Automatyczne etykietowanie wolumenów

17 styczeń 2011 autor: **gani**

Celem artykułu jest przedstawienie tego, w jaki sposób zautomatyzować proces tworzenia woluminów plikowych oraz jak formatować etykiety woluminów.

Wstęp

Praca z Bacula niekoniecznie musi składać się z wpisywania setek komend odpowiedzialnych za takie a takie operacje. Część z cyklicznie wykonywanych zadań administracyjnych da się w mniejszym lub większym stopniu zautomatyzować. W niniejszym artykule chciałbym poruszyć temat automatyzacji etykietowania woluminów.

Na czym polega automatyczne etykietowanie woluminów

Podczas wykonywania backupu demon magazynowania sięga do wyznaczonej mu puli woluminów, następnie wybiera odpowiedni wolumin i dokonuje na niego operacji zapisu danych backupu. Ważnym jest, aby w takiej puli istniał co najmniej jeden wolumin zdolny do zapisu na nim danych. Jeśli demon magazynowania nie znajdzie żadnego woluminu w puli (bo np. administrator zapomniał wcześniej stworzyć tam woluminy) lub woluminy istnieją w puli, lecz żaden z nich nie jest zdolny do ”przyjęcia” danych (np. wszystkie woluminy są oznaczone statusem **Full** oraz nie mają możliwości poddania się procesowi recyklingu) to Bacula wstrzyma backup i zgłosi żądanie przygotowania woluminu, na który mogłaby wykonać backup. Gdyby w tej sytuacji zastosować automatyczne etykietowanie woluminów, to Bacula nie wstrzymałaby backupu lecz automatycznie utworzyła nowy wolumin.

Zastosowanie

Automatyczne etykietowanie woluminów zostało zaprojektowane dla urządzeń plikowych. Nie ma większego sensu stosowanie tego rozwiązania dla napędu taśm magnetycznych, gdyż jego obsługa wymaga ręcznej interwencji zmiany taśmy oraz jej podmontowania, przez co automatyzacja nie ma w tym przypadku zastosowania. Natomiast urządzenia ze zmieniarką taśm nie są wcale wspierane dla automatycznego etykietowania.

Jak to działa

Do skonfigurowania automatycznego etykietowania woluminów w najprostszej postaci wystarczającym jest użycie dwóch dyrektyw:

Label Media = yes/no – dyrektywa definiowana dla zasobu **Device** w pliku konfiguracyjnym demona magazynowania. Jeśli jest ustawiona na **yes**, to nadaje urządzeniu zdolność do automatycznego etykietowania woluminów. Jeśli jest ustawiona na **no**, to urządzenie nie będzie zdolne do zautomatyzowanego tworzenia woluminów.

Label Format = ”ciąg_znaków” - dyrektywa definiowana w zasobie **Pool** w pliku konfiguracyjnym zarządcy (Director). Określa ciąg znaków, jaki zostanie użyty jako prefix w nazwie etykiety dla automatycznie tworzonych wolumenów. Dyrektywa **Label Format** może składać się z liter, cyfr i znaków specjalnych takich jak myślnik (-), podkreślenie (_), dwukropek (:) oraz kropka (.). Jeśli nie użyto żadnego z counter’ów (liczników), to etykiety wolumenów utworzonych automatycznie będą składać się z prefixu z doklejonym identyfikatorem wolumenu (**MediaId**) wraz z wiodącymi zerami, razem tworząc czterocyfrową liczbę.

Przykład

Konfiguracja urządzenia w pliku konfiguracyjnym demona magazynowania:

```
Device {
  Name = FileStorage
  Media Type = File
  Archive Device = /home/gani/bacula/urzadzenie1
  LabelMedia = yes
  Random Access = Yes
  AutomaticMount = yes
  RemovableMedia = no
}
```

Konfiguracja puli woluminów w pliku konfiguracyjnym zarządcy (Director):

```
Pool {
  Name = "Test Pool"
  Pool Type = Backup
  Recycle = no
  Maximum Volume Jobs = 1
  Maximum Volumes = 4
  Label Format = "Pliki-"
}
```

Wolumeny puli woluminów o nazwie **Test Pool** mają możliwość pomieszczenia czterech backupów, gdyż w powyższej konfiguracji każdy wolumin będzie mógł pomieścić dokładnie jeden backup, a maksymalna liczba woluminów w puli **Test Pool** ustawiona jest na cztery. W puli nie znajduje się żaden wolumin.

Wykonuję cztery backupy przy użyciu urządzenia **FileStorage** do puli **Test Pool**.

```
*run job=BackupClient1 pool="Test Pool" yes
*run job=BackupClient1 pool="Test Pool" yes
*run job=BackupClient1 pool="Test Pool" yes
*run job=BackupClient1 pool="Test Pool" yes
```

Po tych operacjach stan puli przedstawia się następująco:

```
+---------+------------+--------------+
| MediaId | VolumeName | VolumeStatus |
+---------+------------+--------------+
| 4       | Pliki-0004 | Used         |
| 5       | Pliki-0005 | Used         |
| 6       | Pliki-0006 | Used         |
| 7       | Pliki-0007 | Used         |
+---------+------------+--------------+
```

Powyższe cztery wolumeny zostały utworzone automatycznie, każdy w chwili uruchomienia kolejnego backupu. Ciekawym może wydać się fakt, że numeracja zaczyna się od liczby 4. Stało się tak, ponieważ oprócz puli **Test Pool** posiadam trzy inne pule woluminów, w których znajdują się taśmy (w każdej puli jedna taśma) przez co przy uruchomieniu pierwszego z wyżej wymienionych backupów kolejną wolną liczbą (identyfikatorem woluminu) był identyfikator o wartości 4.

Oto stan wszystkich moich puli woluminów:

Pool: **Sys-Full**:

```
+---------+------------+--------------+
| MediaId | VolumeName | VolumeStatus |
+---------+------------+--------------+
| 1       | Full-0001  | Append       |
+---------+------------+--------------+
```

Pool: **DB-Full**:

```
+---------+------------+--------------+
| MediaId | VolumeName | VolumeStatus |
+---------+------------+--------------+
| 2       | DB-0002    | Append       |
+---------+------------+--------------+
```

Pool: **Sys-Inc**:

```
+---------+------------+--------------+
| MediaId | VolumeName | VolumeStatus |
+---------+------------+--------------+
| 3       | Inc-0003   | Append       |
+---------+------------+--------------+
```

Pool: **Test Pool**:

```
+---------+------------+--------------+
| MediaId | VolumeName | VolumeStatus |
+---------+------------+--------------+
| 4       | Pliki-0004 | Used         |
| 5       | Pliki-0005 | Used         |
| 6       | Pliki-0006 | Used         |
| 7       | Pliki-0007 | Used         |
+---------+------------+--------------+
```

Powyższy przykład przedstawia najprostsze użycie automatycznego etykietowania woluminów. Może wydawać się nieco kłopotliwy, choćby z faktu, że numeracja etykiet pomieszała się pomiędzy pulami woluminów i przy większej ilości woluminów może stać się zupełnie niejasna. Z pomocą mogą przyjść countery.

Countery

Countery to swego rodzaju liczniki, których można używać indywidualnie dla każdej puli woluminów, przez co w każdej puli numeracja woluminów będzie taka, jaką sobie użytkownik zażyczy, np. numeracja od zera dla każdej z puli.

W pliku konfiguracyjnym serwisu zarządcy counter reprezentowany jest przez zasób o nazwie **Counter** i zawierać może następujące dyrektywy:

Name = nazwa_countera – nazwa licznika, jaką będzie można się posługiwać przy jego użyciu.

Minimum = liczba – minimalna wartość licznika. Innymi słowy jest to wartość, od której licznik zacznie zliczanie. Jeśli dyrektywa nie zostanie zdefiniowana, to domyślną jej wartością jest 0.

Maximum = liczba – maksymalna wartość licznika. Jest to górna wartość, do której licznik będzie zliczał. Domyślną wartością jest wartość 0. Maksymalna liczba jaką można zdefiniować w tej dyrektywie to 2147483648.

Catalog = nazwa - nazwa zasobu **Catalog** wskazującego bazę danych, do której licznik będzie zapisywał swoje wartości. Jeśli dyrektywa nie zostanie zdefiniowana, to licznik użyje bazy danych, z którą obecnie współpracuje Bacula.

Przykład

Posłużę się tą samą konfiguracją puli woluminów oraz urządzenia, która została przedstawiona w poprzednim rozdziale. Usuwam wszystkie taśmy z puli o nazwie **Test Pool** i definiuję licznik dla tej puli.

```
Counter {
  Name = LicznikTestowy
  Minimum = 0
  Maximum = 100
}
```

oraz używam licznika w puli Test Pool:

```
Pool {
  Name = "Test Pool"
  Pool Type = Backup
  Recycle = no
  Maximum Volume Jobs = 1
  Maximum Volumes = 4
  Label Format = "Pliki-${LicznikTestowy+}"
}
```

(znak + w użyciu licznika oznacza inkrementację)

Znów wykonuję 4 backupy z użyciem pustej puli woluminów:

```
*run job=BackupClient1 pool="Test Pool" yes
*run job=BackupClient1 pool="Test Pool" yes
*run job=BackupClient1 pool="Test Pool" yes
*run job=BackupClient1 pool="Test Pool" yes
```

Wynikiem tych operacji jest pojawienie się w puli **Test Pool** czterech nowych woluminów:

```
+---------+------------+--------------+
| MediaId | VolumeName | VolumeStatus |
+---------+------------+--------------+
| 4       | Pliki-0    | Used         |
| 5       | Pliki-1    | Used         |
| 6       | Pliki-2    | Used         |
| 7       | Pliki-3    | Used         |
+---------+------------+--------------+
```

Na listingu da się zauważyć różnicę pomiędzy poprzednim użyciem automatycznego etykietowania bez użycia coutera – brakuje wiodących zer.

W celu dodania wiodących zer posłużę się znakami specjalnymi, dzięki którym jest możliwe formatowanie etykiet powstałych automatycznie.

Oto kilka ze znaków specjalnych dla formatowania liczników:

- o - wycięcie części etykiety (tzw. substring),
- p - dodanie do zmiennej dodatkowych znaków,
- l - konwersja zmiennej na małe litery,
- u - konwersja zmiennej na duże litery.

Pełna lista wszystkich możliwych do użycia znaków specjalnych znajduje się pod adresem:

http://bacula.org/5.0.x-manuals/en/misc/misc/Variable_Expansion.html#SECTION00340000000000000000

Potrzebnym mi znakiem specjalnym jest litera **p**, która zgodnie z powyższym wyjaśnieniem doda żądane znaki do licznika. Moja konfiguracja ma teraz postać:

```
Counter {
  Name = LicznikTestowy2
  Minimum = 0
  Maximum = 100
}

Pool {
  Name = "Test Pool"
  Pool Type = Backup
  Recycle = no
  Maximum Volume Jobs = 1
  Maximum Volumes = 4
  Label Format = "Pliki-${LicznikTestowy2+:p/4/0/r}"
}
```

a zapis **${LicznikTestowy2+:p/4/0/r}** oznacza:

- p - dodaj brakujące znaki,
- 4 - maksymalna ilość znaków,
- 0 - zero jest moim znakiem wiodącym,
- r - dopisuj znaki wiodące od lewej strony (inaczej nie byłyby wiodące ;-)).

Znów usuwam wszystkie woluminy z puli **Test Pool** i wykonuję 4 backupy. Nowo stworzone woluminy mają postać:

```
+---------+------------+--------------+
| MediaId | VolumeName | VolumeStatus |
+---------+------------+--------------+
| 4       | Pliki-0000 | Used         |
| 5       | Pliki-0001 | Used         |
| 6       | Pliki-0002 | Used         |
| 7       | Pliki-0003 | Used         |
+---------+------------+--------------+
```

Rozszerzanie zmiennych

Oprócz prefiksów i cyfr składających się na automatycznie tworzone etykiety istnieje możliwość użycia wbudowanych zmiennych Bacula. I tak na przykład w nazwie automatycznie etykietowanych woluminów można zawrzeć nazwę klienta, nazwę zadania, pełną datę (lub jej część) oraz wiele innych wartości środowiskowych.

Przykład

```
Pool {
  Name = "Test Pool"
  Pool Type = Backup
  Recycle = no
  Maximum Volume Jobs = 1
  Maximum Volumes = 4
  Label Format = "Pliki-${Client}-${Storage}-${JobId}-${Job}"
}
```

Po wykonaniu czterech backupów o nazwie BackupClient1 do puli **Test Pool**, pula zawiera teraz następujące woluminy:

```
+---------+-------------------------------------------------+--------------+
| MediaId | VolumeName                                      | VolumeStatus |
+---------+-------------------------------------------------+--------------+
| 4       | Pliki-gani-desktop-fd-File-37-BackupClient1     | Used         |
| 5       | Pliki-gani-desktop-fd-File-38-BackupClient1     | Used         |
| 6       | Pliki-gani-desktop-fd-File-39-BackupClient1     | Used         |
| 7       | Pliki-gani-desktop-fd-File-40-BackupClient1     | Used         |
+---------+-------------------------------------------------+--------------+
```

gdzie nazwy woluminów składają się z:

- Pliki – prefix podany w dyrektywie Label Format,
- gani-desktop-fd – nazwa klienta (**File Daemon**),
- File – nazwa **Storage**,
- 37,38,39,40 – identyfikatory wykonanych zadań czyli tzw. **JobId**,
- BackupClient1 – nazwa backupu.

Pełna lista zmiennych środowiskowych Bacula znajduje się pod adresem:

http://bacula.org/5.0.x-manuals/en/misc/misc/Variable_Expansion.html#SECTION00320000000000000000

Podsumowanie

Jak zapewne czytelnik zauważył, automatyczne etykietowanie woluminów nie jest niczym trudnym do skonfigurowania (odpowiedzialne za to są - bagatela - dwie dyrektywy: **Label Media** oraz **Label Format**). Możliwość używania wbudowanych zmiennych udostępnia etykietowanie z wykorzystaniem nazw własnych (np. klienta, storage, media type itp.). Używanie couterów daje za to szansę na zapanowanie nad inkrementacją liczników i możliwość uporządkowania nazw wolumenów.

W chwili obecnej coraz popularniejszym sposobem na formatowanie nazw automatycznie tworzonych wolumenów jest użycie oskryptowania w języku Python, które udostępnia Bacula, lecz uważam, że w funkcjonalności modułu Python jest jeszcze dużo do zrobienia. Formatowanie etykiet opisane w tym artykule daje np. tą przewagę nad skryptami Python, że działa ”od ręki” jak się to mówi, bez potrzeby kompilacji Bacula z obsługą Python’a oraz instalacji pakietów Python.

Opisane przeze mnie przykłady tworzyły wolumeny po ręcznym uruchomieniu backupów. Do pełnej automatyzacji opisywanego procesu brakuje oczywiście harmonogramów zadań.