Kibana — wykorzystanie danych z Metricbeat

Propozycja dokumentu: gotowe widoki, KQL, Lens, alerty i korelacje dla metryk z Metricbeat.  
*Data: 2025-08-21 07:22*

# 1) Cel dokumentu

* Pokazać co i jak wyciągać z danych Metricbeat w Kibanie, gdy agent już wysyła metryki.
* Dostarczyć gotowe widoki, zapytania KQL, przepisy Lens i alerty do szybkiego użycia.
* Zakres: hosty Windows/Linux (CPU, pamięć, dyski, sieć, procesy/usługi), wybrane moduły (np. windows, mssql).

# 2) Gdzie są dane i jak je przeglądać

* Data view: metricbeat-\* (w nowszych wdrożeniach także metrics-\*).
* Najważniejsze pola (ECS + Metricbeat): @timestamp, host.name, event.dataset, metricset.name, agent.type.
* CPU: system.cpu.total.norm.pct; Pamięć: system.memory.actual.used.pct (lub system.memory.used.pct), system.memory.total;
* Dyski: system.filesystem.used.pct, system.filesystem.mount\_point; IO: system.diskio.\*;
* Sieć: system.network.in.bytes / out.bytes, system.network.name;
* Procesy/usługi: process.name, process.pid, process.cpu.\*, process.memory.\*, windows.service.state/name.
* Szybki start: Kibana → Discover → wybierz data view → Timepicker (np. “Last 24 hours”).

# 3) Gotowe widoki (po imporcie dashboardów)

* Metricbeat System Overview — CPU, memory, filesystem, network dla hostów.
* Host metrics / Host overview — przegląd per host.name.
* Modułowe (np. MSSQL, Nginx, Redis) — jeśli moduły zostały włączone.

Własne dashboardy dla konkretnych ról (DB/Web/App) — patrz sekcje 5–6.

# 4) Typowe pytania, na które odpowie Kibana

* Które hosty mają najwyższe zużycie CPU/RAM? (trend i ranking).
* Gdzie kończy się wolne miejsce na dyskach? (thresholdy 80/90/95%).
* Jakie procesy zjadają zasoby? (CPU%/RSS, zestawienie top N).
* Czy występują anomalie w przepływie sieci? (nagłe piki in/out bytes).
* Które usługi Windows są zatrzymane?

# 5) Przepisy „Lens” (wizualizacje 30–90 s)

A. CPU% per host (trend + ranking)

* Filter: event.dataset:"system.cpu"; Metric: Average of system.cpu.total.norm.pct (×100 w formacie %); Break down by: host.name; Format: Line chart.

B. Pamięć — zużycie w %

* Filter: event.dataset:"system.memory"; Metric: Average of system.memory.actual.used.pct (lub used.pct); Break down by: host.name; Visualization: Line/Bar.

C. Dyski — krytyczne wolumeny

* Filter: event.dataset:"system.filesystem"; Metric: Max of system.filesystem.used.pct; Break down by: host.name; Add table z mount\_point / device.

D. Sieć — ruch przychodzący/wychodzący

* Filter: event.dataset:"system.network"; Metric: Sum of system.network.in.bytes i out.bytes; Break down by: host.name; Line chart (możesz użyć funkcji „Rate/Differences”, jeśli dostępna).

E. Top procesy CPU

* Filter: event.dataset:"system.process"; Table: columns → @timestamp, host.name, process.name, process.pid, process.cpu.total.pct; sort desc po CPU; (ew. dodaj panel z top N).

F. Usługi Windows — stan

* Filter: event.dataset:"windows.service"; Metric: Last value of windows.service.state; Breakdown: windows.service.name; Wykres: Bar/Heatmap (UP/DOWN).

# 6) Propozycje dashboardów własnych

„Host — zdrowie i pojemność” — panele:

* CPU p95 (system.cpu.total.norm.pct) + RAM avg (system.memory.actual.used.pct) per host.name.
* Filesystem — top mount\_point z system.filesystem.used.pct (max).
* Network — sum in/out bytes per host.name (trend).
* Tabela „Ostatnie alarmy metryczne” — widok pod alerty z sekcji 8.

„Procesy — kto zjada zasoby” — panele:

* Top process.name po process.cpu.total.pct + tabela szczegółów (PID/host).
* Procesy o dużym memory RSS (process.memory.rss.bytes).
* Trend CPU konkretnego procesu (filter process.name:"java").

# 7) KQL — gotowe „cookbook” (kopiuj-wklej)

event.dataset : "system.cpu" and system.cpu.total.norm.pct >= 0.9 and @timestamp >= now-15m # CPU > 90%

event.dataset : "system.memory" and system.memory.actual.used.pct >= 0.9 # RAM > 90% (lub system.memory.used.pct)

event.dataset : "system.filesystem" and system.filesystem.used.pct >= 0.9 # dysk zajęty ≥ 90%

event.dataset : "system.network" and system.network.in.bytes >= 1073741824 # ruch >= 1 GB (okres okna)

event.dataset : "system.process" and process.name : "java" # procesy java

event.dataset : "windows.service" and windows.service.state : "stopped" # zatrzymane usługi

# 8) Alerty — gotowy pakiet reguł

A. CPU przeciążone

* KQL: event.dataset:"system.cpu" and system.cpu.total.norm.pct >= 0.9; Okno: 5 min; Group by: host.name.

B. Mało pamięci

* KQL: event.dataset:"system.memory" and system.memory.actual.used.pct >= 0.9; Okno: 5–10 min; Group by: host.name.

C. Wolne miejsce na dysku < 10%

* KQL: event.dataset:"system.filesystem" and system.filesystem.used.pct >= 0.9; Group by: host.name + mount\_point; warunek: More than 0 matches.

D. Usługa zatrzymana

* KQL: event.dataset:"windows.service" and windows.service.state:"stopped"; Group by: host.name + windows.service.name; Okno: 5 min.

E. Host „zamilkł” (brak metryk)

* Typ: Less than 1 match; KQL: host.name:"\*" (lub konkretny host); Okno: 5–10 min; interpretacja: host down/agent nie raportuje.

# 9) Korelacje i łączenie z innymi danymi

* Filebeat: sprawdź logi hosta/aplikacji w tych samych oknach czasu; koreluj piki CPU z błędami (log.level:error).
* APM: porównaj p95 latency usług z CPU/RAM hosta (host.name / service.node.name).
* Heartbeat: zestaw spadki dostępności z pikiem CPU/dysku na backendzie.
* Packetbeat: jeśli wysokie opóźnienia — sprawdź flows i błędy HTTP/DNS.

# 10) Raportowanie i współdzielenie

* Saved Search: „CPU > 90% (15m)” i „Disk used ≥ 90%”.
* Dashboard PDF: harmonogram (Reporting) dla tygodniowego przeglądu pojemności i zdrowia hostów.
* Skrócone linki z filtrami (np. host.role, host.name, environment).

# 11) Dobre praktyki pracy z metrykami

* Spójne tagi: fields.env (prod/stage/dev), fields.role (db/web/app), host.name zgodny z CMDB.
* Interwał agregacji dopasuj do wolumenu (Auto/1m/5m) — zbyt drobny może zjadać zasoby.
* Uważaj na kardynalność (np. process.command\_line) w tabelach/terms.
* W panelach procentowych ustaw format % (×100 przy norm.pct).
* Retencja/ILM: definiuj polityki dla metricbeat-\* / metrics-\* (gorąca→ciepła→kasowanie).

# 12) „Starter pack” (do importu / odtworzenia)

Saved Searches:

CPU > 90% — Last 15m: event.dataset:"system.cpu" and system.cpu.total.norm.pct >= 0.9 and @timestamp >= now-15m

DISK Used ≥ 90%: event.dataset:"system.filesystem" and system.filesystem.used.pct >= 0.9

RAM > 90%: event.dataset:"system.memory" and system.memory.actual.used.pct >= 0.9

Dashboardy:

* Host — Health & Capacity (sekcja 6).
* Processes — CPU & Memory (sekcja 6).

Alerty:

* CPU\_HIGH\_90 (sekcja 8A).
* MEM\_HIGH\_90 (sekcja 8B).
* DISK\_FULL\_90 (sekcja 8C).
* WIN\_SERVICE\_STOPPED (sekcja 8D).
* HOST\_SILENCE (sekcja 8E).

# 13) Aneks — pola przydatne w analizie

* system.cpu.total.norm.pct — CPU% (0..1).
* system.memory.actual.used.pct — RAM% (0..1).
* system.filesystem.used.pct — zajętość dysku (0..1).
* system.network.in.bytes / out.bytes — bajty (liczniki).
* process.name, process.pid, process.cpu.total.pct, process.memory.rss.bytes — procesy.
* windows.service.name / windows.service.state — usługi.

# Co dalej?

Mogę przygotować pakiet Saved Objects (dashboardy, saved searches, alerty) dopasowany do Twoich hostów/zasobów oraz wersji Kibany.