Logi binarne: archiwizacja i czyszczenie

Poziom: średni **Dotyczy: MySQL 5.7–8.0, MariaDB 10.x–11.x** **Czas: ~25 min**

Binlog nie jest logiem błędów ani logiem zapytań — jest zapisem każdej zmiany danych, na którym stoi zarówno replikacja, jak i odtwarzanie do punktu w czasie, i który domyślnie serwer wcale nie musi prowadzić.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
PITR	Point-In-Time Recovery: odtworzenie bazy do dowolnej chwili, np. 14:31
SQL	Structured Query Language: język zapytań do relacyjnych baz danych

Fiszka

TYLKO ODCZYT

Wylistowanie plików binloga: SHOW BINARY LOGS

```
SHOW BINARY LOGS;
```

Przykład:

```
+------------------+-----------+
| Log_name         | File_size |
+------------------+-----------+
| my01-bin.000041 | 104857802 |
| my01-bin.000042 | 104857600 |
| my01-bin.000043 |  18402913 |
```

```
+------------------+-----------+
3 rows in set (0.00 sec)
```

show = FortiOS: wypisuje konfigurację, w podstawowej postaci tylko ustawienia różniące się od domyślnych, a z wariantem full-configuration także wartości domyślne

TYLKO ODCZYT

Odczytanie surowej zawartości pliku binloga: SHOW BINLOG EVENTS

```
SHOW BINLOG EVENTS IN 'my01-bin.000043';
```

Przykład:

```
+--------------------+-----+------------+-----------+-------------
+---------------------------------------+
| Log_name           | Pos | Event_type | Server_id | End_log_pos |
Info                                    |
+--------------------+-----+------------+-----------+-------------
+---------------------------------------+
| my01-bin.000043 |   4 | Format_desc|         1 |         126 | Server ver: 8.0.35,
Binlog ver: 4      |
| my01-bin.000043 | 126 | Query      |         1 |         234 |
BEGIN                                   |
| my01-bin.000043 | 234 | Table_map  |         1 |         298 | table_id: 112
(produkcja.zamowienia)   |
| my01-bin.000043 | 298 | Update_rows|         1 |         512 | table_id: 112 flags:
STMT_END_F        |
+--------------------+-----+------------+-----------+-------------
+---------------------------------------+
4 rows in set (0.00 sec)
```

show = FortiOS: wypisuje konfigurację, w podstawowej postaci tylko ustawienia różniące się od domyślnych, a z wariantem full-configuration także wartości domyślne

TYLKO ODCZYT

Usunięcie logów starszych niż wskazana data: PURGE BINARY LOGS BEFORE

```
PURGE BINARY LOGS BEFORE NOW() - INTERVAL 3 DAY;
```

Z poziomu CLI SQL.

□□ **CLI** = Command Line Interface: praca przez wpisywanie komend w terminalu · **SQL** = Structured Query Language: język zapytań do relacyjnych baz danych

□□ TYLKO ODCZYT

Usunięcie logów starszych niż wskazany plik: PURGE BINARY LOGS TO

```
PURGE BINARY LOGS TO 'my01-bin.000040';
```

Wariant z nazwą pliku zamiast daty.

□□ ZMIENIA SYSTEM

Konwersja binloga na czytelny SQL: mysqlbinlog

```
mysqlbinlog SciezkaDoPliku > SciezkaDoPliku.sql
```

□□ **SQL** = Structured Query Language: język zapytań do relacyjnych baz danych · **mysqlbinlog** = narzędzie konwertujące binarną zawartość pliku binloga MySQL/MariaDB na czytelny SQL

□□ ZMIENIA SYSTEM

Odtworzenie do punktu w czasie po zakresie dat: mysqlbinlog --start-datetime

```
mysqlbinlog --start-datetime="2026-03-14 02:00:00" --stop-datetime="2026-03-14 09:41:00"
my01-bin.000043 | mysql -u NazwaUsera -p
```

Wariant z zakresem czasowym, typowy przy PITR po przywróceniu backupu pełnego z nocy — odtwarzamy wszystko, co wydarzyło się między backupem a chwilą tuż przed awarią.

□□ **PITR** = Point-In-Time Recovery: odtworzenie bazy do dowolnej chwili, np. 14:31 · **mysqlbinlog** = narzędzie konwertujące binarną zawartość pliku binloga MySQL/MariaDB na czytelny SQL

ZMIENIA SYSTEM

Odtworzenie do punktu w czasie po pozycji bajtowej: mysqlbinlog --stop-position

```
mysqlbinlog --stop-position=298 my01-bin.000043 | mysql -u NazwaUsera -p
```

Wariant z pozycją bajtową zamiast czasu, precyzyjniejszy, gdy znamy dokładną pozycję z `SHOW BINLOG EVENTS` (na przykład tuż przed błędnym `DELETE` bez `WHERE`).

mysqlbinlog = narzędzie konwertujące binarną zawartość pliku binloga MySQL/MariaDB na czytelny SQL

Z życia wzięte

Z ŻYCIA WZIĘTE

Ktoś z zespołu Oli uruchamia na my01 `UPDATE` bez `WHERE` na tabeli cen w `sklep_b2b` tuż przed piętnastą. Odtwarzamy backup pełny sprzed nocy, a potem `mysqlbinlog` z zakresem czasu aż do minuty tuż przed błędnym poleceniem, żeby nie stracić zamówień złożonych rano. Cennik wraca do stanu sprzed pomyłki, a zamówienia z całego dnia zostają nietknięte.

Czym jest binlog

Pojęcie: binary log (binlog). Sekwencyjny zapis wszystkich poleceń zmieniających dane — `INSERT`, `UPDATE`, `DELETE`, `CREATE`, `DROP` — utrwalany w osobnych plikach, niezależnie od tego, czy transakcja została jeszcze zatwierdzona. Binlog jest podstawą dwóch mechanizmów: replikacji, opisanej na stronie Replikacja, gdzie replika po prostu odtwarza u siebie strumień zdarzeń przeczytany z binloga źródła, oraz odtwarzania do punktu w czasie (PITR), gdzie po przywróceniu backupu pełnego odtwarzamy zdarzenia z binloga aż do wybranego momentu tuż przed awarią.

W przeciwieństwie do redo logu InnoDB, binlog nie jest domyślnie włączony na wszystkich instalacjach — MySQL i MariaDB uruchamiają się bez niego, dopóki nie ustawimy `log_bin` w `my.cnf`. Bez binloga replikacja jest niemożliwa, a odtworzenie da się cofnąć wyłącznie do momentu ostatniego backupu pełnego, bez żadnej granulacji pomiędzy.

Binlog kontra redo log InnoDB

Oba logi zapisują zmiany, oba są sekwencyjne, i to jedyne, co mają wspólnego — mylenie ich jest częstym błędem osoby przychodzącej z innego silnika, gdzie jeden log robi obie te role naraz.

Cecha	Binlog	Redo log InnoDB
Poziom	logiczny — zapisane polecenie albo wiersze zmian	fizyczny — zmiana konkretnej strony na dysku
Zależność od silnika	niezależny od silnika składowania	wyłącznie InnoDB
Cel	replikacja, PITR, audyt zmian	crash recovery samego InnoDB
Domyślnie włączony	nie (do włączenia w `my.cnf`)	tak, zawsze
Cykliczność	rotacja przy rozmiarze/restarcie, kasowany dopiero po `PURGE` albo wygaśnięciu	nadpisywany cyklicznie po każdym checkpoincie
Możliwość odtworzenia na innej wersji silnika	tak, format logiczny	nie, format wewnętrzny InnoDB

Formaty ROW, STATEMENT, MIXED

> **Pojęcie: format binloga.** Sposób, w jaki zmiana trafia do logu — deklarujemy parametrem `binlog_format` w `my.cnf`.

STATEMENT zapisuje samo polecenie SQL tak, jak zostało wykonane — `UPDATE zamowienia SET status = 'wyslane' WHERE data < '2026-01-01'`. Log jest kompaktowy, ale niebezpieczny przy funkcjach niedeterministycznych (`NOW()`, `RAND()`, `UUID()`), bo replika wykonując to samo polecenie może dostać inny wynik niż źródło.

ROW zapisuje nie polecenie, tylko faktyczne zmienione wiersze — który wiersz, jakie wartości przed i po zmianie. Bezpieczny niezależnie od użytych funkcji, kosztem większego rozmiaru logu przy operacjach masowych (`UPDATE` na milionie wierszy zapisuje milion wpisów, nie jedno polecenie). Domyślny od MySQL 5.7.7.

MIXED zaczyna od `STATEMENT` i automatycznie przełącza się na `ROW` dla poleceń rozpoznanych jako niedeterministyczne — kompromis, dziś rzadziej wybierany niż samo `ROW`, bo różnica w rozmiarze logu przestała być istotnym problemem przy dzisiejszych pojemnościach dysków.

Format	Rozmiar logu	Bezpieczeństwo replikacji	Domyślny od

STATEMENT	najmniejszy	ryzykowny przy funkcjach niedeterministycznych	—
ROW	największy	bezpieczny zawsze	MySQL 5.7.7
MIXED	pośredni	bezpieczny, przełącza się automatycznie	—

Ustawienia archiwizacji logów transakcyjnych

Binlogi z automatu nie odkładają się na dysk, dopóki nie włączymy tego w pliku konfiguracyjnym `my.cnf` / `my.ini`. W pliku wyszukujemy sekcję:

```
# ***** Group Replication Related *****
# Specifies the base name to use for binary log files. With binary logging
# enabled, the server logs all statements that change data to the binary
# log, which is used for backup and replication.
```

Usuwamy znak komentarza `#` przed sekcją albo dopisujemy poniższe parametry:

```
binlog-format = ROW
expire-logs-days = 2
log-bin="SciezkaDoFolderuLogow"
```

`log-bin` włącza binlog i ustawia ścieżkę oraz prefiks nazwy plików, `binlog-format` ustawia format opisany wyżej, `expire-logs-days` ustawia automatyczne usuwanie starych plików po dwóch dniach — parametr przestarzały od MySQL 8.0 na rzecz `binlog_expire_logs_seconds` opisanego niżej. Zmiana `log-bin` wymaga restartu serwera.

SHOW BINARY LOGS i SHOW BINLOG EVENTS

> **Pojęcie: SHOW BINARY LOGS.** Zwraca listę aktualnie istniejących plików binloga wraz z ich rozmiarem.

> **Pojęcie: SHOW BINLOG EVENTS.** Zwraca surową zawartość pojedynczego pliku binloga w formie czytelnej dla człowieka — pozycje bajtowe zdarzeń i ich typy, przydatne przy ręcznym znajdowaniu punktu, do którego chcemy odtwarzać.

Purge logów starszych niż 3 dni

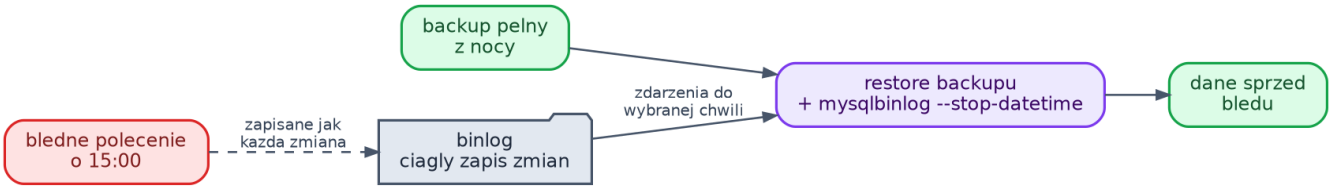
> **Pojęcie: PURGE BINARY LOGS.** Trwale usuwa pliki binloga starsze niż wskazana data (`BEFORE`) albo wszystkie starsze niż wskazany plik (`TO`), zarówno z dysku, jak i z pliku indeksu `.index` , w którym serwer trzyma listę aktywnych plików.

Automatyczne odpowiedniki tego ręcznego czyszczenia to parametry `expire-logs-days` (do 8.0) albo `binlog_expire_logs_seconds` (od 8.0, jednostka sekundowa, domyślnie 30 dni) — ustawione w `my.cnf` , zwalniają z ręcznego wykonywania `PURGE` po każdym backupie. Ręczny `PURGE` pozostaje przydatny wtedy, gdy trzeba natychmiast odzyskać miejsce na dysku, nie czekając na automatyczne wygaśnięcie.

Czyszczenie logu, który jest jeszcze potrzebny replice niepodłączonej od dłuższego czasu, kończy się tym, że replika po powrocie nie może dogonić źródła — brakującego pliku binloga nie da się już odzyskać, a jedynym wyjściem zostaje ponowne zbudowanie repliki od pełnego backupu.

mysqlbinlog — odczyt i PITR

> **Pojęcie: mysqlbinlog.** Narzędzie CLI konwertujące binarną zawartość pliku binloga na czytelny SQL, gotowy albo do przeczytania okiem, albo do wykonania klientem `mysql` przy odtwarzaniu do punktu w czasie.



Backup pełny z nocy plus binlog aż do wybranej chwili odtwarzają dane sprzed błędnego polecenia (kliknięcie otwiera pełny rozmiar)

Częste błędy

Włączenie binloga bez ustawienia wygasania. `log-bin` bez `expire-logs-days`/`binlog_expire_logs_seconds` i bez ręcznego `PURGE` w harmonogramie oznacza katalog logów rosnący bez końca, aż do zapełnienia dysku — to samo zdarzenie, które kończy się awarią całej instancji, bo `mysqld` nie potrafi zapisać kolejnej zmiany.

PURGE logów wciąż potrzebnych replice. Usunięcie pliku binloga, którego pozycja replikacji jeszcze nie minęła u którejkolwiek podłączonej repliki, zrywa replikację nieodwracalnie — przed czyszczeniem ręcznym warto sprawdzić pozycję najwolniejszej repliki przez `SHOW REPLICA STATUS` opisane na stronie Replikacja.

Powiązane: Replikacja · mysqldump i backup logiczny · Log transakcyjny: WAL, redo, transaction log

Źródła: dokumentacja MySQL 8.0, rozdział „The Binary Log" i „mysqlbinlog — Utility for Processing Binary Log Files" · procedura autora

---

Revision #43
Created 2025-03-09 16:55:36 UTC by Tomasz Konieczny
Updated 2026-10-26 14:08:24 UTC by Tomasz Konieczny