**Kibana – mini‑słownik pojęć  
(Beats: Filebeat/Metricbeat/Winlogbeat/Packetbeat/Heartbeat + APM/RUM)**

*Praktyczna ściąga do interfejsu UI Kibany*

*Data: 21 sierpnia 2025*

# Wprowadzenie

Ten dokument to zwięzły mini‑słownik pojęć i elementów interfejsu Kibany, skrojony pod strumienie danych z Beats (Filebeat, Metricbeat, Winlogbeat, Packetbeat, Heartbeat) oraz APM/RUM (.NET i Angular). Zawiera krótkie definicje, kontekst użycia oraz przykładowe zapytania KQL. Nagłówki mają style zgodne z Word (Heading 1/2), więc możesz z nich łatwo wygenerować spis treści w Wordzie.

# Podstawy UI (widoczne niemal wszędzie)

* Data view (dawniej index pattern) – logiczny widok danych (np. „logs-\*”, „metrics-\*”), definiuje dostępne pola i pozwala filtrować KQL-em.
* Data stream – nowocześniejsze podejście do indeksowania; posiada pola data\_stream.type/dataset/namespace i automatyczny rollover.
* ECS (Elastic Common Schema) – wspólny schemat pól (np. host.name, service.name, event.dataset), ułatwia korelację logów/metryk/śladów.
* KQL (Kibana Query Language) – język filtrów w pasku wyszukiwania. Filtry możesz też dodawać jako „kafle” poniżej paska.
* Timepicker – globalny wybór zakresu czasu (prawy górny róg). Wpływa na wszystkie widoczne widżety i listy.
* Filters – włączalne/wyłączalne kafle filtrów bez edycji KQL.
* Saved objects – zapisane wyszukiwania, wizualizacje, dashboardy, SLO itp.
* Spaces & Roles – wydzielanie obszarów i nadawanie uprawnień w Kibanie.

## Przykłady KQL

agent.type != "filebeat"

http.response.status\_code >= 500

service.name : "orders-api" and service.environment : "prod"

# Fleet & Integracje (Elastic Agent/Beats)

* Fleet – zarządzanie agentami i integracjami centralnie z poziomu Kibany.
* Agent – proces zbierający dane (zastępuje wiele osobnych Beats; jednak Beats nadal są wspierane).
* Agent policy – polityka dla grupy agentów; zawiera integracje i ustawienia wejść (inputs).
* Integration – pakiet dla konkretnego źródła (System, Windows, Nginx, APM, Synthetics).
* Input – definicja wejścia w integracji (logi plikowe, Winlog, Prometheus, http/tcp/icmp).
* Namespace – kontekst (np. „dev”, „prod”), trafia do data\_stream.namespace.
* Dataset – rodzaj danych (np. system.syslog, windows.sysmon), trafia do event.dataset oraz data\_stream.dataset.

# Logi (Filebeat / Winlogbeat)

* Logs → Stream – ciągły strumień logów z przeglądarką „Log rate”, „Highlights” i szybkimi filtrami.
* event.module / event.dataset – moduł i dataset (np. system.syslog, windows.sysmon).
* log.level – poziom logowania (info, warn, error, fatal).
* host.name / agent.hostname – identyfikacja hosta lub agenta.
* message – surowa treść logu.

## Przykłady KQL

service.name : "api" and log.level : "error"

event.dataset : "windows.sysmon"

# Metryki (Metricbeat)

* Infrastructure → Hosts/Pods/Containers – inwentarz hostów i klastrów z CPU/memory/load.
* Metrics Explorer – ad‑hoc wykresy metryk z Group by i agregacjami (avg, max, p95).
* Typowe pola: host.cpu.usage, system.memory.actual.used.pct, system.load.1.
* Anomalies (ML) – wykrywanie nietypowych odchyleń metryk (opcjonalnie).

## Przykłady KQL

host.cpu.usage > 0.9

container.image.name : "\*nginx\*"

# Uptime / Synthetics (Heartbeat)

* Monitors – ICMP, TCP, HTTP oraz Browser (synthetic journeys).
* Status – Up/Down; wskaźniki Availability i czasy odpowiedzi (p95/p99).
* Waterfall (dla Browser) – przebieg ładowania, kroki (steps), zrzuty ekranu.
* TLS – szczegóły certyfikatu (termin ważności, issuer).
* Locations – miejsca wykonywania testów (np. regiony chmurowe).

## Przykłady KQL

monitor.type : "http"

monitor.status : "down"

# Sieć (Packetbeat)

* Network → Flows – ruch między hostami (bytes in/out, liczba połączeń).
* Network → HTTP/DNS/… – widoki protokołowe (HTTP, DNS, TLS, MySQL itp.).
* source.\* / destination.\* – adresy i porty źródła/celu.
* client.\* / server.\* – rola uczestników połączenia (dla niektórych protokołów czytelniejsza niż src/dst).
* Kluczowe pola: http.response.status\_code, dns.question.name, tls.server.hash.sha256.

## Przykład KQL

network.transport : "tcp" and http.response.status\_code >= 500

# APM (.NET & RUM/Angular)

* Services – lista usług z metrykami: Latency, Throughput, Failed transaction rate, Errors, Dependencies, Instances.
* Transactions – operacje biznesowe (np. żądania HTTP, page‑load/route‑change w RUM).
* Spans – pod‑operacje (zapytania DB, wywołania HTTP, render).
* Traces – pełne ślady rozproszone (transakcje + spany).
* Service map – graf zależności usług (automatycznie budowany na podstawie śladów).
* Environments – dev/stage/prod (filtr w APM).
* Service version – wersja releasu aplikacji (diagnostyka regresji).
* RUM (Angular) – page‑load/route‑change oraz Core Web Vitals: LCP, INP (następca FID), CLS, TTFB.
* Source maps – mapy źródeł JS do czytelnych stack‑trace’ów w błędach.

## Przykłady KQL

service.name : "orders-api" and event.outcome : "failure"

transaction.type : "page-load" and transaction.duration.us > 4000000

# Alerty, Reguły, Akcje

* Rules – definicje alertów (np. APM error rate, Uptime monitor down, Log threshold).
* Connectors – kanały powiadomień (e‑mail, Slack, webhook, Teams, itp.).
* Action groups – różne stany alertu (firing, recovered) z przypisanymi akcjami.
* Throttle / Mute / Snooze – ograniczenie częstotliwości wysyłek i czasowe wyciszenie.
* Alert status – Active, Recovered, Acknowledged.
* Cases – sprawy incydentowe łączące alerty, komentarze, załączniki.

## Wzorzec: próg dla logów

log.level : "error" and service.name : "api"

# SLO w Observability (Respond quicker with SLOs)

* SLO (Service Level Objective) – cel jakości (np. 99,9% availability albo p95 latency ≤ 4 s).
* SLI (Service Level Indicator) – metryka obliczania SLO (np. odsetek 2xx, p95).
* Error budget – 1 − SLO; dopuszczalny margines awarii w oknie czasu.
* Burn rate – tempo „spalania” budżetu błędów; wysokie oznacza szybką degradację.
* SLO alerts – reguły na burn‑rate z krótkim i długim oknem dla czułości i stabilności.

# Analytics & Wizualizacje

* Discover – szybkie przeszukiwanie danych tabelarycznie; zapisywanie zapytań; eksport CSV.
* Lens – wizualizacje typu „drag‑and‑drop” (agregacje, breakdowny, porównania).
* Dashboards – kompozycje kafelków; wspólny timepicker, KQL i filtry drill‑down.
* TSVB (Time Series Visual Builder) – zaawansowane serie czasowe i panelem Markdown.
* Canvas – prezentacje na żywo (opcjonalnie, bardziej „slajdowe”).

# Zarządzanie danymi

* Index Management – zarządzanie indeksami i statystykami segmentów.
* Index Lifecycle Management (ILM) – polityki rollover → warm → cold → delete (ważne przy data streams).
* Ingest Pipelines – przetwarzanie przy wlocie (processors: grok, rename, geoip, dissect).
* Transforms – zdenormalizowane widoki/analityki (np. metryki per serwis/dzień).

# Najważniejsze pola (Beats + APM) – ściąga

* agent.type – np. filebeat, metricbeat, winlogbeat, packetbeat, heartbeat, apm.
* event.module / event.dataset – np. system.syslog, windows.sysmon.
* data\_stream.type / data\_stream.dataset / data\_stream.namespace – np. logs‑, metrics‑, „default”.
* service.name / service.environment / service.version – identyfikacja usług (APM i logi aplikacji).
* host.name, cloud.\* – kontekst maszyny/chmury.
* http.\*, url.\*, user\_agent.\*, source.\*, destination.\* – kluczowe dla ruchu web/sieć.
* transaction.\*, trace.\*, span.\* – ślady APM i RUM.

# Aneks: przykłady KQL (kopiuj‑wklej)

event.dataset : "nginx.access" and http.response.status\_code >= 500

data\_stream.type : "logs" and log.level : ("error" or "fatal")

agent.type : "heartbeat" and monitor.status : "down"

service.environment : "prod" and event.outcome : "failure"

source.ip : "10.0.0.0/8" and destination.port >= 1024