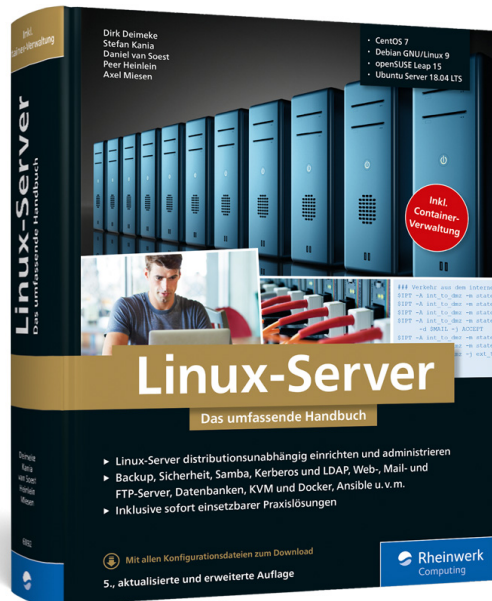


Dirk Deimeke, Stefan Kania, Daniel van Soest,  
Peer Heinlein, Axel Miesen

## Backup und Recovery mit Bacula/Bareos

Zusatzinformationen zu  
»Linux-Server – Das umfassende Handbuch«



# Kapitel 7

## Backup und Recovery

### 7.7 Backup und Recovery mit Bacula/Bareos

Die Backupsoftware Bacula/Bareos gilt als Open-Source-Lösung mit Enterprise-Niveau und hat sich einen gewissen Fanklub erarbeitet. Sie unterstützt Linux- und Windows-Clients, kann mit Tape-Libraries umgehen, unterstützt Verfahren wie *Backup2Disk2Tape* und kann auch komplexe Backupstrategien mit Kombinationen aus Vollbackup sowie differenziellen und inkrementellen Backups umsetzen.

Aufgrund von internen Meinungsverschiedenheiten über die genaue Ausrichtung des Projekts hat sich ein Teil der Entwickler im Jahr 2012 von Bacula gelöst und entwickelt dieses Projekt nun unter dem Namen Bareos nach eigenen Vorstellungen weiter. Wir stellen Ihnen in diesem Beispiel Enterprise-Backup am Beispiel von Bareos vor, die gezeigten Beispiele sollten aber auch mit dem alten Bacula-Zweig noch ebenso funktionieren.

Installieren Sie zunächst den Bareos-Server. Da die Programmierer von Bareos langjährige Linux-Spezialisten und noch dazu ehemalige Mitarbeiter der SUSE-Distribution sind, sind fertige Repositorys mit formvollendeten RPM- und DEB-Installationspakete natürlich Ehrensache. Dabei sind die stabilen »fertigen« Versionen als RPM/DEB-Paket nur mit Support-Vertrag zu bekommen; der »nightly build« ist jedoch ausreichend stabil und grundlegend vorher getestet und kann als freie Software genutzt werden.

Das Bareos-System besteht dabei aus drei Komponenten:

1. *Bareos Director*: die Schaltzentrale, die alle Aktionen koordiniert und auslöst und die anderen Komponenten steuert. Er wird über `/etc/bareos/bareos-dir.conf` konfiguriert.
2. *Bareos Storage-Daemon*: übernimmt das Schreiben und Lesen der Daten auf dem Server oder angeschlossenen Speichermedien. Der Storage-Daemon kann ebenfalls auf dem Bareos Director laufen, kann jedoch auch auf einem oder mehreren eigenständigen Systemen laufen, die vom Director angesprochen und koordiniert werden. Er wird über `/etc/bareos/bareos-sd.conf` konfiguriert.
3. *Bareos File-Daemon*: ist jeweils auf dem zu sichernden Client installiert, liest die zu sichernden Daten und sendet sie an den Storage-Daemon. Er wird über `/etc/bareos/bareos-fd.conf` konfiguriert.

Die nachfolgenden Beispiele zeigen Ihnen, wie Sie Ihre Clients zunächst auf die Festplatte eines zentralen Bareos-Servers sichern, wie also Director und Storage-Daemon auf dem gleichen System laufen, was insoweit auch nicht unüblich ist. Achten Sie bei der Installation des Bareos-Servers darauf, dass genügend freier Speicherplatz unter `/var/lib/bareos/storage` zur Verfügung steht. Fügen Sie unter openSUSE das Bareos-Repository hinzu – sollten Sie eine andere openSUSE-Version einsetzen, müssen Sie die URL natürlich entsprechend anpassen –, und installieren Sie dann mittels zypper Server und Datenbank.

```
backup:~ # zypper addrepo --refresh \
  http://download.bareos.org/bareos/experimental/nightly/openSUSE_Leap_42.1/
backup:~ # zypper install bareos bareos-database-mysql
```

#### Listing 7.45 Installation von Bareos unter openSUSE

Auch unter Ubuntu können Sie mit folgenden Kommandos auf ein Repository mit fertigen Paketen zurückgreifen:

```
user@backup:~$ URL=http://download.bareos.org/bareos/experimental/\
nightly/xUbuntu_16.04/
user@backup:~$ sudo printf "deb $URL /\n" > /etc/apt/sources.list.d/bareos.list
user@backup:~$ sudo wget -q $URL/Release.key -O- | apt-key add -
user@backup:~$ sudo apt update && sudo apt install bareos bareos-database-postgresql
```

#### Listing 7.46 Einrichtung von Bareos unter Ubuntu

Und ganz analog für Debian und CentOS:

```
http://download.bareos.org/bareos/experimental/nightly/Debian_8.0/
http://download.bareos.org/bareos/experimental/nightly/CentOS_7/
```

#### Listing 7.47 Paketquellen für andere Distributionen

Anschließend müssen Sie eine Datenbank aufsetzen, in der Bareos seine Metadaten verwaltet. Bareos unterstützt dabei bevorzugt MySQL und PostgreSQL; wir zeigen im Folgenden den Weg hier am Beispiel der weiter verbreiteten MySQL-Datenbank. Der MySQL-Server muss dazu gegebenenfalls vorab gestartet werden:

```
backup:~ # /usr/lib/bareos/scripts/create_bareos_database
backup:~ # /usr/lib/bareos/scripts/make_bareos_tables
backup:~ # /usr/lib/bareos/scripts/grant_bareos_privileges
```

#### Listing 7.48 Initialisierung der MySQL-DB für Bareos

Falls Sie den root-Account Ihrer MySQL-Datenbank mit einem Passwort geschützt haben (löblich!), können Sie die Initialisierungsskripte über den Parameter `p` nach dem notwendigen root-Passwort fragen lassen. Alle Aufrufparameter der Skripte werden 1:1 an das darunterliegende `mysql`-Kommando durchgereicht.

```
backup:~ # /usr/lib/bareos/scripts/create_bareos_database -p
Creating mysql database
Enter password:
```

**Listing 7.49** Verwendung von Passwörtern bei der Datenbankinitialisierung

Wenn Sie PostgreSQL einsetzen, müssen Sie die Initialisierung der Datenbank als PostgreSQL-Nutzer vornehmen.

```
user@backup:~ > su postgres -c /usr/lib/bareos/scripts/create_bareos_database
[...]
user@backup:~ > su postgres -c /usr/lib/bareos/scripts/make_bareos_tables
[...]
user@backup:~ > su postgres -c /usr/lib/bareos/scripts/grant_bareos_privileges
[...]
```

**Listing 7.50** Initialisierung der PostgreSQL-DB für Bareos

Ist alles fehlerfrei durchlaufen, können Sie den Bareos-Server starten. Für alle drei Komponenten existieren eigene Start-/Stop-Skripte.

```
backup:~ # systemctl start bareos-dir
backup:~ # systemctl start bareos-sd
backup:~ # systemctl start bareos-fd
```

**Listing 7.51** Start aller Bareos-Dienste

Ergänzt werden diese drei Komponenten durch das Tool `bconsole`, über das Sie als Administrator alle wesentlichen Steuerungsaufgaben in Ihrem Bareos-System wahrnehmen können (und müssen). Wir werden die Bedienung von `bconsole` auf den nächsten Seiten Schritt für Schritt darlegen; die vollständige Kommandoübersicht können Sie sich über das Kommando `bconsole help` anzeigen lassen.

### 7.7.1 Das erste Backup: Der Bareos-Server sichert sich selbst

Als allerersten Server sollten Sie Ihren Bareos-Server einmal auf sich selbst sichern. Das ist nicht nur ein guter Testkandidat für den allerersten Durchlauf, sondern auch hier kann ernsthafter Datenverlust eintreten; zudem wollen Sie die Metadaten-Datenbank von Bareos zukünftig gut gesichert wissen.

#### Anlegen eines virtuellen Backupmediums

Backupserver denken in wechselbaren Backupmedien: Tapes, Festplatten und andere Sicherungsmedien. Auch wenn Sie anfangs einfach nur auf die Festplatte Ihres Bareos-Servers sichern wollen, um die Daten vom zu sichernden Client herunterzubekommen, müssen Sie auf dem Storage-Daemon zuerst ein (virtuelles) Backupmedium (Volume) anlegen. Bareos

legt dann unter `/var/lib/bareos/storage` diese Volumes an, die aus Sicht von Bareos wie (virtuelle) Bänder benutzt werden. Bareos kümmert sich damit eigenständig darum, wo in welcher Datei eine Sicherung vorgenommen wird und aus welchen Speicherdateien das Backup zu rekonstruieren ist. Greifen Sie zum Anlegen des Volumes für das Storage Device auf das Tool `bconsole` zurück, führen Sie darin das Kommando `label` aus, und legen Sie ein filebasiertes Volume mit dem Namen `test` an.

```
backup:~ # bconsole
Connecting to Director desktop-booster:9101
1000 OK: bareos-dir Version: 16.2.3 (09 June 2016)
Enter a period to cancel a command.
*label
Automatically selected Catalog: MyCatalog
Using Catalog "MyCatalog"
Automatically selected Storage: File
Enter new Volume name: test
Defined Pools:
    1: Scratch
    2: Incremental
    3: Full
    4: Differential
Select the Pool (1-4): 1
Connecting to Storage daemon File at desktop-booster:9103 ...
Sending label command for Volume "test" Slot 0 ...
3000 OK label. VolBytes=197 Volume="test" Device="FileStorage" (/var/lib/bareos/\
storage)
Catalog record for Volume "test", Slot 0 successfully created.
Requesting to mount FileStorage ...
3906 File device "'FileStorage' (/var/lib/bareos/storage)" is always mounted.
```

#### Listing 7.52 Anlegen eines Volumes auf der Festplatte

Nun haben Sie *ein* Backupvolume angelegt, das 50 GByte groß werden darf. Wenn Ihnen das für den Alltagsbetrieb nicht reicht, müssten Sie mit dem `label`-Kommando weitere Volumes hinzufügen. Oder Sie überlassen diese Aufgabe Ihrem Bareos-Server, der dies auch automatisch erledigen kann. Suchen Sie dafür die Definition des File-Pools in der Datei `/etc/bareos/bareos-dir.conf`, und ergänzen Sie diese um den zusätzlichen Eintrag `LabelFormat=File-n`:

```
# File Pool definition
Pool {
    Name = File
    Pool Type = Backup
    Recycle = yes                # Bareos can automatically recycle Volumes
```

```

AutoPrune = yes                # Prune expired volumes
Volume Retention = 365 days    # one year
Maximum Volume Bytes = 50G     # Limit Volume size to something reasonable
Maximum Volumes = 100         # Limit number of Volumes in Pool
LabelFormat = "File-n"
}

```

**Listing 7.53** Automatisches Anlegen weiterer Volumes

Bareos wird nun weitere Volumes nach dem Namensschema `File-<nummer>` anlegen, sodass auch umfangreicheren Backups nichts im Wege steht.

**Definition der zu sichernden Dateien**

In der Datei `/etc/bareos/bareos-dir.conf` finden Sie verschiedene *FileSets* vorbereitet. Neben dem *Full Set*, das für Linux-Server geeignet ist, haben auch Windows-Server unter dem Namen *Windows All Drives* einen Eintrag vorbereitet. Diese *FileSets* bestehen jeweils aus einer Liste der zu berücksichtigenden Dateien, typischerweise mit `»/«` als Verweis auf die root-Partition, und zudem aus einer *Exclude-Liste*, die Systemdateien und die Backupdaten selbst vom Backup ausschließt.

```

# List of files to be backed up
FileSet {
  Name = "Full Set"
  Include {
    Options {
      signature = MD5
    }
    File = /
  }
  Exclude {
    File = /var/lib/bareos
    File = /var/lib/bareos/storage
    File = /proc
    File = /tmp
    File = /.journal
    File = /.fsck
  }
}

```

**Listing 7.54** Das Standard-FileSet für einen Linux-Server

Sie können ein solches *FileSet* anpassen und weitere Dateien *exclude*n, Sie können es aber auch kopieren und mit einem neuen Namen versehen und parallel zum normalen *Full Set* anbieten.

## Durchführung des ersten Backups

In der Beispielkonfiguration des Bareos Directors ist bereits eine Job-Definition *DefaultJob* enthalten, die

1. (fast) den kompletten Server (FileSet = Full Set),
2. täglich inkrementell (Level=Incremental),
3. monatlich voll (Schedule=WeeklyCycle),
4. den lokalen eigenen Rechner (Client = backup-fd) und
5. in die virtuelle Tape-Library (Storage = File)

sichert. Sie finden diese Job-Definition in *bareos-dir.conf*:

```
JobDefs {
  Name = "DefaultJob"
  Type = Backup
  Level = Incremental
  Client = backup-fd
  FileSet = "Full Set"
  Schedule = "WeeklyCycle"
  Storage = File
  Messages = Standard
  Pool = File
  Priority = 10
  Write Bootstrap = "/var/lib/bareos/%c.bsr"
}
```

**Listing 7.55** Die Job-Definition regelt Umfang, Art und Zeitpunkt des Backups.

Diese Job-Definition ist ein Template, das in einem Job benutzt werden kann. In Listing 7.56 sehen Sie den dazugehörigen Job.

```
# Define the main nightly save backup job
# By default, this job will back up to disk in /var/lib/bareos/storage
Job {
  Name = "BackupClient1"
  JobDefs = "DefaultJob"
}
```

**Listing 7.56** Der Job definiert den eigentlichen Backupjob.

Später können Sie innerhalb eines solchen Jobs auch Parameter der generischen Job-Definition individuell abändern, um zum Beispiel für individuelle Hosts Backups nur für individuelle Zeiten durchzuführen. Dieses Vollbackup würde bereits über den Scheduler regelmäßig ausgeführt werden, solange der Bareos Director gestartet ist.

Unabhängig davon können Sie über die `bconsole` mit dem Kommando `run` den Backupprozess für `BackupClient1` manuell anstoßen.

```
backup:~ # bconsole
Connecting to Director backup:9101
1000 OK: backup-dir Version: 13.2.2 (12 November 2013)
Enter a period to cancel a command.
*run
A job name must be specified.
The defined Job resources are:
    1: BackupClient1
    2: BackupCatalog
    3: RestoreFiles
Select Job resource (1-3): 1
Run Backup job
JobName: BackupClient1
Level: Incremental
Client: backup-fd
Format: Native
FileSet: Full Set
Pool: File (From Job resource)
Storage: File (From Job resource)
When: 2014-07-10 15:57:45
Priority: 10
OK to run? (yes/mod/no): yes
Job queued. JobId=1
You have messages.
```

#### Listing 7.57 Manuelles Starten eines Jobs

Ebenfalls über die `bconsole` können Sie den Status der aktuell laufenden Jobs kontrollieren:

```
*status jobs
Automatically selected Storage: File
Connecting to Storage daemon File at backup:9103

backup-sd Version: 13.2.2 (12 November 2013) x86_64-suse-linux-gnu suse \
openSUSE 13.1 (Bottle) (x86_64)
Daemon started 10-Jul-14 15:53. Jobs: run=33, running=0.
Heap: heap=270,336 smbytes=308,312 max_bytes=312,869 bufs=125 max_bufs=138
Sizes: boffset_t=8 size_t=8 int32_t=4 int64_t=8 mode=0 bwlimit=0kB/s

Running Jobs:
Writing: Incremental Backup job BackupClient1 JobId=34 Volume="test"
      pool="File" device="FileStorage" (/home/bareos/storage)
```



```

spooling=0 despooling=0 despool_wait=0
Files=0 Bytes=0 AveBytes/sec=0 LastBytes/sec=0
FDReadSeqNo=7 in_msg=6 out_msg=5 fd=5

```

### Listing 7.58 Den Status eines Jobs überprüfen

#### Wiederherstellen von Dateien aus dem Backup

Wenn Sie Dateien aus einem Backup wiederherstellen möchten, nutzen Sie in der `bconsole` zunächst das Kommando `restore`, um die richtige Sicherung vom richtigen Zeitpunkt auszuwählen:

##### **\*restore client=backup-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

The defined FileSet resources are:

- 1: Catalog
- 2: Full Set

Select FileSet resource (1-2): 2

JobId	Level	JobFiles	JobBytes	StartTime	VolumeName
10	F	373,443	13,493,754,496	2014-07-10 23:05:03	test
19	D	268	659,590,404	2014-07-13 23:05:02	test
22	I	132	480,770,461	2014-07-14 23:05:03	test
25	I	127	495,149,515	2014-07-15 23:05:02	test
28	I	157	491,662,830	2014-07-16 23:05:02	test

```
| 31 | I | 895 | 992,323,574 | 2014-07-17 23:05:03 | test |
+-----+-----+-----+-----+-----+-----+-----+
You have selected the following JobIds: 10,19,22,25,28,31
```

```
Building directory tree for JobId(s) 10,19,22,25,28,31 ...
+++++
334,797 files inserted into the tree.
```

You are now entering file selection mode where you add (mark) and remove (unmark) files to be restored. No files are initially added, unless you used the "all" keyword on the command line.  
Enter "done" to leave this mode.

```
cwd is: /
$
```

#### Listing 7.59 Dateien aus einem Bareos-Backup wiederherstellen

Sie können sich nun analog zu den Linux-Kommandos über `cd` und `ls` in diesem Backuparchiv virtuell bewegen.

Alle Dateien (oder Verzeichnisse), die Sie wiederherstellen wollen, können Sie über das Kommando `mark` markieren – sie werden anschließend im Dateilisting mit einem Sternchen markiert, wie hier bei `test` zu sehen:

```
$ markdir home
1 directory marked.
$ mark test
1 file marked.
$ ls
dev
etc/
+home/
lib/
*test
[...]
```

#### Listing 7.60 Zur Wiederherstellung markierte Dateien

Haben Sie selektiert, welche Dateien Sie aus dem Backup zurückspielen wollen, können Sie den Dateibrowser mit dem Kommando `done` verlassen und den Startschuss zur Wiederherstellung geben.

```
$ done
Bootstrap records written to /var/lib/bareos/backup-dir.restore.1.bsr
```

The job will require the following

Volume(s)	Storage(s)	SD Device(s)
test	File	FileStorage

Volumes marked with "\*" are online.

1 file selected to be restored.

Automatically selected Client: backup-fd

Run Restore job

```

JobName:      RestoreFiles
Bootstrap:    /var/lib/bareos/backup-dir.restore.1.bsr
Where:        /tmp/bareos-restores
Replace:      Always
FileSet:      Full Set
Backup Client: backup-fd
Restore Client: backup-fd
Format:       Native
Storage:      File
When:         2014-07-10 16:09:19
Catalog:     MyCatalog
Priority:      10
Plugin Options: *None*
OK to run? (yes/mod/no): yes
Job queued. JobId=2

```

#### Listing 7.61 Die Wiederherstellung der Dateien durch Bareos

Das eigentliche Wiederherstellen kann (je nach Datenmenge) länger dauern und findet darum als separater Job im Hintergrund statt.

```
*status jobs
```

```
Automatically selected Storage: File
```

```
Connecting to Storage daemon File at backup:9103
```

```
backup-sd Version: 13.2.2 (12 November 2013) x86_64-suse-linux-gnu suse
```

```
openSUSE 13.1 (Bottle) (x86_64)
```

```
Daemon started 10-Jul-14 15:53. Jobs: run=2, running=0.
```

```
Heap: heap=270,336 smbytes=162,986 max_bytes=236,613 bufs=80 max_bufs=107
```

```
Sizes: boffset_t=8 size_t=8 int32_t=4 int64_t=8 mode=0 bwlimit=0kB/s
```

```
Running Jobs:
```

No Jobs running.

====

Jobs waiting to reserve a drive:

====

Terminated Jobs:

JobId	Level	Files	Bytes	Status	Finished	Name
1	Full	670	56.67 M	OK	10-Jul-14 15:58	BackupClient1
2		0	0	OK	10-Jul-14 16:09	RestoreFiles

[...]

Logfile:

10-Jul 16:09 backup-dir JobId 2: Bareos booster-dir 13.2.2 (12Nov13):

```
Build OS:          x86_64-suse-linux-gnu suse
                  openSUSE 13.1 (Bottle) (x86_64)
JobId:            2
Job:              RestoreFiles.2014-07-10_16.09.26_05
Restore Client:   backup-fd
Start time:       10-Jul-2014 16:09:28
End time:         10-Jul-2014 16:09:29
Files Expected:   1
Files Restored:   1
Bytes Restored:   31,160
Rate:             31.2 KB/s
FD Errors:        0
FD termination status: OK
SD termination status: OK
Termination:      Restore OK
```

**Listing 7.62** Die Wiederherstellung von Dateien als Job im Hintergrund

Ist alles erfolgreich durchlaufen, haben Sie in `/tmp/bareos-restores` die zurückgespielten Dateien.

### Regelmäßige Backups einrichten

Welche Backups in welchen Intervallen derzeit aktiv sind, können Sie über das `bconsole`-Kommando `status` in Erfahrung bringen, wenn Sie dort den Status Scheduler abfragen. Bareos gibt Ihnen dabei auch gleich einen kalendarischen Ausblick auf die nächsten sieben Tage:

**\*status**

Status available for:

- 1: Director
- 2: Storage
- 3: Client
- 4: Scheduler
- 5: All

Select daemon type for status (1-5): 4

Scheduler Jobs:

```

Schedule                Jobs Triggered
=====
WeeklyCycle
                        BackupClient1

WeeklyCycleAfterBackup
                        BackupCatalog

====

```

Scheduler Preview for 7 days:

```

Date                Schedule                Overrides
=====
Thu 10-Jul-2014 23:05 WeeklyCycle                Level=Incremental
Thu 10-Jul-2014 23:10 WeeklyCycleAfterBackup    Level=Full
Fri 11-Jul-2014 23:05 WeeklyCycle                Level=Incremental
Fri 11-Jul-2014 23:10 WeeklyCycleAfterBackup    Level=Full
Sat 12-Jul-2014 23:05 WeeklyCycle                Level=Incremental
Sat 12-Jul-2014 23:10 WeeklyCycleAfterBackup    Level=Full
Sun 13-Jul-2014 23:05 WeeklyCycle                Level=Differential
Sun 13-Jul-2014 23:10 WeeklyCycleAfterBackup    Level=Full
Mon 14-Jul-2014 23:05 WeeklyCycle                Level=Incremental
Mon 14-Jul-2014 23:10 WeeklyCycleAfterBackup    Level=Full
Tue 15-Jul-2014 23:05 WeeklyCycle                Level=Incremental
Tue 15-Jul-2014 23:10 WeeklyCycleAfterBackup    Level=Full
Wed 16-Jul-2014 23:05 WeeklyCycle                Level=Incremental
Wed 16-Jul-2014 23:10 WeeklyCycleAfterBackup    Level=Full
====

```

**Listing 7.63** Wissen, was läuft: Bareos kann in die Zukunft blicken.

Wann welche Jobs laufen, ist in der *bareos-dir.conf* in den verschiedenen Schedule-Ressourcen definiert, die Sie auch kopieren und unter eigenem Namen anpassen können.

```
# When to do the backups, full backup on first sunday of the month,
# differential (i.e. incremental since full) every other sunday,
# and incremental backups other days
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
}
```

**Listing 7.64** Die Sektion »WeeklyCycle« kombiniert sinnvoll verschiedene Backupmethoden.

Das Beispiel `WeeklyCycle` führt demnach aus:

1. ein Vollbackup an jedem ersten Sonntag im Monat,
2. differenzielle Backups zu diesem Vollbackup an jedem anderen Sonntag und
3. täglich fortlaufend kleine inkrementelle Backups.

Dies ist an sich ein guter Rhythmus, um Schnelligkeit, Sicherheit und Effizienz unter einen Hut zu bekommen. Sofern Sie keine besonderen Ansprüche haben, sollten Sie diesen Modus kurzerhand beibehalten.

## 7.7.2 Einrichtung eines zu sichernden externen Clients

Auf den Clients, die Sie durch Bareos sichern möchten, müssen Sie ebenfalls die oben genannten Bareos-Repositorys einrichten und daraus den *linux file daemon* installieren, den Sie im Paket `bareos-fd` (openSUSE) bzw. `bareos-filedaemon` (Ubuntu) finden. Für Windows-Clients können Sie auf die Pakete *winbareos* zurückgreifen:

<http://download.bareos.org/bareos/release/latest/windows/>

Damit Client und Server sich verstehen, muss der Bareos-Client den Namen des Directors wissen, dem Zugriff gewährt werden soll. Außerdem werden sich Client und Director über ein gemeinsames Passwort autorisieren, das Sie hier schon einmal hinterlegen sollten.

Passen Sie darum in der Datei `/etc/bareos/bareos-fd.conf` die Ressource Director an, und tragen Sie als Namen den schon in der `bareos-dir.conf` verwendeten Director-Namen sowie das Passwort, das Sie später auch in der Clientressource verwenden, ein.

```
# List Directors who are permitted to contact this File daemon
Director {
  Name = backup-dir
  Password = "MfYsw3t+uqi2kotDR56ZeEf3nSh411R7SPDokhaMq6vU"
}
```

**Listing 7.65** Autorisierung des Directors nebst Passwort auf dem Client

## Einrichten von Jobs zur Sicherung der Clients

Für Ihren neuen Client müssen Sie nun auf dem Director einen Job definieren, damit der Client in die normale Sicherung mit aufgenommen wird. Geben Sie ihm einen aussagekräftigen Namen, benennen Sie den Namen des Clients wie in dessen Konfiguration *bareos-fd.conf*, und verweisen Sie auf die Standard-Job-Definition. Achtung: Vermeiden Sie stets Leerzeichen in den verwendeten Namen!

```
Job {
    Name = "Akademie-FullbackupWeekly"
    Client = akademie-fd
    JobDefs = "DefaultJob"
}

# Client (File Services) to backup
Client {
    Name = akademie-fd
    Address = 192.168.1.20 # FQDN sollte bevorzugt werden wenn möglich
    Password = "MfYsw3t+uqi2kotDR56ZeEf3nSh4l1R7SPDokhaMq6vU" # password for FD
    File Retention = 30 days          # 30 days
    Job Retention = 6 months          # six months
    AutoPrune = no                    # Prune expired Jobs/Files
}
```

### Listing 7.66 Einrichtung eines Jobs für den neuen Client

Prüfen Sie, ob Ihre Konfigurationsdateien syntaktisch korrekt sind, und lassen Sie dann den Bareos Director die neuen Konfigurationen einlesen.

```
root@bareos:~ # bareos-dir -t -c /etc/bareos/bareos-dir.conf
root@bareos:~ # service bareos-dir reload
```

### Listing 7.67 Prüfen der Konfiguration und Neustart des Directors

Ihr zusätzlicher neuer Rechner muss nun auch im Schedule auftauchen, den Sie wie gehabt über die *bconsole* abfragen können.

```
*status schedule
```

```
Scheduler Jobs:
```

```
Schedule          Jobs Triggered
=====
WeeklyCycle
                  BackupClient1
                  Akademie-FullbackupWeekly
```

```
WeeklyCycleAfterBackup
BackupCatalog
```

**Listing 7.68** Der neue Job ist im Schedule »WeeklyCycle« eingeplant.

Über das `bconsole`-Kommando `status client=<hostname>` können Sie prüfen, ob die Netzwerkkommunikation zum Client reibungslos funktioniert. Ist alles in Ordnung, können Sie nun auf das Eintreten des nächsten Schedule-Zeitpunkts warten oder wie gehabt über das `bconsole`-Kommando `run` einen Lauf außer der Reihe initiieren.


```
backup:~ # bconsole
*run job=Akademie-FullbackupWeekly
*status jobs
```

Running Jobs:

```
Backup Job Akademie_FullbackupWeekly.2014-07-10_16.37.07_05 waiting for
Client connection.
```

```
Writing: Full Backup job Akademie_FullbackupWeekly JobId=7 Volume="test"
  pool="File" device="FileStorage" (/home/bareos/storage)
  spooling=0 despooling=0 despool_wait=0
  Files=0 Bytes=0 AveBytes/sec=0 LastBytes/sec=0
  FDSocket closed
```

**Listing 7.69** Manuelles Starten eines Backups vom neu eingerichteten Client

Server und Client müssen beide in der Lage sein, eine Verbindung zum jeweils anderen Partner auf den TCP-Ports 9101, 9102 und 9103 aufzubauen. Ist eine Firewall zwischen Ihren Systemen aktiv, müssen Sie diese Ports freischalten (siehe Tabelle 7.1)! 

Quelle	Ziel	Zielport
bconsole	Director	9101
Director	File-Daemon	9102
Director	Storage-Daemon	9103
File-Daemon	Storage-Daemon	9103

**Tabelle 7.1** Die für Bareos-Dienste verwendeten TCP-Ports

### 7.7.3 Sichern von MySQL und LDAP

Die Sicherung von Datenbanken wie MySQL oder von Verzeichnisdiensten wie LDAP ist immer eine etwas heikle Sache. Sie können sich sicher ausmalen, wie Ihr Backup aussieht, wenn



Index- und Datenbankdateien inhaltlich verschieden sind, weil noch während des Sicherungsprozesses in die Datenbank geschrieben worden ist.

Wenn Sie zum Zeitpunkt des Backups Ihren Datenbank- oder LDAP-Server nicht kurzerhand stoppen können, sollten Sie solche Dienste darum mit den dafür zur Verfügung stehenden Tools zunächst in normale externe Dateien exportieren lassen; Tools wie `mysqldump` kümmern sich darum, dass dieser Export auch dann inhaltlich konsistent ist, wenn er während des laufenden Betriebs geschehen ist.

Ihr Backup wird diese Exportdateien dann wie jede andere normale Datei mit in die Sicherung übernehmen. Im Disaster-Fall müssen Sie zunächst den Rechner wiederherstellen und im zweiten Schritt gegebenenfalls aus diesem Dump heraus die Datenbank wiederherstellen, beispielsweise durch `mysqlimport`. Am besten ist es, wenn Sie den Export Ihrer Datenbank durch ein kleines Shell-Skript automatisieren und dieses dann in der Job-Ressource in *bareos-dir.conf* als `RunBeforeJob` ausführen lassen.

```
# Safely dump this database
RunBeforeJob = "mysqldump --all-databases > /var/lib/mysql/mysqldump.dat"
# This deletes the dump afterwards
RunAfterJob = "rm /var/lib/mysql/mysqldump.dat"
```

**Listing 7.70** Kurz vor der Datensicherung stößt Bareos einen Export der MySQL-Datenbank an.

#### 7.7.4 Bareos und ReaR als Dreamteam

Nachdem wir in diesem Kapitel zuerst ReaR als Disaster-Recovery-System und nun Bareos als Backuplösung kennengelernt haben, kostet es jetzt nur noch wenig Mühe, beide Lösungen zum Dreamteam miteinander zu vereinen.

1. Installieren Sie Bareos auf Ihrem Linux-Client wie in diesem Kapitel gezeigt, und prüfen Sie, ob die normale Datensicherung funktioniert.
2. Installieren Sie dann ReaR, und richten Sie Bareos als Backupmethode ein.

```
# Bootfähigen USB-Stick erzeugen
OUTPUT=USB
OUTPUT_URL="usb:///dev/disk/by-label/REAR-000"
# Backup von Bareos vornehmen lassen
BACKUP=BAREOS
```

**Listing 7.71** Disaster Recovery mit ReaR und Bareos

3. Sorgen Sie nun über das Kommando `rear mkrescue` dafür, dass ReaR ein Rescue-Image erzeugt, das zugleich auch Bareos mitsamt lauffähiger Konfiguration beinhaltet. Im Disaster-Fall können Sie dieses Image booten, damit ReaR einerseits das Grundsystem wiederherstellt und andererseits Bareos zum Zurückspielen der Daten veranlasst.

### 7.7.5 Bacula/Bareos für Fortgeschrittene

Im Rahmen dieses Buches können wir bei Bacula/Bareos leider nur an der Oberfläche kratzen und Ihnen ein grundlegendes Verständnis geben, wie diese Komponenten ineinandergreifen und zu bedienen sind. Themen wie die Einbindung echter Tape-Libraries oder komplexere Backupmechanismen wie *Backup2Disk2Tape* sprengen hier den Rahmen. Aber das ist nicht weiter schlimm: In den Konfigurationsdateien sind bereits viele Beispiele vorbereitet, und auf <http://doc.bareos.org/master/html/bareos-manual-main-reference.html> finden Sie eine umfangreiche Online-Referenz zur weiteren Vertiefung. Auch das von dem Bareos-Autor Philipp Storz geschriebene Buch »Bacula: Backup-Strategien und -Lösungen im Netzwerk« (ISBN 978-3-941841-41-3) ist natürlich eine für Bacula und Bareos gleichermaßen gut geeignete, tief gehende Lektüre zum Thema.