Host, element danych, wyzwalacz, akcja: jak to się składa

Poziom: podstawy **Dotyczy: Zabbix 6.0 LTS / 6.4** **Czas: ~25 min**

Bez tego rozdziału reszta konfiguracji Zabbixa jest zestawem magicznych zaklęć — tu układamy pojęcia, z których zbudowana jest każda strona monitoringu w tym systemie.

Skróty użyte na tej stronie (5)

Skrót	Co znaczy
LLD	Low-Level Discovery: automatyczne wykrywanie w Zabbixie np. dysków i interfejsów
LTS	Long Term Support: wersja z długim wsparciem poprawkami
SMS	Short Message Service: krótka wiadomość tekstowa na telefon, która dociera także bez internetu
SNMP	Simple Network Management Protocol: protokół odczytu stanu urządzeń sieciowych
WWW	World Wide Web: strony i aplikacje otwierane w przeglądarce po HTTP albo HTTPS

Z życia wzięte

Z ŻYCIA WZIĘTE

Nowy administrator w zespole pyta Tomka o 13:00, dlaczego usunięcie jednego elementu danych z hosta `app01` nagle zostawiło na pulpicie czerwony alarm o „elemencie nieobsługiwanym", chociaż nic już go nie zbiera. Tomek tłumaczy, że element i wyzwalacz to dwie osobne rzeczy, więc usunięcie elementu bez usunięcia powiązanego wyzwalacza zostawia ten wyzwalacz żywy, tylko bez danych do oceny. Razem usuwają oba, po czym alarm znika z listy problemów.

Host — punkt odniesienia dla wszystkiego innego

Pojęcie: host. Pojedynczy monitorowany obiekt — zwykle serwer albo urządzenie sieciowe, ale równie dobrze abstrakcyjna jednostka bez fizycznego adresu (np. „monitoring zewnętrznej strony WWW"). Host sam w sobie nie zbiera niczego — jest kontenerem, do którego przypisujemy elementy danych, wyzwalacze i szablony. Host należy do co najmniej jednej grupy hostów, a grupy służą wyłącznie do organizacji uprawnień i filtrowania widoków, nie mają wpływu na to, co i jak jest zbierane.

Element danych — pojedyncza metryka

Pojęcie: element danych (item). Definicja pojedynczej metryki zbieranej z hosta — jedna linia mówiąca „co zbieramy, jak często i skąd". Element sam w sobie tylko gromadzi wartości w historii, nie ocenia niczego i nie wysyła żadnych powiadomień — to zadanie wyzwalaczy, opisanych niżej.

Pojęcie: klucz elementu (item key). Identyfikator mówiący agentowi (albo innemu mechanizmowi zbierania), jaką konkretnie wartość zwrócić — np. `vfs.fs.size[/,pfree]` (procent wolnego miejsca w systemie plików zamontowanym w `/`) albo `system.cpu.load[percpu,avg1]` (obciążenie procesora, wartość znormalizowana per rdzeń, średnia z minuty). Parametry w nawiasach kwadratowych doprecyzowują, o którą konkretnie instancję metryki chodzi.

Pojęcie: interwał aktualizacji. Jak często element jest odpytywany — ustawiany osobno dla każdego elementu, typowo od kilkudziesięciu sekund (metryki krytyczne, np. dostępność usługi) do kilkunastu minut (metryki wolno zmieniające się, np. wersja zainstalowanego oprogramowania). Krótszy interwał oznacza dokładniejszą historię, ale też szybszy przyrost bazy danych — kompromis, który świadomie dobieramy per element, a nie ustawiamy jednakowo wszędzie.

Pojęcie: typ elementu. Skąd faktycznie pochodzi wartość — najczęstsze typy to Zabbix agent (pasywny, serwer pyta agenta), Zabbix agent (active) (aktywny, agent sam wysyła), SNMP agent, Zabbix trapper (wartość wysyłana z zewnątrz komendą `zabbix_sender`, bez cyklicznego odpytywania) oraz Zabbix internal (metryki samego procesu Zabbixa, np. długość kolejki).

Wyzwalacz — logika oceny

Pojęcie: wyzwalacz (trigger). Wyrażenie logiczne oceniające wartości jednego lub kilku elementów danych i przyjmujące jeden z dwóch stanów: PROBLEM (warunek spełniony) albo OK (warunek niespełniony). Sam wyzwalacz niczego nie wysyła — jedynie zmienia stan, który następnie mogą odczytać akcje.

Pojęcie: wyrażenie wyzwalacza. Formuła odwołująca się do funkcji historycznych na elemencie danych, np.:

```
last(/NazwaHosta/vfs.fs.size[/,pfree])<15
```

Wyrażenie to nie porównuje bieżącej wartości bezpośrednio, tylko wynik funkcji `last()` (ostatnia zebrana wartość) — inne popularne funkcje to `avg()` (średnia z okna czasowego, np. `avg(...,5m)`, przydatna do wygładzania krótkich skoków) oraz `nodata()` (brak jakiejkolwiek wartości przez zadany czas, kluczowe do wykrycia sytuacji, w której element w ogóle przestał być zbierany, a nie tylko przekroczył próg).

Pojęcie: poziom istotności (severity). Etykieta wyzwalacza od Not classified przez Warning, Average, High, aż po Disaster — sama w sobie nic nie robi, ale jest głównym kryterium filtrowania w akcjach: politykę „wysyłaj SMS tylko dla Disaster, e-mail dla wszystkiego od Average w górę" budujemy właśnie na tym polu.

Uważaj: wyzwalacz w stanie PROBLEM, dla którego żadna akcja nie ma dopasowanego warunku, nigdy nie wygeneruje powiadomienia — stan będzie widoczny na pulpicie, ale cicho, co jest częstą przyczyną przeoczonych awarii przy źle poukładanych warunkach akcji.

Akcja — most między stanem a reakcją

> **Pojęcie: akcja (action).** Reguła łącząca zdarzenie (najczęściej zmianę stanu wyzwalacza na PROBLEM albo z powrotem na OK) z konkretną reakcją — wysłaniem powiadomienia, wykonaniem zdalnej komendy albo utworzeniem zgłoszenia w zewnętrznym systemie.



Wartość zebrana przez element danych trafia do wyzwalacza, który przy zmianie stanu tworzy zdarzenie odczytywane przez akcję (kliknięcie otwiera pełny rozmiar)

> **Pojęcie: warunek akcji (condition).** Filtr decydujący, czy dana akcja w ogóle się uruchomi dla konkretnego zdarzenia — np. „grupa hostów = Serwery produkcyjne" AND „poziom istotności ≥ Average". Wiele warunków łączonych jest logiką AND/OR, konfigurowaną osobno dla typu warunku.

> **Pojęcie: operacja akcji (operation).** Konkretna reakcja wykonywana, gdy warunki są spełnione — wysłanie wiadomości do zdefiniowanego użytkownika albo grupy użytkowników wybranym medium (e-mail, SMS, webhook), z opcjonalnym opóźnieniem eskalacji (np. „jeśli po 30 minutach nikt nie potwierdził, powiadom też drugą osobę").

Szablon i dziedziczenie

> **Pojęcie: szablon (template).** Gotowy zestaw elementów danych, wyzwalaczy, reguł wykrywania i pulpitów, przypinany do hosta zamiast konfigurowania każdej z tych rzeczy ręcznie dla każdej maszyny osobno. Podłączenie szablonu MySQL do stu serwerów baz danych oznacza sto identycznie skonfigurowanych zestawów monitoringu, utrzymywanych z jednego miejsca.

> **Pojęcie: dziedziczenie szablonu.** Zmiana wprowadzona w szablonie propaguje się automatycznie na wszystkie podłączone do niego hosty — dodanie nowego elementu do szablonu MySQL od razu pojawia się na każdym hoście z tym szablonem, bez ręcznego powielania zmiany. Element odziedziczony z szablonu można na poziomie konkretnego hosta nadpisać (np. inny próg wyzwalacza dla wyjątkowo obciążonego serwera), ale wtedy przestaje on podążać za dalszymi zmianami tego konkretnego pola w szablonie.

Reguła wykrywania niskopoziomowego i prototypy

> **Pojęcie: reguła wykrywania niskopoziomowego (LLD, low-level discovery).** Mechanizm automatycznego wykrywania zmiennej liczby podobnych obiektów na hoście — systemów plików, interfejsów sieciowych, procesorów, kontenerów Dockera — i tworzenia dla każdego z nich elementów danych i wyzwalaczy bez ręcznego dodawania po jednym na sztukę.

> **Pojęcie: prototyp elementu / wyzwalacza.** Szablon elementu albo wyzwalacza powiązany z regułą LLD, zawierający makro (np. `{#FSNAME}`), które reguła podstawia konkretną wartością dla każdego wykrytego obiektu. Dodanie nowego dysku do serwera z aktywną regułą LLD systemów plików skutkuje automatycznym pojawieniem się nowego elementu monitorującego zajętość tego dysku przy najbliższym cyklu wykrywania — bez żadnej ręcznej interwencji.

Makra i kolejność rozstrzygania

> **Pojęcie: makro.** Zmienna podstawiana w miejscu użycia — we frameworku szablonów pozwala napisać jedną definicję wyzwalacza z progiem `{$FS.PFREE.MIN}` zamiast osobnej definicji z wpisaną na sztywno liczbą dla każdego hosta z osobna.

Wartość makra jest rozstrzygana w ściśle określonej kolejności, od najbardziej szczegółowej do najbardziej ogólnej:

1. wartość ustawiona bezpośrednio na hoście,

2. wartość dziedziczona z szablonu podłączonego do hosta,
3. wartość globalna, ustawiona raz dla całej instalacji Zabbixa,
4. wartość domyślna zdefiniowana przy tworzeniu samego makra.

Dzięki tej kolejności ustawiamy jeden globalny próg `{$FS.PFREE.MIN} = 15` dla wszystkich hostów naraz, a dla pojedynczego serwera z wyjątkowo ciasnym dyskiem nadpisujemy go lokalnie na hoście na `5`, bez dotykania definicji globalnej ani szablonu.

Zdarzenie i problem

> **Pojęcie: zdarzenie (event).** Pojedynczy zapis powstający w chwili zmiany stanu wyzwalacza — jedno zdarzenie przy przejściu w PROBLEM, drugie przy powrocie do OK. To zdarzenia, a nie same wyzwalacze, są tym, co faktycznie odczytują akcje przy podejmowaniu decyzji o powiadomieniu.

> **Pojęcie: problem (problem).** Zdarzenie typu PROBLEM, które nie zostało jeszcze rozwiązane (wyzwalacz nie wrócił do stanu OK) — lista aktywnych problemów jest głównym, pierwszym widokiem otwieranym po zalogowaniu do frontendu, pokazującym wszystko, co w tej chwili wymaga uwagi w całej monitorowanej infrastrukturze.

> **Pojęcie: potwierdzenie problemu (acknowledge).** Ręczne oznaczenie przez administratora, że dany problem jest już zauważony i ktoś się nim zajmuje — nie zmienia to stanu samego wyzwalacza (wciąż jest PROBLEM, dopóki metryka faktycznie nie wróci do normy), ale pozwala odróżnić na liście problemów te, którymi już ktoś się zajął, od tych wciąż nietkniętych.

Częste błędy

Traktowanie elementu i wyzwalacza jako jednej rzeczy. Element tylko zbiera i zapisuje wartość — usunięcie elementu bez usunięcia powiązanych z nim wyzwalaczy zostawia wyzwalacze w stanie „element nieobsługiwany", generujące własny, osobny alarm.

Nadpisanie wartości na hoście i zapomnienie o tym. Wyjątek ustawiony bezpośrednio na hoście przestaje reagować na zmiany w szablonie — pół roku później nikt nie pamięta, dlaczego jeden serwer ma inny próg niż cała reszta grupy, a dokumentacja tego wyjątku nie istnieje.

Zbyt czuła reguła LLD. Reguła wykrywania odpytywana zbyt często na hoście z dużą liczbą dynamicznie tworzonych obiektów (np. kontenery uruchamiane i usuwane co kilka minut) potrafi generować i kasować setki elementów dziennie, co samo w sobie obciąża serwer i zaśmieca historię krótkotrwałymi seriami danych.

□□ Ćwiczenia

> Zadania sprawdzające, bez potrzeby dostępu do żywej instalacji Zabbixa.

Ćwiczenie 1 — złożenie łańcucha pojęć

Dla zdania „gdy wolne miejsce na `/var` spadnie poniżej 10%, wyślij e-mail administratorom" wskazujemy, który fragment odpowiada elementowi danych, który wyzwalaczowi, a który akcji.

Odpowiedź

Element danych to cykliczne zbieranie wartości wolnego miejsca na `/var` (np. `vfs.fs.size[/var,pfree]`). Wyzwalacz to wyrażenie logiczne `last(...)<10`, zmieniające stan na PROBLEM po przekroczeniu progu. Akcja to reguła łącząca zmianę stanu tego wyzwalacza (warunek) z wysłaniem e-maila do grupy administratorów (operacja).

Pytania kontrolne

1. Dlaczego zmiana progu w makrze globalnym nie wpływa na host, który ma ten sam próg nadpisany lokalnie?

Odpowiedź

Bo kolejność rozstrzygania makr sprawdza najpierw wartość ustawioną bezpośrednio na hoście i na niej się zatrzymuje — wartość globalna jest brana pod uwagę tylko wtedy, gdy ani host, ani żaden podłączony do niego szablon nie definiuje własnej wartości tego makra.

2. Po co w ogóle rozdzielać element danych od wyzwalacza, skoro oba dotyczą tej samej metryki?

Odpowiedź

Bo jeden element danych bywa źródłem dla wielu niezależnych wyzwalaczy o różnych progach i różnej istotności (np. Warning przy 20% wolnego miejsca i Disaster przy 5%), a rozdzielenie zbierania od oceny pozwala też zmieniać próg alarmowy bez dotykania definicji samego zbierania danych i bez utraty już zgromadzonej historii.

Powiązane: Architektura Zabbixa: serwer, proxy, agent, frontend · Monitorowanie typowych usług · Utrzymanie serwera Zabbix

Źródła: dokumentacja Zabbix 6.0, rozdział „Data collection", „Triggers" i „Discovery"

Revision #37
Created 2026-09-25 07:40:32 UTC by Tomasz Konieczny
Updated 2026-10-26 14:09:47 UTC by Tomasz Konieczny