Zabbix: SMART dysków na Windows przez agenta 2

Poziom: średnio **Dotyczy: Zabbix agent2 7.0 LTS, smartmontools 7.x, Windows Server** **Czas: ~30 min**

Dokładamy do agenta zainstalowanego według strony Agent, instalacja i konfiguracja monitoring zdrowia dysków SMART, ręcznie i tylko tam, gdzie ma to sens: dyski systemowe tak, dyski w puli Storage Spaces nie.

ZAŁOŻENIA SCENARIUSZA

przykład pokazujemy na hoście Hyper-V `hv01` (Windows Server 2022, węzeł klastra `hvclu01`), bo SMART ma sens tylko na serwerze fizycznym; `fs01` jest maszyną wirtualną na `hvclu01` i fizycznych dysków nie widzi. Zakładamy, że `hv01` ma dwa systemowe dyski NVMe podłączone wprost do płyty, a dyski puli `S2D-Pula` za kontrolerem w trybie HBA, jak na stronie RAID i kontrolery: sprzętowy, programowy i HBA dla S2D. Na `hv02` postępujemy tak samo.

Skróty użyte na tej stronie (9)

Skrót	Co znaczy
DOS	Disk Operating System: bardzo prosty, jednozadaniowy system, na którym stały pierwsze wersje Windows przed Windows NT
HBA	Host Bus Adapter: karta, która tylko wystawia dyski systemowi, bez składania ich w macierz; wymagana w trybie przezroczystym dla S2D
JSON	JavaScript Object Notation: tekstowy format danych z klamrami i cudzysłowami
JSONPath	Zapis ścieżki, która wskazuje jedną wartość w dokumencie JSON, np. $.value to pole value na samym wierzchu
LLD	Low-Level Discovery: automatyczne wykrywanie w Zabbixie np. dysków i interfejsów
LTS	Long Term Support: wersja z długim wsparciem poprawkami
NVMe	Non-Volatile Memory Express: protokół komunikacji z dyskami SSD podłączonymi przez złącze PCIe, znacznie szybszy niż starsze SATA i SAS, typowy dla szybkich dysków systemowych

Skrót	Co znaczy
RAID	Redundant Array of Independent Disks: sposób łączenia kilku fizycznych dysków w jeden, widziany przez system jako pojedynczy dysk, zwykle z dodatkową odpornością na awarię jednego z nich
SMART	Self-Monitoring, Analysis and Reporting Technology: wbudowany w dysk mechanizm zbierający liczniki o jego kondycji, na przykład temperaturę i liczbę realokowanych sektorów, odczytywany programami takimi jak smartctl

Fiszka

□□ TYLKO ODCZYT

Sprawdzenie wersji smartctl: smartctl --version

```
smartctl --version
```

Wtyczka SMART agenta2 potrzebuje smartmontools w wersji przynajmniej 7.1, starsze wersje nie odczytują poprawnie części atrybutów z nowszych dysków NVMe.

Przykład:

```
smartctl 7.4 2023-08-01 r5530 [x86_64-w64-mingw32-2022]
Copyright (C) 2002-23, Bruce Allen, Christian Franke, www.smartmontools.org
```

□□ **SMART** = Self-Monitoring, Analysis and Reporting Technology: wbudowany w dysk mechanizm zbierający liczniki o jego kondycji, na przykład temperaturę i liczbę realokowanych sektorów, odczytywany programami takimi jak smartctl · **NVMe** = Non-Volatile Memory Express: protokół komunikacji z dyskami SSD podłączonymi przez złącze PCIe, znacznie szybszy niż starsze SATA i SAS, typowy dla szybkich dysków systemowych

□□ TYLKO ODCZYT

Wylistowanie widocznych dysków: smartctl --scan-open

```
smartctl --scan-open
```

Wypisuje ścieżki dysków, które `smartctl` w ogóle potrafi odpytać z tego hosta. Te same ścieżki wpisujemy potem w kluczu ręcznej pozycji.

Przykład:

```
/dev/sda -d nvme # /dev/sda, NVMe device
/dev/sdb -d nvme # /dev/sdb, NVMe device
```

PLIK KONFIGURACYJNY

Wskazanie ścieżki do smartctl: Plugins.Smart.Path

```
Plugins.Smart.Path=C:\smartmontools\bin\smartctl.exe
```

> (gdzie `C:\smartmontools\bin\smartctl.exe` to pełna ścieżka do pliku `smartctl.exe`, u nas celowo w katalogu bez spacji w nazwie)

Pojęcie: zmienna PATH. Lista katalogów, w których system szuka programu, gdy wpisujemy samą jego nazwę bez pełnej ścieżki. Instalator smartmontools dopisuje swój katalog do PATH konta, którym instalowaliśmy program, ale usługa agenta na Windows działa na zupełnie innym koncie, Local System, z własną, osobną listą katalogów. Bez jawnego `Plugins.Smart.Path` agent nie znajdzie `smartctl.exe` wcale, mimo że ręczne uruchomienie tego samego polecenia w naszej własnej sesji działa bez problemu.

Jeśli smartmontools stoi już w domyślnym `C:\Program Files\smartmontools`, spacja w nazwie katalogu potrafi psuć parsowanie ścieżki w pliku konfiguracyjnym. Prościej zainstalować program w katalogu bez spacji, jak w przykładzie wyżej, niż walczyć z cudzysłowami albo skróconą nazwą DOS 8.3.

DOS = Disk Operating System: bardzo prosty, jednozadaniowy system, na którym stały pierwsze wersje Windows przed Windows NT

ZMIENIA SYSTEM

Restart usługi agenta po zmianie konfiguracji: Restart-Service

```
Restart-Service -Name "Zabbix Agent 2"
```

Agent, tak samo jak na Linuksie, nie przeładowuje konfiguracji sam z siebie po edycji pliku.

◫◫ KLIKAMY W INTERFEJSIE

Wyłączenie automatycznego wykrywania dysków SMART na hoście hv01

```
Data collection → Hosts → hv01 → Discovery rules → SMART disk discovery → Status: Disabled
```

Wyłączamy regułę na poziomie samego hosta `hv01`, nie całego szablonu, żeby inne hosty z prostszym magazynem korzystały z automatycznego wykrywania normalnie.

◫◫ **SMART** = Self-Monitoring, Analysis and Reporting Technology: wbudowany w dysk mechanizm zbierający liczniki o jego kondycji, na przykład temperaturę i liczbę realokowanych sektorów, odczytywany programami takimi jak smartctl

◫◫ ZMIENIA SYSTEM

Dodanie ręcznej pozycji dla dysku systemowego: smart.disk.get

```
Data collection → Hosts → hv01 → Items → Create item
Nazwa: SMART, dysk systemowy NVMe0
Typ: Zabbix agent
Klucz: smart.disk.get["/dev/sda", ""]
Typ informacji: Tekst
Interwał aktualizacji: 1h
```

> **Pojęcie: RAID (Redundant Array of Independent Disks).** Sposób łączenia kilku fizycznych dysków tak, żeby system widział je jako jeden, z nadmiarowością albo bez niej, obsługiwany zwykle przez osobny kontroler sprzętowy. Drugi parametr klucza to właśnie typ takiego kontrolera, przez jaki dysk jest widoczny. Pusty łańcuch `""` oznacza zwykły, pojedynczy dysk bez kontrolera RAID pomiędzy nim a systemem, dokładnie nasz przypadek dla dysku NVMe podłączonego wprost do płyty głównej.

Powtarzamy tę samą pozycję dla drugiego dysku systemowego, z kluczem `smart.disk.get["/dev/sdb", ""]`.

◫◫ **RAID** = Redundant Array of Independent Disks: sposób łączenia kilku fizycznych dysków w jeden, widziany przez system jako pojedynczy dysk, zwykle z dodatkową odpornością na awarię jednego z nich · **NVMe** = Non-Volatile Memory Express: protokół komunikacji z dyskami SSD podłączonymi przez złącze PCIe, znacznie szybszy niż starsze SATA i SAS, typowy dla szybkich dysków systemowych

TYLKO ODCZYT

Test z serwera: zabbix_get

```
zabbix_get -s AdresHosta -k 'smart.disk.get["/dev/sda", ""]'
```

Sprawdzamy z poziomu serwera albo proxy, że agent w ogóle potrafi zwrócić dane SMART z konkretnego dysku, zanim zaczniemy budować na tym elementy zależne i wyzwalacze.

Przykład:

```
{"model_name":"Dysk NVMe
systemowy","serial_number":"S3X1NX0K123456","smart_status":{"passed":true},"temperature"
:{"current":38},"power_on_time":{"hours":4102}}
```

SMART = Self-Monitoring, Analysis and Reporting Technology: wbudowany w dysk mechanizm zbierający liczniki o jego kondycji, na przykład temperaturę i liczbę realokowanych sektorów, odczytywany programami takimi jak smartctl · **zabbix_get** = ręczne odpytanie agenta Zabbixa

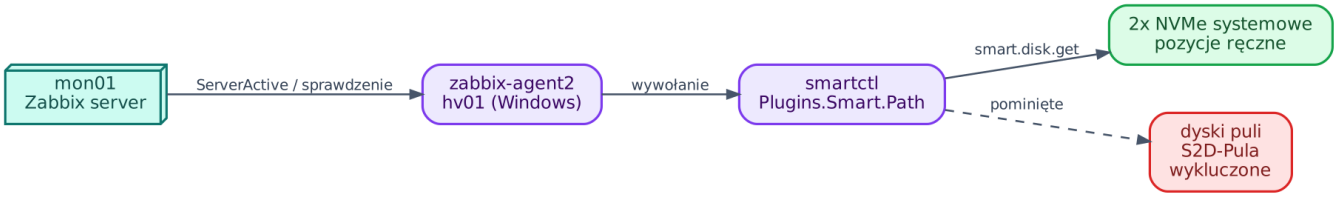
Z życia wzięte

Z ŻYCIA WZIĘTE

O 11:15 wykrywanie SMART na `hv01` zaczyna zgłaszać dziwne, niespójne odczyty dla dysków w puli `S2D-Pula`, a jeden z administratorów dyżurnych chce już budować na tym wyzwalacz. Tomek sprawdza `smartctl --scan-open` i przypomina zespołowi, że dyski za sterownikiem puli nigdy nie były wiarygodnym źródłem dla `smartctl`, bo kontroler nie oddaje ich jako pojedynczych, zwykłych nośników. Wyłącza regułę LLD tylko na `hv01` i zostawia zdrowie puli pod obserwacją `Get-PhysicalDisk`, tak jak ustalono przy pierwszym wdrożeniu tej strony.

Jak to działa

Pojęcie: wtyczka SMART agenta2 (Smart plugin). Rozszerzenie wbudowane w `zabbix-agent2`, które samo nie czyta danych z dysku, tylko za każdym razem uruchamia zewnętrzny program `smartctl` z pakietu smartmontools i parsuje jego wynik do formatu JSON. Bez zainstalowanego `smartctl` na hoście wtyczka nie ma czego wywołać, więc pierwszym krokiem zawsze jest sprawdzenie samego `smartctl`, nie konfiguracja Zabbixa.



Schemat drogi danych SMART: agent 2 wywołuje smartctl, który czyta dyski NVMe wprost, a dyski puli Storage Spaces pomija (kliknięcie otwiera pełny rozmiar)

> **Pojęcie: wykrywanie niskopoziomowe dysków (reguła smart.disk.discovery).** Reguła LLD szablonu „SMART by Zabbix agent 2", która sama odpytuje `smartctl --scan-open` i dla każdego znalezionego dysku tworzy komplet elementów danych (temperatura, liczba godzin pracy, poszczególne atrybuty SMART) razem z wyzwalaczami. Działa dobrze dla dysków podłączonych wprost, bo dla nich `smartctl` widzi jeden fizyczny nośnik i jeden spójny zestaw atrybutów.

Na `hv01` mamy dwa rodzaje dysków: dwa systemowe NVMe podłączone wprost do kontrolera i dyski w puli Storage Spaces Direct `S2D-Pula`, obsługiwane przez sterownik puli, nie wprost przez kontroler dyskowy widziany przez `smartctl`. Zdrowie dysków puli i tak śledzimy przez `Get-PhysicalDisk` z poziomu samego Storage Spaces Direct, łącznie ze stanami takimi jak `Predictive Failure`, opisanymi na stronie S2D: wymiana dysku bez przestoju, gdy pula jest prawie pełna. Zamiast naprawiać albo weryfikować wykrywanie SMART dla przypadku, który i tak monitorujemy innym, bardziej wiarygodnym mechanizmem, wyłączamy regułę LLD na tym hoście i dokładamy tylko dwie ręczne pozycje dla dysków systemowych.

Wyzwalacze

Szablon „SMART by Zabbix agent 2" ma gotowe wyzwalacze reagujące na bity w polu stanu zwracanym przez `smartctl`, między innymi „Some Attributes have been <= threshold", „Device open failed" i „DISK FAILING". Dla naszych dwóch ręcznych pozycji budujemy własne, prostsze wyzwalacze na dwóch danych, które realnie nas interesują.

Wyzwalacz na temperaturę, korzystający z tych samych makr co szablon:

```
avg(/hv01/smart.disk.get["/dev/sda", ""],5m) > {$SMART.TEMPERATURE.MAX.WARN}
```

Domyślne progi w szablonie to 50 stopni dla ostrzeżenia i 65 dla stanu krytycznego. Dopasowujemy je do konkretnego modelu dysku, jeśli producent zaleca inne wartości.

Wyzwalacz na stan zdrowia dysku, wyciągnięty z tego samego surowego JSON-a przez element zależny:

```
last(/hv01/smart.disk.status.nvme0) = 0
```

Element `smart.disk.status.nvme0` jest typu „Element zależny", z elementem nadrzędnym `smart.disk.get["/dev/sda", ""]` i przetwarzaniem wstępnym typu JSONPath, czyli ścieżką wskazującą jedno konkretne pole wewnątrz większego dokumentu JSON, tutaj `$.smart_status.passed`. Dzięki temu nie odpytujemy `smartctl` drugi raz tylko po to, żeby wyciągnąć jedno pole z danych, które i tak już pobraliśmy.

Częste błędy

Objaw	Przyczyna	Co robimy
Element `smart.disk.get` nieobsługiwany, błąd o nieznalezieniu programu	usługa agenta na koncie Local System nie widzi zmiennej PATH ustawionej przy instalacji smartmontools	ustawiamy `Plugins.Smart.Path` na pełną ścieżkę do `smartctl.exe` i restartujemy usługę agenta
`Plugins.Smart.Path` ustawiony, a mimo to błąd parsowania ścieżki	spacja w nazwie katalogu (`C:\Program Files\smartmontools`) źle się parsuje w pliku konfiguracyjnym	instalujemy smartmontools w katalogu bez spacji, na przykład `C:\smartmontools`, albo używamy skróconej nazwy DOS 8.3
Dyski puli Storage Spaces nie pojawiają się w wykrywaniu albo pojawiają się z niewiarygodnymi danymi	dyski za sterownikiem puli nie są dla `smartctl` zwykłym, pojedynczym nośnikiem	nie próbujemy ich monitorować przez SMART, zdrowie puli śledzimy przez `Get-PhysicalDisk` i mechanizmy Storage Spaces
`zabbix_get` z serwera zwraca błąd, mimo że `smartctl` uruchomiony ręcznie na hoście działa	ręczny test w sesji administratora ma inny PATH niż usługa agenta na koncie Local System	testujemy zawsze przez sam element Zabbixa (`zabbix_get` albo podgląd ostatnich danych), nie przez ręczne uruchomienie `smartctl` we własnej sesji

□□ Ćwiczenia

1. Host ma jeden dysk NVMe systemowy i trzy dyski w puli Storage Spaces. Ile ręcznych pozycji SMART dodajemy według zasady z tej strony?

Odpowiedź

Jedną, tylko dla dysku systemowego NVMe. Dyski puli Storage Spaces wykluczamy z monitoringu SMART niezależnie od tego, ile ich jest, bo ich zdrowie śledzimy przez Storage

Spaces, nie przez `smartctl`.

1. `smart.disk.get["/dev/sda", ""]` zwraca dane poprawnie po ręcznym uruchomieniu `smartctl` w sesji administratora, ale ten sam element w Zabbixie jest nieobsługiwany. Co sprawdzamy najpierw?

Odpowiedź

Czy `Plugins.Smart.Path` w konfiguracji agenta wskazuje pełną, poprawną ścieżkę do `smartctl.exe`. Usługa agenta działa na koncie Local System, które ma inny PATH niż nasza sesja administratora, więc samo działające ręczne uruchomienie niczego jeszcze nie gwarantuje.

Powiązane: Agent, instalacja i konfiguracja · Monitorowanie typowych usług · S2D: wymiana dysku bez przestoju, gdy pula jest prawie pełna

Źródła: S.M.A.R.T. monitoring and integration with Zabbix · szablon SMART by Zabbix agent 2, kod źródłowy · ZBX-23750: Plugins.Smart.Path wymagany na Windows · dyskusja na forum Zabbixa o SMART na Windows

---

Revision #27
Created 2026-09-28 06:15:05 UTC by Tomasz Konieczny
Updated 2026-10-26 14:09:52 UTC by Tomasz Konieczny