Heartbeat — dokumentacja wdrożeniowa (PL)

Wersja: Windows (ZIP/offline), integracja z Kibana Uptime & Dashboardy  
*Data: 2025-08-21 06:39*

# 1) Cel i korzyści

* Proste monitorowanie dostępności i czasu odpowiedzi (HTTP/TCP/ICMP).
* Widok w Kibanie: Uptime (status, history, geo) oraz opcjonalne dashboardy.
* Alerty na niedostępność, degradację SLA/SLO, ważność certyfikatów TLS.

# 2) Wymagania i przygotowanie

* Windows x64; konto usługi z prawem uruchamiania Heartbeat.
* Dla ICMP (ping) wymagane uprawnienia administratora i reguła w firewallu.
* Kibana/Elasticsearch: adresy i konto z uprawnieniami do Saved Objects i indeksów.
* TLS: jeśli własne CA — wskaż plik CA w konfiguracji; nie wyłączaj weryfikacji w produkcji.

# 3) Instalacja (ZIP/offline) — Windows

1) Rozpakuj paczkę ZIP do katalogu, np. C:\Elastic\heartbeat.

2) Zainstaluj usługę i uruchom:

cd C:\Elastic\heartbeat  
.\install-service-heartbeat.ps1  
Start-Service heartbeat

# 4) Konfiguracja minimalna (przykłady)

## 4A) HTTP monitor (GET, weryfikacja statusu i treści)

heartbeat.monitors:  
 - type: http  
 id: my-api-health  
 name: My API Health  
 enabled: true  
 schedule: '@every 1m'  
 urls: ["https://api.twojadomena.pl/health"]  
 check.request:  
 method: GET  
 headers:  
 Accept: application/json  
 check.response:  
 status: 200  
 body: '"status":"OK"' # proste sprawdzenie fragmentu treści  
 ssl:  
 verification\_mode: full # full | strict | none (none TYLKO testowo)  
 certificate\_authorities: ["C:/Elastic/certs/ca.crt"]  
  
output.elasticsearch:  
 hosts: ["https://<ES\_HOST>:9200"]  
 username: "<USER>"  
 password: "<PASS>"  
  
setup.kibana:  
 host: "https://<KIBANA\_HOST>:5601"  
 username: "<USER>"  
 password: "<PASS>"

## 4B) TCP monitor (port otwarty i czas odpowiedzi)

heartbeat.monitors:  
 - type: tcp  
 id: tcp-db  
 name: DB Listener  
 enabled: true  
 schedule: '@every 30s'  
 hosts: ["db01.twojadomena.pl:1433"]  
  
output.elasticsearch:  
 hosts: ["https://<ES\_HOST>:9200"]  
 username: "<USER>"  
 password: "<PASS>"

## 4C) ICMP monitor (ping) — wymaga praw admina

heartbeat.monitors:  
 - type: icmp  
 id: ping-frontend  
 name: Ping Frontend  
 enabled: true  
 schedule: '@every 30s'  
 hosts: ["app01.twojadomena.pl"]

## 4D) Wspólne tagi i pola (identyfikacja środowiska)

fields:  
 env: prod  
 role: uptime  
fields\_under\_root: true  
tags: ["heartbeat","synthetic","prod"]

# 5) Kibana — Uptime (preferowane) i dashboardy

* Uptime: Observability → Uptime → przegląd statusów monitorów, historia, mapy, TLS certs.
* Dashboardy: można zaimportować pakietowe dashboardy Heartbeat.

Auto-import dashboardów (opcjonalnie):

C:\Elastic\heartbeat\heartbeat.exe setup --dashboards ^  
 -E setup.kibana.host=https://<KIBANA\_HOST>:5601 ^  
 -E setup.kibana.username=<USER> ^  
 -E setup.kibana.password=<PASS> ^  
 -E output.elasticsearch.hosts=["https://<ES\_HOST>:9200"] ^  
 -E output.elasticsearch.username=<USER> ^  
 -E output.elasticsearch.password=<PASS>

Ręczny import: Kibana → Stack Management → Saved Objects → Import → pliki z katalogu 'kibana/' w paczce Heartbeat.

Data view: `heartbeat-\*` (ew. `synthetics-\*` w nowszych wdrożeniach synthetics).

# 6) Kibana — gdzie patrzeć i jak wyciągać wnioski

* Uptime → Monitors: status UP/DOWN, czas odpowiedzi, historia i geolokalizacja (jeśli używasz multiple-locations).
* Uptime → TLS: przegląd ważności certyfikatów; zdefiniuj alert wygasania.
* Discover/Lens: analiza `monitor.status`, `monitor.name`, `url.full`, `tls.\*`.

# 7) KQL — gotowe filtry

monitor.status : "down" and @timestamp >= now-15m

url.full : "https://api.twojadomena.pl/health" and monitor.status : "down"

tls.server.x509.not\_after <= now+30d # certyfikat wygasa w ciągu 30 dni

# 8) Alerty — wzorce reguł

* Monitor DOWN: `monitor.status: "down"` → warunek „More than 0 matches” w 1–5 min; Group by `monitor.id`.
* Degradacja czasu odpowiedzi: threshold dla średniego response time z Uptime/Lens (np. > 1s przez 5 min).
* Wygasający certyfikat TLS: KQL `tls.server.x509.not\_after <= now+30d`; powiadom wcześniej (30/14/7 dni).

# 9) Dobre praktyki

* Unikalne `id` dla każdego monitora; spójne `name` i tagi (`env`, `role`).
* Dedykowany host dla Heartbeat (jeśli wiele monitorów) lub wykorzystanie wielu lokalizacji dla porównań.
* Nie wyłączaj weryfikacji TLS poza testami; używaj własnego CA w ssl.certificate\_authorities.
* Harmonogramy: rozważ 30–60s dla krytycznych endpointów; dłużej dla mniej istotnych.
* Agreguj monitory per środowisko i aplikację, aby upraszczać alerty i dashboardy.

# 10) Troubleshooting

* Brak danych: `Get-Service heartbeat`, logi w `logs\`, sprawdź `enabled: true` i poprawne `schedule`.
* ICMP nie działa: uruchom usługę z uprawnieniami admina; dodaj regułę firewall; na Windows konieczne prawa do gniazd RAW.
* TLS błędy: dodaj CA w `ssl.certificate\_authorities`; unikaj `verification\_mode: none` w produkcji.
* Puste Uptime/Dashboards: brak data view (`heartbeat-\*`) lub zły zakres czasu w Timepickerze.
* Za dużo danych: zwiększ interwały (`@every 1m/5m`), ogranicz liczbę URLi lub lokacji.

# 11) Szybki checklist (TL;DR)

* Rozpakuj ZIP → `install-service-heartbeat.ps1` → `Start-Service heartbeat`.
* Skonfiguruj monitory (HTTP/TCP/ICMP) + `output.elasticsearch` + `setup.kibana`.
* (Opcjonalnie) Importuj dashboardy: `heartbeat.exe setup --dashboards`.
* Uptime → sprawdź status monitorów; dodaj alerty (DOWN, p95, TLS expiry).