mysqldump i backup logiczny

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

`mysqldump` bierze dane przez zwykłe połączenie SQL, nie przez kopiowanie plików — wynikiem jest zestaw poleceń `CREATE`/`INSERT`, który odtwarza się dowolnym klientem `mysql`, nawet na innej wersji silnika.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
B2B	business to business: sprzedaż i systemy skierowane do innych firm, nie do klientów indywidualnych
MVCC	Multi-Version Concurrency Control: baza trzyma kilka wersji wiersza, żeby czytający nie blokował piszącego
SQL	Structured Query Language: język zapytań do relacyjnych baz danych

Fiszka

OSTROŻNIE · NIEODWRACALNE

Utworzenie skryptu backupu wszystkich baz: NazwaSkryptu.sh

```
#!/bin/bash

USER="NazwaUseraBackupu"
PASSWORD="HasloBackup"
#OUTPUT="SciezkaZapisuPlikow"

#Ewentualne usuwanie starych backupów
#rm "$OUTPUTDIR/*gz" > /dev/null 2>&1
```

```
databases=`mysql -u $USER -p$PASSWORD -e "SHOW DATABASES;" | tr -d "| " | grep -v
Database`

for db in $databases; do
    if [[ "$db" != "information_schema" ]] && [[ "$db" != "performance_schema" ]] && [[
"$db" != "mysql" ]] && [[ "$db" != _* ]] ; then
        echo "Dumping database: $db"
        mysqldump -u $USER -p$PASSWORD --databases $db > `date +%Y%m%d`.$db.sql

       # Ewentualne pakowanie plików SQL
       # gzip $OUTPUT/`date +%Y%m%d`.$db.sql
    fi
done
```

Zamiast jednego pliku ze wszystkim naraz, na potrzeby backupu produkcyjnego wygodniejszy jest jeden plik na bazę — łatwiej odtworzyć pojedynczą bazę bez dotykania reszty, i łatwiej zorientować się w rozmiarze każdej z nich osobno. Poniższy skrypt robi dokładnie to: pyta serwer o listę baz i zrzuca każdą po kolei, pomijając bazy systemowe. Tworzymy plik `NazwaSkryptu.sh`. Konto `$USER` powinno być kontem dedykowanym pod backup, opisanym na stronie Użytkownicy i uprawnienia, nie kontem `root`.

echo = od angielskiego echo: wypisuje podany tekst na standardowe wyjście, często łączone z przekierowaniem do pliku · **mysqldump** = zrzut bazy MySQL do pliku

ZMIENIA SYSTEM

Nadanie prawa wykonywania skryptowi: chmod +x

```
chmod +x NazwaSkryptu.sh
```

Nadajemy uprawnienia do uruchamiania (atrybut `x`) dla pliku.

chmod = change mode: zmień prawa dostępu

TYLKO ODCZYT

Uruchomienie skryptu backupu: ./NazwaSkryptu.sh

```
./NazwaSkryptu.sh
```

Wykonujemy poprzez to polecenie.

Przykład:

```
Dumping database: produkcja
Dumping database: zamowienia_archiwum
```

Skrypt w obecnej formie pomija dodatkowo bazę `sys` tylko pośrednio — trafia w warunek `_*` tylko wtedy, gdy lokalna instalacja nazwała ją z podkreślnikiem; na standardowej instalacji `sys` warto dopisać jawnie do listy wykluczeń, jeśli backup ma pomijać także ją. Można też dodać skrypt do `crontab`, żeby wykonywał się automatycznie o stałej porze.

□□ ZMIENIA SYSTEM

Zrzut jednej bazy: mysqldump

```
mysqldump -h NazwaSerwera -u NazwaUsera -p --single-transaction --routines --triggers
NazwaBazy > SciezkaDoPliku
```

Zrzucamy jedną wskazaną bazę ze strukturą, danymi, procedurami i triggerami, w spójnym stanie dzięki `--single-transaction`.

Przykład:

```
mysqldump -h my01 -u nazwa_usera -p --single-transaction --routines --triggers produkcja
> /home/nazwa_usera/produkcja.sql
```

Polecenie przy sukcesie nic nie wypisuje na ekran — cały wynik trafia do wskazanego pliku.

□□ **mysqldump** = zrzut bazy MySQL do pliku

□□ ZMIENIA SYSTEM

Zrzut jednej tabeli: mysqldump

```
mysqldump -h NazwaSerwera -u NazwaUsera -p NazwaBazy NazwaTabeli > SciezkaDoPliku
```

Zrzucamy strukturę i dane dokładnie jednej tabeli, bez reszty bazy — wygodne przed ryzykowną operacją na tej jednej tabeli.

Przykład:

```
mysqldump -h my01 -u nazwa_usera -p produkcja zamowienia >
/home/nazwa_usera/zamowienia.sql
```

mysqldump = zrzut bazy MySQL do pliku

ZMIENIA SYSTEM

Zrzut samej struktury, bez danych: mysqldump --no-data

```
mysqldump --no-data -h NazwaSerwera -u NazwaUsera -p NazwaBazy > SciezkaDoPliku
```

Zrzucamy wyłącznie definicje tabel, bez ani jednego wiersza danych — przydatne przy przygotowaniu pustej kopii bazy pod środowisko testowe.

Przykład:

```
-- MySQL dump 10.13  Distrib 8.0.35
-- Host: my01    Database: produkcja
```

mysqldump = zrzut bazy MySQL do pliku

TYLKO ODCZYT

Odtworzenie zrzutu: mysql < plik

```
mysql -h NazwaSerwera -u NazwaUsera -p NazwaBazy < SciezkaDoPliku
```

Odtwarzamy zrzut tekstowy zwykłym klientem `mysql`, tak jakby jego treść wpisano ręcznie w sesji — ten sam mechanizm co metakomenda `SOURCE` opisana na stronie Klient mysql i połączenie. Baza docelowa musi już istnieć, `mysql` sam jej nie tworzy, chyba że zrzut wykonano z `--databases`, który zawiera własne `CREATE DATABASE`.

Przykład:

```
mysql -h my01 -u nazwa_usera -p produkcja < /home/nazwa_usera/produkcja.sql
```

Z życia wzięte

Z ŻYCIA WZIĘTE

Skrypt backupu `sklep_b2b` na my01 działa co noc od lat, ale nikt nie sprawdzał, czy obejmuje też procedury składowane. Po awarii dysku i odtworzeniu zrzutu aplikacja sklepu B2B zgłasza błąd o brakującej procedurze liczącej rabaty hurtowe. Poprawiamy skrypt, dopisując flagę `--routines`, i robimy backup ponownie, zanim ktokolwiek zdąży tego potrzebować na produkcji.

Backup logiczny a backup fizyczny

Pojęcie: backup logiczny. Zrzut danych w postaci poleceń SQL, wykonany przez odpytanie działającej instancji, a nie przez skopiowanie plików z dysku — w odróżnieniu od backupu fizycznego, jakim zajmuje się Backup fizyczny: Percona XtraBackup i mariabackup, backup logiczny nie wymaga zatrzymania serwera ani identycznej wersji silnika przy odtwarzaniu, ale jest wolniejszy przy dużych bazach, bo dane trzeba faktycznie odczytać i przepisać na SQL, zamiast skopiować bajt w bajt. Ogólny podział backupów rozwija strona Backup pełny, różnicowy, przyrostowy, logi — `mysqldump` w tym podziale jest zawsze backupem pełnym, nigdy przyrostowym.

Kluczowe flagi mysqldump

--single-transaction — otwiera jedną transakcję z poziomem izolacji `REPEATABLE READ` na cały czas trwania zrzutu, korzystając z MVCC InnoDB, żeby uzyskać spójny obraz danych bez blokowania zapisów na tabelach w trakcie backupu. Działa wyłącznie dla tabel InnoDB — na MyISAM nie ma efektu, bo ten silnik nie zna transakcji.

--routines, --triggers, --events — dołączają do zrzutu odpowiednio procedury składowane i funkcje, triggery oraz zdarzenia zaplanowane (`EVENT`), które domyślnie `mysqldump` pomija poza triggerami (te są domyślnie włączone) — łatwo o sytuację, w której zrzut „wszystkich baz" nie zawiera ani jednej procedury składowanej, bo `--routines` trzeba podać jawnie.

--master-data — dopisuje do zrzutu komentarz z pozycją binloga source'a w chwili wykonania zrzutu, gotowy do użycia przy stawianiu nowej repliki opisanej na stronie Replikacja. Wartość `2` zapisuje pozycję jako komentarz SQL (bezpieczniejsze, bo nieaktywne przy zwykłym imporcie), wartość `1` jako aktywne polecenie `CHANGE REPLICATION SOURCE TO`.

--all-databases — zrzuca wszystkie bazy instancji naraz do jednego pliku, łącznie z bazą `mysql` zawierającą konta — podejście wygodne, ale mniej elastyczne niż zrzut każdej bazy osobno, bo

odtworzenie pojedynczej bazy ze wspólnego pliku wymaga ręcznego wycięcia jej fragmentu.

mysqlpump i mydumper — wspomniane

> **Pojęcie: mysqlpump.** Narzędzie dostarczane przez Oracle od MySQL 5.7.8 jako potencjalny następca `mysqldump` — potrafi zrzucać obiekty równolegle wieloma wątkami, co skraca czas backupu dużych instancji, ale nie jest rozwijane tak aktywnie jak `mysqldump` i nie obsługuje `--single-transaction` w pełni tak samo bezproblemowo. W praktyce rzadko wybierany jako domyślne narzędzie backupu.

> **Pojęcie: mydumper.** Zewnętrzne narzędzie open source (nie wchodzi w skład dystrybucji MySQL/MariaDB), zrzucające dane wieloma wątkami równolegle do osobnych plików na tabelę, z odpowiadającym mu `myloader` do odtwarzania — realny wybór przy bazach na tyle dużych, że jednowątkowy `mysqldump` przestaje mieścić się w oknie backupowym.

Dlaczego --lock-all-tables boli na produkcji

> **Pojęcie: --lock-all-tables.** Flaga wymuszająca globalną blokadę odczytu (`FLUSH TABLES WITH READ LOCK`) na wszystkich tabelach wszystkich baz na czas trwania całego zrzutu — jedyny sposób na spójność między tabelami dla silników bez MVCC (MyISAM), ale na instancji z ruchem produkcyjnym oznacza zamrożenie każdego zapisu do dowolnej bazy na cały czas trwania dumpa, nie tylko tej jednej zrzucanej. Backup bazy ważącej kilkadziesiąt gigabajtów potrafi trwać kilkanaście minut — tyle samo czasu aplikacja produkcyjna stoi z zawieszonymi zapisami. Dla tabel InnoDB `--single-transaction` daje tę samą spójność bez blokowania niczego, dlatego na instalacjach czysto InnoDB `--lock-all-tables` nie ma uzasadnienia.

Częste błędy

Zrzut bez --single-transaction na obciążonej bazie InnoDB. Bez tej flagi `mysqldump` domyślnie i tak stara się o spójność przez blokady tabelowe, co na dużej, aktywnej bazie InnoDB oznacza niepotrzebne blokowanie zapisów przez cały czas trwania zrzutu, mimo że MVCC pozwoliłoby tego uniknąć.

Pominięcie --routines przy migracji na nowy serwer. Zrzut bez tej flagi wygląda na kompletny — wszystkie tabele i dane są na miejscu — dopóki aplikacja nie wywoła procedury składowanej, której po prostu nie ma, bo backup jej nie objął.

Powiązane: Użytkownicy i uprawnienia · Backup fizyczny: Percona XtraBackup i mariabackup · Klient mysql i połączenie · Backup pełny, różnicowy, przyrostowy, logi

Źródła: dokumentacja MySQL 8.0, rozdział „mysqldump — A Database Backup Program" · procedura autora (skrypt backupu wszystkich baz)

Revision #42
Created 2025-03-09 17:02:23 UTC by Tomasz Konieczny
Updated 2026-10-26 14:08:25 UTC by Tomasz Konieczny