Metricbeat — dokumentacja komponentu

Wersja: Windows (ZIP/offline), integracja z Kibana Dashboards  
*Data: 2025-08-20 16:15*

# 1) Cel komponentu

* Zbieranie metryk systemowych i usług na hostach Windows (CPU, pamięć, dyski, procesy, sieć, usługi).
* Dostarczanie danych do analizy w Kibanie (Dashboards, Lens) i pod alerty operacyjne (progi, odchylenia).

# 2) Zakres i funkcje

* Moduł `system`: CPU, memory, filesystem, disk I/O, network, procesy.
* Moduł `windows`: m.in. status usług, dzienniki wydajności (Perf Counters).
* Wsparcie tagowania (`fields`, `tags`) i korelacji z hostem (`host.name`, `agent.id`).
* Integracja z Kibana Dashboards (import automatyczny lub ręczny).
* Obsługa ILM/templating (przez `setup`), data views w Kibanie.

# 3) Wymagania (Windows, offline)

* System: Windows Server / Windows 10+ (x64).
* Uprawnienia: uruchomienie usługi, odczyt PerfMon/WMI, dostęp do plików systemowych metryk.
* Dostęp sieciowy do Elasticsearch i Kibany (hosty, porty, certyfikaty).
* Jeśli własne CA: w konfiguracji wskaż ścieżkę do CA (TLS).

# 4) Instalacja (ZIP/offline) — krok po kroku

1. Skopiuj i rozpakuj paczkę ZIP do: `C:\Elastic\metricbeat`.

2. Skonfiguruj połączenia w `metricbeat.yml` (Elasticsearch, Kibana).

3. Włącz potrzebne moduły (co najmniej `system`, opcjonalnie `windows`).

4. Zainstaluj jako usługę i uruchom.

Przykładowa konfiguracja `metricbeat.yml` (minimum):

metricbeat.modules:  
 - module: system  
 metricsets: [cpu, memory, network, diskio, filesystem, process]  
 period: 10s  
 processes: ['.\*']  
  
 - module: windows  
 metricsets: [service]  
 period: 30s  
  
output.elasticsearch:  
 hosts: ["https://<ES\_HOST>:9200"]  
 username: "<USER>"  
 password: "<PASS>"  
  
setup.kibana:  
 host: "https://<KIBANA\_HOST>:5601"  
 username: "<USER>"  
 password: "<PASS>"

Włączenie modułów (PowerShell, w katalogu `C:\Elastic\metricbeat`):

.\metricbeat.exe modules enable system windows

Instalacja i start usługi (PowerShell):

.\install-service-metricbeat.ps1\nStart-Service metricbeat

Uwaga (TLS): jeśli używasz własnego CA, dodaj w `output.elasticsearch.ssl.certificate\_authorities` oraz w `setup.kibana.ssl.certificate\_authorities` ścieżkę do pliku CA. Tymczasowe `ssl.verification\_mode: none` tylko do testów.

# 5) Dashboardy w Kibanie (dwie ścieżki)

A) Auto-import (rekomendowane):

C:\Elastic\metricbeat\metricbeat.exe setup --dashboards ^  
 -E setup.kibana.host=https://<KIBANA\_HOST>:5601 ^  
 -E setup.kibana.username=<USER> ^  
 -E setup.kibana.password=<PASS> ^  
 -E output.elasticsearch.hosts=["https://<ES\_HOST>:9200"] ^  
 -E output.elasticsearch.username=<USER> ^  
 -E output.elasticsearch.password=<PASS>

(Jeśli używasz Spaces: dodaj `-E setup.kibana.space.id=<SPACE\_ID>`.)

B) Ręczny import:

* Kibana → Stack Management → Saved Objects → Import → wskaż pliki dashboardów z katalogu `kibana\` w paczce Metricbeat (lub wewnątrz modułów).
* Zaznacz „Overwrite” przy aktualizacji.

Gdzie je znaleźć po imporcie: Kibana → Analytics → Dashboard → wyszukaj „Metricbeat …” (np. System Overview, Host metrics).

# 6) Gdzie w Kibanie patrzeć (i jak wyciągać wnioski)

* Dashboards: „Metricbeat System Overview” — szybki stan hosta (CPU, RAM, dyski, sieć, procesy).
* Discover: indeks/data view `metricbeat-\*` (lub `metrics-\*`). Kluczowe pola: `host.name`, `event.dataset`, `metricset.name`, `system.\*`, `windows.service.\*`.
* Analytics → Lens: własne wykresy trendów (np. CPU% per host, wolne miejsce per dysk).

# 7) KQL — gotowe filtry (Discover / Lens)

Najbardziej obciążone hosty (CPU > 90% w 15 min):

event.dataset : "system.cpu" and system.cpu.total.norm.pct >= 0.9 and @timestamp >= now-15m

Mało wolnego miejsca na dysku (<10%):

event.dataset : "system.filesystem" and system.filesystem.used.pct >= 0.9

Usługa Windows zatrzymana:

event.dataset : "windows.service" and windows.service.state : "stopped"

# 8) Lens — szybkie „recepty”

* CPU trend per host: Metric = Average `system.cpu.total.norm.pct` → Break down by `host.name`.
* Wolne miejsce per mount: Metric = Top values of `system.filesystem.used.pct` → Break down by `host.name` i `system.filesystem.mount\_point`.
* Usługi „stopped” w czasie: Metric = Count → Filter `windows.service.state: "stopped"` → Break down by `host.name`.

# 9) Alerty — wzorce reguł

CPU przeciążone:

KQL: event.dataset : "system.cpu" and system.cpu.total.norm.pct >= 0.9  
Warunek: „More than 0 matches” w 5 min, Group by `host.name`.

Mało miejsca na dysku:

KQL: event.dataset : "system.filesystem" and system.filesystem.used.pct >= 0.9  
Group by: `host.name`, `system.filesystem.mount\_point`.

Usługa zatrzymana:

KQL: event.dataset : "windows.service" and windows.service.state : "stopped"  
Group by: `host.name`, `windows.service.name`.

Sieć – wysoka przepustowość (dopasuj próg do środowiska):

KQL: event.dataset : "system.network"  
Rule typu threshold na `system.network.in.bytes` / `system.network.out.bytes` (sum/avg, okno 5–10 min).

# 10) Dobre praktyki

* Nazewnictwo: ustaw `host.name` spójnie z CMDB; dodaj `fields.env` (prod/dev/test) i `fields.role` (db/web/app).
* Okna czasowe: metryki co 10–30 s → dashboardy ustaw na Last 15m/1h; w alertach używaj „rolling window” 5–15 min.
* Szumy: filtrowanie mountów wirtualnych/temp; procesy krótkotrwałe mogą generować outliery.
* ILM: przy pełnym `setup` skonfiguruj retencję metryk (gorąca → ciepła → kasowanie).

# 11) Rozwiązywanie problemów (Troubleshooting)

* Brak danych: sprawdź usługę (`Get-Service metricbeat`), logi w `logs\`, firewall/proxy do ES, Timepicker w Kibanie.
* Brak `windows.service.\*`: upewnij się, że moduł `windows` jest włączony i użytkownik usługi ma dostęp do WMI/PerfMon.
* Błędy TLS: dodaj CA do `output.elasticsearch.ssl.certificate\_authorities` oraz do `setup.kibana.ssl.certificate\_authorities`.
* Wysokie zużycie: podnieś `period` na 30–60 s, ogranicz `processes`, wyłącz zbędne metricsety.
* Space / Data View: jeśli dashboardy „puste”, sprawdź, czy `metricbeat-\*` istnieje w danym Space i czy zakres czasu obejmuje świeże dane.