Metricbeat → Kibana: przewodnik (od danych do wartości)

Krótka, praktyczna ścieżka „od danych do wartości” dla Metricbeat w Kibanie. Zrobisz to w 5 krokach + gotowe przykłady KQL.

# 1) Sprawdź, że dane z Metricbeat faktycznie wpływają

1. Otwórz Discover (lewy pasek → Discover).

2. Ustaw zakres czasu na „Last 15 minutes”.

1. 3. Wpisz filtr KQL:

agent.type : "metricbeat"

Powinieneś zobaczyć dokumenty z polami event.module, event.dataset, host.\*, @timestamp. Uwaga: dane mogą trafiać do indeksów metricbeat-\* (standalone) lub metrics-\* (przez Elastic Agent).

# 2) Szybkie widoki: Infrastructure → Hosts / Inventory

1. Observability → Infrastructure → Hosts – przegląd hostów, filtry KQL (np. host.os.type : "linux"), CPU/mem/dysk/sieć.

2. Observability → Infrastructure → Inventory – kafelki zasobów (hosty/kontenery/usługi) ze stanem i metrykami.

# 3) Metrics Explorer → wykres → alert w 30 sekund

1. Otwórz Metrics Explorer (w Infrastructure).

2. Wybierz metrykę np. system.cpu.total.norm.pct (Average) i rozbij po host.name.

3. Zapisz panel (Save) lub utwórz alert (Create alert) z progiem, np. CPU > 90% przez 5 min (metric threshold rule).

# 4) Własne wizualizacje w Lens (i dashboard)

* Analytics → Discover: zapisz wyszukiwanie (np. event.module : "system").
* Analytics → Visualize → Lens: przeciągnij system.cpu.total.norm.pct → oś Y (Average), rozbij po host.name; ustaw format %.
* Zrób panele dla pamięci (system.memory.used.pct), dysku (system.filesystem.used.pct), procesów (system.process.cpu.total.norm.pct).
* Dodaj panele do własnego Dashboardu.

# 5) Monitoring samej platformy (opcjonalnie)

Metricbeat może zasilać Stack monitoring (Kibana/Elasticsearch), jeśli włączysz odpowiednie moduły/ustawienia.

# Przydatne KQL (kopiuj-wklej w Discover/Hosts/Inventory/Metrics Explorer)

**• Wszystkie zdarzenia Metricbeat:**agent.type : "metricbeat"

**• System/CPU:**event.module : "system" and event.dataset : "system.cpu"

**• Hosty Linux:**host.os.type : "linux"

**• CPU > 90%:**system.cpu.total.norm.pct > 0.9

**• Pamięć zajęta > 80%:**system.memory.used.pct > 0.8

**• Mało wolnego miejsca (filesystem > 90%):**system.filesystem.used.pct > 0.9

**• Krytyczne mount-pointy:**system.filesystem.mount\_point : ("/" or "/var" or "C:\\") and system.filesystem.used.pct > 0.85

**• Intensywny proces .NET:**process.name : "dotnet" and system.process.cpu.total.norm.pct > 0.5

**• Wąska selekcja hostów:**host.name : ("srv-app-01" or "srv-app-02")

**• Host bez swapu:**not system.memory.swap.total

**• Duży ruch sieciowy (in):**system.network.in.bytes > 100000000

**• Wybór po datasetach:**event.dataset : "system.memory" or event.dataset : "system.filesystem"

# Minimalny zestaw alertów (progi startowe)

* CPU: Avg system.cpu.total.norm.pct ≥ 0.9 przez 5 min.
* Pamięć: Avg system.memory.used.pct ≥ 0.85 przez 5 min.
* Dysk: Avg system.filesystem.used.pct ≥ 0.9 przez 5 min.
* Skonfiguruj jako Metric threshold rule i podłącz kanał (e-mail/Slack/webhook).

# Szybkie debugowanie, gdy „czegoś brakuje”

• Brak CPU/memory/filesystem? Upewnij się, że moduł system Metricbeat jest aktywny i że w danych widać event.dataset (np. system.cpu, system.memory).

• Trzymaj się standardu ECS/Beats – niestandardowe typy pól potrafią psuć aplikacje Kibany (Infrastructure/SIEM).