubuntu, Univention Corporate Server
  - Windows 32/64 bit
  - Mac Client
  - FreeBSD
  - UNIX: AIX, HP-UX, Solaris

- Distributions
  - Arch Linux, Debian, FreeBSD, Gentoo, Ubuntu Universe
Network Backup with Bareos
Features

- All common features of a backup solution are supported
- Full, Differential and Incremental backups
- Always Incremental
- Backup Management
  - Volume Management
  - Retention periods
- Flexible Scheduling
- Flexible network setup
  - Director, Storage Daemon, File Daemon
Features

- Different User Interfaces
  - bconsole, bareos-webui (PHP), bat (QT-GUI, deprecated)
Features

• Security
  ▪ Challenge-response authentication
  ▪ TLS
  ▪ Client: Data Encryption
  ▪ Tape: LTO encryption (hardware, keys stored in Bareos Catalog)
  ▪ Audit Log
  ▪ Secure Erase Command
  ▪ ACLs
  ▪ File Daemon: restricted mode
Features

- API / scripting
- Plugin Support
  - C/C++ and Python plugins
- integration with/in other software
  - e.g. Relax-and-Recover
Bareos Architecture
Bareos Architecture
File Daemon

- Runs on Client Computer
- read, write, verify files
- read, write ACLs, attributes
- make VSS snapshots
- checksum calculation
- compression/encryption
- run scripts
- Plugin interface (C++, Python)
Storage Daemon

- device access (disk, tape, cloud)
- media changer control
- read barcodes labels
- Multiple Storage Daemons
  - run Migration and Copy Jobs on/to multiple locations
- handle media errors
- Plugin interface (C++, Python)
Catalog

- stores information about all files, media, jobs
- PostgreSQL/MySQL/SQLite
Director

- handles catalog
- media and pool handling
- scheduling
- trigger jobs
- backup level
- messages, statistics and reports
- run scripts
- Plugin interface (C++, Python)
Network Connectivity

- Normally:
  - Connection are only made when required.
  - Director connects to SD and FD.
    - Tells the SD that it will receive a connection from the FD soon.
    - Tells the FD to connect to the SD.
- Other options:
  - Passive Client:
    - Director tells SD to connect to FD
  - Client Initiated Connection:
    - FD connects to Director
Installation of Bareos
Installing a Bareos Server

1. Install the database of your choice
2. Add Bareos repository
3. Install Bareos packages, matching your database
4. Prepare the Bareos database table
   - /usr/lib/bareos/scripts/create_bareos_database
   - /usr/lib/bareos/scripts/make_bareos_tables
   - /usr/lib/bareos/scripts/grant_bareos_privileges
5. Start the daemons
   => Running Backup Server for Self-Backups
Bareos Packages

Private instance of [http://openbuildservice.org/](http://openbuildservice.org/)

<table>
<thead>
<tr>
<th>Package</th>
<th>CentOS 5</th>
<th>CentOS 6</th>
<th>CentOS 7</th>
<th>Debian 7.0</th>
<th>Debian 8.0</th>
<th>SLE 11 SP4</th>
<th>SLE 12 SP1</th>
<th>win_cross</th>
<th>xUbuntu 12.04</th>
<th>xUbuntu 14.04</th>
</tr>
</thead>
<tbody>
<tr>
<td>bareos</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
</tr>
<tr>
<td>bareos-docs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bareos-vadp-dumpor</td>
<td></td>
<td></td>
<td></td>
<td>succeeded</td>
<td>succeeded</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bareos-vmware-plugin</td>
<td></td>
<td></td>
<td></td>
<td>succeeded</td>
<td>succeeded</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bareos-vmware-vx.disklib</td>
<td></td>
<td></td>
<td></td>
<td>succeeded</td>
<td>succeeded</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bareos-webui</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
</tr>
<tr>
<td>libfastz</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>libgssapi</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>libgssapi5</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lzo</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td>succeeded</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mingw-debugsc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mingw32-wince</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mingw44-wince</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>python-py</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>succeeded</td>
<td></td>
</tr>
<tr>
<td>python-pynvram</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>succeeded</td>
<td></td>
</tr>
<tr>
<td>python-requests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>succeeded</td>
<td></td>
</tr>
<tr>
<td>python-six</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>succeeded</td>
<td></td>
</tr>
<tr>
<td>winbaros-rsi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>succeeded</td>
<td></td>
</tr>
<tr>
<td>winbaros-vcsi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>succeeded</td>
<td></td>
</tr>
</tbody>
</table>
Windows

Windows installer:

- cross-compiled on Linux (http://openbuildservice.org/)
- configuration of the Windows Firewall
- silent install options / OPSI packages
- FD, SD and Director can be selected
- debug package also installs sourcecode
Windows Installation

- FD, SD and Director can be selected
- configuration of the Windows Firewall
- silent install options / OPSI packages
- debug package also installs sourcecode
- cross-compiled on Linux
  - http://openbuildservice.org/
Workflow
Workflow

bconsole: start job

admin@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
A job name must be specified.
The defined Job resources are:
   1: backup-bareos-fd
   2: RestoreFiles
   3: CopyToTape
   4: BackupClient1
   5: BackupCatalog

Select Job resource (1-5): 4
## Workflow

bconsole: start job

<table>
<thead>
<tr>
<th>Run Backup job</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobName:</td>
</tr>
<tr>
<td>Level:</td>
</tr>
<tr>
<td>Client:</td>
</tr>
<tr>
<td>Format:</td>
</tr>
<tr>
<td>FileSet:</td>
</tr>
<tr>
<td>Pool:</td>
</tr>
<tr>
<td>Storage:</td>
</tr>
<tr>
<td>When:</td>
</tr>
<tr>
<td>Priority:</td>
</tr>
</tbody>
</table>

OK to run? (yes/mod/no): **yes**

Job queued. JobId=3

You have messages.

*
Workflow
bconsole: job message

*messages
30-Jan 16:31 bareos-dir JobId 3: No prior Full backup Job record found.
30-Jan 16:31 bareos-dir JobId 3: No prior or suitable Full backup found in catalog. Doing FULL backup.
30-Jan 16:31 bareos-dir JobId 3: Start Backup JobId 3, Job=BackupClient1.2017-01-30_16.31.05_07
30-Jan 16:31 bareos-dir JobId 3: Using Device "FileStorage" to write.
30-Jan 16:31 bareos-sd JobId 3: Volume "File-0001" previously written, moving to end of data.
30-Jan 16:31 bareos-sd JobId 3: Ready to append to end of Volume "File-0001" size=32419543
30-Jan 16:31 bareos-sd JobId 3: Elapsed time=00:00:01, Transfer rate=32.38 M Bytes/second
30-Jan 16:31 bareos-dir JobId 3: Bareos bareos-dir 16.2.4 (01Jul16):
  Build OS: x86_64-suse-linux-gnu suse
  openSUSE Leap 42.1 (x86_64)
  JobId: 3
Workflow
bconsole: start job a second time

Run Backup job
JobName: BackupClient1
Level: Incremental
Client: bareos-fd
Format: Native
FileSet: Full Set
Pool: File (From Job resource)
Storage: File (From Job resource)
When: 2017-01-30 16:40:59
Priority: 10
OK to run? (yes/mod/no): yes
Job queued. JobId=4
You have messages.
*
Workflow

Bconsole job, second run: incremental

*messages
30-Jan 16:41 bareos-dir JobId 4: Start Backup JobId 4, Job=BackupClient1.2017-01-30
30-Jan 16:41 bareos-dir JobId 4: Using Device "FileStorage" to write.
30-Jan 16:41 bareos-sd JobId 4: Volume "File-0001" previously written, moving
30-Jan 16:41 bareos-sd JobId 4: Ready to append to end of Volume "File-0001"
30-Jan 16:41 bareos-sd JobId 4: Elapsed time=00:00:01, Transfer rate=0 Bytes
30-Jan 16:41 bareos-dir JobId 4: Bareos bareos-dir 16.2.4 (01Jul16):
    Build OS: x86_64-suse-linux-gnu suse openSUSE Leap 42.1 (x86_64)
    JobId: 4
    Job: BackupClient1.2017-01-30 16.41.45_08
    Backup Level: Incremental, since=2017-01-30 16:31:08
    Client: "bareos-fd" 16.2.4 (01Jul16) x86_64-suse-linux-gnu,
    FileSet: "Full Set" 2017-01-30 16:29:42
    Pool: "File" (From Job resource)
    Catalog: "MyCatalog" (From Client resource)
    Storage: "File" (From Job resource)
    Scheduled time: 30-Jan-2017 16:41:42
    Start time: 30-Jan-2017 16:41:47
Workflow
Webui: run
Workflow
Webui: list jobs

![Workflow Webui: list jobs](image-url)
Workflow
Webui: list joblog
Workflow
Webui: restore
Bareos Configuration
Bareos Configuration

- Configuration is done in config files
- Each daemon has its own config directory
- usually in `/etc/bareos/daemon.d/resource/*.conf`
  - `/etc/bareos/bareos-dir.d/`
  - `/etc/bareos/bareos-sd.d/`
  - `/etc/bareos/bareos-fd.d/`
- bconsole:
  - `/etc/bareos/bconsole.conf`
FileSet: Definition what to backup

```
FileSet {
    Name = "LinuxAll"
    Include {
        Options {
            Signature = MD5
            One FS = No
            FS Type = btrfs
            FS Type = ext4
            FS Type = zfs
        }
        File = /
    }
    Exclude {
        File = /tmp
    }
}
```
FileSet: let client decide, what to backup

FileSet {
    Name = "LinuxClientDefinedList"
    Include {
        Options {
            Signature = MD5
        }
        File = "\X/etc/bareos/backup-paths.list"
    }
}

\X => \< file_path
/etc/bareos/backup-paths.list:

/home/adam
/home/eva
Schedule: Definition when to run a backup

Schedule {
  Name = "WeeklyCycle"
  Run = Full 1st sun at 23:05
  Run = Differential 2nd-5th sun at 23:05
  Run = Incremental mon-sat at 23:05
}
Client: Definition of a Client

```json
Client {
    Name = bareos-fd
    Address = 192.168.0.1
    Password = "lecCqzgBjxgM0J3+1adiuLzhy0cPGIHrdYMdtGHMbvKX"
}
```
Job: Definition of a Job

- combines the other resources to a runnable backup job

```
Job {
    Name = "backup-bareos-fd"  # name of this resource
    Client = "bareos-fd"       # what client to backup?
    FileSet = "LinuxAll"       # which files to backup?
    Schedule = "WeeklyCycle"   # when to backup?
    Storage = "File"           # where to backup?
    Messages = "Standard"      # where to send messages?
    Full Backup Pool = "Full"  # write Full Backups into "Full" Pool
    Differential Backup Pool = "Differential" # write Diff Backups into "Differential"
    Incremental Backup Pool = "Incremental" # write Incr Backups into "Incremental"
    [...]
}
```
Pool: Full

```yaml
Pool {
  Name = Full
  Pool Type = Backup
  Recycle = yes  # Bareos can automatically recycle Volumes
  AutoPrune = yes  # Prune expired volumes
  Volume Retention = 365 days  # How long should the Full Backups be kept?
  Maximum Volume Bytes = 50G  # Limit Volume size to something reasonable
  Maximum Volumes = 100  # Limit number of Volumes in Pool
  Label Format = "Full-"  # Volumes will be labeled
}
```
Pool: Incremental

Pool {
  Name = Incremental
  Pool Type = Backup
  Recycle = yes # Bareos can automatically recycle Volumes
  AutoPrune = yes # Prune expired volumes
  Volume Retention = 30 days # How long should the Incremental Backups be kept?
  Maximum Volume Bytes = 1G # Limit Volume size to something reasonable
  Maximum Volumes = 100 # Limit number of Volumes in Pool
  Label Format = "Incremental-" # Volumes will be labeled
}
Add A Client

- bareos < 16.2: manually
- bareos >= 16.2:
  - client: install bareos-filedaemon
  - server: "configure add client"
  - server: copy generated client configuration to client
  - client: restart bareos-filedaemon
Add A Client

• Client:
  ▪ add Bareos repository
  ▪ install the package bareos-filedaemon

• Server:

  ```
  linux# bconsole
  *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"
  ```

• creates
  ▪ `/etc/bareos/bareos-dir.d/client/client2-fd.conf`
  ▪ `/etc/bareos/bareos-dir-export/client/client2-fd/bareos-fd.d/director/bareos-dir.conf`

• copy filedaemon configuration to client

  ```
  linux# scp /etc/bareos/bareos-dir-export/client/client2-fd/bareos-fd.d/director/bareos-dir.conf root@client2.example.com:/etc/bareos/bareos-fd.d/director/
  ```

• Client: restart bareos-filedaemon
Add A Client: Verify

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

*estimate listing job=BackupClient1 client=client2-fd
Connecting to Client client2-fd at 192.168.0.2:9102
lrwxrwxrwx 1 root  root  7 2016-09-28 23:14:12 /usr/sbin/a2disconf -> a2enmod
lrwxrwxrwx 1 root  root  7 2016-09-28 23:14:12 /usr/sbin/a2enconf -> a2enmod
...

Add A Job

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

*status schedule job=backup-client2-fd days=3

<table>
<thead>
<tr>
<th>Date</th>
<th>Schedule</th>
<th>Overrides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thu 02-Feb-2017 21:00</td>
<td>WeeklyCycle</td>
<td>Level=Incremental</td>
</tr>
<tr>
<td>Fri 03-Feb-2017 21:00</td>
<td>WeeklyCycle</td>
<td>Level=Incremental</td>
</tr>
<tr>
<td>Sat 04-Feb-2017 21:00</td>
<td>WeeklyCycle</td>
<td>Level=Full</td>
</tr>
</tbody>
</table>

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

*wait jobid=256
JobId=256
JobStatus=OK (T)
Plugin: Backup using Pipes

- uses a pipe to backup a service
- backup as a virtual file

```plaintext
FileSet {
    Name = "postgresql-all"
    Include {
        Options {
            signature = MD5
            compression = gzip
        }
    }
}
```
Plugin: MySQL / MariaDB

- uses Percona xtrabackup
- Incremental backups (for INNODB tables)
- Hotbackup
- Point-In-Time Recovery

```plaintext
FileSet {
  ...
  Plugin = "python:module_path=/usr/lib64/bareos/plugins:module_name=bareos-fd-
  ...
}
```
Plugin: VMware

- VMware Vstorage API support
  - allows backup of VMware virtual machines
  - supports Changed Block Tracking (Incremental backups)
    - only used/changed blocks are backed up/restored

```
FileSet {
  ...

Plugin = "python:module_path=/usr/lib64/bareos/plugins/vmware_plugin:module_name=bareos-fd-vmware:vcserver=vcenter.example.org:dc=mydc1:folder=/webservers:vmname=websrv1:vcuser=backupadmin@ad:vcpass=secret"
  ...
}
```
NDMP support

- Storage systems often provide a NDMP backup interface
  - NetApp, Isilon, ...
- Bareos support NDMP
  - Full and Incremental backups
  - Single File restore
Volume access by native tools

- Access backup data without running Bareos Daemons
- command line tools:
  - bls, bextract, bcopy, bscan, bcrypto
In depth: Always Incremental Backup Scheme
Conventional backup scheme

1. daily incremental backups kept for one week
2. weekly differential backups kept for three weeks
3. monthly full backups kept for half year
Job availability for conventional backup scheme
Data being moved in conventional backup scheme
Problems with conventional backup scheme

1. Full data is copied over the network in regular intervals
2. Identical Data is copied from client multiple times
3. Job history loss caused by retention expiry
always incremental backup scheme

• Basic concept
  ▪ Only changes are copied from the clients - always incremental
  ▪ Existing data from the client is consolidated with the new incremental information (keep history)
  ▪ The consolidation happens without client interaction
  ▪ Minimized number of incrementals is kept to have a defined change history
Two main tasks:

1. Incremental backup job is run every night during the backup window
2. Consolidation job consolidates during the day
How to configure always incremental Backup Job

Job {
    Name = BackupClient1
    Accurate = yes
    Always Incremental = yes
    Always Incremental Job Retention = 7 days
}

Consolidation Job

Job {
    Name = "Consolidate"
    Type = Consolidate
}
The Backup Job

- runs an incremental backup during the backup window
- *Always Incremental* directives configure behaviour
- *Accurate* Backup to notice file deletion
The Consolidation Job

- Loops over all Backup Jobs
- Starts virtual backups according to Always Incremental settings
Job availability with always incremental backup scheme
Job availability compared
Always Incremental Jobdata
Always Incremental Jobdata - Problem

- good: minimal data from the client
- bad: every day the consolidation runs the whole client data is moved during consolidation
- impossible for a large number of clients
Always Incremental Jobdata - Solution

- only consolidate latest incremental during consolidation
- leave the full backup as it is during daily consolidations
- consolidate the full in longer intervals

```java
Job {
    Always Incremental Max Full Age = 21 days
}
```
Always Incremental Max Full Age = 21 days
Always Incremental Max Full Age = 21 days
Always Incremental Max Full Age with multiple clients
Always Incremental Max Full Age with multiple clients and Max Full Consolidations
Always Incremental configuration overview

Backup Job

```json
Job {
    Always Incremental = yes # enabled?
    Always Incremental Job Retention = 7 days # how long is the job history?
    Always Incremental Keep Number = 7 # guaranteed number of incs left?
    Always Incremental Max Full Age = 21 days # if full is older it will be # part of the consolidation
}
```

Consolidation Job

```json
Job {
    Name = "Consolidate"
    Type = Consolidate
    Max Full Consolidations = 1 # how many consolidation jobs # with full included can be started
}
```
Always Incremental summary

- Only incremental Backups are done from the client
  - Minimal network load
  - Minimal backup time
  - In backup window
Always Incremental summary

- Consolidation is done locally on storage
  - Outside of backup window
  - Very fast as local
  - Existing backups are consolidated into new backups
  - No holes in the backup history
- Defined incremental backup history is always available
- Adequate for File Backup, NOT for plugin Backups
ACL support

- Full multi-tenancy support
- Definitions of rules and roles
- Users can only access and see data according to role access
- Prerequisite for WebUI as self-service-portal for restore
Console ACL configuration

```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-client2
    Plugin Options ACL = *all*
    Pool ACL = *all*
    Schedule ACL = *all*
    Storage ACL = *all*
    Where ACL = *all*
}
```
Console ACL Profiles

Profile {
  Name = "webui-admin"
  CommandACL = !.bvfs_clear_cache, !.exit, !.sql
  CommandACL = !configure, !create, !delete, !purge, !prune, !sqlquery, !umount
  CommandACL = *all*
  Job ACL = *all*
  Schedule ACL = *all*
  Catalog ACL = *all*
  Pool ACL = *all*
  Storage ACL = *all*
  Client ACL = *all*
  FileSet ACL = *all*
  Where ACL = *all*
}

Console {
  Name = user2
  Password = secret
  Profile = "webui-admin"
}
Roadmap for Bareos 17.2

- PAM authentication
  - external contribution
  - modification of network handshaking required
  - need carefull testing
- Database performance enhancements
  - filename table denormalization
  - already implemented (customer specific build). Gets integrated as soon as migration process is done.
- python-bareos
  - from bareos-contrib to bareos-core
Roadmap for Bareos 17.2: NDMP

• current status:
  ▪ NDMP backups to Bareos Storage Daemon
  ▪ NDMP Single File restore (thanks to Uni Jena)

• development:
  ▪ NDMP: Backup to storage attached tape-libraries
  ▪ NDMP: Direct Access Restore