Kibana — wykorzystanie danych z Filebeat

Propozycja dokumentu: gotowe widoki, KQL, Lens, alerty i korelacje dla logów z Filebeat.  
*Data: 2025-08-21 07:10*

# 1) Cel dokumentu

* Pokazać co i jak wyciągać z danych Filebeat w Kibanie, gdy agent już wysyła logi.
* Dostarczyć gotowe widoki, zapytania KQL, przepisy Lens i alerty do szybkiego użycia.
* Zakres: logi aplikacyjne (NDJSON/tekst + multiline) i moduły Filebeat (np. IIS).

# 2) Gdzie są dane i jak je przeglądać

* Data view: filebeat-\* (w nowszych wdrożeniach także logs-\*).
* Najważniejsze pola (ECS + Filebeat): @timestamp, message, log.level, log.logger, host.name, agent.type, event.dataset, event.module, labels.\* / app.\* (jeśli dodane), dla IIS: url.\*, http.\*, user.agent.\*, client.ip, source.ip, response.\*, iis.\*.
* Szybki start: Kibana → Discover → wybierz data view → ustaw Timepicker (np. “Last 24 hours”).

# 3) Gotowe widoki (po imporcie dashboardów)

* Filebeat Overview – wolumen logów, top źródła, poziomy logów.
* [Filebeat] IIS logs – rozkład sc-status, cs-uri-stem, time-taken, top klienci i hosty.

Dla niestandardowych logów NDJSON przygotuj własny dashboard (patrz sekcje 5–6).

# 4) Typowe pytania, na które odpowie Kibana

* Czy rośnie liczba błędów? (trend log.level: error/critical, heatmap per host/usługa)
* Które endpointy mają najwięcej 5xx? (IIS/Nginx: http.response.status\_code >= 500)
* Kto i skąd „hałasuje”? (top source.ip / client.ip, rozkład UA/referrer)
* Czy jest „cisza w logach”? (brak zdarzeń dla określonej aplikacji/hosta)
* Jakie błędy aplikacyjne dominują? (agregacja po log.logger / error.type)
* Jak wygląda czas odpowiedzi? (IIS: iis.access.time\_taken / event.duration)

# 5) Przepisy „Lens” (wizualizacje 30–60 s)

A. Trend błędów aplikacji (NDJSON):

* Metric: Count + Filter: log.level:("error" or "critical"); Break down by: app.name (lub labels.app/service.name); Line chart.

B. Top endpointy 5xx (IIS):

* Filter: event.dataset:"iis.access" and http.response.status\_code >= 500; Metric: Count; Break down by: url.path (Top 10); Table z url.path/status/host.name.

C. Czas odpowiedzi (IIS „time-taken”):

* Filter: event.dataset:"iis.access"; Metric: 95th percentile of iis.access.time\_taken (ms); Break down by host.name.

D. „Hałasujące” IP:

* Filter: event.category:"web" (lub dataset IIS); Metric: Sum network.bytes lub Count; Break down by client.ip (Top 10); small multiples po host.name.

E. Cisza w logach (kontrola zdrowia):

* Metric: Count; Filter: app.name:"myapp"; gdy spada do zera → reguła alertowa.

# 6) Propozycje dashboardów własnych

„Aplikacja — zdrowie i błędy (NDJSON)” — panele:

* Trend error (Count) per app.name (linie, 24h).
* Top log.logger (bar) + Top frazy w message (terms na message.keyword, ostrożnie z kardynalnością).
* Heatmap host.name × log.level.
* Tabela „ostatnie błędy”: @timestamp, host.name, log.level, log.logger, message.

„IIS — jakość ruchu” — panele:

* p95 iis.access.time\_taken per host.name.
* Top url.path dla 5xx.
* Rozkład http.response.status\_code (pie/bar).
* Top client.ip i top user.agent.name.
* Trend „4xx vs 5xx vs 2xx” (stacked area) z filtrami po cs-host.

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

log.level : ("error" or "critical") # błędy aplikacji (NDJSON)

app.name : "myapp" and log.level : ("error" or "critical") # błędy konkretnej usługi

event.dataset : "iis.access" and http.response.status\_code >= 500 and @timestamp >= now-15m # IIS 5xx w 15 min

event.dataset : "iis.access" and iis.access.time\_taken >= 3000 # IIS — wolne odpowiedzi (>3 s)

not url.path : "/health" and not url.path : "/ready" # wyklucz szum (healthchecki)

log.logger : "MyCompany.Payments" and log.level : "error" # top błędy specyficznego loggera

app.name : "myapp" and @timestamp >= now-10m # cisza w logach (do alertu typu „Less than 1 match”)

log.level : "debug" and @timestamp >= now-1h # duże wpisy (debug)

event.dataset : "iis.access" and url.referrer.domain : ("\*", "unknown") # podejrzane referrery (IIS)

# 8) Alerty — gotowy pakiet reguł

A. Skok błędów aplikacji

* KQL: log.level:(error or critical) [+ app.name:"myapp"]; Okno 5 min; Warunek: More than 0 matches lub threshold; Group by app.name, host.name.

B. HTTP 5xx

* KQL: event.dataset:"iis.access" and http.response.status\_code >= 500; Group by url.path/cs-host i host.name; Warunek: ratio 5xx > X%.

C. „Cisza w logach”

* KQL: app.name:"myapp"; Typ: Less than 1 match w 10–15 min; Group by host.name.

D. Wolne odpowiedzi

* KQL: event.dataset:"iis.access" and iis.access.time\_taken >= 3000; Group by url.path; Warunek: More than N matches w 5 min.

# 9) Korelacje i łączenie z innymi danymi

* APM/RUM → logi: jeśli aplikacja dopisuje trace.id/transaction.id do logów (np. Serilog), filtruj trace.id:"<id>" i przechodź między śladem APM a logami.
* Host metrics (Metricbeat): koreluj piki log.level:error z CPU/dyskiem hosta (host.name).
* Winlogbeat: powiąż błędy aplikacji z eventami systemowymi (restarty usług, 7045 itp.).

# 10) Raportowanie i współdzielenie

* Saved Search: „Errors — myapp (24h)”.
* Dashboard PDF: harmonogram (Reporting) dla tygodniowego przeglądu jakości.
* Linki z filtrami: skrócone URL-e do gotowych widoków (np. 5xx dla cs-host=api.example.com).

# 11) Dobre praktyki pracy z logami w Kibanie

* Nazewnictwo i tagi: konsekwentne app.name, fields.env (prod/stage/dev), fields.role (web/api/worker).
* Kardynalność: unikaj wykresów po polach o ogromnej liczbie wartości (np. pełne message/stacktrace).
* Czas: dobieraj okno (15m/1h/24h) zależnie od wolumenu; agregacje Auto/1m/5m.
* Panel „Ostatnie błędy”: sortuj @timestamp desc; pokazuj minimalny zestaw kolumn.

# 12) Proponowany „starter pack” (do importu / odtworzenia)

Saved Searches:

Errors — All apps (24h): log.level : ("error" or "critical")

IIS 5xx — Last 1h: event.dataset : "iis.access" and http.response.status\_code >= 500 and @timestamp >= now-1h

Dashboardy:

* App Logs — Health & Errors (4–6 paneli z sekcji 6).
* IIS — Quality (p95 time-taken, 5xx by path, top clients).

Alerty:

* APP\_ERRORS\_SPIKE (sekcja 8A).
* HTTP\_5XX\_SPIKE (sekcja 8B).
* APP\_LOGS\_SILENCE (sekcja 8C).

# 13) Aneks — pola IIS przydatne w analizie

* http.response.status\_code, http.request.method, url.path, url.domain.
* iis.access.time\_taken (ms), client.ip/source.ip, user.agent.name, referrer.

Uwaga: nazwy mogą się różnić zależnie od wersji i parsowania; w Discover przefiltruj „Available fields” po „iis.” i „http.”

# Co dalej?

Mogę przygotować zestaw Saved Objects (dashboardy, saved searches i alerty) dopasowany do Twoich pól (app.name, labels.\*, IIS) do importu w wybranym Space.