

¹Backup mit Bacula - strange version

Axel S. Gruner

17.08.2006

Inhaltsverzeichnis

1	Einleitung	2
1.1	Backupserver	2
1.2	Umgebung	2
1.2.1	Server	2
1.2.2	Clients	2
1.2.2.1	SuSE Linux Enterprise + Vmware Server + WindowsXP	2
1.3	Was ist Bacula?	3
2	Backup	4
2.1	Rsync	4
2.1.1	Script	4
2.2	Bacula	4
2.2.1	Backupintervall	5
2.2.2	Verfall der Bänder	5
2.2.3	Beschriftung der Bänder	5
2.2.4	Handhabung der Bänder	5
2.2.5	Umgang mit Bacula beim Bandwechsel	6
2.2.6	Restore von Dateien	6
3	Installation	7
3.1	Wo wird was installiert?	7
3.2	Die PostgreSQL Datenbank	8
3.3	FreeBSD als Bacula-Client	9
3.4	SuSE Linux als Bacula-Client	9
3.4.1	Installation von Bacula unter SuSE Linux	9
3.4.1.1	Wo wird was installiert?	9
3.5	Windows 2000 als Bacula-Client	10
3.5.1	Installation von Bacula unter Windows2000	10
3.5.1.1	Wo wird was installiert	10
4	Konfiguration	11
4.1	Konfiguration File-Daemon (bacula-fd.conf)	11
4.1.1	Konfiguration Storage-Daemon (bacula-sd.conf)	12
4.1.2	Konfiguration des Director (bacula-dir.conf)	12
4.2	Windows Bacula-Client Konfiguration und Start	21

Kapitel 1

Einleitung

Achtung. Beware of dog. Slippery when wet. Use at your own risk.

Soll heissen, dies ist eine schnelle Zusammenfassung von Bacula unter FreeBSD 6.1, und ist hauptsächlich auf mein Gedächtnis gestützt.

1.1 Backupserver

IP	10.4.1.243
Name	sfbbackup.example.com
Dienste	rsync, bacula-server, bacula-client

1.2 Umgebung

Die Infrastruktur der Server und Clients variiert immer. Daher kann diese beschriebene Umgebung nur als Beispiel dienen und muss entsprechend den Gegebenheiten angepasst werden.

1.2.1 Server

1. **Backupserver** auf dem Bacula läuft und ein LTO2 Laufwerk angeschlossen ist (200/400GB). IBM x345 Server mit RAID1.
2. **Domainserver** mit Samba auf dem sich alle Benutzer anmelden und ihre Daten vorhanden sind

1.2.2 Clients

Alle Clients (bis auf eine Ausnahme) laufen unter WindowsXP. Die Benutzer sind nur auf dem Domaincontroller angelegt und melden sich ausschliesslich an der Domäne an. Daher liegen alle Daten der Benutzer auf dem Domainserver.

1.2.2.1 SuSE Linux Enterprise + Vmware Server + WindowsXP

An die 100 WindowsXP Clients alleine zu verwalten, Probleme zu beheben, updates einzuspielen oder von Viren/Würmer zu befreien, möchte niemand. Daher hat man sich für eine virtuelle Umgebung entschieden.

Dabei wurde auf einem Referenz PC SuSE Linux Enterprise 10 installiert. Dort dann Vmware Server Edition und in der Vmware WindowsXP. Ich möchte dabei nicht auf die Einzelheiten eingehen, dies würde den Umfang dieses Howtos sprengen. WindowsXP wurde auf dem Referenz PC mit der Standard Software eingerichtet (Office, Powerterm, Notes,...). Dieses Image kann nun auf die anderen PCs, auf denen die SuSE Linux Enterprise Version aufgespielt wurde (inkl. Vmware), kopiert werden. Jeder PC bootet SuSE Linux, die Anmeldung mit einem Benutzer erfolgt automatisch, danach wird sofort die Vmware gestartet und in dieser WindowsXP. Der Anwender selbst bekommt dies alles nicht wirklich mit und glaubt an einem regulären WindowsXP PC zu sitzen.

Zwar hat jeder Benutzer nur eingeschränkte Rechte, aber Benutzer sind oftmals sehr kreativ. Ausserdem kümmern sich Viren und Würmer um diese Rechte nicht wirklich. Der Vorteil dieser Installationsart ist nun, das auf dem Referenz PC updates von Microsoft heruntergeladen werden können, einem Test unterzogen werden und danach das neue Vmware Image in der Nacht auf die Clients verteilt werden kann.

1.3 Was ist Bacula?

Bacula ist ein Open-Source Backup-Tool. Die meisten mögen bei "Backup" zuerst an "Amanda" denken, ich denke aber das "Bacula" (<http://www.bacula.org>) mindestens einen Blick Wert ist (wenn nicht mehr).

Die Features sind:

- Viele Plattformen (*BSD, Linux, Windows, Solaris, MacOSX)
- Es kann auf Band oder Festplatte gesichert werden
- Der Scheduler ist äusserst variabel (verschiedene Jobs zu verschiedenen Zeiten)
- Backups können über mehrere Bänder gehen
- Jede Datei bekommt (optional) einen Fingerprint
- Backups können gepackt oder ungepackt gemacht werden
- Autochanger werden unterstützt

Kapitel 2

Backup

Das Backup der Daten der Clients teilt sich in zwei Bereiche auf. Zum einen wird jede Nacht ein Backup der Daten auf Band gemacht, zum anderen werden die Daten alle 10 Minuten vom Domainserver auf den Backupserver abgeglichen. Dabei kommt "rsync" zum Einsatz.

2.1 Rsync

Über einen Cronjob wird alle 10 Minuten ein Script aufgerufen, welches eine synchronisation (rsync) mit dem Domainserver (10.4.1.245) ausführt. Dabei werden die Dateien in den Userverzeichnissen, die nicht mehr auf dem Domainserver bestehen, auf dem Backupserver nicht überschrieben bzw. gelöscht.

```
Script    /usr/local/backup/rsync.sh
crontab  */10 7-20 * * * root /usr/backup/rsync.sh > /dev/null 2>&1
```

2.1.1 Script

```
#!/bin/sh
/usr/local/bin/rsync -e "ssh -p 2222" -avzu ?exclude-
from=/usr/local/backup/exclude-dir \ --
exclude '*.core*' 10.4.1.245:/home/ /usr/home/
```

Rsync läuft dabei über SSH und Port 2222, auf dem sich root ohne Kennwort vom Backupserver auf dem Domainserver einloggen kann. Hierzu wurde eine zweite SSHD Instanz auf dem Domainserver konfiguriert und ein Schlüssel erstellt. Eklig...

2.2 Bacula

Bacula erstellt die Backups auf das angeschlossene LTO2 Laufwerk und folgende Konfigurationsdateien sind dabei wichtig. Dabei beziehe ich mich auf die Pfade wie unter FreeBSD. Bei anderen System sind die Pfade evtl. anders, unter Windows lässt sich der Bacula Client auf der GUI konfigurieren und als Dienst direkt starten.

Server

Configs /usr/local/etc/bacula-dir.conf
 /usr/local/etc/bacula-sd.conf
 /usr/local/etc/bacula-fd.conf

Client

Config /usr/local/etc/bacula-fd.conf

2.2.1 Backupintervall**Backupintervall**

Inkrementell Dienstag bis Freitag
 Full-Backup Woche Jeden zweiten bis fünften Samstag im Monat
 Full-Backup Monat Jeden ersten Samstag im Monat

2.2.2 Verfall der Bänder**Verfall der Bänder**

Inkrementell Nach 6 Tagen
 Full-Backup Woche Nach 30 Tagen
 Full-Backup Monat Nach 365 Tagen

2.2.3 Beschriftung der Bänder**Beschriftung**

Inkrementell Daily-Tue, Daily-Wed, Daily-Thu, Daily-Fri
 Full-Backup Woche Weekly-2, Weekly-3, Weekly-4, Weekly-5
 Full-Backup Monat Monthly-January,..., Monthly-December

2.2.4 Handhabung der Bänder**Inkrementell**

Montagabend Daily-Tue
 Dienstagabend Daily-Wed
 Mittwochabend Daily-Thu
 Donnerstagabend Daily-Fri

Full-Backup Woche

Zweite Woche Weekly-2
 Dritte Woche Weekly-3
 Vierte Woche Weekly-4
 Fünfte Woche Weekly-5

Full-Backup Monat

Freitagabend Monthly-*

Anmerkung zu Full-Backup Woche:

Jeden Freitagabend das entsprechende Weekly Band einlegen, ausser es handelt sich beim Folgetag (Samstag) um den ersten Samstag im Monat!

Anmerkung Full-Backup Monat:

Nur am Freitagabend das entsprechende Monthly Band einlegen wenn der Folgetag (Samstag) der erste Samstag im neuen Monat ist. Dabei wird immer

das Band, welches mit dem Vormonat beschriftet ist, eingelegt (am ersten Samstag im Juli also das Band Monthly-June)

2.2.5 Umgang mit Bacula beim Bandwechsel

- Login auf Bacula Server
- **bconsole** eingeben
- Danach sollte das Prompt nur einen * anzeigen
- **messages** eingeben und nachsehen ob das Backup ok war
- **umount** eingeben (Band wird aus bacula entfernt)
- **exit** eingeben (nun ist man wieder auf der Unix Shell)
- **camcontrol eject sa0** eingeben und warten bis das Band entladen wird
- Neues Band einlegen (siehe 2.2.4)
- **bconsole** eingeben
- Danach sollte das Prompt wieder nur ein * zeigen
- **mount** eingeben
- Es sollte die Meldung kommen, dass das Band mit dem Namen, entsprechend der Aufschrift auf dem Tape, geladen wurde.
- **exit** eingeben

2.2.6 Restore von Dateien

Dabei kann man die wx-console (welche mit dem bacula-client installiert wurden) direkt auf dem Backupserver starten. Oder aber, so man keine GUI auf dem Backupserver hat aber wx-console gerne nutzen möchte, man installiert diese auf seinem PC und verbindet sich dann mit dem Backupserver über die console.

- Karteikarte **Restore** auswählen
- Entsprechend der zu restorenden Daten die Auswahl treffen und Anweisungen befolgen (Welches Band bacula braucht,...)

Kapitel 3

Installation

Für Bacula gibt es unter FreeBSD zwei Ports:

- `/usr/ports/sysutils/bacula-server`
- `/usr/ports/sysutils/bacula-client`

Beide Ports sind dabei auch auf dem Backuoerver zu installieren. Dabei sollten bei `bacula-server` die Optionen “PostgreSQL” und “OpenSSL”, beim Bacula-Client die Optionen “wx-console” und “docs”, gewählt werden.

```
#cd /usr/ports/sysutils/bacula-server
#make install
#cd /usr/ports/sysutils/bacula-client
#make install
```

Wie üblich werden dabei die Startskripte unter `/usr/local/etc/rc.d` installiert und müssen über einen Eintrag in `/etc/rc.conf` aktiviert werden, dass die Dienste nach einem reboot automatisch starten. Dabei wird auch eine Gruppe “bacula” und ein Benutzer “bacula” auf dem System angelegt.

3.1 Wo wird was installiert?

Die Konfigurationsdateien von Bacula sind unter `/usr/local/etc/` zu finden.

```
%ls -l /usr/local/etc/bacula*
/usr/local/etc/bacula-barcodes
/usr/local/etc/bacula-dir.conf
/usr/local/etc/bacula-fd.conf
/usr/local/etc/bacula-sd.conf
```

Bacula ist eine Client-Server Backuplösung die in verschiedene Konfigurationsdateien unterteilt ist:

- “bacula-dir.conf” ist die Konfigurationsdatei des sogenannten “Director” und die komplexeste von allen. In dieser Konfigurationsdatei sind die Clients angegeben und deren Dateien von denen ein Backup gemacht werden soll. Der “bacula-dir” Daemon hält eine Verbindung zu den Clients (bacula-fd) und zum Backup-Medium (bacula-sd).

- Der “storage daemon” (bacula-sd) ist für die Kommunikation mit dem Backup-Medium zuständig. In der dazugehörigen Konfigurationsdatei “bacula-sd.conf” wird das Backup-Medium angegeben.
- Der “file daemon” (bacula-fd) muss auf jedem Rechner im Netzwerk laufen von dem ein Backup gemacht werden soll. Da in diesem Howto der Bacula-Backup-Server auch gleichzeitig als Client arbeiten soll, ist dieser Daemon und die Konfiguration nötig. Bei Rechnern im Netz die ausschliesslich als Client zu sehen sind muss nur dieser Daemon laufen.

Wie schon angesprochen gibt es 3 Daemons bei Bacula. Diese liegen unter:

```
%ls -l /usr/local/sbin/bacula-*
/usr/local/sbin/bacula-dir
/usr/local/sbin/bacula-fd
/usr/local/sbin/bacula-sd
```

3.2 Die PostgreSQL Datenbank

Wird Bacula installiert, so wird der Client 7.4 von PostgreSQL mitinstalliert. Diesen kann man nach der Installation von Bacula mit “pkg_deinstall” wieder löschen, um danach PostgreSQL 8.0 zu installieren.

```
#cd /usr/ports/database/postgresql80-server
#make install
```

Ist diese installiert, so muss die Datenbank initiiert werden, falls diese nicht schon auf dem System vorhanden war.

```
#!/usr/local/etc/rc.d/010.pgsql.sh initdb
```

oder

```
#su -l postgres -c initdb
```

Danach ist der User “root” in PostgreSQL anzulegen:

```
#su postgres
#!/usr/local/bin/createuser root
```

Ist dies geschehen muss die Bacula DB noch angelegt werden:

```
#cd /usr/local/share/bacula
#./create_bacula_database
#./make_bacula_tables
#./grant_bacula_privileges
```

Danach kann man sich die Rechte genauer ansehen:

```
#psql --command \dp bacula
```

Den Eintrag für den Start von PostgreSQL in der /etc/rc.conf nicht vergessen.

3.3 FreeBSD als Bacula-Client

Auf einem FreeBSD Client, welcher auch in den Pool der zu sichernden Client aufgenommen werden soll, ist der Port

```
#/usr/ports/sysutils/bacula-client
```

zu installieren.

3.4 SuSE Linux als Bacula-Client

Unter anderen Linux Distributionen sollte die nachstehende Installation ähnlich verlaufen.

Da Bacula hier nur als Client eingesetzt werden soll, reicht es aus auch nur dem Bacula “file-daemon” zu installieren.

3.4.1 Installation von Bacula unter SuSE Linux

Das benötigte RPM Paket gibt es unter http://sourceforge.net/project/showfiles.php?group_id=50727 und heisst:

```
bacula-client-1.32f-5.i586.su9.rpm
```

Sollte der direkte Link nicht mehr funktionieren, so ist der Link auch über <http://www.bacula.org/> und dort dem Link “Current Downloads” zu erreichen.

Die Installation erfolgt dann über

```
rpm -ihv bacula-client-1.32f-5.i586.su9.rpm
```

Bei mir kam es in diesem Zusammenhang zu Problemen. So fehlten die Abhängigkeiten “termcap” und “libtermcap.so.2”. Beide kann man als RPM unter <http://rpmseek.com/> finden:

```
libtermcap2-2.0.8-35mdk.i586.rpm
termcap-2.0.8-674.i586.rpm
```

3.4.1.1 Wo wird was installiert?

Die Konfigurations- und Startdatei liegt bei SuSE Linux unter `/etc/bacula/`

```
% ls -l /etc/bacula/
bacula-fd.conf
fd
[...]
```

Die Start-up Datei “fd” startet den unter

```
% ls -l /usr/sbin/bac*
/usr/sbin/bacula-fd
```

liegenden File-Daemon.

3.5 Windows 2000 als Bacula-Client

Unter Windows wurde bisher nur der File-Daemon getestet was aber in diesem Howto vollkommen ausreichend ist, da der Windows Rechner nur ein Backup-Client darstellt. Der Windows File-Daemon läuft unter Win98, NT, Windows2000 und XP ohne Probleme.

Bacula sollte bei NT, Windows2000 und XP, in der Lage sein, die Benutzinformationen und Benutzerrechte bei einem restore der Dateien wieder richtig zu setzen. Dies war bei früheren Versionen nicht der Fall.

3.5.1 Installation von Bacula unter Windows2000

Das benötigte Paket gibt es unter http://sourceforge.net/project/showfiles.php?group_id=50727. Dabei sollte nicht die Beta genommen werden, sondern die regulären “binaries”:

```
winbacula-1.32f-4.tar.gz
```

Auch hier, sollte der Link nicht direkt funktionieren, so kann man über <http://www.bacula.org> und dem Link “Current Downloads” auch zum Ziel gelangen. Ist die Datei auf dem lokalen Windows Rechner wie folgt vorgehen:

- Die Datei mit WinZIP öffnen
- Den Inhalt direkt nach C:/ extrahieren (es wird ein Ordner “bacula” angelegt, der die Unterordner “bin” und “working” beinhaltet)
- Unter C:/bacula/bin/ die Datei “bacula-fd.conf.new” in “bacula.fd.conf” umkopieren
- Die Datei “Install” unter C:/bacula/bin/ ausführen (diese legt einen System Service an, so dass bacula beim Neustart des System immer automatisch gestartet wird).

3.5.1.1 Wo wird was installiert

Bei Windows liegen die Dateien alle unter von

```
C:/bacula/
```

Alles weitere zu den Dateien und dem Start des Dienstes, wird im nächsten Kapitel, der Konfiguration, besprochen.

Kapitel 4

Konfiguration

Das Gros der Konfiguration ist in der Datei `bacula-dir.conf` auf dem Backupserver zu erledigen. Der File-Daemon auf den Clients ist schnell konfiguriert, ebenso der Storage Daemon auf dem Backupserver.

4.1 Konfiguration File-Daemon (`bacula-fd.conf`)

```
Director {
  (1)Name = backup
  (2>Password = "passwort"
}
FileDaemon {
  (3)Name = sfapp0-fd
  (4)FDport = 9102
  WorkingDirectory = /var/db/bacula
  Pid Directory = /var/run
}
Messages {
  Name = Standard
  (5)director = backup = all, !skipped
}
```

Alle Clients im Netz sind identisch zu konfigurieren. Dabei ist nur darauf zu achten, dass diese eigene Namen, wie unter (3) angegeben, bekommen, und diese Namen müssen auch entsprechend in der `bacula-dir.conf` auf dem Backupserver vorhanden sein. Im unteren Beispiel der `bacula-dir.conf` sind mehrere Clients eingetragen.

Die Zahlenangaben vor den Zeilen dienen nur der Erklärung und dürfen bei der eigentlichen Konfiguration nicht angegeben werden.

- (1) Dies ist der Name des "Director", welcher in der "bacula-dir.conf" angegeben ist
- (2) Das Passwort wird bei der Installation von Bacula selbst generiert und muss nicht geändert werden
- (3) Dies ist der Name des File-Daemon welcher noch beim "Director" in der "bacula-dir.conf" eine Rolle spielt
- (4) Dies ist Port auf den gelauscht wird und der Director Verbindung mit auf nimmt

- (5) Sendet alle Nachrichten, bis auf die bei übersprungenen Dateien, an den "Director"

4.1.1 Konfiguration Storage-Daemon (bacula-sd.conf)

```
Storage {
(1)Name = FileStorage
(2)SDPort = 9103 # Director's port
WorkingDirectory = "/var/db/bacula"
Pid Directory = "/var/run"
}
Director {
(3)Name = backup
(4)Password = "passwort"
}
Device {
Name = LT02
Description = "LT02"
(5)Media Type = LT02
LabelMedia = yes;                # lets Bacula label unlabeled me-
dia
(6)Archive Device = /dev/sa0
AutomaticMount = yes;           # when device opened, read it
AlwaysOpen = yes
Offline On Unmount = no
Hardware End of Medium = no
BSF at EOM = yes
Backward Space Record = no
Fast Forward Space File = no
TWO EOF = yes
}
Messages {
Name = Standard
director = sfbackup-dir = all
}
```

- (1) Name des Storage-Daemon (wichtig noch in "bacula-dir.conf")
- (2) Port auf den auf den "Director" gelauscht wird
- (3) Name des "Director"
- (4) Passwort welches an den "Director" übertragen werden muss (wird bei der Installation automatisch erstellt)
- (5) Auf welchen Medien-Typ das Backup erstellt wird.
- (6) Auf welches Device das Backup erstellt wird.

4.1.2 Konfiguration des Director (bacula-dir.conf)

```
Director { # define myself
Name = backup
DIRport = 9101 # where we listen for UA connections
QueryFile = "/usr/local/share/bacula/query.sql"
WorkingDirectory = "/var/db/bacula"
PidDirectory = "/var/run"
Maximum Concurrent Jobs = 1
Password = "passwort" # Console password
```

```
Messages = Standard
}
#####
# Job Resource - Angaben zu den Clients / START #
#####
Job {
Name = "Client-Backup"
Type = Backup
Client = sfapp0-fd
FileSet = "Full Set Client"
Schedule = "Client-Backup"
Storage = ClientBackup
Messages = Standard
Pool = Default
Write Bootstrap = "/var/db/bacula/clientbackup.bsr"
Priority = 10
Max Start Delay = 22h
}
Job {
Name = "sfld2-Backup"
Type = Backup
Client = sfld2-fd
FileSet = "Full Set sfld2"
Schedule = "Client-Backup"
Storage = ClientBackup
Messages = Standard
Pool = Default
Write Bootstrap = "/var/db/bacula/sfld2backup.bsr"
Priority = 10
Max Start Delay = 22h
}
Job {
Name = "avenum-Backup"
Type = Backup
Client = avenum-fd
FileSet = "Full Set avenum"
Schedule = "Client-Backup"
Storage = ClientBackup
Messages = Standard
Pool = Default
Write Bootstrap = "/var/db/bacula/avenumbackup.bsr"
Priority = 10
Max Start Delay = 22h
}
Job {
Name = "sffwd0-Backup"
Type = Backup
Client = sffwd0-fd
FileSet = "Full Set sffwd0"
Schedule = "Client-Backup"
```

```
Storage = ClientBackup
Messages = Standard
Pool = Default
Write Bootstrap = "/var/db/bacula/sffwd0backup.bsr"
Priority = 10
Max Start Delay = 22h
}
Job {
Name = "sffwd2-Backup"
Type = Backup
Client = sffwd2-fd
FileSet = "Full Set sffwd2"
Schedule = "Client-Backup"
Storage = ClientBackup
Messages = Standard
Pool = Default
Write Bootstrap = "/var/db/bacula/sffwd2backup.bsr"
Priority = 10
Max Start Delay = 22h
}
Job {
Name = "sfbackup-Backup"
Type = Backup
Client = sfbackup-fd
FileSet = "Full Set sfbackup"
Schedule = "Client-Backup"
Storage = ClientBackup
Messages = Standard
Pool = Default
Write Bootstrap = "/var/db/bacula/sfbackupbackup.bsr"
Priority = 10
Max Start Delay = 22h
}
Job {
Name = "BackupCatalog"
Type = Backup
Client = sfbackup-fd
FileSet = "Catalog"
Schedule = "WeeklyCycleAfterBackup"
Storage = ClientBackup
Messages = Standard
Pool = Default
# This creates an ASCII copy of the catalog
RunBeforeJob = "/usr/local/share/bacula/make_catalog_backup ba-
cula bacula"
# This deletes the copy of the catalog
RunAfterJob = "/usr/backup/bacula/end_of_backup.sh"
Write Bootstrap = "/var/db/bacula/BackupCatalog.bsr"
Max Start Delay = 22h
}
```

```
#####  
# Job Resource - Angabe zu den Clients / ENDE #  
#####  
#####  
# Restore-Angaben /START #  
#####  
Job {  
  Name = "Restore Client"  
  Type = Restore  
  Client = sfapp0-fd  
  FileSet = "Full Set Client"  
  Storage = ClientBackup  
  Messages = Standard  
  Pool = Default  
}  
Job {  
  Name = "Restore sfld2"  
  Type = Restore  
  Client = sfld2-fd  
  FileSet = "Full Set Client"  
  Storage = ClientBackup  
  Messages = Standard  
  Pool = Default  
}  
Job {  
  Name = "Restore avenum"  
  Type = Restore  
  Client = avenum-fd  
  FileSet = "Full Set Client"  
  Storage = ClientBackup  
  Messages = Standard  
  Pool = Default  
}  
Job {  
  Name = "Restore sffwd0"  
  Type = Restore  
  Client = sffwd0-fd  
  FileSet = "Full Set Client"  
  Storage = ClientBackup  
  Messages = Standard  
  Pool = Default  
}  
Job {  
  Name = "Restore sffwd2"  
  Type = Restore  
  Client = sffwd2-fd  
  FileSet = "Full Set Client"  
  Storage = ClientBackup  
  Messages = Standard  
  Pool = Default
```

```

}
Job {
Name = "Restore sfbbackup"
Type = Restore
Client = sfbbackup-fd
FileSet = "Full Set Client"
Storage = ClientBackup
Messages = Standard
Pool = Default
}

```

```
#####
# Restore-Angaben /ENDE #
#####
#####
# Was soll gesichert werden /START #
#####
# List of files to be backed up

```

```
FileSet {
Name = "Full Set Client"
Include {
Options {
signature = MD5
}
File = /usr/home
File = /etc
File = /usr/local
}

```

```
Exclude {
File = /usr/home/foo/bar
File = /usr/home/LSfoo
File = /usr/home/ARCHIV
}
}

```

```
FileSet {
Name = "Full Set sfld2"
Include {
Options {
signature = MD5
}
File = /usr/local/notesdata
File = /opt/ibm
}
}

```

```
FileSet {
Name = "Full Set avenum"
Include {
Options {
signature = MD5
IgnoreCase = yes
}
}

```

```
File = "C:/MB00"
File = "C:/FP32"
File = "D:/avx"
}
}
FileSet {
Name = "Full Set sffwd0"
Include {
Options {
signature = MD5
}
File = /usr/local
File = /etc
File = /var/db
File = /var/named
File = /boot
File = /root
File = /usr/home
}
}
FileSet {
Name = "Full Set sffwd2"
Include {
Options {
signature = MD5
}
File = /usr/local
File = /etc
File = /var/db
File = /var/named
File = /boot
File = /usr/home
File = /root
}
}
FileSet {
Name = "Full Set sfbackup"
Include {
Options {
signature = MD5
}
File = /etc
File = /usr/local/etc
File = /var/db
File = /boot
File = /root
File = /usr/local/italive
File = /usr/local/pgsql
}
}
```

```

FileSet {
  Name = "Catalog"
  Include {
    Options {
      signature = MD5
    }
    File = /var/db/bacula/bacula.sql
  }
}

#####
# Was soll gesichert werden / ENDE #
#####
# Wann soll gesichert werden / START #
#####
Schedule {
  Name = "Client-Backup"
  Run = Level=Full Pool=Monthly 1st sat at 02:01
  Run = Level=Full Pool=Weekly 2nd-5th sat at 02:01
  Run = Level=Incremental Pool=Daily tue-fri at 02:01
}
# This does the catalog. It starts after the WeeklyCycle
Schedule {
  Name = "WeeklyCycleAfterBackup"
  Run = Level=Full Pool=Monthly 1st sat at 03:55
  Run = Level=Full Pool=Weekly 2nd-5th sat at 03:55
  Run = Level=Incremental Pool=Daily tue-fri at 03:55
}
#####
# Wann soll gesichert werden / ENDE #
#####
# File Daemon Password -
Dauer der Sicherungsverwahrung / START #
#####
Client {
  Name = sfapp0-fd
  Address = 10.4.1.245
  FDPort = 9102
  Catalog = MyCatalog
  Password = "passwort"
  File Retention = 30 days
  Job Retention = 6 months
  AutoPrune = yes
}
Client {
  Name = sfld2-fd
  Address = 10.4.1.60
  FDPort = 9102
  Catalog = MyCatalog

```

```
Password = "passwort"
File Retention = 30 days
Job Retention = 6 months
AutoPrune = yes
}
Client {
Name = avenum-fd
Address = 10.2.1.206
FDPort = 9102
Catalog = MyCatalog
Password = "passwort"
File Retention = 30 days
Job Retention = 6 months
AutoPrune = yes
}
Client {
Name = sffwd0-fd
Address = 10.4.1.250
FDPort = 9102
Catalog = MyCatalog
Password = "passwort"
File Retention = 30 days
Job Retention = 6 months
AutoPrune = yes
}
Client {
Name = sffwd2-fd
Address = 10.4.1.251
FDPort = 9102
Catalog = MyCatalog
Password = "passwort"
File Retention = 30 days
Job Retention = 6 months
AutoPrune = yes
}
Client {
Name = sfbackup-fd
Address = 10.4.1.243
FDPort = 9102
Catalog = MyCatalog
Password = "passwort"
File Retention = 30 days
Job Retention = 6 months
AutoPrune = yes
}
#####
# File Daemon Password -
Dauer der Sicherungsverwahrung / ENDE #
#####
#####
```

```

# Storage Daemon Adresse/Password/Sicherungstyp (auf Server) / START #
#####
Storage {
  Name = ClientBackup
  # IP des Storage Daemon auf dem Server
  Address = 10.4.1.243
  SDPort = 9103
  Password = "passwort"
  Device = LT02
  Media Type = LT02
}
#####
# Storage Daemon Adresse/Password/Sicherungstyp (auf Server) / ENDE #
#####
Catalog
{
  Name = MyCatalog
  dbname = bacula; user = bacula; password = ""
}
#####
# Nachrichten / START #
#####
Messages {
  Name = Standard
  mailcommand = "/usr/local/sbin/bsmtp -h localhost -
f \"\\(Bacula) %r\" -s \"Bacula: %t %e of %c %l\" %r"
  operatorcommand = "/usr/local/sbin/bsmtp -h localhost -
f \"\\(Bacula) %r\" -
s \"Bacula: Intervention needed for %j\" %r"
  mail = asg@suedfactoring.com = all, !skipped
  operator = asg@suedfactoring.com = mount
  console = all, !skipped, !saved
  append = "/var/db/bacula/log" = all, !skipped
}
#####
# Nachrichten / ENDE #
#####
#####
# Label-Format / START #
#####
Pool {
  Name = Default
  Pool Type = Backup
}
Pool {
  Name = Daily
  Pool Type = Backup
  AutoPrune = yes

```

```

# VolumeRetention = 6d # recycle in 10 days
VolumeRetention = 4d # recycle in 10 days
Maximum Volumes = 4
Recycle = yes
}
Pool {
Name = Weekly
Use Volume Once = yes
Pool Type = Backup
AutoPrune = yes
VolumeRetention = 30d # recycle in 30 days (default)
Recycle = yes
}
Pool {
Name = Monthly
Use Volume Once = yes
Pool Type = Backup
AutoPrune = yes
VolumeRetention = 365d # recycle in 1 year
Recycle = yes
}
#####
# Label-Format / ENDE #
#####

```

Der Client sfapp0-fd ist dabei der Domainserver.

4.2 Windows Bacula-Client Konfiguration und Start

Die Konfiguration erfolgt in der Datei "C:/bacula/bin/bacula-fd.conf":

```

Director
{
(1)Name = backup
(2>Password = "passwort"
}
FileDaemon
{
(3)Name = Client_2
(4)FDport = 9102
WorkingDirectory = /var/bacula
Pid Directory = /var/run
}
Messages
{
Name = Standard
(5)director = backup = all, !skipped
}

```

- (1) Name des "Director" auf dem Bacula-Backup-Server (siehe 3.1.3).
- (2) Das Passwort wird bei der Installation von Bacula selbst generiert und muss nicht geändert werden

- (3) Dies ist der Name des File-Daemon welcher noch beim "Director" in der "bacula-dir.conf" eine Rolle spielt
- (4) Der Port auf dem der Daemon lauscht
- (5) Nachrichten werden an den "Director" gesendet