Heartbeat – słownik pól z logów (ECS + protokoły)

*Co oznacza każde pole i jak je analizować w Kibanie*

*Wersja: 1.0 | Data: 2025-08-22*

# 1) Jak czytać ten dokument

• Dokument zawiera najczęściej używane pola generowane przez Heartbeat (ECS + specyficzne dla monitorów).

• Dla każdego pola podajemy: nazwa, typ, przykład, opis oraz „Analiza/Wnioski” – czyli co realnie da się z tego pola wyciągnąć w Kibanie (KQL, wizualizacje, alerty).

• Zakres obejmuje monitory: HTTP/HTTPS, TCP, ICMP, a także metadane TLS i pola podsumowań (summary/state).

# 2) Pola wspólne (ECS + Heartbeat)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pole | Typ | Przykład | Opis | Analiza/Wnioski |
| @timestamp | date | 2025-08-22T12:34:56.789Z | Czas zdarzenia. | Oś czasu, histogramy, okna korelacyjne. |
| event.dataset | keyword | heartbeat | Zestaw danych/źródło. | Filtr na konkretny typ danych (np. heartbeat). |
| event.module | keyword | heartbeat | Moduł/event source (czasem nieużywane). | Segmentacja raportów po module. |
| event.kind | keyword | event | Rodzaj zdarzenia wg ECS. | Filtr diagnostyczny w przypadkach nietypowych wpisów. |
| event.category | keyword | network | Kategoria zdarzenia wg ECS. | Budowa wysokopoziomowych paneli. |
| event.type | keyword | connection | Typ zdarzenia wg ECS. | Wskaźnik rodzaju akcji; użyteczne w filtrach. |
| event.outcome | keyword | success / failure | Wynik checka monitora. | Podstawa do SLO/uptime, alertów i SLA. |
| agent.id | keyword | b1b9a6f2-… | ID instancji Heartbeat. | Rozróżnianie wielu agentów/instalacji. |
| agent.name | keyword | hb-euw-prod-01 | Nazwa hosta agenta. | Grupowanie po lokalizacji/klastrze. |
| agent.type | keyword | heartbeat | Rodzaj agenta. | Filtr globalny: agent.type:heartbeat. |
| agent.version | keyword | 8.19.0 | Wersja Heartbeat. | Kontrola zgodności, regresje po aktualizacji. |
| ecs.version | keyword | 8.11.0 | Wersja ECS. | Zgodność pól i dashboardów. |
| host.name | keyword | monitor-vm-01 | Host wykonujący test. | Grupowanie metryk po hoście. |
| observer.geo.name | keyword | Warszawa-dc | Nazwa lokalizacji obserwatora. | Mapy i porównanie lokacji. |
| observer.geo.location | geo\_point | {"lon": 21.01, "lat": 52.23} | Geo punkt obserwatora. | Wizualizacje na mapie, latency vs. geografia. |
| tags | keyword[] | prod, critical | Tagi użytkownika. | Segmentacja SLO/raportów wg krytyczności. |
| labels.\* | object | labels.team: core | Dowolne metadane użytkownika. | Przekrój raportów po ownerze/usłudze. |
| url.scheme | keyword | https | Schemat URL. | Rozdzielenie HTTP/HTTPS; polityki. |
| url.domain | keyword | api.example.com | Domena celu. | Top N hostów; korelacja z DNS/ingresem. |
| url.port | long | 443 | Port TCP celu. | Heatmapy portów/serwisów. |
| url.path | keyword | /health | Ścieżka URL. | Analiza endpointów; regresje po deployu. |
| url.full | keyword | https://api.example.com/health | Pełny URL. | Dedup/GroupBy; outliers content hash. |
| error.message | text | read: connection reset by peer | Opis błędu. | Root cause w panelach awarii. |
| error.type | keyword | io | Klasyfikacja błędu. | Alerting per klasa problemu. |
| monitor.id | keyword | api-health-euw | ID monitora. | Klucz do grupowań i raportów. |
| monitor.name | keyword | API Health EUW | Czytelna nazwa monitora. | Raporty dla biznesu/Slack. |
| monitor.type | keyword | http|tcp|icmp|browser | Typ monitora. | Filtr oraz routing do odpowiednich paneli. |
| monitor.status | keyword | up / down | Status sprawdzenia. | Metryka dostępności, SLA, alerty. |
| monitor.ip | ip | 93.184.216.34 | Adres IP celu po resolve. | Diagnostyka DNS/anycast; zmiany IP. |
| monitor.host | keyword | api.example.com | Host celu. | Gdy DNS zwraca wiele IP – porównania. |
| monitor.duration.us | long (µs) | 128734 | Czas całkowity checka. | SLO latency (p95/p99), regresje wydajności. |
| summary.up | long | 1 | Liczba prób udanych w oknie. | Szybka agregacja uptime. |
| summary.down | long | 0 | Liczba prób nieudanych w oknie. | Szybka agregacja awarii. |
| state.id | keyword | b19a… | Identyfikator stanu monitora. | Korelacja zdarzeń w obrębie jednego check\_group. |
| monitor.check\_group | keyword | aa3b3f… | ID grupy checka. | Łączenie kroków/warstw w jeden przebieg. |

# 3) Pola specyficzne: HTTP/HTTPS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pole | Typ | Przykład | Opis | Analiza/Wnioski |
| http.url | alias→url.full | https://api.example.com/health | Pełny URL zapytania. | Jednoznaczna identyfikacja endpointu. |
| http.response.headers.\* | object | content-type: application/json | Nagłówki odpowiedzi HTTP. | Weryfikacja cache/CORS/CT/STS. |
| http.response.redirects | keyword[] | http→https→… | Lista przekierowań. | Wykrywanie pętli, legacy http→https. |
| http.response.body.hash | keyword | sha1: 3c4a… | Hash treści body. | Detekcja zmian treści mimo 200 OK. |
| http.rtt.write\_request.us | long (µs) | 12000 | Czas wysłania żądania. | Diagnostyka klienta/kolejki socketów. |
| http.rtt.response\_header.us | long (µs) | 18000 | Czas do 1. bajtu odpowiedzi. | TTFB – serwer/edge. |
| http.rtt.content.us | long (µs) | 35000 | Czas pobrania treści. | Problemy sieci/CDN, rozmiar body. |
| http.rtt.validate.us | long (µs) | 6000 | Walidacja odpowiedzi. | Koszt walidatorów/patternów. |
| http.rtt.validate\_body.us | long (µs) | 9000 | Walidacja z odczytem body. | Gdy treść duża – rośnie dynamicznie. |
| http.rtt.total.us | long (µs) | 78000 | Czas całego requestu. | KPI latency do SLO, trendowanie. |

# 4) Pola specyficzne: TCP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pole | Typ | Przykład | Opis | Analiza/Wnioski |
| tcp.port | alias→url.port | 443 | Port usługi. | Mapy portów, compliance/zmiany. |
| tcp.rtt.connect.us | long (µs) | 15000 | Czas zestawienia połączenia TCP. | Problemy z łącznością/handshake/latencją sieci. |

# 5) Pola specyficzne: ICMP (ping)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pole | Typ | Przykład | Opis | Analiza/Wnioski |
| icmp.requests | integer | 1 | Liczba wysłanych EchoRequest. | Wykrywanie strat pakietów przy >1 probe. |
| icmp.rtt.us | long (µs) | 12000 | RTT Echo Request/Reply. | Podstawowa latencja L3; korelacja z TCP/HTTP. |

# 6) Pola TLS/certyfikaty

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pole | Typ | Przykład | Opis | Analiza/Wnioski |
| tls.rtt.handshake.us | long (µs) | 22000 | Czas handshake TLS. | Wykrywanie problemów crypto/ALPN/po stronie serwera. |
| tls.server.x509.not\_before | date | 2025-05-01T00:00:00Z | Data od kiedy certyfikat jest ważny. | Audyt poprawności łańcucha; regresje po odnowieniu. |
| tls.server.x509.not\_after | date | 2026-05-01T00:00:00Z | Data wygaśnięcia certyfikatu. | Alerty „cert expiring in < N dni”. |
| tls.server.version\_number | keyword | 3 | Wersja formatu x509. | Rzadko analizowane; diagnostyka nietypowych certów. |

# 7) Pola podsumowań i stanu (summary/state)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pole | Typ | Przykład | Opis | Analiza/Wnioski |
| summary.up | long | 1 | Liczba udanych checków w oknie. | Szybkie KPI uptime. |
| summary.down | long | 0 | Liczba nieudanych checków w oknie. | Wykrywanie awarii/przestojów. |
| state.id | keyword | b19a… | Identyfikator stanu monitora. | Sklejanie kroków w jedną sesję. |
| monitor.check\_group | keyword | aa3b3f… | Id grupy przebiegu. | Join do synthetics/stepów. |

# 8) Przykładowe kwerendy KQL (Kibana)

• Dostępność monitorów  
 KQL: monitor.status : "down"

• Awarii w ostatnich 15 min  
 KQL: monitor.status : "down" and @timestamp >= now()-15m

• Monitory HTTP z wysokim TTFB  
 KQL: monitor.type : "http" and http.rtt.response\_header.us > 500000

• Łączne opóźnienie > 1s  
 KQL: http.rtt.total.us > 1000000

• Problemy z handshake TLS  
 KQL: exists: tls.rtt.handshake.us and tls.rtt.handshake.us > 500000

• Wygasające certyfikaty w 30 dni  
 KQL: tls.server.x509.not\_after <= now()+30d

• Top domeny z błędami  
 KQL: event.outcome : "failure" | stats count() by url.domain

• Filtr po lokalizacji obserwatora  
 KQL: observer.geo.name : "Warszawa-\*"

• Tylko ICMP (ping)  
 KQL: monitor.type : "icmp"

• HTTP 200 ale zmienia się treść  
 KQL: http.response.body.hash : \* and event.outcome : "success" | stats cardinality(http.response.body.hash) by url.full

# 9) Notatki praktyczne

• Pola i ich dostępność mogą zależeć od wersji agenta i konfiguracji (np. włączone walidatory HTTP, TLS).

• W środowiskach z Elastic Agent/Fleet nazwy data streamów mogą różnić się od klasycznych indeksów heartbeat-\* (stosuj filtry po agent.type/monitor.type).

• Dla monitorów browser (Synthetics) dostępne są też metryki UX (FCP/LCP/CLS) i pola synthetics.step.\*, synthetics.type – nie uwzględnione szczegółowo w tej wersji słownika.