

# Plik bootstrap w teorii i w praktyce

29 kwiecień 2012 autor: **gani**

Celem artykułu jest zaznajomienie czytelnika z tworzeniem i użytkowaniem plików bootstrap. Artykuł zawiera liczne przykłady pracy z tego typu plikami oraz niezbędną teorię do rozpoczęcia własnych prób z wykorzystaniem takich narzędzi jak: **bextract**, **bls** czy **bscan**.

## Wstęp

Istnieje co najmniej kilka sposobów na wykonanie przywracania plików w Bacula czyli tzw. wykonanie **restore**. Te bardziej przyjazne sposoby na restore dotyczą pełnosprawnej instalacji serwerów z Bacula z wykorzystaniem bazy katalogowej (bazy danych) oraz tekstowej konsoli lub któregoś z graficznych narzędzi Baculi. Nieco mniej przyjazne sposoby dotyczą użycia narzędzi pomocniczych takich jak **bextract**, **bscan** czy **bls**. Użyłem frazy „nieco mniej przyjazne” nie tylko w odniesieniu do sposobu korzystania z wymienionych narzędzi, lecz również do sytuacji, w której może zajść potrzeba ich użycia jak np. utrata bazy katalogowej czy też utraty całej instalacji serwerów z Bacula.

Opisane w tym artykule **pliki bootstrap** są jedną z tych funkcjonalności dostarczanych przez Bacula, które na pewno mogą przyspieszyć czas przywrócenia instalacji Bacula do stanu sprzed awarii (czy to bazy katalogowej czy też serwerów z serwisami Bacula) oraz ułatwić pracę z jej narzędziami pomocniczymi. Zanim jednak przedstawię zastosowanie plików bootstrap w praktyce, na początek nieco teorii o **plikach bootstrap**.

## Do czego służą pliki bootstrap

Głównym zadaniem plików bootstrap jest przechowywanie informacji o tym, co powinno być przywrócone w operacji restore. Błędym jednak jest rozumienie plików bootstrap jako **list dołączeń** (ang. include list), gdyż w odróżnieniu od nich pliki bootstrap wiedzą, co przywracać oraz to, gdzie dokładnie znajdują się przywracane dane. Listy dołączeń tego nie potrafią, gdyż znają one jedynie lokalizacje i nazwy plików, które będą przywrócone.

Różnicę pomiędzy plikami bootstrap a listami dołączeń w operacji restore można przedstawić np. w odniesieniu do sytuacji, gdy chcemy zaparzyć kubek kawy. Potrzebne do tego będą: kubek, gorąca woda, pojemnik z kawą, cukier i mleko. Gdyby stworzyć do czynności zaparzenia kawy listę dołączeń i plik bootstrap, to lista dołączeń wie, że dopuszczalne do użycia w tej czynności są: kubek, gorąca woda, kawa, cukier i mleko. Plik bootstrap nie zna tych składników, lecz ma dokładne informacje o tym, gdzie sięgnąć aby zdobyć wszystkie składniki czyli wie o tym, że jeden składnik znajduje się w suszarce, drugi w czajniku, trzeci składnik jest na górnej półce kredensu, czwarty na stole w cukiernicy, piąty w lodówce.

Opisaną powyżej właściwość zapamiętywania przez **pliki bootstrap** tego, gdzie znajdują się określone dane można wykorzystać do zapamiętania informacji o każdym backupie. Proszę wyobrazić sobie sytuację, że w celu zabezpieczenia się przed utratą bazy danych Bacula administrator ustawił wykonywanie backupu bazy danych w harmonogramie zadań (ang. schedule). Któregoś dnia w wyniku awarii serwera administrator utracił bazę danych. W celu odtworzenia stanu bazy danych sprzed awarii potrzebna jest mu taśma z najaktualniejszą kopią zrzutu bazy danych. Posiadając **plik bootstrap** z backupu bazy danych nie musi on główkować nad tym, na której taśmie znajduje się najaktualniejszy backup bazy danych, gdyż to pamięta za niego plik bootstrap. Co więcej, jeśli na taśmie z kopią bazy danych znajduje się jeszcze kilka innych backupów, to **plik bootstrap**

wskaże w którym miejscu taśmy znajduje się backup bazy danych. Nie ma więc w tej sytuacji potrzeby „ręcznego” przywracania całej zawartości taśmy.

Innym przykładem może być awaria serwera z serwisem **zarządcy Bacula** (ang. Director). Pliki konfiguracyjne serwisu zarządcy zawierają cenne informacje, które nie można w prosty sposób odzyskać. Zakładając, że administrator wykonywał **backup plików konfiguracyjnych** zarządcy Bacula wraz z zapisem pliku bootstrap, to przy pomocy narzędzia **bextract** wraz z **plikiem bootstrap** może on szybko odzyskać najaktualniejszą kopię zapasową z konfiguracją zarządcy, która umożliwi mu wznowienie pracy serwisu zarządcy.

## Co zawierają pliki bootstrap

Składnia plików bootstrap zawiera prostą strukturę dyrektyw w postaci:

```
opcja = wartość
```

gdzie:

**opcja** to słowo kluczowe będące jednocześnie jedną z właściwości backupu np. Volume, Client, Job, JobId i inne.

**wartość** to np. nazwa woluminu na którym znajduje się backup, nazwa klienta, z którego został wykonany backup, nazwa lub identyfikator backupu i inne.

Poniżej znajduje się opis większości dopuszczalnych opcji dla pliku bootstrap.

**Volume** - określa wymagany wolumin do przywrócenia backupu. Opcja woluminu ma specjalne znaczenie w plikach bootstrap z dwóch powodów. Pierwszy dotyczy tego, że opcja **Volume** jest obowiązkowa i w każdym pliku bootstrap użytkownik Bacula może znaleźć co najmniej jedną taką opcję. Drugi powód mówi o tym, że choć w plikach bootstrap nie ma widocznie określonych sekcji, to opcja **Volume** rozpoczyna każdą definicję kryteriów dla wydobywania określonych danych z woluminu określonego w opcji **Volume**. Można więc powiedzieć, że to właśnie opcja **Volume** pełni swego rodzaju rolę sekcji. Szkielet pliku bootstrap dla trzech woluminów o nazwach **raz**, **dwa** i **trzy** może wyglądać jak poniżej.

```
Volume = raz
Opcja = wartość
Opcja = wartość
Volume = dwa
Opcja = wartość
Opcja = wartość
Volume = trzy
Opcja = wartość
Opcja = wartość
```

Jeśli **wolumin Bacula** określony w tej sekcji zawiera znak (lub znaki) spacji w nazwie, to w pliku bootstrap powinien on być opatrzony znakami cudzysłowu. Np. dla woluminu o nazwie **Jakis wolumin** definicja w pliku bootstrap powinna wyglądać następująco:

```
Volume = „Jakis wolumin”
```

**Count** - opcja ta wyraża maksymalną liczbę plików jaka zostanie przywrócona. Jeśli w operacji przywracania danych okaże się że odzyskiwany backup zawiera większą ilość

plików, to po przywróceniu ilości plików określonych w tej opcji odzyskiwanie zostanie zatrzymane.

Przykład z ustawieniem maksymalnej liczby plików na 54.

```
Count = 54
```

**VolFile** - opcja, w której można określić numer fizycznego pliku znajdującego się na wolumenie, który to plik zostanie odzyskany. Możliwe jest podanie pojedynczego numeru pliku, listy z numerami plików oraz zakres numerów plików na woluminie.

Przykład dla pliku o numerze 7:

```
VolFile = 7
```

Przykład dla listy numerów plików o wartościach 3, 5 i 12:

```
VolFile = 3,5,12
```

Przykład dla zakresu numerów plików od 44 do 100:

```
VolFile = 44-100
```

Numer pliku, listę numerów plików oraz zakresy można ze sobą łączyć. Np. dla plików o numerach 3,6,7,8,9,14,22 zapis opcji **VolFile** może wyglądać następująco:

```
VolFile = 3,6-9,14,22
```

**VolBlock** - wartość tej opcji wyraża numer bloku na wolumenie. Możliwe jest podanie pojedynczego numeru bloku, listy z numerami bloków oraz zakres numerów bloków na wolumenie. Podobnie jak w przypadku opcji **VolFile**, w opcji **VolBlock** jest możliwe łączenie numerów bloków, listy bloków oraz zakresów. Przy okazji opisu tej opcji warto też wspomnieć o tym, że domyślny **rozmiar bloku** (ang. **blocksize**) z jakim Bacula zapisuje dane na wolumenach to 64512 bajty. O ile wartość ta nie została zmieniona przez użytkownika Bacula, to przy podaniu w opcji **VolBlock** wartości 2-3 czy też 12-13, obydwa z nich będą odnosić się do fragmentu wolumenu o rozmiarze 129024 bajtów (2 \* 64512 bajtów) lecz każdy z tych fragmentów znajduje się w innym miejscu woluminu.

Przykłady dla jednego numeru bloku woluminu, listy bloków woluminu, zakresu bloków oraz połączenia listy bloków z ich zakresami znajdują się poniżej.

```
VolBlock = 8
```

```
VolBlock = 55,60,78
```

```
VolBlock = 4-29
```

```
VolBlock = 5,8,15,19-29
```

**VolSessionTime** - opcja ta określa czas sesji dla woluminu. Za każdym razem, gdy dane zostają zapisane na woluminie poprzez backup, to nadany zostaje czas sesji, który zapisywany jest na woluminie oraz w bazie danych Bacula. Widoczny jest on również w podsumowaniu backupu po jego zakończeniu. W poniższym listingu podsumowania backupu czas **VolSessionTime** ma wartość 1333201882.

```

Build OS:                x86_64-unknown-linux-gnu ubuntu 11.10
JobId:                   93
Job:                     www.2012-03-31_15.59.56_04
Backup Level:           Full
Client:                  „darkstar-fd” 5.2.3 (16Dec11) x86_64-unknown-
linux-gnu,ubuntu,11.10
FileSet:                 „www-fileset” 2012-01-15 04:01:49
Pool:                    „TapeBackup” (From User input)
Catalog:                „MyCatalog” (From Client resource)
Storage:                „TapeDrive” (From Pool resource)
Scheduled time:         31-Mar-2012 15:59:25
Start time:              31-Mar-2012 15:59:58
End time:                31-Mar-2012 16:00:49
Elapsed time:           51 secs
Priority:                 1
FD Files Written:       11,295
SD Files Written:       11,295
FD Bytes Written:       199,859,270 (199.8 MB)
SD Bytes Written:       201,575,247 (201.5 MB)
Rate:                   3918.8 KB/s
Software Compression:   None
VSS:                    no
Encryption:             no
Accurate:               no
Volume name(s):         T0001L1
Volume Session Id:      2
Volume Session Time:    1333201882
Last Volume Bytes:      546,158,592 (546.1 MB)
Non-fatal FD errors:    0
SD Errors:              0
FD termination status:  OK
SD termination status:  OK
Termination:           Backup OK

```

Dopuszczalną wartością opcji **VolSessionTime** jest wartość liczbowa. Opcja ta nie wspiera list z czasami sesji ani też zakresów czasów sesji.

Poniżej znajduje się przykład użycia opcji **VolSessionTime**.

```
VolSessionTime = 1333201882
```

**VolSessionId** - opcja określa identyfikator sesji dla wolumenu. Wartość tej opcji może być podawana jako pojedyncza wartość, lista lub zakres. Możliwe jest również łączenie list i zakresów z identyfikatorami sesji. Każdej wartości **VolSessionId** odpowiada dokładnie jeden czas sesji dla wolumenu (**VolSessionTime**), które - gdy występują razem w jednym **pliku bootstrap** i w jednej sekcji wolumena - odpowiadają dokładnie jednemu unikalnemu backupowi na wolumenie.

Poniżej znajduje się przykład pojedynczej wartości identyfikatora sesji, listy identyfikatorów sesji, zakresu identyfikatorów sesji oraz połączenia wszystkich wymienionych.

```
VolSessionId = 4
```

```
VolSessionId = 4,6,8
```

```
VolSessionId = 1-7
```

```
VolSessionId = 2-5,7,9,10-20,25
```

**JobId** - opcja wyraża identyfikator backupu, listę lub zakres identyfikatorów backupów. Możliwe jest podanie kombinacji tych trzech typów wartości. Ta opcja nie jest używana jako kryterium do przywracania danych z woluminów, gdyż nie wyraża ona w żaden sposób położenia backupu o jakimś identyfikatorze na woluminie. Może za to być skutecznie użyta np. przy użyciu narzędzia bscan. W przypadku używania wielu serwisów zarządcy (ang. director) wartość **JobId** może nie być wartością unikalną dla woluminu. Poniżej zamieszczam kilka przykładów definicji opcji **JobId**.

```
JobId = 121
```

```
JobId = 3,8
```

```
JobId = 1-49,51-100
```

```
JobId = 1-102,111
```

**Job** - przy pomocy tej opcji można podać w pliku bootstrap nazwę backupu (lub backupów) które mają zostać wybrane z wolumenu. Jako, że nazwa backupu (**Job**) podobnie jak identyfikator backupu nie wyraża położenia backupu na woluminie, to nie jest ona używana do przywracania danych w pliku bootstrap. Do tego celu nadaje się za to para opcji **VolSessionId** i **VolSessionTime**. W opcji **Job** dopuszczalne jest użycie znaków wieloznacznych (wildcards) w celu dopasowania nazw backupów do określonego wzorca. W definicji opcji **Job** jako wartości dopuszczalne jest podanie pojedynczej nazwy backupu lub listy z nazwami backupów (przykłady znajdują się poniżej).

```
Job = „Katalog domowy”
```

```
Job = Home,Logs,„katalog Web”
```

**Client** - opcja ta określa nazwę (lub nazwy) klienta (lub klientów) z których został wykonany poszukiwany na woluminie backup. W wartości tej opcji dopuszczalne jest użycie znaków wieloznacznych (wildcards). Ta opcja również nie jest używana w pliku bootstrap do operacji przywracania danych. Poniżej zamieszczam przykład pojedynczej definicji klienta w pliku bootstrap oraz przykład listy nazw klientów.

```
Client = gani-fd
```

```
Client = gani-fd, ads-fd,„inny klient”
```

**FileIndex** - opcja mówi o tym, jakie pliki o jakich indeksach będą brane pod uwagę w pliku bootstrap. Dla każdej sesji zapisu na wolumin (dla każdego backupu) indeksy zapisywanych plików są numerowane od liczby 1. Dla każdego pliku z backupu przypisany jest jeden unikalny w obrębie danej sesji indeks.

Np. w skład backupu wchodzi następujące pliki:

```
/home/gani/dokument.txt
/home/gani/notatka.txt
/home/gani/backup.png
/home/gani/restore.xcf
/home/gani/job.log
/home/gani/bacula.log
```

Dla takiego backupu pliki będą numerowane indeksami w następujący sposób:

```
1 /home/gani/dokument.txt
2 /home/gani/notatka.txt
3 /home/gani/backup.png
4 /home/gani/restore.xcf
5 /home/gani/job.log
6 /home/gani/bacula.log
```

Jeśli chciałbym uwzględnić w pliku bootstrap tylko pliki: **dokument.txt**, **restore.xcf**, **job.log** oraz bacula.log to definicja opcji **FileIndex** mogłaby wyglądać jak poniżej.

```
FileIndex = 1,4-6
```

Jak widać na załączonym przykładzie dopuszczalne w opcji **FileIndex** są wartości: pojedynczy indeks, listy indeksów oraz zakresy indeksów jak i możliwość łączenia wszystkich wymienionych.

Poznane już opcje **VolSessionId**, **VolSessionTime** oraz **FileIndex** zapisane w pliku bootstrap mogą jednoznacznie identyfikować pojedynczy plik (lub wiele plików) na wolumenie. Te trzy opcje są zapisywane na wolumenie oraz w bazie danych Bacula dla każdego zapisanego na wolumenie pliku.

**FileRegexp** - opcja definiuje wyrażenie regularne, przy pomocy którego można wybierać pliki, które zostaną przywrócone przy akcji **restore** z użyciem pliku bootstrap.

Oto prosty przykład opcji FileRegexp:

```
FileRegexp=~/etc/apache2/apache.*
```

**Slot** - ta opcja - jeśli jest zdefiniowana - dotyczy numeru slotu, z którego pochodzi wolumen. Definicja tej opcji może być używana dla wolumenów pochodzących z urządzeń ze zmieniającą taśmą. Dla każdej sekcji wolumena w pliku bootstrap możliwe jest podanie tylko jednego numeru slotu, gdyż każda taka taśma w bazie danych Bacula jest powiązana z dokładnie jednym numerem slotu urządzenia ze zmieniającą.

## Jak stworzyć plik bootstrap

### 1. Tworzenie pliku bootstrap poprzez komendę restore

Pierwszym ze sposobów na stworzenie **pliku bootstrap** - przy założeniu, że serwisy Baculi są sprawne - jest użycie komendy **restore** w tekstowej konsoli Bacula. Potrzebne do tego będzie posiadanie co najmniej jednego zasobu **Job** z dyrektywą **Type** ustawioną na **Restore (Job typu Restore)**, np.:

```
Job {
  Name = „RestoreFiles”
```

```
Type = Restore
Client = darkstar-fd
FileSet= „DBDump FileSet”
Storage = VTL
Pool = Full-VTL1
Messages = Standard
Where = /tmp/bacula-restores
}
```

W celu stworzenia pliku bootstrap potrzeba w konsoli Bacula wykonać komendę restore w sposób podobny do przedstawionego poniżej. W tym wypadku przywracane będą dwa pliki z backupu o **JobId** równym 109:

```
restore jobid=109
```

Jeśli backup o identyfikatorze 109 zawiera pliki, to czytelnik powinien zostać przekierowany do „*podkonsoli*”, w której będzie mógł zaznaczyć pliki do przywrócenia z backupu. Wejście do tej „*podkonsoli*” można rozpoznać po znaku zachęty składającego się ze znaku \$.

```
Building directory tree for JobId(s) 109 ...
3 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: /
$
```

Zaznaczam dwa pliki: **syslog** oraz **bacula.log**, które znajdują się w backupie w lokalizacji **/var/log/**

```
$ ls
var/
$ cd var/log
cwd is: /var/log/
$ ls
bacula.log
mail.log
syslog
$ mark bacula.log
1 file marked.
$ mark syslog
1 file marked.
```

Kończę selekcję plików poprzez wpisanie komendy **done**.

```
$ done

Bootstrap records written to /usr/local/bacula/var/bacula/working/
darkstar-dir.restore.2.bsr
```

```

The job will require the following
  Volume(s)                Storage(s)                SD Device(s)
=====
=====

    *VTL_0003_0010          VTL                        Virtual Tape
Library

Volumes marked with „*” are online.

2 files selected to be restored.

Defined Clients:
  1: darkstar-fd
  2: darkstarus-fd
  3: tymlas-fd
Select the Client (1-3): 1
Run Restore job
JobName:          RestoreFiles
Bootstrap:        /usr/local/bacula/var/bacula/working/darkstar-dir.
restore.2.bsr
Where:            /tmp/bacula-restores
Replace:          always
FileSet:          DBDump FileSet
Backup Client:    darkstar-fd
Restore Client:   darkstar-fd
Storage:          VTL
When:             2012-04-26 04:06:43
Catalog:          MyCatalog
Priority:          10
Plugin Options:   *None*
OK to run? (yes/mod/no):

```

Po wpisaniu komendy **done** pojawia się podsumowanie opcji, które zostaną użyte do restore. Widać tu również, że został stworzony **plik bootstrap** o nazwie **darkstar-dir.restore.2.bsr**. Informacja o tym, gdzie można znaleźć stworzony plik bootstrap znajduje się dokładnie w tej linii z powyższego listingu:

```

Bootstrap:          /usr/local/bacula/var/bacula/working/darkstar-dir.
restore.2.bsr

```

## UWAGA!

Plik bootstrap został stworzony, lecz aby go zachować **NIE** powinno się potwierdzić podsumowania opcji restore lecz wpisać po prostu **no**:

```

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

```



Gdybym potwierdził start restore poprzez wpisanie **yes**, to po wykonaniu zadania restore, plik bootstrap zostałby skasowany.

Zawartość stworzonego przed chwilą pliku bootstrap wygląda tak:

```
Storage="VTL"
Volume="VTL_0003_0010"
MediaType="Plik"
Device="Virtual Tape Library"
Slot=40
VolSessionId=2
VolSessionTime=1335100342
VolAddr=6195610-6719686
FileIndex=1
FileIndex=3
Count=2
```

W zawartości stworzonego pliku można zauważyć kilka z poznanych już opcji jak np.: **Volume, Slot, VolSessionId, VolSessionTime, FileIndex, Count**.

## 2. Tworzenie pliku bootstrap poprzez dyrektywę Write Bootstrap

O ile w poprzednim przykładzie przy operacji **restore** na pełnosprawnej instalacji serwisów Bacula łatwiej byłoby wykonać po prostu przywracanie danych niż posiłkować się **plikiem bootstrap** i narzędziem **bextract**, o tyle gdyby instalacja Baculi była uszkodzona (np. poprzez utratę bazy danych Bacula) to użycie pliku bootstrap mogłoby znacznie przyspieszyć odzyskiwanie danych z woluminów czy też odtworzenie bazy danych ze stanu sprzed awarii.

Aby zabezpieczyć backupy w ten sposób, że dla każdego backupu tworzony byłby plik bootstrap potrzebne będzie użycie dyrektywy **Write Bootstrap** w zasobie **Job**. Dyrektywa **Write Bootstrap** definiuje lokalizację, do której ma zostać zapisany plik bootstrap z wykonywanego backupu. Dla zasobu Job o nazwie „Dokumenty” może to wyglądać w następujący sposób:

```
Job {
    Name = „Dokumenty”
    ...
    Write Bootstrap = „/mnt/jakas_partycja/Dokumenty.bsr”
}
```

(trzy kropki użyte w listingu oznaczają, że jest to wycinek konfiguracji Bacula)

W powyższym przykładzie użyłem rozszerzenia dla pliku bootstrap o nazwie **.bsr**. Nie jest to żaden wymóg, aby pliki bootstrap posiadały takie rozszerzenie, lecz jest to sprawa konwencji nazewnictwa, która przyjmuje rozszerzenie **.bsr** dla tego typu plików. Trzymając się tej konwencji dla wszystkich plików bootstrap łatwiej będzie rozpoznać tego typu pliki.

Popularnym sposobem na zapis plików bootstrap jest definiowanie dyrektywy **Write Bootstrap** z wartością, która wykorzystuje słowa kluczowe w następujący sposób:

```
Job {
  Job = Dokumenty
  Client = gani-fd
  ...
  Write Bootstrap = „/mnt/jakas _ partycja/%c _ %n.bsr”
}
```

(trzy kropki użyte w listingu oznaczają, że jest to wycinek konfiguracji Bacula)

Wynikowy plik bootstrap będzie miał postać:

```
/mnt/jakas _ partycja/gani-fd _ Dokumenty.bsr
```

Dzieje się tak za sprawą użycia słów kluczowych:

**%c** - za który podstawiana jest nazwa użytego klienta do backupu,

**%n** - za który podstawiana jest nazwa zasobu **Job**, z którego backup został wykonany.

Po więcej informacji na temat możliwych do użycia słów kluczowych odsyłam czytelnika do oficjalnej dokumentacji Bacula.

## UWAGA!

Dla kopii pełnych wykorzystujących zapis pliku bootstrap poprzez dyrektywę **Write Bootstrap** za każdym wykonaniem kopii typu **Full** wszystkie wcześniej zapisane opcje pliku bootstrap są czyszczone. Dla **kopii przyrostowych** i **różnicowych** takiego backupu nowe opcje są dopisywane na koniec treści istniejącego pliku bootstrap.

## Przykład

Pierwszy backup wykonał kopię pełną i zapisał do pliku bootstrap poniższe wartości:

```
Volume="VTL _ 0002 _ 0010"
MediaType="Plik"
Slot=25
VolSessionId=2
VolSessionTime=1329556067
VolAddr=203-82743088
FileIndex=1-1
```

Drugi backup z tego samego zasobu **Job** wykonał **kopię przyrostową** i dopisał do pliku bootstrap swoją sekcję **Volume** i opcje tej sekcji. Teraz zawartość pliku bootstrap wygląda tak:

```
Volume="VTL _ 0002 _ 0010"
MediaType="Plik"
Slot=25
VolSessionId=2
VolSessionTime=1329556067
VolAddr=203-82743088
FileIndex=1-1
Volume="VTL _ 0002 _ 0015"
MediaType="Plik"
```

```
Slot=30
VolSessionId=1
VolSessionTime=1329568155
VolAddr=203-56846875
FileIndex=1-1
```

Trzeci backup z tego samego zasobu **Job** wykonał **kopię różnicową** i dopisał do pliku bootstrap swoją sekcję **Volume** i opcje tej sekcji. Zawartość pliku bootstrap wygląda teraz tak:

```
Volume="VTL _ 0002 _ 0010"
MediaType="Plik"
Slot=25
VolSessionId=2
VolSessionTime=1329556067
VolAddr=203-82743088
FileIndex=1-1
Volume="VTL _ 0002 _ 0015"
MediaType="Plik"
Slot=30
VolSessionId=1
VolSessionTime=1329568155
VolAddr=203-56846875
FileIndex=1-1
Volume="VTL _ 0002 _ 0020"
MediaType="Plik"
Slot=10
VolSessionId=2
VolSessionTime=1329682123
VolAddr=203-95613455
FileIndex=1-3
```

Gdyby teraz wykonać kopię pełną z tego samego zasobu **Job**, to powyższy plik uległby najpierw wyczyszczeniu, a następnie na koniec backupu zostałyby zapisane dla backupu tylko opcje dla aktualnej **kopii pełnej**.

Opisana zależność pomiędzy zawartością pliku bootstrap dla **kopii pełnych** (full), **przyrostowych** (incremental) i **różnicowych** (differential) wynika z takich faktów, jak:

- do odtworzenia kopii pełnej potrzebna jest tylko kopia pełna,
- do odtworzenia danych z kopii przyrostowych potrzebne są wszystkie ostatnio wykonane kopie przyrostowe i kopia pełna, na której bazowały kopie przyrostowe,
- do odtworzenia danych z kopii różnicowych potrzebne są jedna kopia różnicowa oraz jedna kopia pełna, na której kopia różnicowa bazowała.

### 3. Ręczne tworzenie pliku bootstrap

Jako, że pliki bootstrap to najzwyczajniejsze pliki tekstowe, to możliwe jest również stworzenie samemu takiego pliku przy użyciu edytora tekstu. Informacje potrzebne do tego, aby ręcznie utworzyć plik bootstrap można czerpać np. z dziennika systemowego Bacula czyli z tzw. „logów”.

Przykładowy **log** z podsumowaniem backupu może mieć postać:

```
Build OS:                x86_64-unknown-linux-gnu ubuntu 11.10
JobId:                   93
Job:                     www.2012-03-31_15.59.56_04
Backup Level:           Full
Client:                  „darkstar-fd” 5.2.3 (16Dec11) x86_64-unknown-
linux-gnu,ubuntu,11.10
FileSet:                 „www-fileset” 2012-01-15 04:01:49
Pool:                   „TapeBackup” (From User input)
Catalog:                „MyCatalog” (From Client resource)
Storage:                „TapeDrive” (From Pool resource)
Scheduled time:         31-Mar-2012 15:59:25
Start time:             31-Mar-2012 15:59:58
End time:               31-Mar-2012 16:00:49
Elapsed time:           51 secs
Priority:                1
FD Files Written:       11,295
SD Files Written:       11,295
FD Bytes Written:       199,859,270 (199.8 MB)
SD Bytes Written:       201,575,247 (201.5 MB)
Rate:                   3918.8 KB/s
Software Compression:   None
VSS:                    no
Encryption:             no
Accurate:               no
Volume name(s):         T0001L1
Volume Session Id:      2
Volume Session Time:    1333201882
Last Volume Bytes:      546,158,592 (546.1 MB)
Non-fatal FD errors:    0
SD Errors:              0
FD termination status:  OK
SD termination status:  OK
Termination:           Backup OK
```

Patrząc na listing z podsumowaniem backupu, można wykorzystać - na potrzeby stworzenia własnego pliku bootstrap dla tego backupu - następujące opcje:

```
JobId:                   93
Job:                     www
Volume name(s):         T0001L1
Volume Session Id:      2
Volume Session Time:    1333201882
```

Na podstawie tych informacji można stworzyć plik bootstrap dla backupu o nazwie **www** o następującej zawartości:

```
# Backup o nazwie „www” oraz identyfikatorze 93
Volume = T0001L1
VolSessionId = 2
VolSessionTime = 1333201882
```

Użycie takiego pliku bootstrap do restore przy użyciu narzędzia **bextract** i taśmy o etykiecie **T0001L1** spowoduje wypakowanie zawartości backupu www o identyfikatorze 93 z taśmy **T0001L1**.

## Zależności opcji w pliku bootstrap

Opcje zdefiniowane w pliku bootstrap mają pomiędzy sobą zależność operatora logicznego **AND**. Oznacza to, że dla każdego wolumena wpisanego w plik bootstrap każda z opcji musi być spełniona, aby możliwe było znalezienie danych na wolumenie. Zakładając, że plik bootstrap wygląda tak:

```
Volume="VTL _ 0003 _ 0010"  
VolSessionId=2  
VolSessionTime=1335100342  
FileIndex=1
```

to odczytanie zależności pomiędzy wszystkimi opcjami można czytać w poniższy sposób.

*Jeśli nazwa wolumena jest równa VTL\_0003\_0010 i opcja VolSessionId jest równa 2 i opcja VolSessionTime jest równa 1335100342 i opcja FileIndex jest równa 1 to przywróć dane spełniające wszystkie te kryteria.*

Krótszy zapis odczytania powyższego przykładu z wykorzystaniem operatora logicznego **AND** wygląda tak:

```
Volume="VTL _ 0003 _ 0010"
```

**AND**

```
VolSessionId=2
```

**AND**

```
VolSessionTime=1335100342
```

**AND**

```
FileIndex=1
```

## UWAGA!

Istnieją również sytuacje, w których oprócz operatora logicznego **AND** zostanie również użyty operator **OR**. Dotyczy to sytuacji, gdy używa się wielu wartości w jednej opcji (listy wartości i zakresy) oraz wtedy, gdy ta sama opcja zostaje użyta więcej niż jeden raz. Poniżej zamieszczam przykłady dla obydwu tych przypadków.

### Przykład 1 (wiele wartości w jednej opcji)

```
Volume="VTL _ 0003 _ 0010"  
VolSessionId=2  
VolSessionTime=1335100342  
FileIndex=1,5-10,12
```

Przykład można przeczytać jak poniżej.

```
Volume="VTL _ 0003 _ 0010"
```

AND

VolSessionId=2

AND

VolSessionTime=1335100342

AND

FileIndex=1 OR 5 OR 6 OR 7 OR 8 OR 9 OR 10 OR 12

## Przykład 2 (wiele tych samych opcji)

Volume="VTL \_ 0003 \_ 0010"

VolSessionId=2

VolSessionTime=1335100342

FileIndex=1

FileIndex=7

FileIndex=15

Przykład można przeczytać jak poniżej.

Volume="VTL \_ 0003 \_ 0010"

AND

VolSessionId=2

AND

VolSessionTime=1335100342

AND

FileIndex=1

OR

FileIndex=7

OR

FileIndex=15

Można też zadać sobie pytanie o to, w jaki sposób traktowane są sekcje **Volume**, jeśli w pliku bootstrap występuje więcej niż jedna definicja opcji **Volume**. Otóż są one połączone ze sobą operatorem logicznym **OR** (przykład poniżej).

Volume="VTL \_ 0003 \_ 0010"

VolSessionId=2

VolSessionTime=1335100342

FileIndex=1

OR

Volume="VTL \_ 0003 \_ 0011"

VolSessionId=3

```
VolSessionTime=1335101421
FileIndex=5
```

### OR

```
Volume="VTL_0003_0012"
VolSessionId=4
VolSessionTime=1335101847
FileIndex=7
```

## Użycie pliku bootstrap z narzędziem bextract

W tym przykładzie zakładam, że baza danych Bacula uległa zniszczeniu. Administrator Bacula zabezpieczył się przed tego typu awarią poprzez backup bazy danych Bacula. Dysponuje on plikiem bootstrap z backupu bazy danych oraz kompletem wolumenów plikowych zarządzanych przez urządzenie wirtualnej biblioteki taśmowej realizowanej przy użyciu programu **Vchanger**. Administrator zagląda w plik bootstrap z backupu bazy katalogowej Bacula, który wygląda następująco:

```
# cat DBDump.bsr
Volume="VTL_0002_0010"
MediaType="Plik"
Slot=25
VolSessionId=2
VolSessionTime=1329556067
VolAddr=203-82743088
FileIndex=1-1
```

Wydaje on komendę, której składnia wygląda następująco:

```
bextract -c {lokalizacja_pliku_konfiguracyjnego_bacula-sd.conf} -b
{lokalizacja_pliku_bootstrap} {nazwa_urzadzenia} {lokalizacja_
docelowa}
```

gdzie:

**{lokalizacja\_pliku\_konfiguracyjnego\_bacula-sd.conf}** - oznacza lokalizację pliku konfiguracyjnego daemon'a magazynowania (bacula-sd),

**{lokalizacja\_pliku\_bootstrap}** - lokalizacja, w której znajduje się plik bootstrap z backupu bazy danych,

**{nazwa\_urzadzenia}** - nazwa urządzenia z zasobu Device, które ma zostać użyte do operacji rozpakowania backupu bazy danych,

**{lokalizacja\_docelowa}** - docelowe miejsce, do którego zostaną wypakowane pliki backupu.

Administrator wykonuje komendę zgodną z przedstawionym powyżej wzorem:

```
# bextract -c /usr/local/bacula/etc/bacula-sd.conf -b /usr/local/bacula/
var/bacula/working/DBDump.bsr Drive-2 /tmp
```

Na standardowe wyjście zostaje wydrukowany poniższy listing, z którego można przeczytać, że został przywrócony jeden plik (dump bazy danych Bacula).

```
bextract: butil.c:287 Using device: „Drive-2” for reading.
28-Apr 18:55 bextract JobId 0: 3301 Issuing autochanger „loaded? drive
1” command.
28-Apr 18:55 bextract JobId 0: 3302 Autochanger „loaded? drive 1”, result
is Slot 25.
28-Apr 18:55 bextract JobId 0: Ready to read from volume „VTL _ 0002 _ 0010”
on device „Drive-2” (/media/backup/VTL/1/drive1).
28-Apr 18:55 bextract JobId 0: Forward spacing Volume „VTL _ 0002 _ 0010”
to file:block 0:203.
bextract JobId 0: -rw----- 1 root      root          82681035 2012-02-18
10:12:32 /tmp/usr/local/bacula/var/bacula/working/bacula.sql
28-Apr 18:55 bextract JobId 0: End of Volume at file 0 on device „Drive-2”
(/media/backup/VTL/1/drive1), Volume „VTL _ 0002 _ 0010”
28-Apr 18:55 bextract JobId 0: End of all volumes.
1 files restored.
```

Odzyskany dump bazy danych znajduje się w lokalizacji:

```
/tmp/usr/local/bacula/var/bacula/working/bacula.sql
```

Teraz pozostaje administratorowi użycie narzędzia z serwera bazodanowego, które przywróci strukturę bazy danych sprzed awarii.

## Użycie narzędzia bls z plikiem bootstrap

Ten sam administrator z przykładu użycia narzędzia bextract chciałby sprawdzić przy użyciu pliku bootstrap jakie pliki znajdują się w innym backupie o nazwie www. Niestety administrator nie dysponuje plikiem bootstrap z tego backupu. Posiada on za to woluminy z backupami oraz dziennik systemowy Bacula czyli tzw. log z backupu. Log wygląda jak poniżej:

```
22-Apr 07:21 darkstar-dir JobId 107: Start Backup JobId 107,
Job=www.2012-04-22_07.20.58_03
22-Apr 07:21 darkstar-dir JobId 107: Using Volume „VTL _ 0003 _ 0009” from
,Scratch’ pool.
22-Apr 07:21 darkstar-dir JobId 107: Using Device „Drive-1”
22-Apr 07:21 darkstar-sd JobId 107: 3307 Issuing autochanger „unload slot
38, drive 0” command.
22-Apr 07:21 darkstar-sd JobId 107: 3304 Issuing autochanger „load slot
39, drive 0” command.
22-Apr 07:21 darkstar-sd JobId 107: 3305 Autochanger „load slot 39, drive
0”, status is OK.
22-Apr 07:21 darkstar-sd JobId 107: Wrote label to prelabeled Volume
„VTL _ 0003 _ 0009” on device „Drive-1” (/media/backup/VTL
22-Apr 07:21 darkstar-dir JobId 107: Max Volume jobs=1 exceeded. Marking
Volume „VTL _ 0003 _ 0009” as Used.
22-Apr 07:21 darkstar-sd JobId 107: Job write elapsed time = 00:00:21,
Transfer rate = 7.204 M Bytes/second
Build OS:          x86_64-unknown-linux-gnu ubuntu 11.10
JobId:             107
Job:               www.2012-04-22_07.20.58_03
Backup Level:     Full
```



```
Client:                „darkstar-fd” 5.2.3 (16Dec11) x86_64-unknown-
linux-gnu,ubuntu,11.10
FileSet:               „www-fileset” 2012-01-15 04:01:49
Pool:                  „Full-VTL4” (From User input)
Catalog:              „MyCatalog” (From Client resource)
Storage:              „VTL” (From Pool resource)
Scheduled time:       22-Apr-2012 07:20:36
Start time:           22-Apr-2012 07:21:00
End time:             22-Apr-2012 07:21:23
Elapsed time:         23 secs
Priority:              1
FD Files Written:     11,294
SD Files Written:     11,294
FD Bytes Written:     149,583,399 (149.5 MB)
SD Bytes Written:     151,299,354 (151.2 MB)
Rate:                 6503.6 KB/s
Software Compression: None
VSS:                  no
Encryption:           no
Accurate:             no
Volume name(s):       VTL_0003_0009
Volume Session Id:    1
Volume Session Time: 1335070457
Last Volume Bytes:    151,748,341 (151.7 MB)
Non-fatal FD errors: 0
SD Errors:            0
FD termination status: OK
SD termination status: OK
Termination:         Backup OK
```

Dzięki posiadaniu podsumowania backupu z pliku dziennika systemowego, administrator może samodzielnie stworzyć plik bootstrap. Potrzebne mu będą do tego poniższe cztery linie z podsumowania backupu, z których administrator stworzy plik bootstrap:

```
22-Apr 07:21 darkstar-sd JobId 107: 3304 Issuing autochanger „load slot
39, drive 0” command.
Volume name(s):       VTL_0003_0009
Volume Session Id:    1
Volume Session Time: 1335070457
```

Przy pomocy programu **touch** tworzy on więc plik o dowolnej nazwie:

```
# touch /tmp/moj_bootstrap_www.bsr
```

i wpisuje do niego taką zawartość:

```
Volume = VTL_0003_0009
Slot=39
VolSessionId = 1
VolSessionTime = 1335070457
```

Teraz wystarczy uruchomić narzędzie **bls** w następujący sposób:

```
# /usr/local/bacula/sbin/bls -c /usr/local/bacula/etc/bacula-sd.conf -b  
/tmp/moj_bootstrap_www.bsr Drive-1
```

Otrzymany i przycięty - z racji dużej ilości plików - listing wygląda jak poniżej:

```
bls: butil.c:287 Using device: „Drive-1” for reading.  
28-Apr 19:28 bls JobId 0: 3301 Issuing autochanger „loaded? drive 0”  
command.  
28-Apr 19:28 bls JobId 0: 3302 Autochanger „loaded? drive 0”, result is  
Slot 39.  
28-Apr 19:28 bls JobId 0: Ready to read from volume „VTL_0003_0009” on  
device „Drive-1” (/media/backup/VTL/0/drive0).  
bls JobId 107: -rw-r--r--    1 root      root          371 2012-01-21  
00:45:00 /var/www/bweb/last.png  
bls JobId 107: -rw-r--r--    1 root      root          672 2012-01-21  
00:45:00 /var/www/bweb/T.png  
bls JobId 107: -rw-r--r--    1 root      root          537 2012-01-21  
00:45:00 /var/www/bweb/tape.png  
bls JobId 107: -rw-r--r--    1 root      root         1379 2012-01-21  
00:45:00 /var/www/bweb/A.png  
bls JobId 107: -rw-r--r--    1 root      root         1406 2012-01-21  
00:45:00 /var/www/bweb/favicon.ico  
bls JobId 107: -rw-r--r--    1 root      root         1379 2012-01-21  
00:45:00 /var/www/bweb/cancel.png  
bls JobId 107: -rw-r--r--    1 root      root         1434 2012-01-21  
00:45:00 /var/www/bweb/colorscm.png  
bls JobId 107: -rw-r--r--    1 root      root         2211 2012-01-21  
00:45:00 /var/www/bweb/filename.png  
bls JobId 107: -rw-r--r--    1 root      root         1603 2012-01-21  
00:45:00 /var/www/bweb/AT.png  
bls JobId 107: -rw-r--r--    1 root      root         1379 2012-01-21  
00:45:00 /var/www/bweb/f.png  
bls JobId 107: -rw-r--r--    1 root      root           61 2012-01-21  
00:45:00 /var/www/bweb/first.gif  
bls JobId 107: -rw-r--r--    1 root      root         1030 2012-01-21  
00:45:00 /var/www/bweb/file_d.png  
bls JobId 107: -rw-r--r--    1 root      root         1876 2012-01-21  
00:45:00 /var/www/bweb/chart.png  
bls JobId 107: -rw-r--r--    1 root      root         1027 2012-01-21  
00:45:00 /var/www/bweb/save.png  
bls JobId 107: -rw-r--r--    1 root      root          791 2012-01-21  
00:45:00 /var/www/bweb/prune.png  
bls JobId 107: -rw-r--r--    1 root      root         2660 2012-01-21  
00:45:00 /var/www/bweb/bweb.css  
bls JobId 107: -rw-r--r--    1 root      root          103 2012-01-21  
00:45:00 /var/www/bweb/pix.png  
bls JobId 107: -rw-r--r--    1 root      root         1291 2012-01-21  
00:45:00 /var/www/bweb/label.png  
bls JobId 107: -rw-r--r--    1 root      root         1094 2012-01-21  
00:45:00 /var/www/bweb/R.png
```

```
[ ----- SKRÓCENIE LISTINGU ----- ]
bls JobId 107: -rw-r--r--    1 root      root          177 2011-11-23
21:21:26 /var/www/index.html
bls JobId 107: -rw-----    1 root      root          160 2011-12-03
22:48:12 /var/www/.bconsole_history
bls JobId 107: -rw-r--r--    1 root      root           79 2012-01-21
00:52:26 /var/www/.htaccess
bls JobId 107: drwxr-xr-x    7 gani      gani         4096 2012-04-15
12:37:28 /var/www/
28-Apr 19:31 bls JobId 0: End of Volume at file 0 on device „Drive-1” (/
media/backup/VTL/0/drive0), Volume „VTL_0003_0009”
28-Apr 19:31 bls JobId 0: End of all volumes.
11294 files found.
```

## Użycie narzędzia bscan z plikiem bootstrap

Odtworzenie bazy danych Bacula przy użyciu narzędzia **bscan** może okazać się mocno czasochłonnym procesem (zwłaszcza, gdy administrator ma do przeskanowania kilkadziesiąt lub kilkaset wolumenów). Potrzebuje on podać narzędziu **bscan** wszystkie nazwy woluminów. Przy dużej ilości woluminów może okazać się, że długość wydawanej komendy przekroczy ilość dopuszczalnych znaków możliwych do podania jako parametry. Tutaj również można posłużyć się plikiem bootstrap. Wystarczy stworzyć własny plik bootstrap np. poprzez program **touch**:

```
# touch /tmp/wszystkie_woluminy.bsr
```

a następnie zdefiniować w stworzonym pliku wszystkie potrzebne nazwy wolumenów w sposób podobny do poniższego:

```
Volume="wolumin_1"
Volume="wolumin_2"
Volume="wolumin_3"
Volume="wolumin_4"
Volume="wolumin_5"
Volume="wolumin_6"
```

Teraz wystarczy wydać komendę podobną do tej:

```
# bscan -c /usr/local/bacula/etc/bacula-sd.conf -b /tmp/wszystkie_woluminy.bsr -m -s /media/jakies_urzadzenie
```

i oczekiwać na zakończenie procesu odtworzenia struktury tabel bazy danych Bacula.

## Podsumowanie

Mam nadzieję, że choć trochę przybliżyłem czytelnikowi temat plików bootstrap. Myślę, że warto samodzielnie poeksperymentować z tym typem plików, ich generowaniem, tworzeniem i używaniem.

W opracowaniu niniejszego artykułu pomocna była mi dokumentacja Bacula, w której czytelnik - jeśli ma na to ochotę - może znaleźć więcej informacji na poruszany tutaj temat.