Metricbeat – Pola w logach i co z nich wyciągniesz (KQL + alerty)

Praktyczny katalog najważniejszych pól pojawiających się w zdarzeniach Metricbeat (ECS + moduł system). Dla każdego pola: zwięzły opis, na co patrzeć w analizie oraz przykład KQL/alertu. Zestaw jest kuratorski (najczęstsze pola do operacyjnej obserwowalności).

# 1) Metadane zdarzenia (ECS / Base)

|  |  |  |  |
| --- | --- | --- | --- |
| Pole | Opis | Co daje analiza | Przykład KQL / alert |
| @timestamp | Chwila zebrania metryki (czas zdarzenia). | Oś czasu, okna agregacji, korelacja z logami/APM; wykrywanie luk w danych. | @timestamp >= now()-15m |
| event.module | Moduł zbierający metryki (np. system, redis, elasticsearch). | Szybkie filtrowanie i budowa paneli tylko z jednego źródła metryk. | event.module : "system" |
| event.dataset | Konkretna paczka metryk (np. system.cpu, system.memory). | Segmentacja CPU vs pamięć vs dysk; routing do dashboardów. | event.dataset : "system.cpu" |
| event.category / event.type / event.kind | Klasyfikacja zdarzenia (kategoria, typ, rodzaj). | Ujednolicenie widoków i alertów między różnymi źródłami. | event.category : "host" |
| ecs.version | Wersja schematu ECS użyta w dokumencie. | Kompatybilność pól po aktualizacjach agentów. | ecs.version : "8.\*" |
| data\_stream.type / dataset / namespace | Identyfikacja data streamu (np. metrics-system.cpu-prod). | Retencja/ILM i separacja środowisk (prod/dev/test). | data\_stream.dataset : "system.cpu" and data\_stream.namespace : "prod" |
| metricset.name | Nazwa zebranych metryk (np. cpu, memory). | Diagnostyka brakujących paczek metryk. | metricset.name : "cpu" |
| metricset.period | Interwał zbierania metryk (ms). | Dopasowanie okien alertów do częstotliwości zbioru (np. 60s). | metricset.period >= 60000 |

# 2) Kontekst źródła (agent / host / cloud / container / service)

|  |  |  |  |
| --- | --- | --- | --- |
| Pole | Opis | Co daje analiza | Przykład KQL / alert |
| agent.type / agent.version / agent.name | Typ i wersja agenta oraz jego nazwa. | Inwentaryzacja agentów, regresje po upgrade. | agent.type : "metricbeat" and agent.version >= "8.13.0" |
| host.name / host.hostname | Nazwa hosta (DNS/logiczna). | Rozbicie wykresów per host; pivot do szczegółów. | host.name : ("srv-app-01" or "srv-app-02") |
| host.id / host.ip | Stabilny identyfikator i adres(y) IP hosta. | Łączenie danych z CMDB; śledzenie reimagingu. | host.ip : "10.0.0.\*" |
| host.os.type / host.os.version | Rodzina systemu i wersja (linux/windows/darwin). | Inne progi alertów i inne panele dla OS. | host.os.type : "linux" |
| cloud.provider / region / instance.id | Kontekst chmury (dostawca, region, instancja). | Heatmapy kosztów i prawidłowe rightsizing. | cloud.provider : "aws" and cloud.region : "eu-central-1" |
| container.id / container.name | Identyfikator i nazwa kontenera. | Pivot z hosta do poziomu kontenera (noisy neighbor). | container.name : "api-gateway" |
| kubernetes.pod.name / node.name | Kontekst Kubernetes (pod/węzeł). | Triangulacja problemów per pod i node. | kubernetes.pod.name : "orders-\*" |
| service.name / service.environment | Nazwa usługi i środowisko (prod/dev). | SLO i alerty per usługa, spójność z APM. | service.name : "checkout-api" and service.environment : "prod" |

# 3) Moduł system – CPU i Load

|  |  |  |  |
| --- | --- | --- | --- |
| Pole | Opis | Co daje analiza | Przykład KQL / alert |
| system.cpu.total.pct | Całkowite wykorzystanie CPU (0–1). | Szybki wskaźnik obciążenia; dobre do trendów. | system.cpu.total.pct > 0.9 |
| system.cpu.total.norm.pct | CPU znormalizowane per rdzeń (0–1). | Porównywalność między hostami o różnej liczbie rdzeni; progi alertów. | avg(system.cpu.total.norm.pct) > 0.9 |
| system.cpu.user.pct / system.cpu.system.pct | Udział czasu w trybie user/system. | Diagnostyka: CPU‑bound aplikacja vs narzut jądra/I/O. | system.cpu.user.pct > 0.7 |
| system.cpu.cores | Liczba rdzeni logicznych. | Normalizacja metryk i planowanie pojemności. | system.cpu.cores >= 4 |
| system.load.1 / .5 / .15 (Unix) | Średnie obciążenie systemu z 1/5/15 min. | Kolejka zadań; load≈liczba rdzeni → granica nasycenia. | system.load.5 > system.cpu.cores |

# 4) Moduł system – Pamięć

|  |  |  |  |
| --- | --- | --- | --- |
| Pole | Opis | Co daje analiza | Przykład KQL / alert |
| system.memory.used.pct | Udział zajętej pamięci (0–1). | Wczesne ostrzeganie o ryzyku OOM. | avg(system.memory.used.pct) >= 0.85 |
| system.memory.actual.used.pct | Użycie „rzeczywiste” (bez cache/buffers). | Rozróżnia problem realny vs cache; lepsze decyzje o skalowaniu. | system.memory.actual.used.pct > 0.8 |
| system.memory.total / free | Całkowita i wolna pamięć RAM. | Kontekst do procentów; planowanie pojemności. | system.memory.free < 1073741824 # <1 GiB |
| system.memory.swap.total / used.pct | Wielkość i użycie swap. | Wykrywanie thrashingu; tuning swappiness. | system.memory.swap.used.pct > 0.3 |

# 5) Moduł system – Filesystem i IO dysku

|  |  |  |  |
| --- | --- | --- | --- |
| Pole | Opis | Co daje analiza | Przykład KQL / alert |
| system.filesystem.used.pct | Wypełnienie systemu plików (0–1). | Prewencja „disk full”; trendy wzrostu. | system.filesystem.used.pct > 0.9 |
| system.filesystem.mount\_point / type | Punkt montowania i typ FS (ext4/xfs/ntfs...). | Priorytetyzacja krytycznych wolumenów (/ /var /data / C:\). | system.filesystem.mount\_point : ("/" or "/var" or "C:\\") |
| system.diskio.read.bytes / write.bytes | I/O bajty odczytu/zapisu per urządzenie. | Diagnoza bottlenecków storage; korelacja z latencją. | system.diskio.write.bytes > 100000000 |
| system.diskio.read.time / write.time | Czas odczytu/zapisu (ms). | Wzrost opóźnień IO → degradacja aplikacji. | system.diskio.read.time > 5000 |

# 6) Moduł system – Sieć

|  |  |  |  |
| --- | --- | --- | --- |
| Pole | Opis | Co daje analiza | Przykład KQL / alert |
| system.network.name | Nazwa interfejsu sieciowego. | Rozkład ruchu per interfejs; diagnoza błędów na konkretnym NIC. | system.network.name : ("eth0" or "ens\*") |
| system.network.in.bytes / out.bytes | Przepływ bajtów RX/TX w interwale zbioru. | Nasycenie łącza; spike’i ruchu. | system.network.in.bytes > 100000000 |
| system.network.in.errors / out.errors | Błędy RX/TX. | Jakość łącza/kabla/sterownika; problemy CRC. | system.network.out.errors > 0 |
| system.network.in.packets / out.packets | Pakiety RX/TX. | Charakter ruchu (małe vs duże pakiety), PPS‑y. | system.network.out.packets > 1000000 |

# 7) Moduł system – Procesy, Uptime, Użytkownicy

|  |  |  |  |
| --- | --- | --- | --- |
| Pole | Opis | Co daje analiza | Przykład KQL / alert |
| process.name / process.pid | Nazwa i PID procesu. | Identyfikacja sprawcy obciążenia; whitelisting krytycznych procesów. | process.name : "dotnet" |
| system.process.cpu.total.norm.pct | Użycie CPU przez proces (znormalizowane 0–1). | Alarmy na runaway process; porównania między hostami. | system.process.cpu.total.norm.pct > 0.5 |
| system.process.memory.rss.bytes / size | Pamięć rezydentna i rozmiar procesu. | Wykrywanie wycieków pamięci i nietypowych wzrostów. | system.process.memory.rss.bytes > 1073741824 # >1 GiB |
| system.process.state | Stan procesu (running/sleeping/zombie...). | Diagnostyka wiesza się/nie odpowiada vs normalny idle. | system.process.state : "zombie" |
| process.command\_line | Pełna linia poleceń procesu. | Walidacja parametrów uruchomienia (porty, config). | process.command\_line : "\*--environment=prod\*" |
| system.uptime.\* | Czas działania hosta (uptime). | Wykrywanie restartów, okna serwisowe. | exists:system.uptime.\* |
| system.users.\* | Użytkownicy zalogowani na host. | Wykrywanie interaktywnych logowań na serwery produkcyjne. | exists:system.users.name |

# 8) Starter – progi alertów (Metric threshold)

• CPU: avg(system.cpu.total.norm.pct) ≥ 0.90 przez 5 min – wysokie, stałe obciążenie.

• RAM: avg(system.memory.used.pct) ≥ 0.85 przez 5 min – ryzyko OOM/thrashing.

• FS: avg(system.filesystem.used.pct) ≥ 0.90 – prewencja 'disk full'.

• Proces krytyczny: system.process.cpu.total.norm.pct ≥ 0.5 dla process.name : "<usługa>".

• Sieć: system.network.in.errors > 0 lub out.errors > 0 – degradacja łącza/NIC.