# Graylog: częste problemy

**Poziom: operacyjny**

**Dotyczy: Graylog 7.1.9, MongoDB 8.0, OpenSearch 2.19.6 w Dockerze na docker01 (docelowo log01)**

**Czas: ~30 min**

*Zestawienie objawów, na które zespół IT trafia najczęściej: brak logów z hosta, czas w złej strefie, dysk OpenSearch pełny, rosnący dziennik, za mały heap, błąd po aktualizacji i zgubione pola. Tabela daje szybką odpowiedź, a trzy najczęstsze problemy mają krótkie procedury krok po kroku.*

### Skróty użyte na tej stronie (7)

| Skrót | Co znaczy |
|---|---|
| **CLI** | Command Line Interface: praca przez wpisywanie komend w terminalu |
| **DNS** | Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP |
| **GB** | Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej |
| **IT** | Information Technology: dział albo praca związana z komputerami i infrastrukturą firmy |
| **RFC** | Request for Comments: oficjalny dokument opisujący standard internetowy, na przykład sposób działania danego protokołu; w ITIL także Request for Change, czyli wniosek o zmianę |
| **TCP** | Transmission Control Protocol: protokół sieciowy z potwierdzaniem dostarczenia |
| **UTC** | Coordinated Universal Time: uniwersalny czas bez strefy, punkt odniesienia dla wszystkich stref czasowych na świecie |

# Fiszka

Komendy wykonujemy z CLI bash jako user z uprawnieniami sudo na hoście z Graylogiem. Polecenia do OpenSearch wołamy z wnętrza kontenera `graylog-opensearch`, bo port `9200` nie jest wystawiony na hosta.

**TYLKO ODCZYT**

## Sprawdzenie stanu kontenerów: docker compose ps

```
docker compose -f /docker/compose/graylog/compose.yaml ps
```

Przykład:

```
NAME                IMAGE                                  STATUS
graylog             graylog/graylog:7.1.9                  Up 3 days
graylog-mongodb     mongo:8.0                              Up 3 days
graylog-opensearch  opensearchproject/opensearch:2.19.6    Restarting (137) 12 seconds
ago
```

Status `Restarting (137)` oznacza, że proces został zabity, najczęściej przez brak pamięci.

**docker** = narzędzie do kontenerów

**TYLKO ODCZYT**

## Sprawdzenie logu Grayloga: docker compose logs

```
docker compose -f /docker/compose/graylog/compose.yaml logs --tail 100 graylog
```

Pierwsze miejsce diagnozy większości objawów z tabeli niżej. To samo polecenie z nazwą `opensearch` zamiast `graylog` pokazuje log silnika.

**docker** = narzędzie do kontenerów

**TYLKO ODCZYT**

## Sprawdzenie stanu klastra OpenSearch: _cluster/health

```
docker exec graylog-opensearch curl -s "localhost:9200/_cluster/health?pretty"
```

Przykład:

```
{
  "cluster_name" : "docker-cluster",
  "status" : "yellow",
  "number_of_nodes" : 1,
  "unassigned_shards" : 2
}
```

Przy jednym węźle bez replik zdrowy stan to `green`. `yellow` albo `red` oznacza shardy bez przydziału, najczęściej z powodu dysku albo błędu po aktualizacji.

**docker** = narzędzie do kontenerów

**TYLKO ODCZYT**

## Sprawdzenie zajętości dysku w OpenSearch: _cat/allocation

```
docker exec graylog-opensearch curl -s "localhost:9200/_cat/allocation?v"
```

Przykład:

```
shards disk.indices disk.used disk.avail disk.total disk.percent host
    42      262.1gb   284.9gb     15.1gb      300gb           94 172.18.0.3
```

Progi: 85 procent (koniec przydziału nowych shardów), 90 (przenoszenie shardów), 95 (indeksy tylko do odczytu).

**docker** = narzędzie do kontenerów

**TYLKO ODCZYT**

## Sprawdzenie, które indeksy mają blokadę: filter_path

```
docker exec graylog-opensearch curl -s
"localhost:9200/_all/_settings?filter_path=*.settings.index.blocks&pretty"
```

Przykład:

```
{
  "fw_12" : {
    "settings" : {
      "index" : {
        "blocks" : {
          "read_only_allow_delete" : "true"
        }
      }
    }
  }
```

```
}
```

Pusty wynik `{ }` oznacza brak blokad.

**docker** = narzędzie do kontenerów

**TYLKO ODCZYT**

## Zdjęcie blokady tylko do odczytu: read_only_allow_delete

```
docker exec graylog-opensearch curl -s -X PUT "localhost:9200/_all/_settings" \
  -H 'Content-Type: application/json' \
  -d '{"index.blocks.read_only_allow_delete": null}'
```

Wykonujemy dopiero po zwolnieniu miejsca poniżej 90 procent, inaczej blokada wróci przy pierwszym zapisie. Wartość `null` przywraca ustawienie domyślne, nie ustawia `false`.

**docker** = narzędzie do kontenerów

**TYLKO ODCZYT**

## Sprawdzenie rozmiaru dziennika Grayloga: du

```
sudo du -sh /docker/apps/graylog/graylog/journal
```

Przykład:

```
4.8G	/docker/apps/graylog/graylog/journal
```

Domyślne limity dziennika to 12 godzin i 5 GB (`message_journal_max_age` i `message_journal_max_size`). Wartość blisko 5 GB to dziennik pełny. Stan na żywo widać też w System → Nodes.

**GB** = Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej · **du** = disk usage: ile miejsca zajmuje katalog

**TYLKO ODCZYT**

## Sprawdzenie, czy kontener zabił brak pamięci: docker inspect

```
docker inspect -f 'OOM={{.State.OOMKilled}} restarty={{.RestartCount}}' graylog-opensearch
```

Przykład:

```
OOM=true restarty=7
```

**docker** = narzędzie do kontenerów · **OOM** = Out Of Memory: brak pamięci; OOM killer to mechanizm jądra zabijający proces, gdy jej zabraknie

**TYLKO ODCZYT**

## Sprawdzenie wersji OpenSearch: filter_path

```
docker exec graylog-opensearch curl -s "localhost:9200/?filter_path=version.number&pretty"
```

Przykład:

```
{
  "version" : {
    "number" : "2.19.6"
  }
}
```

**docker** = narzędzie do kontenerów

**TYLKO ODCZYT**

## Sprawdzenie czasu hosta: timedatectl i chronyc

```
timedatectl
chronyc tracking
```

Szukamy trzech rzeczy: strefa `Europe/Warsaw`, wiersz `System clock synchronized: yes` i niewielkie `System time` w wyniku `chronyc`. Synchronizację czasu opisujemy w Czas w Linuksie: chrony.

**timedatectl** = time date control: pokaż albo ustaw czas, datę i strefę czasową systemu oraz stan synchronizacji zegara · **chronyc** = chrony client: stan synchronizacji zegara

□□ TYLKO ODCZYT

## Sprawdzenie połączenia hosta z odbiorcą: ss

```
ss -tn | grep 1514
```

Przykład:

```
ESTAB 0 0 10.20.10.41:51204 10.20.10.42:1514
```

Brak wiersza przy TCP oznacza, że host w ogóle nie ma otwartego połączenia z wejściem Grayloga.

□□ **TCP** = Transmission Control Protocol: protokół sieciowy z potwierdzaniem dostarczenia ·**ss** = socket statistics: kto nasłuchuje i jakie są połączenia

□□ PLIK KONFIGURACYJNY

## Ustawienie strefy czasowej konta administratora: GRAYLOG_ROOT_TIMEZONE

Plik `/docker/compose/graylog/compose.yaml`, w sekcji `environment` usługi `graylog`:

```
    environment:
      - GRAYLOG_ROOT_TIMEZONE=Europe/Warsaw
```

Dopisujemy do istniejącej listy. Dotyczy tylko konta `admin`, bo konta domenowe mają własną strefę w System → Users → Edytuj użytkownika. Graylog zapisuje czasy zawsze w UTC, strefa zmienia tylko ich wyświetlanie.

□□ **UTC** = Coordinated Universal Time: uniwersalny czas bez strefy, punkt odniesienia dla wszystkich stref czasowych na świecie

□□ PLIK KONFIGURACYJNY

## Podniesienie heap OpenSearch i Grayloga: Xms i Xmx

Plik `/docker/compose/graylog/compose.yaml`:

```
  opensearch:
    environment:
```

```
      - OPENSEARCH_JAVA_OPTS=-Xms4g -Xmx4g
  graylog:
    environment:
      - GRAYLOG_SERVER_JAVA_OPTS=-Xms2g -Xmx2g -server -XX:+UseG1GC -XX:-
OmitStackTraceInFastThrow
```

Dopisujemy do istniejących list `environment`. Wartości `-Xms` i `-Xmx` ustawiamy równe. Zasada producenta: heap najwyżej połowa pamięci hosta i nie więcej niż 31 GB, a kontenery na jednym hoście muszą się razem zmieścić w tej połowie. Po zmianie `docker compose up -d`.

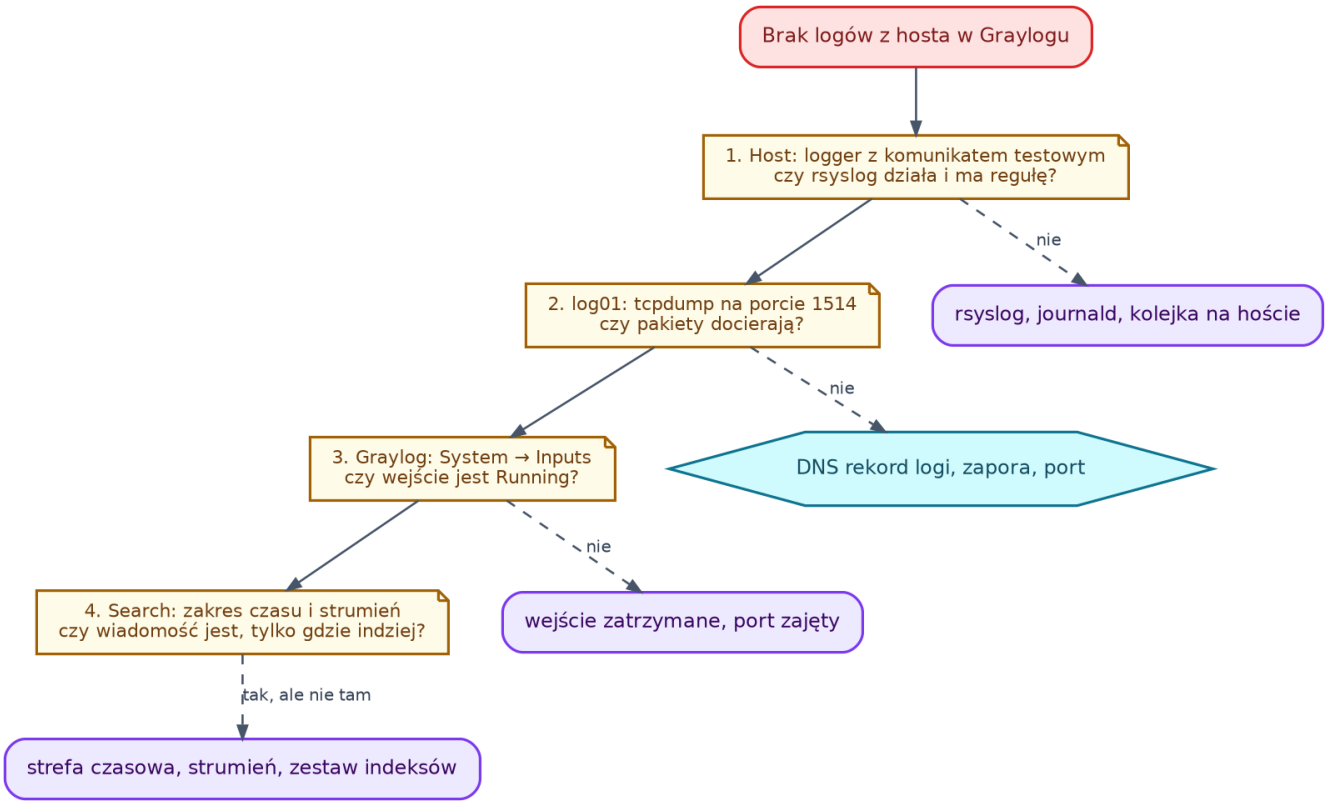
□□ **GB** = Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej

# Z życia wzięte

**□□ Z ŻYCIA WZIĘTE**

Piątek, 16:10. Panel „Zapora fw01” pokazuje zero wiadomości od 14:05, a zespół czeka na wynik kontroli ruchu przed weekendem. Tomek zaczyna od `docker compose ps`: wszystko `Up`. Dopiero `_cat/allocation` pokazuje 96 procent zajętości dysku, a lista blokad wskazuje `read_only_allow_delete` na wszystkich indeksach `fw`. Przez te dwie godziny dziennik Grayloga buforował przychodzące wiadomości, więc po zwolnieniu miejsca i zdjęciu blokady logi z okna wracają do indeksu. Gdyby Tomek zaczął od restartu kontenerów, straciłby dziesięć minut i nie dowiedziałby się niczego.

# Częste błędy




*Kolejność diagnozy braku logów: komunikat testowy na hoście, pakiety na log01, stan wejścia w Graylogu, zakres czasu i strumień w wyszukiwaniu (kliknięcie otwiera pełny rozmiar)*

| Objaw | Przyczyna | Co robimy |
|---|---|---|
| Brak logów z jednego hosta, reszta płynie | brak reguły wysyłki w `rsyslog.conf`, usługa zatrzymana albo zła nazwa odbiorcy | procedura 1 niżej, składnia reguł na stronie Wysyłka logów z hostów |
| Brak logów ze wszystkich hostów naraz | rekord `logi` wskazuje zły adres, wejście zatrzymane albo zapora blokuje port | `dig +short`, System → Inputs, `ncat -zv`, po przeprowadzce patrz Graylog: przeprowadzka z docker01 na log01 |
| Logi mają czas przesunięty o godzinę lub dwie albo „z przyszłości” | nadawca wysyła czas lokalny bez strefy (starszy format syslog RFC 3164), a Graylog przyjmuje go jako UTC albo zegar hosta jest zły | procedura 3 niżej |
| W oknie Search „nie ma” świeżych logów, choć host pisze | wiadomości mają czas w przyszłości i zakres „ostatnie 5 minut” ich nie obejmuje | zakres absolutny sięgający jutra, potem naprawa strefy u źródła |
| Zapis do indeksu odrzucany, w logu `FORBIDDEN/12/index read-only / allow delete (api)` | dysk OpenSearch przekroczył 95 procent (flood stage) i indeksy dostały blokadę | procedura 2 niżej |

| Objaw | Przyczyna | Co robimy |
|---|---|---|
| Dziennik Grayloga w System → Nodes rośnie, rośnie liczba niezatwierdzonych wiadomości | OpenSearch nie przyjmuje zapisu (blokada, mała pamięć, dysk) albo przetwarzanie reguł jest wolniejsze od napływu | najpierw przyczyna po stronie OpenSearch, dziennika nie kasujemy, bo to są wiadomości, których jeszcze nie zapisano |
| Kontener `graylog-opensearch` w pętli restartów, `Restarting (137)` albo `OutOfMemoryError` w logu | heap za mały dla ilości danych albo kontenery razem większe niż pamięć hosta | `docker inspect` na OOM, podnosimy `-Xms` i `-Xmx` według Fiszki, sprawdzamy `free -h` na hoście |
| Wyszukiwanie wolne, interfejs „mieli” przy dużym zakresie czasu | heap Grayloga za mały, zbyt szeroki zakres, zapytanie bez wąskiego pola | zakres czasu, strumień w zapytaniu, heap Grayloga |
| Po aktualizacji OpenSearch Graylog nie widzi silnika, w logu błąd wersji | tag obrazu spoza macierzy zgodności (OpenSearch 3.x, albo wyższa niż wspierana w 2.19.x) | przypinamy tag z macierzy zgodności dla wersji Grayloga, patrz Kopia zapasowa i aktualizacja Grayloga |
| Po aktualizacji OpenSearch kontener nie startuje, a powrót do starego tagu nie pomaga | dane zostały zapisane nowszą wersją, a OpenSearch nie czyta danych nowszych niż własna wersja | odtwarzamy dane z kopii zrobionej przed aktualizacją, nie próbujemy „cofać” tagu na tych samych danych |
| Po aktualizacji OpenSearch klaster `red` albo `yellow` | shardy jeszcze się odtwarzają, za mało pamięci albo `vm.max_map_count` wrócił do domyślnego | `_cluster/health`, odczekanie na odtworzenie, `sysctl vm.max_map_count` |
| Pola, po których szukaliśmy, są puste albo zniknęły | zmiana formatu logu u źródła, potok niepodpięty do strumienia, zła kolejność procesorów | symulator w System → Pipelines na świeżej wiadomości, patrz Graylog: codzienna praca z logami |
| W System → Overview rośnie licznik błędów indeksowania (Indexer failures) | konflikt typu pola w indeksie, na przykład `level` raz liczba, raz tekst | dajemy polom z potoku przedrostek, żeby nie nadpisywały pól Grayloga, patrz Graylog: codzienna praca z logami |

# Procedury dla trzech najczęstszych problemów

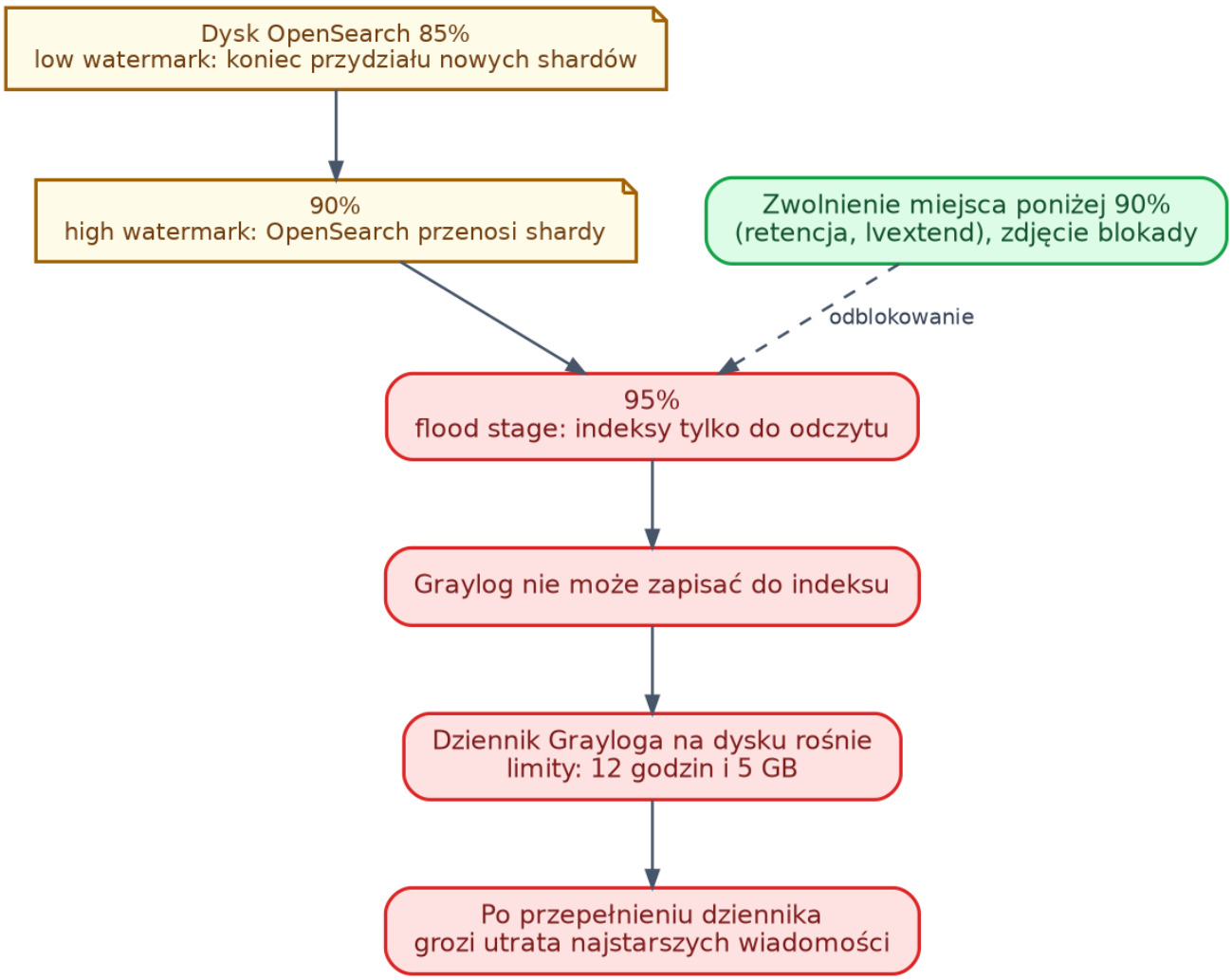
## Procedura 1: brak logów z jednego hosta

1. Na hoście wysyłamy komunikat testowy: `logger -p local0.info "test $(hostname)"` i sprawdzamy `systemctl status rsyslog` oraz składnię `sudo rsyslogd -N1 -f /etc/rsyslog.conf`.
2. Na hoście sprawdzamy, czy nazwa jest dobra i port osiągalny: `dig +short logi.zielona.example`, `ncat -zv logi.zielona.example 1514` i `ss -tn | grep 1514`.

3. Na serwerze z Graylogiem uruchamiamy `sudo tcpdump -n -i any port 1514` i powtarzamy `logger` z hosta. Pakiety widać albo ich nie widać, co od razu dzieli problem na stronę sieci i stronę Grayloga.
4. Jeśli pakietów nie ma, a host ma kolejkę dyskową, sprawdzamy `ls -lh /var/lib/rsyslog/`: rosnący plik `fwdRule1` oznacza, że host buforuje i dosyła, gdy odbiorca wróci. Szukamy przyczyny po drodze: DNS, `fw01`, port.
5. Jeśli pakiety są, w Graylogu sprawdzamy System → Inputs (wejście w stanie Running), a w Search zakres 15 minut i `source:NazwaHosta`. Gdy wiadomości jest, ale nie tam, gdzie patrzymy, sprawdzamy strumienie i strefę czasową z procedury 3.

# Procedura 2: indeksy tylko do odczytu po zapełnieniu dysku

1. Potwierdzamy przyczynę: `_cat/allocation` (zajętość powyżej 95 procent) i `filter_path` z blokadami. Jeżeli blokad nie ma, a zapis nadal nie działa, szukamy gdzie indziej (heap, wersja).
2. Zwalniamy miejsce na jeden z dwóch sposobów. Skracamy retencję zestawu, który zajmuje najwięcej (System → Indices → zestaw → Edit, mniejsze Max days in storage), i usuwamy najstarszy indeks z poziomu zestawu indeksów w Graylogu, a nie wprost przez OpenSearch, żeby Graylog zaktualizował własne zapisy o indeksach. Albo powiększamy wolumen, o ile w grupie jest wolne miejsce: `sudo lvextend -L +50G --resizefs /dev/mapper/vgOpenSearch-lvOpenSearch`.
3. Gdy zajętość jest poniżej 90 procent, sprawdzamy blokady ponownie. OpenSearch 2.x zdejmuje blokadę sam po spadku poniżej progu high watermark, a jeżeli blokada została, zdejmujemy ją komendą z Fiszki.
4. Sprawdzamy w System → Nodes, że dziennik Grayloga się opróżnia, a w Search, że nowe wiadomości pojawiają się na bieżąco. Jeżeli zapis nie wraca, w System → Indices → zestaw → Maintenance wykonujemy Rotate active write index.
5. Dopisujemy w Zabbixie alarm na zajętość wolumenu `lvOpenSearch` przy 75 procentach, żeby następnym razem dowiedzieć się o problemie godziny, a nie dni przed blokadą.




*Dysk OpenSearch przechodzi progi 85, 90 i 95 procent, przy 95 indeksy stają się tylko do odczytu, Graylog przestaje zapisywać, dziennik rośnie, a zwolnienie miejsca poniżej 90 procent i zdjęcie blokady przywraca zapis (kliknięcie otwiera pełny rozmiar)*

> Dziennika Grayloga (`/docker/apps/graylog/graylog/journal`) nie kasujemy, żeby „przyspieszyć powrót”: to wiadomości przyjęte, ale jeszcze niezapisane w indeksie. Zachowanie przy przepełnieniu dziennika do potwierdzenia w dokumentacji producenta, zakładamy najgorszy przypadek, czyli utratę najstarszych niezapisanych wiadomości. Dlatego alarmujemy przed progiem, nie po nim.

# Procedura 3: logi w złej strefie czasowej albo „z przyszłości”

1. Na hoście źródłowym sprawdzamy `timedatectl` i `chronyc tracking`: strefa `Europe/Warsaw`, zegar zsynchronizowany. Zły zegar naprawiamy u źródła i na tym kończymy, w Graylogu nie ma czego poprawiać.
2. W Graylogu otwieramy jedną nową wiadomość i porównujemy pole `timestamp` z czasem na hoście. Różnica równa dokładnie jednej albo dwóm godzinom wskazuje strefę, różnica minut wskazuje zegar.
3. Gdy host wysyła czas lokalny w starszym formacie syslog (RFC 3164), nie niosącym strefy, najlepszą naprawą jest zmiana formatu u źródła na RFC 5424 z czasem w formacie RFC 3339 (z przesunięciem strefy), czyli szablon `GraylogFormat` opisany na stronie Wysyłka logów z hostów. W ustawieniach wejścia syslog w Graylogu może też być pole Time Zone dla wiadomości bez strefy, jeśli występuje w naszej wersji, i wtedy wskazujemy `Europe/Warsaw`.
4. Strefę wyświetlania ustawiamy osobno: dla kont domenowych w System → Users, dla konta `admin` zmienną `GRAYLOG_ROOT_TIMEZONE` z Fiszki. To nie zmienia danych w indeksach, tylko ich pokazywanie.
5. Wiadomości z przyszłości znajdujemy zakresem absolutnym sięgającym jutra. Uwaga na zmianę czasu z letniego na zimowy: w nocy z 24 na 25 października 2026 różnica z UTC zmienia się z dwóch godzin na jedną, więc źródła z ręcznie ustawionym przesunięciem zaczną „kłamać” o godzinę.

# □□ Ćwiczenia

1. Panel pokazuje, że logi z `app01` urwały się o 14:00, a `docker compose ps` pokazuje trzy działające kontenery. Wskazujemy trzy pierwsze rzeczy do sprawdzenia, w kolejności.

**Odpowiedź**

Najpierw `_cat/allocation` i blokady indeksów, bo pełny dysk OpenSearch wyłącza zapis przy działających kontenerach. Potem `logger` na `app01` i `tcpdump` na serwerze z Graylogiem, żeby ustalić, czy pakiety w ogóle docierają. Na końcu System → Inputs i zakres czasu w Search, bo wiadomości mogą być, tylko z innym czasem.

1. Po aktualizacji OpenSearch kontener nie startuje. Tomek cofa tag w `compose.yaml` na poprzedni, ale kontener dalej się nie uruchamia. Wyjaśniamy dlaczego i co robimy.

**Odpowiedź**

Nowsza wersja zdążyła zapisać dane w formacie, którego starsza nie czyta, a OpenSearch nie wspiera cofania wersji na takich danych. Zamiast cofać tag, odtwarzamy katalog danych albo migawkę wykonaną przed aktualizacją, z tego powodu kopia przed aktualizacją jest

obowiązkowym krokiem na stronie o kopii i aktualizacji.

1. Logi z nowego serwera pojawiają się w Graylogu z czasem o dwie godziny wcześniejszym niż na zegarze. Zegar serwera jest poprawny. Wskazujemy najbardziej prawdopodobną przyczynę i najlepszą naprawę.

**Odpowiedź**

Serwer wysyła czas lokalny w formacie bez strefy, a Graylog traktuje go jako UTC, stąd różnica dwóch godzin latem. Najlepiej zmienić format wysyłki na RFC 5424 z czasem RFC 3339, który niesie przesunięcie strefy, a dopiero w drugiej kolejności ustawić strefę na wejściu.

**Powiązane:** Graylog: codzienna praca z logami · Graylog: przeprowadzka z docker01 na log01 · Kopia zapasowa i aktualizacja Grayloga · Wysyłka logów z hostów

**Źródła:** go2docs.graylog.org: „Compatibility Matrix", „Additional Configuration" (heap), „Search syntax reference" · Graylog `server.conf` w gałęzi 7.1 (`message_journal_max_age`, `message_journal_max_size`, `root_timezone`) · dokumentacja OpenSearch, progi dysku 85, 90 i 95 procent oraz automatyczne zdejmowanie blokady `read_only_allow_delete` w 2.x · community.graylog.org i wpisy poradnikowe o przepełnionym dysku i blokadzie indeksów (sprawdzone 03.10.2026)

---

Revision #14
Created 2026-10-03 12:08:25 UTC by Tomasz Konieczny
Updated 2026-10-26 14:15:54 UTC by Tomasz Konieczny