Role, uprawnienia i monitoring JMX

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

Poza rolami odpytywanymi CQL-em Cassandra udostępnia też uprawnienia do własnych metryk i operacji przez JMX — a monitoring zewnętrzny (Zabbix, Prometheus) potrzebuje własnego konta z dokładnie tymi uprawnieniami, których potrzebuje, i żadnych innych.

Skróty użyte na tej stronie (2)

Skrót	Co znaczy
JMX	Java Management Extensions: mechanizm Javy do odpytywania i sterowania działającą aplikacją z zewnątrz, tędy Cassandra wystawia swoje metryki
JVM	Java Virtual Machine: maszyna wirtualna, w której działają programy w Javie

Fiszka

ZMIENIA SYSTEM

Utworzenie roli wyłącznie jako zbioru uprawnień: CREATE ROLE bez logowania

```
CREATE ROLE jmx_monitoring WITH LOGIN = false;
```

Tworzymy rolę, która nigdy nie posłuży do zalogowania się do bazy — istnieje wyłącznie po to, żeby zgromadzić pod jedną nazwą zestaw uprawnień do MBeanów JMX, opisanych niżej, i nadać je później kontu monitorującemu jednym poleceniem `GRANT`.

JMX = Java Management Extensions: mechanizm Javy do odpytywania i sterowania działającą aplikacją z zewnątrz, tędy Cassandra wystawia swoje metryki

ZMIENIA SYSTEM

Utworzenie roli logującej się z hasłem: CREATE ROLE z hasłem

```
CREATE ROLE zbx_monitor WITH PASSWORD = 'HasloUzytkownika' AND LOGIN = true AND
SUPERUSER = false;
```

Tworzymy konto, którym faktycznie będzie logować się zewnętrzny system monitoringu — `SUPERUSER = false` jest tu istotne z zasady najmniejszych uprawnień: konto monitoringu nie potrzebuje pełnych uprawnień administratora bazy, potrzebuje wyłącznie dostępu do odczytu metryk przez JMX, nadanego osobno poniższymi poleceniami.

JMX = Java Management Extensions: mechanizm Javy do odpytywania i sterowania działającą aplikacją z zewnątrz, tędy Cassandra wystawia swoje metryki

ZMIENIA SYSTEM

Nadanie uprawnienia do odczytu metryk JMX: GRANT SELECT ON ALL MBEANS

```
GRANT SELECT ON ALL MBEANS TO jmx_monitoring;
```

> **Pojęcie: MBean.** Obiekt zarządzalny udostępniany przez JVM przez interfejs JMX (Java Management Extensions) — Cassandra wystawia przez MBeany większość swoich metryk operacyjnych (te same dane, które widzimy przez `nodetool`) oraz część operacji administracyjnych. `SELECT` na MBeanach odpowiada odczytowi ich atrybutów — dokładnie temu, czego potrzebuje system monitoringu odpytujący metryki cyklicznie.

JMX = Java Management Extensions: mechanizm Javy do odpytywania i sterowania działającą aplikacją z zewnątrz, tędy Cassandra wystawia swoje metryki · **MBean** = obiekt zarządzalny wystawiony przez aplikację javową przez JMX, przez niego odczytujemy metryki albo wywołujemy operacje administracyjne · **JVM** = Java Virtual Machine: maszyna wirtualna, w której działają programy w Javie

ZMIENIA SYSTEM

Nadanie uprawnienia do odczytu diagnostyki JVM: GRANT EXECUTE ON MBEAN HotSpotDiagnostic

```
GRANT EXECUTE ON MBEAN 'com.sun.management:type=HotSpotDiagnostic' TO jmx_monitoring;
```

Nadajemy uprawnienie do wywołania konkretnej operacji na konkretnym MBeanie — w tym przypadku diagnostyki samej maszyny wirtualnej Javy, niezależnej od Cassandry jako takiej. W przeciwieństwie do `SELECT ON ALL MBEANS`, `GRANT EXECUTE` na pojedynczym MBeanie jest precyzyjne co do jednej konkretnej operacji, zgodnie z zasadą najmniejszych uprawnień.

JVM = Java Virtual Machine: maszyna wirtualna, w której działają programy w Javie

ZMIENIA SYSTEM

Nadanie uprawnienia do odczytu informacji o snitchu: GRANT EXECUTE ON MBEAN EndpointSnitchInfo

```
GRANT EXECUTE ON MBEAN 'org.apache.cassandra.db:type=EndpointSnitchInfo' TO
jmx_monitoring;
```

Pozwalamy koncie monitorującemu odpytać informacje o snitchu opisanym na stronie Architektura Cassandry: pierścień, węzeł, replikacja — przynależność węzłów do racków i datacenter, którą wiele dashboardów monitoringu pokazuje obok samych metryk liczbowych.

ZMIENIA SYSTEM

Nadanie uprawnienia do odczytu wątków JVM: GRANT EXECUTE ON MBEAN Threading

```
GRANT EXECUTE ON MBEAN 'java.lang:type=Threading' TO jmx_monitoring;
```

Pozwalamy odczytać stan wątków JVM — liczbę aktywnych wątków, ewentualne zakleszczenia — uzupełniająco wobec metryk specyficznych dla Cassandry.

JVM = Java Virtual Machine: maszyna wirtualna, w której działają programy w Javie

ZMIENIA SYSTEM

Nadanie uprawnienia do operacji na całej usłudze: GRANT EXECUTE ON MBEAN StorageService

```
GRANT EXECUTE ON MBEAN 'org.apache.cassandra.db:type=StorageService' TO jmx_monitoring;
```

`StorageService` jest MBeanem, przez który `nodetool` faktycznie komunikuje się z węzłem — nadanie uprawnienia do niego pozwala kontu monitorującemu wywoływać te same operacje

odczytowe, które wykonuje `nodetool status` czy `nodetool info`, bez potrzeby logowania się jako pełny administrator.

⚠ ZMIENIA SYSTEM

Nadanie uprawnienia do przeglądania listy MBeanów: GRANT DESCRIBE ON ALL MBEANS

```
GRANT DESCRIBE ON ALL MBEANS TO jmx_monitoring;
```

Pozwalamy koncie odkrywać, jakie MBeany w ogóle istnieją i jakie mają atrybuty i operacje — przydatne dla systemów monitoringu, które same automatycznie wykrywają dostępne metryki, zamiast mieć je zaszyte na sztywno w konfiguracji.

⚠ ZMIENIA SYSTEM

Przypisanie zestawu uprawnień do konta logującego się: GRANT roli do roli

```
GRANT jmx_monitoring TO zbx_monitor;
```

Łączymy zbiór uprawnień zgromadzony pod rolą `jmx_monitoring` z kontem `zbx_monitor`, którym faktycznie loguje się system monitoringu — od tego momentu `zbx_monitor` dziedziczy wszystkie uprawnienia nadane wcześniej `jmx_monitoring`, bez potrzeby powtarzania sześciu osobnych poleceń `GRANT` dla każdego kolejnego konta monitorującego, jeśli powstanie taka potrzeba.

Cała sekwencja wykonana w jednej sesji `cqlsh` wygląda tak:

```
dba@cqlsh> CREATE ROLE jmx_monitoring WITH LOGIN = false;
dba@cqlsh> GRANT SELECT ON ALL MBEANS TO jmx_monitoring;
dba@cqlsh> GRANT EXECUTE ON MBEAN 'com.sun.management:type=HotSpotDiagnostic' TO
jmx_monitoring;
dba@cqlsh> GRANT EXECUTE ON MBEAN 'org.apache.cassandra.db:type=EndpointSnitchInfo' TO
jmx_monitoring;
dba@cqlsh> GRANT EXECUTE ON MBEAN 'java.lang:type=Threading' TO jmx_monitoring;
dba@cqlsh> GRANT EXECUTE ON MBEAN 'org.apache.cassandra.db:type=StorageService' TO
jmx_monitoring;
dba@cqlsh> GRANT DESCRIBE ON ALL MBEANS TO jmx_monitoring;
```

```
dba@cqlsh> CREATE ROLE zbx_monitor WITH PASSWORD = 'hasło' AND LOGIN = true AND
SUPERUSER = false;
dba@cqlsh> GRANT jmx_monitoring TO zbx_monitor;
```

Rozbicie na rolę pośrednią (`jmx_monitoring`) i konto logujące się (`zbx_monitor`) nie jest konieczne — te same sześć uprawnień dałoby się nadać wprost kontu `zbx_monitor`. Ma ono jednak sens tam, gdzie tego samego zestawu uprawnień do JMX będzie potrzebować więcej niż jedno konto (na przykład osobne konta dla różnych narzędzi monitorujących) — wtedy zmiana zestawu uprawnień w jednym miejscu (`jmx_monitoring`) obejmuje automatycznie wszystkie konta, które go dziedziczą.

JMX = Java Management Extensions: mechanizm Javy do odpytywania i sterowania działającą aplikacją z zewnątrz, tędy Cassandra wystawia swoje metryki

Z życia wzięte

Z ŻYCIA WZIĘTE

Zespół monitoringu chce podłączyć Zabbixa do klastra cass01 do cass03, żeby widzieć te same metryki, które administrator widzi ręcznie przez nodetool. Zamiast dawać kontu monitorującemu pełne uprawnienia superusera, zakładamy rolę zbx_monitor bez SUPERUSER i nadajemy jej dokładnie SELECT ON ALL MBEANS, tyle, ile potrzebuje do odczytu metryk przez JMX. Miesiąc później hasło tego konta trafia przypadkiem do publicznego repozytorium konfiguracji Zabbixa, ale ryzyko ogranicza się do odczytu metryk, nie do administracji klastrem.

Role w Cassandrze

Pojęcie: rola (role) w Cassandrze. Odpowiednik konta użytkownika i jednocześnie grupy uprawnień naraz — tak jak w PostgreSQL, gdzie rola i użytkownik są tym samym obiektem, co opisuje ogólnie strona Konta, role i uprawnienia: GRANT, REVOKE, zasada najmniejszych uprawnień. Rola w Cassandrze może służyć do logowania (`LOGIN = true`, z hasłem) albo istnieć wyłącznie jako zbiór uprawnień nadawany innym rolom (`LOGIN = false`) — ten drugi wariant jest wygodnym sposobem grupowania powiązanych uprawnień pod jedną nazwą, zamiast nadawać je wszystkie osobno przy każdym koncie, które ich potrzebuje.

Częste błędy

Nadanie kontu monitorującemu roli `SUPERUSER`. Konto, które ma wyłącznie odczytywać metryki, nie potrzebuje uprawnień administracyjnych do danych — nadanie `SUPERUSER` łamie zasadę najmniejszych uprawnień bez żadnej korzyści dla samego monitoringu.

Nadawanie uprawnień wprost kontu logującemu się zamiast przez rolę pośrednią, gdy z tego samego zestawu skorzysta więcej niż jedno konto. Prowadzi do rozjazdu uprawnień między kontami, które powinny mieć identyczny dostęp, gdy ktoś zapomni powtórzyć jedno z poleceń `GRANT` przy tworzeniu drugiego konta.

Powiązane: Konta, role i uprawnienia: GRANT, REVOKE, zasada najmniejszych uprawnień · Architektura Cassandry: pierścień, węzeł, replikacja · nodetool: stan klastra i operacje administracyjne

Źródła: dokumentacja Apache Cassandra, rozdz. „Security" i „JMX" · notatki własne z konfiguracji monitoringu Zabbix

---

Revision #42
Created 2026-09-24 23:34:59 UTC by Tomasz Konieczny
Updated 2026-10-26 14:13:21 UTC by Tomasz Konieczny