Konto monitorujące dla Zabbixa

□□ Typ: procedura **□□ Dotyczy: Oracle Database 19c, szablon Zabbix Oracle**

□ Czas: ~20 min

Efekt: konto `ZABBIX` (albo `C##ZABBIX` w bazie kontenerowej) z odczytem do widoków wydajnościowych i słownikowych, wystarczającym dla oficjalnego szablonu monitoringu Zabbixa, bez uprawnień do modyfikacji czegokolwiek w bazie.

□□ Skróty użyte na tej stronie (5)

Skrót	Co znaczy
CDB	Container Database: baza-kontener w Oracle 12c i nowszych, trzyma w sobie bazy PDB
ODBC	Open Database Connectivity: uniwersalny sposób łączenia się z różnymi bazami danych tym samym interfejsem
PDB	Pluggable Database: baza wpinana do kontenera CDB w Oracle
README	Read Me: plik tekstowy z opisem projektu, wykresu lub szablonu, czytany jako pierwszy
WMS	Warehouse Management System: system zarządzania magazynem

Zanim zaczniemy

> **□□ Z ŻYCIA WZIĘTE**
>
> Zanim zespół dorzuci ora01 do Zabbixa, Marek pyta, czy monitoring nie spowolni WMS w godzinach szczytu. Tomek zakłada konto ZABBIX z samym `SELECT_CATALOG_ROLE` i garścią uprawnień do widoków `v$`, bez żadnego prawa zapisu. Kiedy szablon zaczyna pobierać metryki co minutę, obciążenie instancji WMSPRD nie zmienia się zauważalnie.

Wymaganie	Jak sprawdzić
Dostęp jako sysdba do instancji docelowej	`sqlplus / as sysdba`
Ustalone hasło dla konta ZABBIX	wg polityki haseł ustalonej dla kont technicznych

Wymaganie	Jak sprawdzić
Wiedza, czy baza jest klasyczna czy kontenerowa (CDB)	`SELECT cdb FROM v$database;`

Uważaj: konto monitorujące dostaje wyłącznie uprawnienia `SELECT` i rolę `SELECT_CATALOG_ROLE` — nigdy `DBA` ani uprawnień do zapisu. Szablon Zabbixa tylko odczytuje metryki, więc szersze uprawnienia byłyby czystym ryzykiem bez żadnej korzyści.

Procedura ma dwa warianty w zależności od architektury instancji — klasyczny (baza niekontenerowa) i kontenerowy (CDB z użytkownikiem wspólnym `C##`). Rozróżnienie CDB/PDB opisuje strona Architektura Oracle: instancja, baza, SGA, PGA, procesy tła.

Przebieg — wariant klasyczny (baza niekontenerowa)

1. Utworzenie konta

```
CREATE USER zabbix IDENTIFIED BY "HasloZabbixa";
```

```
User created.
```

2. Nadanie prawa logowania

```
GRANT CONNECT, CREATE SESSION TO zabbix;
```

```
Grant succeeded.
```

3. Nadanie roli odczytu słownika danych

```
GRANT SELECT_CATALOG_ROLE TO zabbix;
```

> **Pojęcie: SELECT_CATALOG_ROLE.** Rola wbudowana dająca odczyt do praktycznie całego słownika danych (`DBA_*`) jedną komendą, zamiast nadawania `SELECT` na dziesiątki widoków z osobna. Szablon Zabbixa odpytuje spory zestaw widoków `DBA_*`, więc ta rola jest tu

wygodniejsza niż wyliczanie ich pojedynczo.

```
Grant succeeded.
```

4. Nadanie odczytu do widoków dynamicznych wydajności

Każdy `GRANT` uruchamiamy osobno — to nie jest sekwencja, w której kolejność ma znaczenie, ale gdy któryś widok nie istnieje w danej wersji silnika (rzadkie, ale zdarza się przy starszych patchach), pojedynczy błąd nie przerywa reszty.

```
GRANT SELECT ON v_$database TO zabbix;
GRANT SELECT ON v_$instance TO zabbix;
GRANT SELECT ON v_$session TO zabbix;
GRANT SELECT ON v_$process TO zabbix;
GRANT SELECT ON v_$sysmetric TO zabbix;
GRANT SELECT ON v_$sgastat TO zabbix;
GRANT SELECT ON v_$pgastat TO zabbix;
GRANT SELECT ON v_$system_parameter TO zabbix;
GRANT SELECT ON v_$osstat TO zabbix;
GRANT SELECT ON v_$log TO zabbix;
GRANT SELECT ON v_$datafile TO zabbix;
GRANT SELECT ON v_$archive_dest TO zabbix;
GRANT SELECT ON v_$recovery_file_dest TO zabbix;
GRANT SELECT ON v_$restore_point TO zabbix;
GRANT SELECT ON v_$active_session_history TO zabbix;
GRANT SELECT ON v_$asm_diskgroup TO zabbix;
GRANT SELECT ON sys.dba_data_files TO zabbix;
GRANT SELECT ON DBA_USERS TO zabbix;
GRANT SELECT ON DBA_TABLESPACES TO zabbix;
GRANT SELECT ON DBA_TABLESPACE_USAGE_METRICS TO zabbix;
```

```
Grant succeeded.
Grant succeeded.
...
Grant succeeded.
```

Przebieg — wariant kontenerowy (CDB, konto C##ZABBIX)

W architekturze CDB konto wspólne musi mieć nazwę z prefiksem `C##` — wymóg silnika opisany na stronie Architektura Oracle: instancja, baza, SGA, PGA, procesy tła. Zamiast widoków `DBA_*`, ograniczonych do bieżącego kontenera, nadajemy uprawnienia do ich odpowiedników `CDB_*`, widocznych ze wszystkich PDB naraz.

1. Utworzenie konta wspólnego

```
CREATE USER c##zabbix IDENTIFIED BY "HasloZabbixa" CONTAINER=ALL;
```

```
User created.
```

2. Nadanie prawa logowania

```
GRANT CONNECT, CREATE SESSION TO c##zabbix;
```

```
Grant succeeded.
```

3. Nadanie roli odczytu słownika i jej ograniczenie do bieżącego kontenera

```
GRANT SELECT_CATALOG_ROLE TO c##zabbix;
```

```
Grant succeeded.
```

```
ALTER USER c##zabbix SET CONTAINER_DATA=ALL CONTAINER=CURRENT;
```

`CONTAINER_DATA=ALL` pozwala kontu wspólnemu widzieć dane wszystkich kontenerów w widokach `CDB_*`, zamiast tylko tego, w którym akurat jest zalogowane.

```
User altered.
```

4. Nadanie odczytu do widoków dynamicznych i CDB_*

```
GRANT SELECT ON v_$database TO c##zabbix;
GRANT SELECT ON v_$instance TO c##zabbix;
GRANT SELECT ON v_$session TO c##zabbix;
GRANT SELECT ON v_$process TO c##zabbix;
GRANT SELECT ON v_$parameter TO c##zabbix;
GRANT SELECT ON v_$system_parameter TO c##zabbix;
GRANT SELECT ON v_$sysmetric TO c##zabbix;
GRANT SELECT ON v_$sgastat TO c##zabbix;
GRANT SELECT ON v_$pgastat TO c##zabbix;
GRANT SELECT ON v_$osstat TO c##zabbix;
GRANT SELECT ON v_$log TO c##zabbix;
GRANT SELECT ON v_$datafile TO c##zabbix;
GRANT SELECT ON v_$archive_dest TO c##zabbix;
GRANT SELECT ON v_$recovery_file_dest TO c##zabbix;
GRANT SELECT ON v_$active_session_history TO c##zabbix;
GRANT SELECT ON v_$asm_diskgroup TO c##zabbix;
GRANT SELECT ON v_$asm_diskgroup_stat TO c##zabbix;
GRANT SELECT ON v_$sort_segment TO c##zabbix;
GRANT SELECT ON v_$pdbs TO c##zabbix;
GRANT SELECT ON DBA_USERS TO c##zabbix;
GRANT SELECT ON CDB_TABLESPACES TO c##zabbix;
GRANT SELECT ON CDB_DATA_FILES TO c##zabbix;
GRANT SELECT ON CDB_TEMP_FILES TO c##zabbix;
GRANT SELECT ON CDB_FREE_SPACE TO c##zabbix;
```

```
Grant succeeded.
Grant succeeded.
...
Grant succeeded.
```

5. Wycofanie SELECT_CATALOG_ROLE po ustabilizowaniu monitoringu (opcjonalnie)

W jednym z wdrożeń, po przejściu na jawnie wyliczone uprawnienia do widoków `CDB_*`, szeroka rola `SELECT_CATALOG_ROLE` z kroku 3 przestała być potrzebna i została wycofana — zasada najmniejszych uprawnień z czasem, nie tylko na starcie:

```
REVOKE SELECT_CATALOG_ROLE FROM c##zabbix;
```

```
Revoke succeeded.
```

Ten krok jest opcjonalny i wykonujemy go dopiero po potwierdzeniu, że szablon Zabbixa faktycznie działa na jawnie nadanych uprawnieniach z kroku 4 — nie przed weryfikacją.

Funkcja pomocnicza wymagana przez szablon Zabbixa: punkty przywracania w CDB (opcjonalnie)

Widok `V$RESTORE_POINT` w architekturze CDB nie jest bezpośrednio widoczny w formie, jakiej oczekuje oficjalny szablon Zabbixa dla Oracle — dlatego szablon (zobacz źródło w sekcji Źródła) definiuje własny typ obiektowy i funkcję potokową (pipelined), która odczytuje `V$RESTORE_POINT` i zwraca jej wiersze w ujednoliconej formie, niezależnie od kontenera.

```
CREATE OR REPLACE TYPE zbx_mon_restore_point_row AS OBJECT (
  scn                          NUMBER,
  database_incarnation#        NUMBER,
  guarantee_flashback_database VARCHAR2(3),
  storage_size                 NUMBER,
  time                         TIMESTAMP(9),
  restore_point_time           TIMESTAMP(9),
  preserved                    VARCHAR2(3),
  name                         VARCHAR2(128),
  pdb_restore_point            VARCHAR2(3),
  clean_pdb_restore_point      VARCHAR2(3),
  pdb_incarnation#             NUMBER,
  replicated                   VARCHAR2(3),
  con_id                       NUMBER
);
/
```

```
CREATE OR REPLACE TYPE zbx_mon_restore_point_tab IS TABLE OF zbx_mon_restore_point_row;
/

CREATE OR REPLACE FUNCTION zbx_mon_restore_point RETURN zbx_mon_restore_point_tab PIPELINED AS
BEGIN
  FOR i IN (SELECT * FROM v$restore_point) LOOP
    PIPE ROW (zbx_mon_restore_point_row(
      i.scn, i.database_incarnation#, i.guarantee_flashback_database, i.storage_size,
      i.time, i.restore_point_time, i.preserved, i.name, i.pdb_restore_point,
      i.clean_pdb_restore_point, i.pdb_incarnation#, i.replicated, i.con_id));
  END LOOP;
  RETURN;
END;
/
```

```
Type created.
Type created.
Function created.
```

Instalujemy tę funkcję tylko wtedy, gdy konkretna wersja szablonu Zabbixa jej wymaga — sprawdzamy to w dokumentacji szablonu, nie zakładamy z góry.

Weryfikacja

Sprawdzenie, że konto się loguje i widzi widoki

```
CONNECT zabbix/HasloZabbixa@NazwaInstancji
SELECT instance_name, status FROM v$instance;
```

Przykład (wariant klasyczny):

```
SQL> CONNECT zabbix/********@WMSPRD
Connected.
SQL> SELECT instance_name, status FROM v$instance;
```

```
INSTANCE_NAME    STATUS
---------------- ------------
WMSPRD           OPEN
```

Sprawdzenie kompletu nadanych uprawnień

```
SELECT privilege, table_name FROM dba_tab_privs WHERE grantee = 'ZABBIX' ORDER BY table_name;
```

Przykład:

```
PRIVILEGE  TABLE_NAME
---------- ------------------------------
SELECT     V_$ARCHIVE_DEST
SELECT     V_$DATABASE
SELECT     V_$INSTANCE
SELECT     V_$SESSION
...
```

Sprawdzenie, że konto nie ma nic ponad odczyt

```
SELECT privilege FROM dba_sys_privs WHERE grantee = 'ZABBIX';
```

Przykład:

```
PRIVILEGE
----------------------------------------
CREATE SESSION
```

Jedyne uprawnienie systemowe, jakie konto monitorujące powinno mieć bezpośrednio, to `CREATE SESSION` — wszystko inne dochodzi przez `SELECT_CATALOG_ROLE` albo jawne `GRANT SELECT` na widoki wymienione wyżej.

Wycofanie zmian

```
DROP USER zabbix CASCADE;
```

albo, w wariancie kontenerowym:

```
DROP USER c##zabbix CASCADE;
```

`CASCADE` usuwa razem z kontem wszystkie obiekty, jakie zdążyło ono utworzyć — dla konta czysto monitorującego, bez własnych tabel, nie zmienia to nic poza samym kontem, ale trzymamy się tej klauzuli jako standardu, żeby procedura była identyczna niezależnie od tego, czy ktoś w międzyczasie dołożył kontu cokolwiek własnego.

```
User dropped.
```

Gdy coś nie zadziała

Objaw	Przyczyna	Naprawa
`ORA-01031: insufficient privileges` przy GRANT na v_$...	wykonanie GRANT zalogowanym jako zwykły DBA bez SYSDBA, na wersji z zaostrzonymi uprawnieniami do widoków v$	logujemy się `sqlplus / as sysdba`
`ORA-00942: table or view does not exist` przy CREATE USER z prefiksem C##	próba utworzenia konta z prefiksem C## na bazie, która nie jest CDB	sprawdzamy `SELECT cdb FROM v$database;` przed wyborem wariantu
Szablon Zabbixa zgłasza brak dostępu do konkretnej metryki mimo wykonanych GRANT-ów	widok dodany w nowszej wersji szablonu, którego nie ma na liście w kroku 4	sprawdzamy README aktualnej wersji szablonu i donadajemy brakujący `GRANT SELECT`
`ORA-01920: user name 'C##ZABBIX' conflicts with another user or role name`	konto już istnieje z poprzedniej próby instalacji	`DROP USER c##zabbix CASCADE;` i powtórzenie procedury od początku

Powiązane: Użytkownicy, role i uprawnienia · Architektura Oracle: instancja, baza, SGA, PGA, procesy tła · Co monitorujemy w bazie danych

Źródła: notatki własne, README szablonu Zabbix Oracle ODBC (git.zabbix.com, gałąź release/7.0)

Revision #42
Created 2026-09-24 23:34:22 UTC by Tomasz Konieczny
Updated 2026-10-26 14:08:53 UTC by Tomasz Konieczny