Backup i odtwarzanie w Cassandrze

Poziom: średni **Dotyczy: Apache Cassandra 3.x / 4.x** **Czas: ~20 min**

Backup w Cassandrze wykonujemy per węzeł, nie per klaster — co jest naturalną konsekwencją tego, że żaden pojedynczy węzeł nie przechowuje pełnego zbioru danych.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
CQL	Cassandra Query Language: język zapytań Cassandry, podobny do SQL
CSV	w Windowsie: Cluster Shared Volume, dysk widoczny naraz dla wszystkich węzłów klastra; w danych: Comma-Separated Values, plik z wartościami oddzielonymi przecinkami
RF	Replication Factor: liczba kopii tego samego wiersza trzymanych naraz w klastrze Cassandry

Fiszka

ZMIENIA SYSTEM

Utworzenie migawki danych węzła: nodetool snapshot

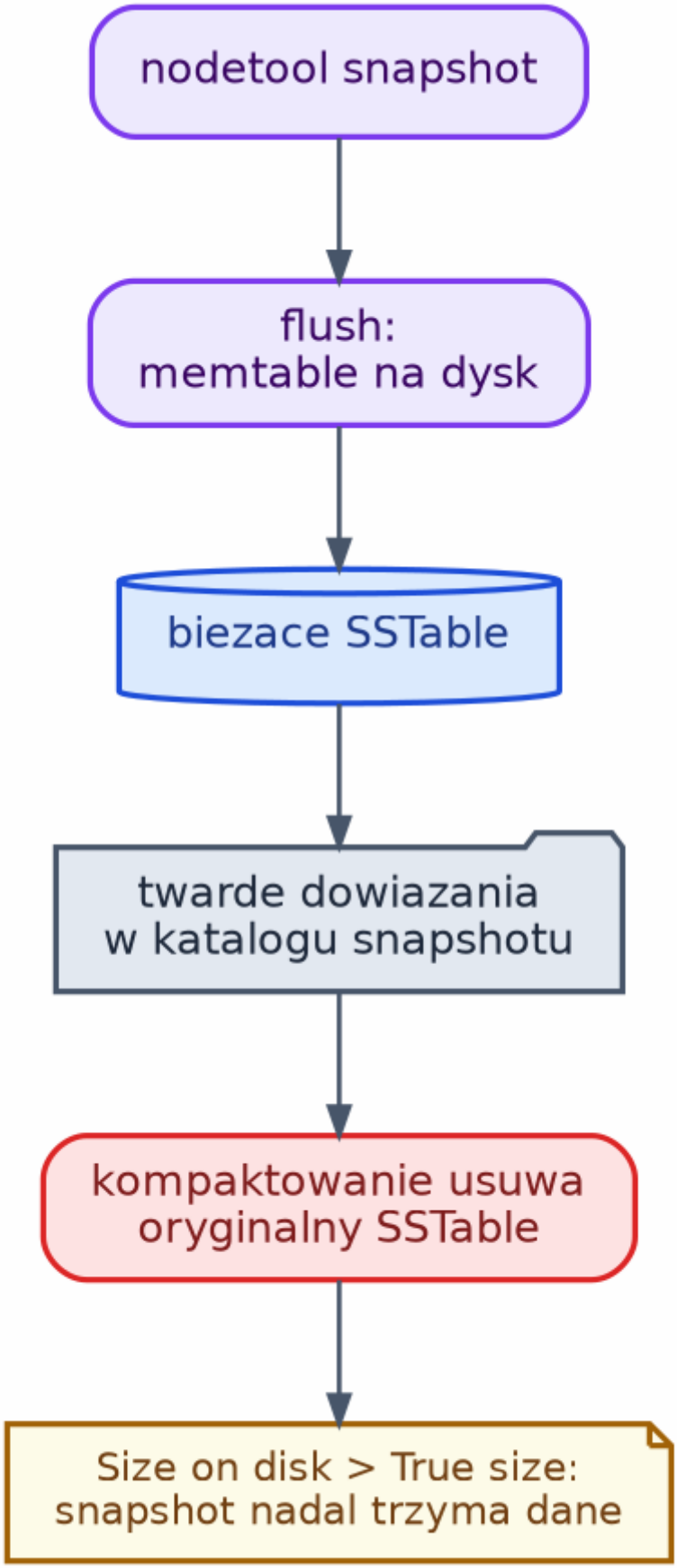
```
sudo nodetool snapshot -t NazwaSnapshota NazwaKeyspace
```

Tworzymy snapshot wskazanego keyspace (albo wszystkich, jeśli nazwę keyspace pominiemy) pod nazwą przekazaną flagą `-t` — bez niej Cassandra nadaje nazwę na podstawie znacznika czasu.

Przykład:

```
Requested creating snapshot(s) for [aplikacja] with snapshot name [2026-03-14-0300] and
options {skipFlush=false}
Snapshot directory: 2026-03-14-0300
```

Domyślnie `nodetool snapshot` wykonuje najpierw `flush`, opisany na stronie nodetool: stan klastra i operacje administracyjne — snapshot obejmuje więc też dane, które w chwili wywołania leżały jeszcze w memtable, nie tylko już zapisane SSTable.


nodetool snapshot
flush:
memtable na dysk
biezace SSTable
twarde dowiazania
w katalogu snapshotu
kompaktowanie usuwa
oryginalny SSTable
Size on disk > True size:
snapshot nadal trzyma dane

□□ *Snapshot jako twarde dowiązania do SSTable: po kompaktowaniu oryginału snapshot nadal trzyma dane na dysku (kliknięcie otwiera pełny rozmiar)*

□□ **nodetool** = narzędzie administracyjne węzła Cassandry

□□ TYLKO ODCZYT

Wylistowanie istniejących migawek: nodetool listsnapshots

```
sudo nodetool listsnapshots
```

Zwracamy wszystkie snapshoty przechowywane obecnie na węźle, wraz z ich rozmiarem rzeczywistym i rozmiarem zajmowanym na dysku.

Przykład:

```
Snapshot Details:
Snapshot name  Keyspace name  Column family name      True size  Size on disk
2026-03-14-0300 aplikacja     klienci                   1.03 KiB      1.07 KiB
2026-03-14-0300 aplikacja     zamowienia_wg_klienta    233.92 MiB      4.73 GiB
2026-03-01-0300 aplikacja     klienci                   1.03 KiB      1.07 KiB
2026-03-01-0300 aplikacja     zamowienia_wg_klienta     59.07 MiB    116.14 MiB
```

Kolumna `Size on disk` bywa znacznie większa niż `True size` dla starszych snapshotów — różnica to dane, które formalnie należą już tylko do snapshotu, bo oryginalne SSTable, z którym dzielił dowiązanie, zostało w międzyczasie usunięte przez kompaktowanie.

□□ **nodetool** = narzędzie administracyjne węzła Cassandry

□□ TYLKO ODCZYT

Usunięcie migawki: nodetool clearsnapshot

```
sudo nodetool clearsnapshot -t NazwaSnapshota
```

Usuwamy wskazany snapshot, zwalniając miejsce zajmowane wyłącznie przez niego — bez argumentu `-t` komenda usuwa wszystkie snapshoty na węźle, co bywa łatwym sposobem na odzyskanie miejsca po serii zapomnianych migawek testowych.

□□ **nodetool** = narzędzie administracyjne węzła Cassandry

TYLKO ODCZYT

Przesłanie plików SSTable do żyjącego klastra: sstableloader

```
sstableloader -d 10.20.10.36,10.20.10.37,10.20.10.38
/var/lib/cassandra/data/NazwaKeyspace/NazwaTabeli/
```

> **Pojęcie: sstableloader.** Narzędzie strumieniujące zawartość plików SSTable bezpośrednio do żyjącego klastra przez normalny mechanizm replikacji — w przeciwieństwie do kopiowania plików na miejsce, `sstableloader` nie wymaga, żeby topologia klastra docelowego pokrywała się z topologią klastra źródłowego, bo dane trafiają tam, gdzie należą zgodnie z aktualnymi tokenami klastra docelowego, a nie tam, gdzie leżały wcześniej. Sensowny sposób przeniesienia danych między klastrami o różnej liczbie węzłów albo odtworzenia pojedynczej tabeli bez odtwarzania całego węzła.

sstableloader = narzędzie strumieniujące pliki SSTable Cassandry bezpośrednio do żyjącego klastra przez normalny mechanizm replikacji

Z życia wzięte

Z ŻYCIA WZIĘTE

Przed dużą aktualizacją schematu telemetrii robimy snapshot wszystkich trzech węzłów cass01 do cass03 na wszelki wypadek. nodetool snapshot wykonuje najpierw flush, więc migawka obejmuje też dane, które w tej chwili leżały jeszcze w pamięci, nie tylko to, co już trafiło na dysk jako SSTable. Aktualizacja przechodzi bez problemu, ale tydzień później zespół Oli pyta, czy można bezpiecznie skasować stare snapshoty, żeby odzyskać miejsce na dysku zajęte przez trzy migawki, których nikt już nie potrzebuje.

Po co to istnieje

Ogólne rozróżnienie backupu pełnego, różnicowego, przyrostowego i logów opisuje strona Backup pełny, różnicowy, przyrostowy, logi — mechanizmy tam opisane działają na poziomie pojedynczej instancji z pełnym zbiorem danych. W Cassandrze każdy węzeł przechowuje tylko swój wycinek danych (i repliki wycinków sąsiadów, zgodnie z replication factor), więc backup węzła nigdy nie jest

backupem całego zbioru danych klastra — jest backupem tego, co ten konkretny węzeł akurat przechowuje.

Snapshot: twarde dowiązania do SSTable

> **Pojęcie: snapshot w Cassandrze.** Migawka stanu danych węzła w danej chwili, utworzona nie przez kopiowanie plików, tylko przez utworzenie twardych dowiązań (hard links) do istniejących już plików SSTable opisanych na stronie Utrzymanie klastra: repair, compaction, czyszczenie. Ponieważ SSTable nigdy nie są modyfikowane w miejscu — kompaktowanie zawsze tworzy nowe pliki i usuwa stare — twarde dowiązanie do istniejącego SSTable pozostaje spójne nawet wtedy, gdy oryginalny plik zostanie później usunięty przez kompaktowanie: system plików zwolni go dopiero, gdy zniknie też dowiązanie w katalogu snapshotu. Dzięki temu snapshot powstaje niemal natychmiastowo, niezależnie od rozmiaru danych, i nie zajmuje dodatkowego miejsca na dysku w chwili utworzenia — miejsce zaczyna być realnie zużywane dopiero, gdy kompaktowanie chciałoby usunąć plik, do którego snapshot wciąż się odwołuje.

Backupy przyrostowe

> **Pojęcie: incremental backups.** Mechanizm włączany parametrem `incremental_backups: true` w `cassandra.yaml`, opisanym na stronie Konfiguracja i diagnostyka — po włączeniu, każdy nowo utworzony SSTable (czyli każdy zrzut memtable) dostaje automatycznie dodatkowe twarde dowiązanie w podkatalogu `backups/` obok właściwych danych tabeli. W przeciwieństwie do snapshotu, który jest migawką jednorazową wykonaną na żądanie, backup przyrostowy narasta w tle, cały czas, i wymaga własnego zewnętrznego mechanizmu kopiowania tych plików poza węzeł oraz czyszczenia katalogu `backups/` po wykonaniu takiej kopii — sam mechanizm w Cassandrze nic z tymi plikami dalej nie robi.

Odtwarzanie danych

Kopiowanie plików SSTable na miejsce. Najprostszy sposób odtworzenia węzła ze snapshotu — zatrzymujemy usługę `cassandra` na docelowym węźle, kopiujemy pliki snapshotu z powrotem do właściwego katalogu danych tabeli (`data_file_directories`, opisany na stronie Konfiguracja i diagnostyka), zastępując to, co tam obecnie leży, i uruchamiamy usługę ponownie. Jeśli usługa musi pozostać uruchomiona, alternatywą jest skopiowanie plików do katalogu danych i wywołanie `nodetool refresh NazwaKeyspace NazwaTabeli` — komenda każe węzłowi ponownie przeskanować katalog danych tabeli i załadować nowo pojawione SSTable bez restartu.

Kopiowanie mniejszych zbiorów danych

Dla pojedynczych tabel niewielkiego rozmiaru pełny mechanizm snapshotu i `sstableloader` bywa nadmiarowy — `COPY TO` i `COPY FROM` w `cqlsh`, opisane na stronie cqlsh: klient wiersza poleceń, pozwalają wyeksportować albo zaimportować dane tabeli jako zwykły plik CSV, kosztem tego, że operują przez zwykłe zapytania CQL, wiersz po wierszu, więc nie skalują się do wolumenów, dla których sensowny jest `sstableloader`.

Dlaczego backup jest per węzeł

Snapshot wykonany na jednym węźle zawiera wyłącznie te partycje, za które ten węzeł jest odpowiedzialny — swoje własne oraz te, które przechowuje jako replika sąsiadów, zgodnie z replication factor opisanym na stronie Architektura Cassandry: pierścień, węzeł, replikacja. Odtworzenie spójnego stanu całego klastra ze snapshotów wymaga więc jednej z dwóch rzeczy: albo odtworzenia snapshotów wszystkich węzłów naraz, wykonanych w tym samym momencie (albo wystarczająco blisko siebie w czasie, żeby różnica nie miała znaczenia biznesowego), albo odtworzenia jednego węzła i pozwolenia mechanizmom uzgadniania stanu — read repair i `nodetool repair`, opisanym na stronie Architektura Cassandry: pierścień, węzeł, replikacja — dociągnąć go do zgodności z resztą żyjącego klastra. Odtworzenie pojedynczego węzła ze starego snapshotu bez późniejszego repair pozostawia go z danymi cofniętymi w czasie względem reszty klastra — read repair z czasem to skoryguje, ale do tego momentu odczyty z poziomem spójności `ONE` trafiające akurat na ten węzeł mogą zwrócić dane nieaktualne.

Pełne omówienie backupu pełnego, różnicowego, przyrostowego i logów transakcyjnych, wspólne dla wszystkich silników tej półki, prowadzi strona Backup pełny, różnicowy, przyrostowy, logi.

Częste błędy

Odtworzenie snapshotu jednego węzła bez uruchomienia repair. Węzeł wraca do klastra z danymi sprzed momentu snapshotu — bez repair pozostanie w tyle za resztą klastra na czas zależny wyłącznie od tego, kiedy read repair przypadkiem trafi na jego rozjechane wiersze.

Traktowanie `incremental_backups` jako kompletnego rozwiązania backupowego. Mechanizm tylko tworzy dowiązania lokalnie na węźle — bez zewnętrznego procesu kopiującego te pliki poza węzeł i poza jego dysk, awaria samego dysku zabiera backup razem z danymi właściwymi.

Zapominanie o `nodetool clearsnapshot`. Snapshoty testowe zostawione na produkcyjnym węźle rosną wraz z każdym kompaktowaniem, które próbuje usunąć plik wciąż trzymany przez zapomniany snapshot — objawia się to jako zużycie dysku rosnące szybciej, niż wynikałoby to z faktycznego wolumenu danych.

□□ Ćwiczenia

> Pytania sprawdzają rozumienie backupu rozproszonego, nie wymagają uruchomionego klastra.

Pytania kontrolne

1. Dlaczego snapshot Cassandry powstaje niemal natychmiastowo, niezależnie od rozmiaru danych na węźle?

Odpowiedź

Bo nie kopiuje danych — tworzy twarde dowiązania do już istniejących plików SSTable. Koszt czasowy takiej operacji nie zależy od rozmiaru danych, tylko od liczby plików, do których trzeba utworzyć dowiązanie.

2. Klaster ma trzy węzły i RF = 3 (każdy węzeł ma pełną kopię danych). Klaster ma pięć węzłów i RF = 3 (żaden węzeł nie ma pełnej kopii). W którym przypadku snapshot pojedynczego węzła wystarczy do odtworzenia całego zbioru danych, a w którym nie?

Odpowiedź

Przy trzech węzłach i RF = 3 każdy węzeł faktycznie przechowuje kopię całego zbioru danych, więc snapshot jednego z nich technicznie zawiera wszystko. Przy pięciu węzłach i RF = 3 żaden pojedynczy węzeł nie ma pełnego zbioru — snapshot obejmuje tylko partycje, dla których ten węzeł jest jedną z replik, więc odtworzenie całości wymaga snapshotów ze wszystkich pięciu węzłów.

Powiązane: Backup pełny, różnicowy, przyrostowy, logi · Architektura Cassandry: pierścień, węzeł, replikacja · nodetool: stan klastra i operacje administracyjne · cqlsh: klient wiersza poleceń

Źródła: dokumentacja Apache Cassandra, rozdz. „Backups" · `nodetool help snapshot`

Revision #42
Created 2026-09-24 23:34:57 UTC by Tomasz Konieczny
Updated 2026-10-26 14:13:17 UTC by Tomasz Konieczny