Architektura i zarządzanie instancją

Procesy, SID, sapcontrol i hdbsql.

- SAP HANA: cała baza w pamięci RAM
- Architektura SAP HANA: instancja, numer, nazwa SID, procesy
- sapcontrol i zarządzanie instancją
- hdbsql: klient wiersza poleceń

SAP HANA: cała baza w pamięci RAM

Poziom: podstawy **Dotyczy: SAP HANA 2.0** **Czas: ~10 min**

Ta strona otwiera książkę „SAP HANA": skąd wziął się pomysł trzymania całej bazy w pamięci RAM, jak to wygląda w jednym obrazku i dlaczego w Zielonej stoi na razie tylko pilotaż, a nie produkcja.

Skróty użyte na tej stronie (8)

Skrót	Co znaczy
ERP	Enterprise Resource Planning: system planowania zasobów przedsiębiorstwa, łączący księgowość, magazyn i inne procesy firmy w jednej aplikacji
HANA	SAP HANA: baza danych w pamięci firmy SAP, używana do dużych systemów księgowych i magazynowych
HPI	Hasso Plattner Institute: instytut badawczy w Poczdamie założony przez współzałożyciela SAP, w którym powstały pierwsze prototypy bazy in-memory pod SAP HANA
IT	Information Technology: dział albo praca związana z komputerami i infrastrukturą firmy
RAM	Random Access Memory: pamięć operacyjna
SAP	Systeme, Anwendungen und Produkte in der Datenverarbeitung: niemiecki producent oprogramowania klasy ERP, właściciel silnika HANA i przejętego wcześniej Sybase ASE
SID	w Oracle: System Identifier, nazwa instancji; w Windows: Security Identifier, numer konta
SQL	Structured Query Language: język zapytań do relacyjnych baz danych

Skąd się wzięło

> **Pojęcie: baza in-memory.** Silnik, w którym dane robocze trzymane są cały czas w pamięci RAM, a nie tylko wczytywane z dysku na żądanie, tak jak w klasycznym silniku dyskowym.

Pomysł, żeby całą bazę danych dużej firmy trzymać w pamięci, nie tylko jej najczęściej używany fragment, wyszedł z badań Hassona Plattnera, współzałożyciela SAP, prowadzonych razem z zespołem Hasso Plattner Institute w Poczdamie na przełomie lat dwutysięcznych. Plattner

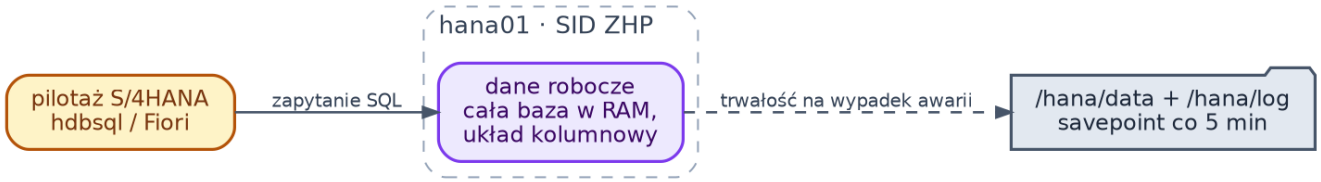
zauważył, że pamięć RAM robi się na tyle tania i pojemna, że przestaje być ograniczeniem nawet dla baz wielkości tych, na których stoi księgowość korporacji. Zespół HPI budował na tej podstawie prototypy łączące przetwarzanie transakcyjne z analitycznym na tych samych danych, bez osobnego przenoszenia ich do hurtowni raportowej co noc.

Pierwszy publiczny pokaz tych prac odbył się na konferencji SAPPHIRE w 2010 roku, a SAP wprowadził wtedy HANA jako produkt komercyjny. Nazwa rozwijana jest jako High Performance Analytic Appliance, choć dziś SAP używa jej częściej jako samodzielnej marki niż jako skrótu do rozwijania. Część branży przyjęła zapowiedź bazy trzymanej w całości w pamięci sceptycznie, jako pomysł trudny do utrzymania na produkcji dużej firmy.

Kolejnym krokiem było przeniesienie na HANA głównego systemu ERP samego SAP. W 2015 roku firma wypuściła S/4HANA, następcę starszego SAP ERP, zbudowanego od podstaw pod silnik in-memory zamiast pod klasyczny silnik dyskowy. Migracja z poprzedniej generacji na S/4HANA stała się dla wielu firm korzystających z SAP jednym z większych projektów IT dekady, bo oznaczała nie tylko zmianę silnika bazy, ale i przebudowę części procesów księgowych pod nowy model danych.

Jak to działa w jednym obrazku

Zapytanie od aplikacji trafia bezpośrednio do danych trzymanych w pamięci RAM, ułożonych kolumnowo zamiast wierszowo, więc odpowiedź wraca bez czekania na dysk. Dysk wchodzi do gry dopiero w tle: co kilka minut HANA zapisuje na niego spójny obraz zmian, żeby przetrwać restart albo awarię zasilania.



Zapytanie SQL trafia do danych w pamięci RAM na hana01, a dysk dostaje tylko okresowy zapis kontrolny (kliknięcie otwiera pełny rozmiar)

Dlaczego Zielona

Z ŻYCIA WZIĘTE

> Pan Zbigniew wrócił z konferencji branżowej przekonany, że S/4HANA to przyszłość księgowości w Zielonej, i kazał postawić pilotaż, zanim podejmie decyzję o migracji z Optimy. Na `hana01` stanęła instalacja z SID `ZHP`, z próbką danych sprzedażowych z ostatnich dwóch lat. Pół roku później pilotaż wciąż działa, a decyzja o dalszym kroku wciąż czeka na budżet.

Zielona nie ma dziś systemu SAP na produkcji, ale ma dokładnie taki pilotaż, jaki zwykle poprzedza decyzję o wejściu w ten świat: mały serwer, jedna instancja, dane testowe i pytanie „czy to się firmie opłaci" zawieszone gdzieś między działem IT a zarządem. Instalacja na `hana01` pozwala zespołowi nauczyć się podstaw obsługi HANA, zanim ktokolwiek podejmie decyzję wartą wieloletniego kontraktu licencyjnego. To samo w sobie tłumaczy, po co ta książka istnieje: administrator, który nigdy wcześniej nie widział HANA, może w każdej chwili dostać telefon, że pilotaż trzeba obsłużyć.

Mapa tej książki

Książka „SAP HANA" składa się z sześciu rozdziałów i zakłada znajomość słownika z książki „Bazy danych: fundamenty".

Rozdział	Na jakie pytanie odpowiada
SAP HANA: cała baza w pamięci RAM	Skąd wzięła się HANA i czym różni się od silnika dyskowego
Architektura SAP HANA	Co to jest instancja, SID, numer instancji i z jakich procesów składa się HANA
sapcontrol i zarządzanie instancją	Jak wystartować, zatrzymać i sprawdzić stan instancji
hdbsql	Jak połączyć się z bazą i wykonać zapytanie z linii poleceń
Użytkownicy i hasła	Jak założyć konto i nadać uprawnienia
Replikacja systemowa (HSR)	Jak HANA chroni się przed awarią całego serwera
Diagnostyka i miejsce na dysku	Co robimy, gdy brakuje pamięci albo miejsca na dysku

Powiązane: Architektura SAP HANA · Baza danych, silnik, instancja, schemat

Źródła: SAP News Center, „SAP HANA Turns 10" (news.sap.com, marzec 2020) · SAP, „What is SAP HANA?" (sap.com) · Wikipedia, „SAP HANA" i „SAP S/4HANA" · SAP Community, „What did the acronym HANA originally stand for?"

Architektura SAP HANA: instancja, numer, nazwa SID, procesy

Poziom: podstawy **Dotyczy: SAP HANA 2.0 SPS ≥ 04** **Czas: ~20 min**

HANA nie jest jednym procesem jak `mysqld` czy `postgres` — to zestaw kilkunastu współpracujących procesów, zidentyfikowanych parą SID plus numer instancji, o której administrator SAP myśli jako o jednej całości.

Skróty użyte na tej stronie (7)

Skrót	Co znaczy
HANA	SAP HANA: baza danych w pamięci firmy SAP, używana do dużych systemów księgowych i magazynowych
HTTP	Hypertext Transfer Protocol: protokół stron WWW
MDC	Multitenant Database Container: tryb SAP HANA, w którym jedna instalacja obsługuje bazę systemową i wiele baz tenant naraz
RAM	Random Access Memory: pamięć operacyjna
SAP	Systeme, Anwendungen und Produkte in der Datenverarbeitung: niemiecki producent oprogramowania klasy ERP, właściciel silnika HANA i przejętego wcześniej Sybase ASE
SID	w Oracle: System Identifier, nazwa instancji; w Windows: Security Identifier, numer konta
SQL	Structured Query Language: język zapytań do relacyjnych baz danych

Z życia wzięte

> **Z ŻYCIA WZIĘTE**
>
> Tomek dostaje zadanie sprawdzenia, czy pilotaż SAP na hana01 w ogóle żyje, ale polecenie hdbsql z numerem instancji 00 nie chce pokazać żadnych danych aplikacji. Po chwili okazuje

się, że łączył się z System DB zamiast z tenantem ZHP, więc SHOW TABLES pokazywało pustkę zamiast danych pilotażu. Dopiero dodanie flagi -d ZHP pokazuje właściwe tabele i Tomek rozumie różnicę między jedną instalacją a wieloma bazami w środku.

Baza in-memory i kolumnowa

Pojęcie: baza in-memory. Silnik, w którym dane robocze trzymane są cały czas w pamięci RAM, a nie tylko buforowane z dysku na żądanie jak w klasycznym silniku dyskowym (InnoDB, silnik PostgreSQL). Dysk w HANA nadal istnieje i jest niezbędny — trzyma trwałe kopie danych (`data volumes`) i log transakcyjny (`log volumes`) na wypadek awarii — ale ścieżka odczytu i zapisu w normalnej pracy nie czeka na dysk tak, jak w silnikach klasycznych. Konsekwencja praktyczna: rozmiar pamięci RAM serwera HANA musi pomieścić realnie całą aktywną bazę danych, nie tylko jej gorący podzbiór, inaczej system w ogóle nie wystartuje albo zacznie odrzucać zapytania z brakiem pamięci, co rozwija strona Diagnostyka i miejsce na dysku.

Pojęcie: baza kolumnowa. W przeciwieństwie do klasycznego układu wierszowego, gdzie wszystkie kolumny jednego wiersza leżą obok siebie na dysku, układ kolumnowy trzyma osobno całą kolumnę dla wszystkich wierszy tabeli. HANA jest silnikiem kolumnowym niemal wyłącznie — układ ten sprzyja zapytaniom analitycznym czytającym niewiele kolumn z bardzo wielu wierszy (typowe dla raportowania SAP) i pozwala na dużo skuteczniejszą kompresję niż układ wierszowy, bo wartości w jednej kolumnie są zwykle do siebie podobne. Kosztem jest gorsza wydajność pojedynczych zapisów punktowych pojedynczego wiersza w porównaniu z silnikiem wierszowym — kompromis akceptowalny, bo HANA projektowana jest pod obciążenia SAP łączące transakcje z analityką w czasie rzeczywistym (SAP nazywa to podejściem OLTP i OLAP na tych samych danych).

SID i numer instancji

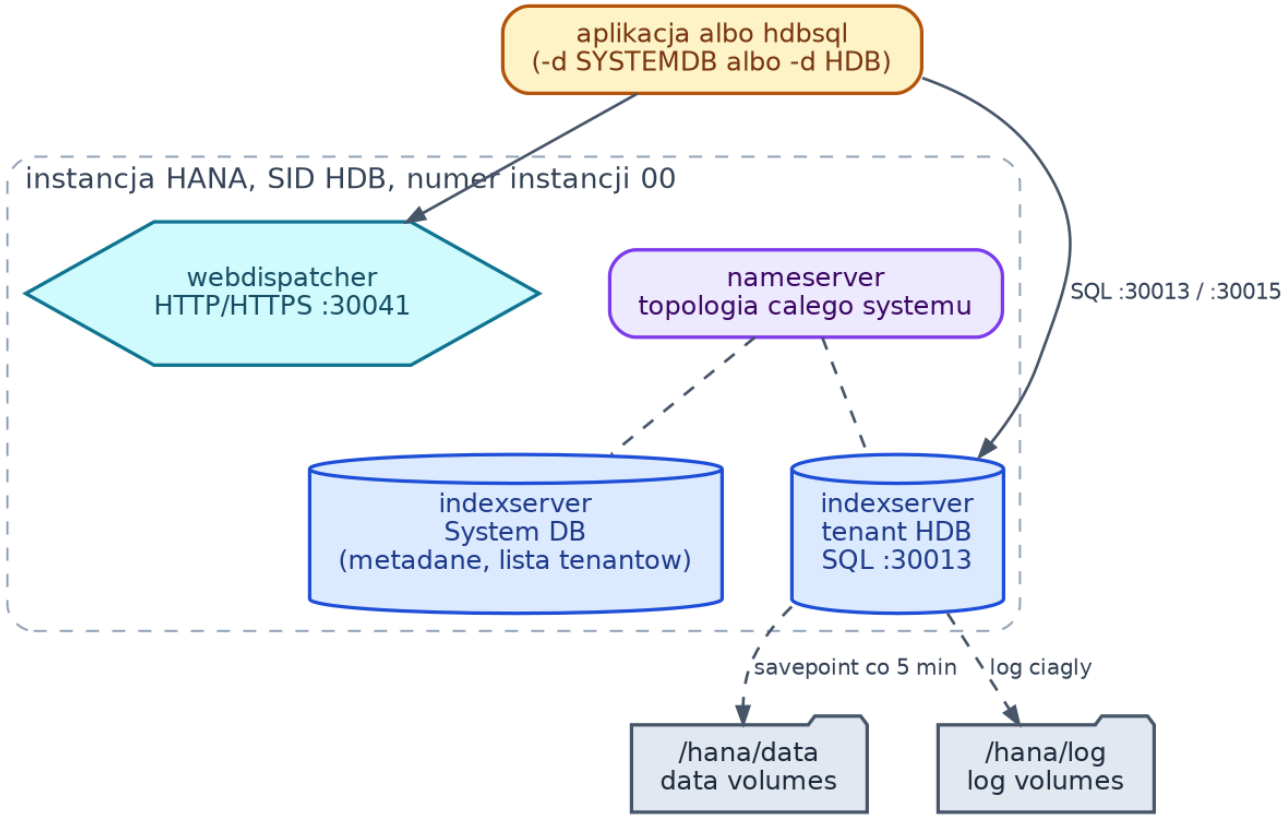
Pojęcie: SID (System ID). Trzyznakowy identyfikator instalacji SAP, pisany wielkimi literami (na przykład `HDB` albo `PRD`), jednoznacznie odróżniający jedną instalację od drugiej na tym samym serwerze albo w tej samej sieci. SID przewija się przez ścieżki katalogów, nazwy usług systemowych i nazwę użytkownika administracyjnego.

> **Pojęcie: numer instancji.** Dwucyfrowa liczba (`00`–`97`) odróżniająca poszczególne instancje SAP działające na tym samym SID-zie i tym samym hoście — większość instalacji ma jedną instancję bazodanową o jednym numerze, ale środowiska z wieloma komponentami (baza plus serwer aplikacyjny) mają po jednym numerze na każdy z nich. Numer instancji wchodzi wprost do numeracji portów, opisanej niżej.

Pełna identyfikacja jednej instancji HANA to zawsze para: `NazwaSID` plus `NumerInstancji`, na przykład `HDB` `00`.

Procesy

> **Pojęcie: instancja HANA.** Zestaw współpracujących procesów systemowych, uruchamianych i zatrzymywanych razem jako jedna całość, mimo że w `ps aux` widać ich kilkanaście osobno. Najważniejsze z nich:



Instancja HANA jako zestaw procesow, System DB i tenant DB, oraz zapis na data i log volumes (kliknięcie otwiera pełny rozmiar)

nameserver (`hdbnameserver`) — proces centralny, trzyma topologię całego systemu: które usługi działają, na którym hoście, w jakiej roli. Pierwszy proces startujący i ostatni zatrzymywany. To ten proces obsługuje polecenia `hdbnsutil` opisane na stronach Użytkownicy i hasła i Replikacja systemowa (HSR).

indexserver (`hdbindexserver`) — właściwy silnik bazy danych, przechowujący i przetwarzający dane tabel. W trybie MDC (opisanym niżej) każda baza tenant ma własny, osobny proces `indexserver`.

preprocessor (`hdbpreprocessor`) — obsługuje wyszukiwanie pełnotekstowe i analizę tekstu, wykorzystywane przez funkcje text mining i fuzzy search.

xsengine (`hdbxsengine`) — silnik aplikacyjny XS Classic, wykonujący logikę biznesową bezpośrednio wewnątrz bazy (model SAP HANA XS). W nowszych wdrożeniach jego rolę częściowo przejmuje XSA (XS Advanced), działający jako osobna warstwa poza rdzeniem bazy.

compileserver (`hdbcompileserver`) — kompiluje modele kalkulacji (calculation views) i procedury do postaci wykonywalnej przez `indexserver`.

webdispatcher (`hdbwebdispatcher`) — punkt wejścia HTTP/HTTPS dla ruchu do aplikacji HANA XS i do niektórych narzędzi administracyjnych.

Status wszystkich procesów naraz sprawdzamy poleceniem `sapcontrol -function GetProcessList`, opisanym na stronie sapcontrol i zarządzanie instancją.

Porty 3NN13 i 3NN15

> **Pojęcie: numeracja portów HANA.** Porty usług HANA nie są ustalone na sztywno, tylko wyliczane ze wzoru zawierającego numer instancji (`NN`) — dzięki temu wiele instancji na tym samym hoście nie koliduje portami. Dwa porty pojawiają się najczęściej w dokumentacji i w konfiguracji klienta:

Wzór portu	Przykład dla instancji 00	Usługa
3`NN`13	30013	SQL (indexserver, baza systemowa albo pojedynczy tenant w trybie bez wielu baz)
3`NN`15	30015	SQL (indexserver, dodatkowy port w konfiguracjach MDC)
3`NN`07	30007	nameserver, port wewnętrzny
3`NN`41	30041	HTTP (XS Classic)

Klient `hdbsql`, opisany na stronie hdbsql: klient wiersza poleceń, łączy się właśnie na jeden z portów SQL, wskazany numerem instancji, nie numerem portu wprost.

Katalogi /hana/shared, /usr/sap/SID i użytkownik sidadm

> **Pojęcie: /hana/shared.** Katalog (zwykle osobny wolumen albo punkt montowania) trzymający binaria wykonywalne HANA, wspólne dla wszystkich hostów instalacji wieloserwerowej (scale-out) — jeden zestaw plików wykonywalnych montowany przez wszystkie węzły klastra.

> **Pojęcie: /usr/sap/SID.** Katalog specyficzny dla danej instalacji o danym SID-zie, trzymający konfigurację, profile startowe i lokalne dane instancji (`/usr/sap/NazwaSID/HDBNumerInstancji/`), niezależnie na każdym hoście.

> **Pojęcie: sidadm.** Konto systemowe administrujące daną instancją, którego nazwa powstaje z SID-u pisanego małymi literami plus dopisek `adm` — dla SID `HDB` będzie to `hdbadm`, dla `PRD` będzie to `prdadm`. Wszystkie polecenia administracyjne HANA — `sapcontrol` , `HDB` , `hdbnsutil` , `hdbsql` z uprawnieniami lokalnymi — wykonujemy zalogowani na to konto, nigdy na `root` , poza wyjątkowymi operacjami wymagającymi uprawnień systemowych.

System DB i tenant DB w trybie MDC

> **Pojęcie: MDC (Multitenant Database Containers).** Tryb pracy HANA, domyślny od HANA 2.0, w którym jedna instalacja (jeden zestaw procesów, jeden SID, jeden numer instancji) obsługuje wewnętrznie wiele logicznie odseparowanych baz danych — tak jak jeden klaster PostgreSQL potrafi trzymać wiele baz. Przed MDC (HANA 1.0 i wczesne HANA 2.0 w trybie single-container) jedna instalacja odpowiadała zawsze dokładnie jednej bazie danych.

> **Pojęcie: System DB.** Specjalna baza administracyjna istniejąca zawsze w trybie MDC, niezależnie od liczby baz tenant — nie trzyma danych aplikacji, tylko metadane całej instalacji: listę tenantów, ich stan, globalną konfigurację. Łączymy się z nią do zadań administracyjnych obejmujących całą instalację, na przykład tworzenie nowego tenanta.

> **Pojęcie: tenant DB.** Pojedyncza baza danych wewnątrz instalacji MDC, izolowana od pozostałych tenantów własnym zestawem użytkowników, uprawnień i (w typowej konfiguracji) własnym procesem `indexserver`. Aplikacja SAP łączy się zawsze z konkretnym tenantem, nigdy z System DB.

Rozróżnienie, do której bazy się łączymy, ma znaczenie praktyczne przy każdym poleceniu `hdbsql` — flaga `-d` wskazuje nazwę bazy docelowej (`SYSTEMDB` albo nazwę tenanta), co rozwija strona hdbsql: klient wiersza poleceń.

Persystencja — savepoint i log volumes

> **Pojęcie: savepoint.** Mimo że HANA pracuje głównie w pamięci, dane w RAM same w sobie nie przetrwałyby restartu ani awarii zasilania — co jakiś czas (domyślnie co 5 minut, parametr `savepoint_interval_s`) `indexserver` zapisuje spójny obraz wszystkich zmienionych od ostatniego savepointu stron danych z pamięci na dysk, do `data volumes`. Między kolejnymi savepointami trwałość zapewnia log transakcyjny zapisywany na bieżąco do `log volumes`, na tej samej zasadzie co redo log InnoDB opisany dla MySQL na stronie Log transakcyjny: WAL, redo, transaction log — po awarii HANA wczytuje ostatni savepoint z dysku, a potem odtwarza z logu wszystkie zmiany zapisane już po nim.

Ten podział tłumaczy, dlaczego `/hana/data` i `/hana/log` rozdziela się zwykle na osobne wolumeny — pierwszy przyjmuje duże, rzadkie zapisy sekwencyjne (savepoint), drugi ciągły strumień małych zapisów przy każdej transakcji, opisany szerzej na stronie Diagnostyka i miejsce na dysku.

Scale-up i scale-out — wspomniane

> **Pojęcie: scale-up.** Instalacja HANA działająca na jednym serwerze fizycznym (albo jednej maszynie wirtualnej) — cała pamięć i wszystkie procesy `indexserver` (dla System DB i każdego tenanta) współdzielą ten sam host. Typowa konfiguracja dla większości instalacji produkcyjnych SAP.

> **Pojęcie: scale-out.** Instalacja rozłożona na wiele hostów, z których każdy trzyma fragment danych (jedną partycję) i uruchamia własny `indexserver` — używana tam, gdzie rozmiar bazy przekracza pojemność pamięci pojedynczego dostępnego serwera. Katalog `/hana/shared` opisany wyżej jest tym, co spaja hosty scale-out w jedną logiczną instalację — wspólny zestaw binariów montowany przez wszystkie węzły, podczas gdy `/hana/data` i `/hana/log` pozostają lokalne dla każdego hosta.

Częste błędy

Mylenie numeru instancji z numerem portu. Numer instancji (`00`, `03`...) to dwucyfrowy identyfikator, z którego dopiero wylicza się faktyczny numer portu według wzoru `3NN13`/`3NN15` — podanie numeru instancji tam, gdzie narzędzie oczekuje numeru portu, kończy się błędem połączenia.

Próba połączenia z System DB tam, gdzie trzeba połączyć się z tenantem. Zapytanie wykonane na System DB nie widzi żadnych danych aplikacji — nowa osoba bywa zaskoczona pustym wynikiem `SHOW TABLES`, nie zdając sobie sprawy, że jest podłączona do niewłaściwej bazy w ramach tej samej instalacji.

Powiązane: sapcontrol i zarządzanie instancją · hdbsql: klient wiersza poleceń · Replikacja systemowa (HSR)

Źródła: dokumentacja SAP HANA, „SAP HANA Administration Guide", rozdział „SAP HANA Architecture Overview" i „Multitenant Database Containers"

sapcontrol i zarządzanie instancją

Poziom: podstawy **Dotyczy: SAP HANA 2.0 SPS ≥ 04** **Czas: ~20 min**

`sapcontrol` jest wspólnym narzędziem administracyjnym całego świata SAP, nie tylko HANA — te same funkcje działają identycznie na serwerze aplikacyjnym ABAP czy Javy, co upraszcza pracę administratora obsługującego wiele typów instalacji naraz.

Skróty użyte na tej stronie (4)

Skrót	Co znaczy
ACL	Access Control List: lista, kto ma jakie prawa do pliku lub zasobu
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
HANA	SAP HANA: baza danych w pamięci firmy SAP, używana do dużych systemów księgowych i magazynowych
SAP	Systeme, Anwendungen und Produkte in der Datenverarbeitung: niemiecki producent oprogramowania klasy ERP, właściciel silnika HANA i przejętego wcześniej Sybase ASE

Fiszka

TYLKO ODCZYT

Sprawdzenie stanu procesów instancji: sapcontrol -function GetProcessList

```
sapcontrol -nr NumerInstancji -function GetProcessList
```

Sprawdzamy stan wszystkich procesów wchodzących w skład instancji naraz, każdy ze statusem koloru w kolumnie `dispstatus`.

Przykład:

```
sapcontrol -nr 00 -function GetProcessList
```

Przykład:

```
14.03.2026 09:41:02
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
hdbdaemon, HDB Daemon, GREEN, Running, 2026 03 10 08:00:00, 4:12:34, 4021
hdbnameserver, HDB Nameserver, GREEN, Running, 2026 03 10 08:00:05, 4:12:29, 4102
hdbcompileserver, HDB Compileserver, GREEN, Running, 2026 03 10 08:00:12, 4:12:22, 4210
hdbpreprocessor, HDB Preprocessor, GREEN, Running, 2026 03 10 08:00:18, 4:12:16, 4289
hdbindexserver, HDB Indexserver-HDB, GREEN, Running, 2026 03 10 08:00:20, 4:12:14, 4318
hdbxsengine, HDB XSEngine-HDB, GREEN, Running, 2026 03 10 08:00:25, 4:12:09, 4402
hdbwebdispatcher, HDB Web Dispatcher, GREEN, Running, 2026 03 10 08:00:30, 4:12:04, 4501
```

□□ **sapcontrol** = sterowanie instancją SAP

□□ **TYLKO ODCZYT**

Sprawdzenie stanu wszystkich instancji instalacji: sapcontrol -function GetSystemInstanceList

```
sapcontrol -nr NumerInstancji -function GetSystemInstanceList
```

Sprawdzamy stan wszystkich instancji wchodzących w skład instalacji (przydatne przy topologii z wieloma numerami instancji na kilku hostach), nie tylko procesów jednej z nich.

Przykład:

```
14.03.2026 09:41:15
GetSystemInstanceList
OK
hostname, instanceNr, httpPort, httpsPort, startPriority, features, dispstatus
hana01, 00, 50013, 50014, 1, HDB|HDB_WORKER, GREEN
```

□□ **sapcontrol** = sterowanie instancją SAP

□□ **TYLKO ODCZYT**

Uruchomienie instancji: HDB start

```
HDB start
```

□□ TYLKO ODCZYT

Zatrzymanie instancji: HDB stop

```
HDB stop
```

Wykonujemy z poziomu CLI Bash/SH, zalogowani na odpowiedniego użytkownika (`sidadm`).

Przykład:

```
StartService
Impromptu CCC initialization by 'rscpCInit'.
See SAP note 1266393.
OK
StartWait
OK
```

□□ **CLI** = Command Line Interface: praca przez wpisywanie komend w terminalu

□□ TYLKO ODCZYT

Sprawdzenie stanu instancji bez zmian: HDB info

```
HDB info
```

Sprawdzenie stanu bez uruchamiania ani zatrzymywania czegokolwiek.

Przykład:

```
USER    PID   COMMAND
hdbadm 4021  hdbdaemon
hdbadm 4102  hdbnameserver
hdbadm 4318  hdbindexserver -port 30003
```

□□ TYLKO ODCZYT

Uruchomienie instancji przez sapcontrol: sapcontrol -function Start

```
sapcontrol -nr NumerInstancji -function Start
```

sapcontrol = sterowanie instancją SAP

TYLKO ODCZYT

Uruchomienie instancji z oczekiwaniem na potwierdzenie: sapcontrol -function StartWait

```
sapcontrol -nr NumerInstancji -function StartWait 600 2
```

sapcontrol = sterowanie instancją SAP

TYLKO ODCZYT

Zatrzymanie instancji przez sapcontrol: sapcontrol -function Stop

```
sapcontrol -nr NumerInstancji -function Stop
```

sapcontrol = sterowanie instancją SAP

TYLKO ODCZYT

Zatrzymanie instancji z oczekiwaniem na potwierdzenie: sapcontrol -function StopWait

```
sapcontrol -nr NumerInstancji -function StopWait 600 2
```

Te same operacje przez `sapcontrol` bezpośrednio, z rozróżnieniem wariantów synchronicznych (`Wait`) i asynchronicznych.

sapcontrol = sterowanie instancją SAP

TYLKO ODCZYT

Przejście do katalogu konfiguracyjnego SAP: cdcoc

```
cdcoc
```

Wykonujemy z poziomu CLI Bash/SH na odpowiednim userze (`sidadm`).

SAP = Systeme, Anwendungen und Produkte in der Datenverarbeitung: niemiecki producent oprogramowania klasy ERP, właściciel silnika HANA i przejętego wcześniej Sybase ASE ·**CLI** = Command Line Interface: praca przez wpisywanie komend w terminalu

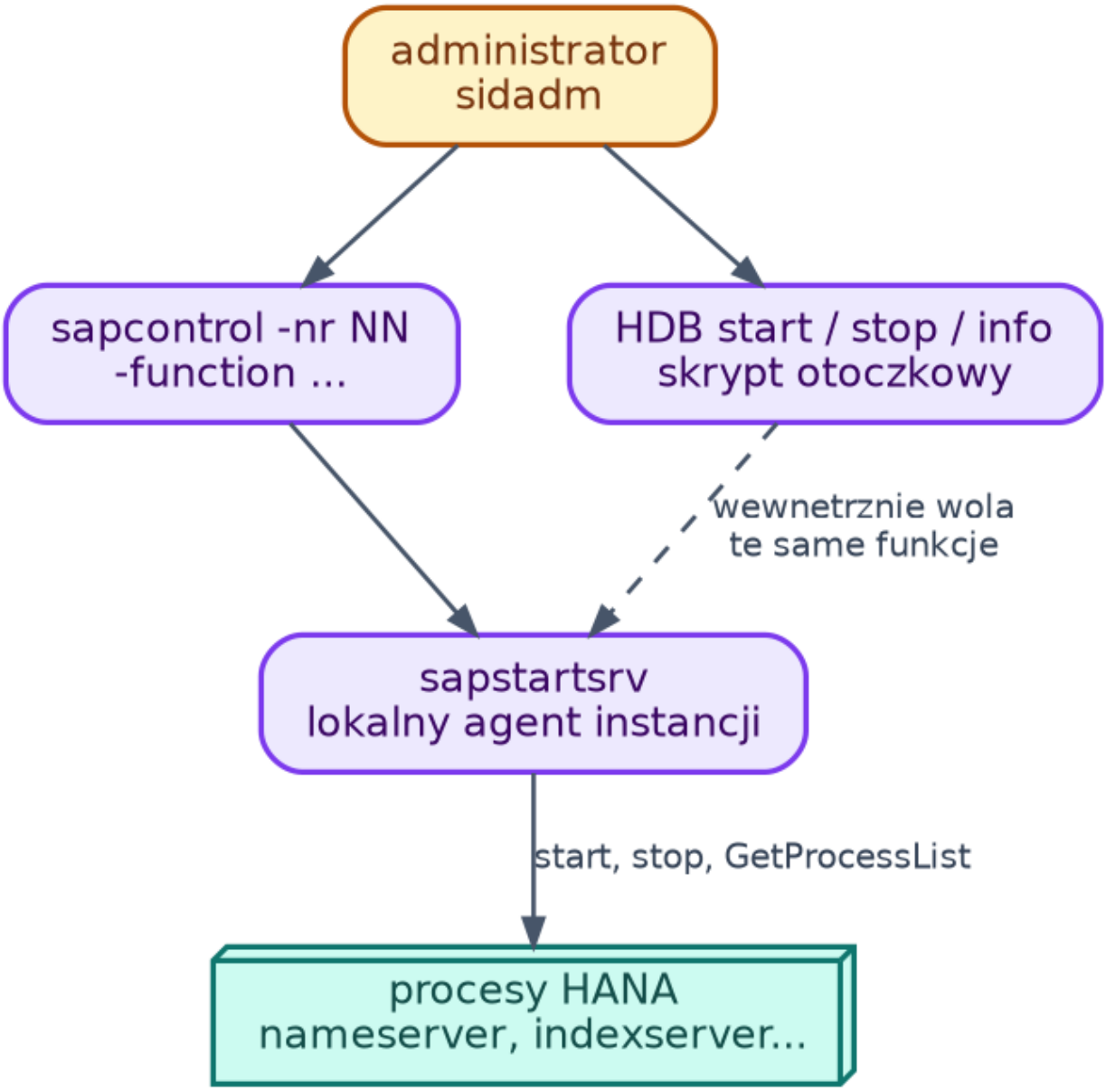
Z życia wzięte

Z ŻYCIA WZIĘTE

W czwartek rano hana01 nie odpowiada na zapytania z pilotażu ZHP, a HDB start kończy się pozornym sukcesem. Tomek sprawdza sapcontrol -function GetProcessList i widzi, że proces hdbindexserver ma status RED, mimo że reszta procesów świeci na zielono. Dopiero zajrzenie do logu tego procesu pokazuje brak miejsca na dysku danych, o czym sam HDB start w ogóle nie uprzedził. Po zwolnieniu miejsca i ponownym starcie wszystkie procesy wracają do statusu GREEN.

Czym jest sapcontrol

Pojęcie: sapcontrol. Narzędzie CLI komunikujące się z lokalnym agentem `sapstartsrv`, który z kolei zarządza rzeczywistymi procesami danej instancji SAP opisanymi na stronie Architektura SAP HANA: instancja, numer, nazwa SID, procesy. Każde wywołanie `sapcontrol` wymaga wskazania numeru instancji flagą `-nr` i nazwy funkcji do wykonania flagą `-function` — bez działającego procesu `sapstartsrv` dla danej instancji polecenie kończy się błędem połączenia, niezależnie od tego, czy same procesy bazy danych działają.



sapcontrol i HDB wolaja te same funkcje sapstartsrv, ktory zarzadza procesami HANA (kliknięcie otwiera pełny rozmiar)

Wykonujemy polecenia z poziomu CLI Bash/SH, zalogowani na odpowiedniego użytkownika administracyjnego danej instancji (`sidadm`, opisanego na poprzedniej stronie).

Weryfikacja statusu danej instancji

Pojęcie: status GREEN. Kolor w kolumnie `dispstatus` oznacza, że dany proces działa poprawnie. Pozostałe kolory: `YELLOW` — proces w trakcie startu albo zatrzymywania, `RED` — proces nie działa albo zgłasza błąd, `GRAY` — stan nieznany, zwykle tuż po starcie samego `sapstartsrv`. Instancję uznajemy za w pełni sprawną dopiero wtedy, gdy każdy proces z listy pokazuje `GREEN`.

START / STOP instancji

Pojęcie: HDB. Skrypt otoczkowy (wrapper) w katalogu domowym `sidadm`, upraszczający najczęstsze operacje na instancji bez podawania numeru instancji ani pełnej składni `sapcontrol` — wewnętrznie i tak woła te same funkcje `sapstartsrv`. `HDB start` czeka na pełne wystartowanie wszystkich procesów przed zwróceniem sterowania, `HDB stop` analogicznie czeka na pełne zatrzymanie.

Odpowiedniki przez sam sapcontrol

Pojęcie: warianty Wait. `Start` i `Stop` zwracają sterowanie natychmiast, nie czekając na faktyczne osiągnięcie docelowego stanu — do skryptów automatyzujących restart bezpieczniejsze są warianty `StartWait` / `StopWait`, przyjmujące limit czasu w sekundach (pierwszy parametr) i interwał sprawdzania stanu (drugi parametr), zwracające sterowanie dopiero po faktycznym potwierdzeniu statusu wszystkich procesów.

Przejście do katalogu konfiguracyjnego SAP

> **Pojęcie: aliasy cd konta sidadm.** Profil startowy konta `sidadm` (`.sapenv.sh` / `.sapenv.csh`, ładowany automatycznie przy logowaniu) definiuje rodzinę krótkich aliasów `cd*`, przenoszących bezpośrednio do często odwiedzanych katalogów instalacji bez wpisywania pełnej ścieżki za każdym razem — `cdcoc` w tej instalacji przenosi do katalogu konfiguracji (`custom/config` wewnątrz struktury `/usr/sap/NazwaSID`). Dokładny zestaw i nazwy tych aliasów bywają dopisywane ręcznie przez administratora przy zakładaniu środowiska, więc nie są identyczne na każdej instalacji — sprawdzenie zawartości `.sapenv.sh` w katalogu domowym `sidadm` pokazuje pełną listę zdefiniowanych skrótów na danym serwerze.

Zmienne środowiskowe sidadm

Profil `sidadm` ustawia też zmienne środowiskowe, z których korzystają wszystkie narzędzia CLI HANA opisane na kolejnych stronach:

Zmienna	Zawartość
`SAPSYSTEMNAME`	SID instalacji
`DIR_INSTANCE`	pełna ścieżka do katalogu instancji, `/usr/sap/NazwaSID/HDBNumerInstancji`
`PATH`	rozszerzony o katalog `exe` instalacji, skąd biorą się binaria `hdbsql`, `hdbnsutil`

Częste błędy

Wykonywanie poleceń administracyjnych jako root zamiast sidadm. Procesy HANA działają z uprawnieniami konta `sidadm` — polecenie `sapcontrol` uruchomione jako `root` bez dodatkowej konfiguracji `sapcontrol`-owego mechanizmu ACL kończy się błędem braku uprawnień albo w ogóle nie znajduje działającego `sapstartsrv` danej instancji.

Uznanie instancji za sprawną po samym HDB start bez sprawdzenia GetProcessList. `HDB start` kończy się sukcesem, gdy `sapstartsrv` zdąży wystartować podstawowe procesy w oczekiwanym czasie — proces, który wystartował i natychmiast padł (typowo `hdbindexserver` przy braku miejsca na dysku, opisanym na stronie Diagnostyka i miejsce na dysku), bywa niewidoczny bez jawnego sprawdzenia statusu `GetProcessList` chwilę później.

Powiązane: Architektura SAP HANA: instancja, numer, nazwa SID, procesy · hdbsql: klient wiersza poleceń · Diagnostyka i miejsce na dysku

Źródła: dokumentacja SAP, „sapcontrol Web Service Interface" · SAP Note 1428653 „SAPCONTROL: Functions description" · procedura autora

hdbsql: klient wiersza poleceń

Poziom: podstawy **Dotyczy: SAP HANA 2.0 SPS ≥ 04** **Czas: ~20 min**

`hdbsql` odpowiada w świecie HANA temu, czym `psql` jest dla PostgreSQL albo `mysql` dla MySQL — z jedną istotną różnicą: hasła do niego w praktyce produkcyjnej w ogóle nie wpisujemy wprost.

Skróty użyte na tej stronie (5)

Skrót	Co znaczy
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
HANA	SAP HANA: baza danych w pamięci firmy SAP, używana do dużych systemów księgowych i magazynowych
MDC	Multitenant Database Container: tryb SAP HANA, w którym jedna instalacja obsługuje bazę systemową i wiele baz tenant naraz
SAP	Systeme, Anwendungen und Produkte in der Datenverarbeitung: niemiecki producent oprogramowania klasy ERP, właściciel silnika HANA i przejętego wcześniej Sybase ASE
SQL	Structured Query Language: język zapytań do relacyjnych baz danych

Fiszka

TYLKO ODCZYT

Sprawdzenie listy baz danych instalacji MDC: SYS.M_DATABASES

```
SELECT DATABASE_NAME, ACTIVE_STATUS, HOST FROM SYS.M_DATABASES;
```

Sprawdzamy, jakie bazy (System DB i tenanty) istnieją w tej instalacji i czy są aktywne — pierwsze pytanie po podłączeniu do nieznanej instalacji w trybie MDC opisanym na stronie Architektura SAP HANA: instancja, numer, nazwa SID, procesy.

Przykład:

```
DATABASE_NAME  ACTIVE_STATUS  HOST
SYSTEMDB       YES            hana01
HDB            YES            hana01
```

MDC = Multitenant Database Container: tryb SAP HANA, w którym jedna instalacja obsługuje bazę systemową i wiele baz tenant naraz · **SAP** = Systeme, Anwendungen und Produkte in der Datenverarbeitung: niemiecki producent oprogramowania klasy ERP, właściciel silnika HANA i przejętego wcześniej Sybase ASE · **HANA** = SAP HANA: baza danych w pamięci firmy SAP, używana do dużych systemów księgowych i magazynowych · **SID** = w Oracle: System Identifier, nazwa instancji; w Windows: Security Identifier, numer konta

TYLKO ODCZYT

Sprawdzenie stanu usług instancji: M_SERVICES

```
SELECT HOST, PORT, SERVICE_NAME, PROCESS_ID, ACTIVE_STATUS FROM M_SERVICES;
```

Sprawdzamy z poziomu SQL to samo, co `sapcontrol -function GetProcessList` pokazuje z poziomu systemowego — listę usług HANA wraz z portami i statusem, wygodne, gdy jesteśmy już podłączeni klientem SQL i nie chcemy przełączać się do powłoki systemowej.

Przykład:

```
HOST       PORT   SERVICE_NAME       PROCESS_ID  ACTIVE_STATUS
hana01   30001  nameserver         4102        YES
hana01   30003  indexserver        4318        YES
hana01   30007  xsengine           4402        YES
hana01   30010  compileserver      4210        YES
```

SQL = Structured Query Language: język zapytań do relacyjnych baz danych · **HANA** = SAP HANA: baza danych w pamięci firmy SAP, używana do dużych systemów księgowych i magazynowych

TYLKO ODCZYT

Sprawdzenie zajętości dysków: M_DISKS

```
SELECT PATH, USED_SIZE, TOTAL_SIZE, USAGE_TYPE FROM M_DISKS;
```

Sprawdzamy wykorzystanie wolumenów danych i logu w bajtach — pierwsze miejsce, gdzie szukamy przyczyny alertu o zapełniającym się `/hana/log`, opisanego szerzej na stronie

Diagnostyka i miejsce na dysku.

Przykład:

```
PATH             USED_SIZE      TOTAL_SIZE      USAGE_TYPE
/hana/data/HDB   214748364800   536870912000    DATA
/hana/log/HDB     42949672960    64424509440    LOG
```

TYLKO ODCZYT

Sprawdzenie aktywnych połączeń: M_CONNECTIONS

```
SELECT CONNECTION_ID, HOST, PORT, USER_NAME, CONNECTION_STATUS, CLIENT_HOST
FROM M_CONNECTIONS
WHERE CONNECTION_STATUS = 'RUNNING';
```

Sprawdzamy, kto jest aktualnie podłączony do bazy i które połączenie faktycznie wykonuje zapytanie w danej chwili, w odróżnieniu od połączeń bezczynnych.

Przykład:

```
CONNECTION_ID  HOST       PORT   USER_NAME     CONNECTION_STATUS  CLIENT_HOST
482            hana01   30003  NAZWA_USERA   RUNNING            app01
```

TYLKO ODCZYT

Wylistowanie tabel w bieżącej bazie: TABLES

```
SELECT SCHEMA_NAME, TABLE_NAME, RECORD_COUNT FROM M_TABLES WHERE SCHEMA_NAME =
'NazwaSchematu';
```

Sprawdzamy tabele danego schematu wraz z przybliżoną liczbą wierszy, bez sięgania po pełny `SELECT COUNT(*)` na każdej z osobna.

Przykład:

```
SCHEMA_NAME     TABLE_NAME    RECORD_COUNT
NAZWA_SCHEMATU  ZAMOWIENIA    48213
NAZWA_SCHEMATU  KLIENCI       9284
```

Z życia wzięte

Z ŻYCIA WZIĘTE

Ola chce podłączyć zewnętrzny skrypt monitorujący do hana01, ale nie chce trzymać hasła jawnym tekstem w cronie. Zapisuje dane logowania kluczem MONITORING w hdbuserstore i skrypt łączy się poleceniem hdbsql -U MONITORING bez podawania czegokolwiek więcej. Miesiąc później podczas przeglądu bezpieczeństwa pan Wojtek sprawdza listę procesów na serwerze i nie znajduje ani jednego hasła widocznego w ps aux, co akurat go zadowala.

Parametry połączenia

```
hdbsql -i NumerInstancji -n NazwaSerwera -u NazwaUsera -p HasloUzytkownika -d NazwaBazy
```

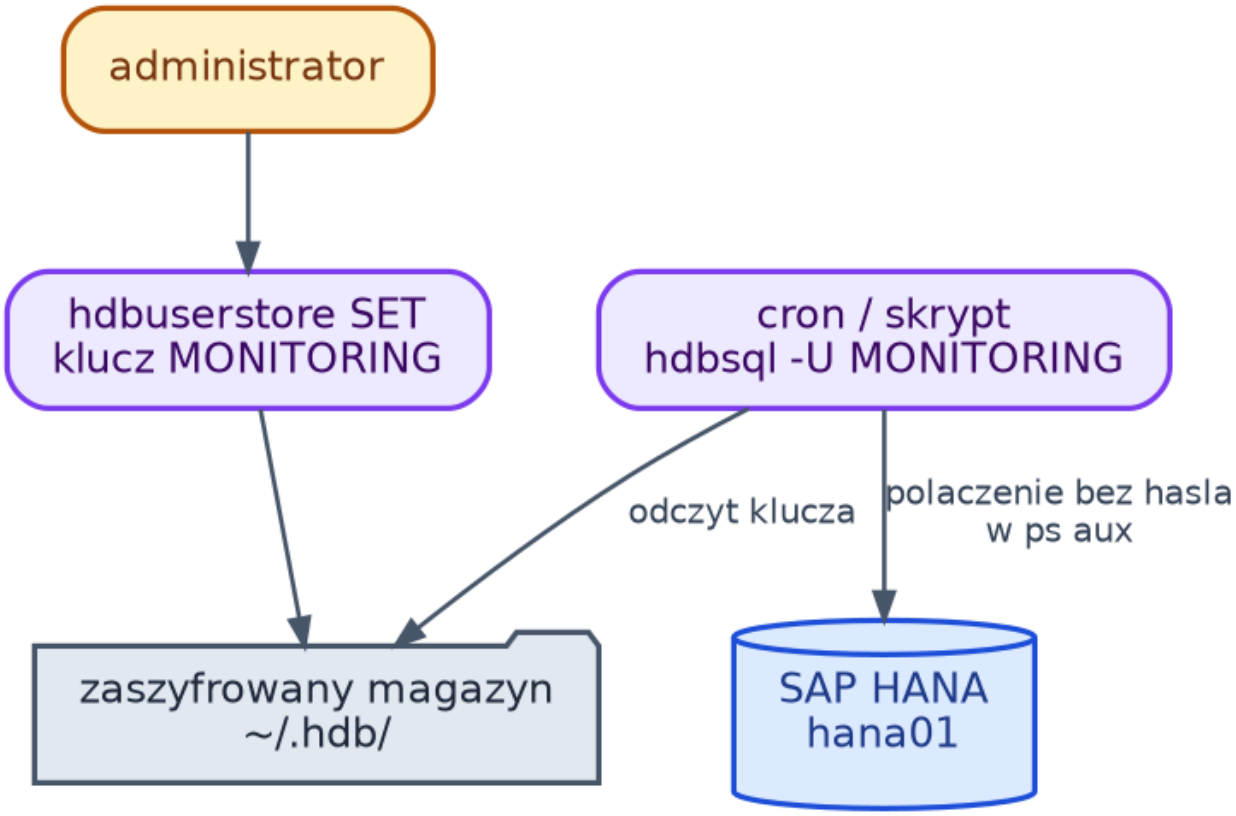
Flaga `-i` wskazuje numer instancji (nie numer portu — port HANA sam wylicza się z numeru instancji według wzoru opisanego na stronie Architektura SAP HANA: instancja, numer, nazwa SID, procesy), `-n` wskazuje host, `-u` i `-p` dane logowania, `-d` nazwę bazy docelowej — `SYSTEMDB` dla bazy systemowej albo nazwę konkretnego tenanta w instalacji MDC.

Przykład połączenia do tenanta na zdalnym hoście:

```
hdbsql -i 00 -n hana01 -u NazwaUsera -p HasloUzytkownika -d NazwaTenanta
```

hdbuserstore — dlaczego nie podajemy hasła w wierszu poleceń

Pojęcie: hdbuserstore. Narzędzie CLI zapisujące dane logowania (host, port, użytkownik, hasło) w zaszyfrowanym magazynie w katalogu domowym użytkownika systemowego, pod nazwaną kluczem referencją — odpowiednik `mysql_config_editor` z MySQL albo `.pgpass` z PostgreSQL, tylko bardziej rygorystycznie egzekwowany w środowiskach SAP, gdzie hasło w wierszu poleceń widoczne w `ps aux` jest traktowane jako naruszenie polityki bezpieczeństwa, nie tylko niedogodność.



Zapisanie klucza w hdbuserstore i połączenie skryptu z HANA bez hasła w wierszu poleceń (kliknięcie otwiera pełny rozmiar)

Zapisanie klucza:

```
hdbuserstore SET NazwaKlucza NazwaSerwera:NumerPortu@NazwaBazy NazwaUsera HasloUzytkownika
```

Przykład:

```
hdbuserstore SET MONITORING hana01:30013@SYSTEMDB nazwa_usera HasloUzytkownika
```

Wylistowanie zapisanych kluczy — bez ujawniania samych haseł:

```
hdbuserstore LIST
```

Przykład:

```
DATA FILE       : /home/nazwa_usera/.hdb/hana01/SSFS_HDB.DAT
KEY FILE        : /home/nazwa_usera/.hdb/hana01/SSFS_HDB.KEY

KEY MONITORING
```

```
  ENV : hana01:30013
  USER: NAZWA_USERA
  DATABASE: SYSTEMDB
```

Połączenie z użyciem klucza, bez podawania czegokolwiek poza samą nazwą klucza:

```
hdbsql -U NazwaKlucza
```

Ten sam mechanizm klucza wykorzystują zadania `cron` i skrypty backupu, o których mowa na stronie Diagnostyka i miejsce na dysku, żeby łączyć się z bazą bez interaktywnego pytania o hasło i bez trzymania go jawnym tekstem w treści skryptu.

Częste błędy

Podanie hasła flagą -p w skrypcie widocznym dla innych userów systemowych. Tak jak w każdym innym silniku, argument wywołania procesu jest widoczny w liście procesów systemu operacyjnego dla innych kont — `hdbuserstore` istnieje właśnie po to, żeby tego uniknąć w skryptach automatyzujących.

Połączenie z SYSTEMDB tam, gdzie potrzebny jest tenant. Zapytania do widoków `M_*` na System DB pokazują stan całej instalacji, nie danych konkretnego tenanta — próba odpytania tabeli aplikacji z poziomu System DB kończy się błędem, że taki obiekt nie istnieje, mimo że w tenantcie istnieje naprawdę.

Powiązane: Architektura SAP HANA: instancja, numer, nazwa SID, procesy · sapcontrol i zarządzanie instancją · Diagnostyka i miejsce na dysku

Źródła: dokumentacja SAP HANA, „SAP HANA SQL Command Network Protocol Reference" i „SAP HANA Administration Guide", rozdział „hdbuserstore"