

Kopia przyrostowa w Bacula

16 kwiecień 2012 autor: **gani**

Artykuł prezentuje dogłębne spojrzenie na kopię przyrostową w Bacula. Kierowany jest zarówno do osób poznających backup przyrostowy jak i dla tych, którzy już go dobrze znają.

Wstęp

W niniejszym artykule chciałbym przyjrzeć się bliżej jednemu z typów kopii zapasowych obsługiwanych przez Bacula - **kopii przyrostowej**. Gdyby spojrzeć jedynie na ogólną definicję kopii przyrostowej, to temat ten nie wydaje się być niczym trudnym. Definicja ta mówi o tym, że **kopia przyrostowa** (ang. incremental backup) polega na zapisaniu kopii danych, które zmieniły się od czasu wykonania kopii dowolnego typu (pełna, przyrostowa, różnicowa). W odniesieniu do Bacula znaczenie tej definicji nieco się rozszerza.

Definiowanie kopii przyrostowej

Za określenie kopii przyrostowej odpowiada dyrektywa **Level** z wartością **Incremental**. Np. dla zadania o nazwie *Katalogi web* może to wyglądać jak poniżej.

```
Job {
    Name = „Katalogi web”
    Type = Backup
    Level = Incremental
    ...
}
```

(trzy kropki oznaczają, że jest to wycinek zasobu)

Poziom backupu można również nadpisywać poprzez **harmonogram zadań** (ang. schedule) lub bezpośrednio z linii komend tekstowej konsoli Bacula o nazwie **bconsole**.

Warunki wykonania kopii przyrostowej

Istnieje pięć wytycznych, które po kolei są sprawdzane przez Bacula przed wykonaniem backupu przyrostowego. Jeśli którykolwiek z nich nie zostanie spełniony to backup przyrostowy zostanie przełączony w backup pełny (ang. full backup). Są to:

1. Ta sama nazwa backupu

Zadania backupu w Bacula są definiowane w zasobie **Job**, np. dla backupu o nazwie *Katalog domowy* zasób **Job** może wyglądać jak poniżej:

```
Job {
    Name = „Katalog domowy”
    ...
}
```

(trzy kropki oznaczają, że jest to wycinek zasobu)

Jeśli zostanie wykonana **kopia pełna** z zadania *Katalog domowy*, a następnie zostanie zmieniona nazwa zadania na np. *Katalog Jacka* i nastąpi próba wykonania backupu przyrostowego, to w konsekwencji tego pierwsza kopia przyrostowa z zadania *Katalog Jacka* zostanie przełączona w kopię pełną.

2. Istnieje poprawnie wykonana kopia pełna, przyrostowa lub różnicowa

Każda kopia przyrostowa musi mieć swój **backup bazowy** czyli taki, który będzie punktem odniesienia dla kopii przyrostowej. Nazwa „przyrostowa” wskazuje, że dane przyrastają, lecz zanim to się stanie, to musi istnieć punkt odniesienia, względem którego nastąpi „przyrost”. Tym punktem odniesienia może być kopia pełna (ang. **full backup**), kopia różnicowa (ang. **differential backup**) lub inna **kopia przyrostowa** tego samego zadania.

Ważne jest, aby backup bazowy dla kopii przyrostowej był poprawnie zakończony. Jeśli backup bazowy będzie zakończony błędem lub anulowany, to wykonanie kopii przyrostowej z takiego backupu będzie skutkowało przełączeniem się kopii przyrostowej w kopię pełną.

3. Taki sam klient

W zasobie Job określa się również komputer klienta, z którego chce wykonać się kopię przyrostową. Np. dla komputera zdefiniowanego w zasobie **Client** o nazwie *serwer-1* będzie to:

```
Job {
  Name = „Katalog domowy”
  Client = „serwer-1”
  ...
}
```

(trzy kropki oznaczają, że jest to wycinek zasobu)

Jeśli po wykonaniu kopii dowolnego typu (**full**, **incremental**, **differential**) z klienta *serwer-1* nastąpi zmiana klienta na innego np.:

```
Job {
  Name = „Katalog domowy”
  Client = „serwer-77”
  ...
}
```

(trzy kropki oznaczają, że jest to wycinek zasobu)

to wykonanie kopii przyrostowej przełączy ją w kopię pełną. Dopiero po wykonaniu kopii pełnej, możliwe będzie wykonanie kopii przyrostowej.

4. Zasób FileSet nie uległ zmianie

W zasobie **FileSet** definiowane są m.in. lokalizacje, z których zadanie backupu czerpie informacje o tym, gdzie na komputerze klienta leżą dane, którymi klient ma się „zająć”. Np.:

```
FileSet {
  Name = „Dokumenty”
  Include {
    File = „/home/jacek/faktury”
    File = „/home/jacek/wydatki”
  }
}
```

Jakakolwiek zmiana lokalizacji skutkuje tym, że przy próbie wykonania kopii przyrostowej, backup przełączy się w kopię pełną.

5. Kopia przyrostowa wykonywana jest nie dłużej niż czas Max Full Interval

Dyrektywa Max Full Interval określa odstęp czasowy, po upływie którego kopia zostanie przełączona automatycznie w kopię pełną. Jeśli dyrektywa ta została użyta w definicji zasobu Job to kopia przyrostowa wykonana po upływie czasu określonego w Max Full Interval przełączy się automatycznie w kopię pełną.

Zasada wybierania plików do kopii przyrostowej

Podczas wykonywania backupu Bacula do selekcji plików bazuje na czasach modyfikacji plików oraz na czasach modyfikacji ich atrybutów. Są to czasy nazywane kolejno **MTIME** (ang. modification time) i **CTIME** (ang. change time) i porównywane są z czasem zakończenia backupu bazowego dla backupu przyrostowego. Jeśli czas zakończenia backupu bazowego dla backupu przyrostowego jest mniejszy niż czas **MTIME** lub **CTIME** pliku, wtedy plik zostanie zapisany w kopii przyrostowej.

Przykład

Ostatni backup pełny pliku o nazwie ml.pl wykonał się dnia 2012-04-15 o godzinie 01:00:54.

Czas MTIME pliku *ml.pl* to 2011-12-26 01:25:30.

Czas CTIME pliku *ml.pl* to 2012-04-16 02:58:20.

Plik ml.pl zostanie uwzględniony w kopii przyrostowej gdyż jego czas CTIME jest nowszy niż czas zakończenia ostatniego pełnego backupu pliku o *ml.pl*.

UWAGA!

Niektóre programy antywirusowe mogą modyfikować czas **CTIME**. Takie działanie może zakłócić wykonywanie kopii Baculi. Warto więc upewnić się, że używany program antywirusowy nie zachowuje się w ten sposób. Jeśli natomiast program antywirusowy modyfikuje **CTIME**, to jednym z rozwiązań może być sprawdzenie w dokumentacji programu antywirusowego, czy istnieje możliwość wyłączenia tej opcji.

NOTKA

Czasy plików **MTIME** i **CTIME** można sprawdzić np. przy użyciu komendy stat w następujący sposób:

```
$ stat ml.pl
  File: `ml.pl'
  Size: 150          Blocks: 8          IO Block: 4096   zwykły plik
Device: 822h/2082d  Inode: 659380       Links: 1
Access: (0775/-rwxrwxr-x)  Uid: ( 1000/      gani)   Gid: ( 1000/      gani)
Access: 2011-12-26 01:24:54.827116028 +0100
Modify: 2011-12-26 01:25:30.511116025 +0100
Change: 2012-04-16 02:58:20.506822999 +0200
```

Dwie ostatnie linie zawierają czasy MTIME i CTIME dla pliku *ml.pl*.

Podsumowanie

Zachęcam do własnych eksperymentów z backupem przyrostowym. Myślę, że mimo kilku jego wad (jak choćby bardziej skomplikowane odtwarzanie niż w przypadku odtwarzania samej kopii pełnej) jest wart zainteresowania.

Opisując backup przyrostowy w Bacula ominąłem temat kopii przyrostowej z użyciem accurate backup ze względu na to, że został on opisany w artykule pt. [Accurate Backup - sposób na dokładną kopię bezpieczeństwa](#).