mongosh: powłoka administracyjna

Poziom: podstawy **Dotyczy: MongoDB 5.0–8.0, mongosh** **Czas: ~20 min**

Interaktywna powłoka administracyjna oparta na JavaScripcie, przez którą wykonujemy niemal każdą operację administracyjną na MongoDB — logowanie, przegląd baz, statystyki i diagnostykę bieżącej aktywności.

Skróty użyte na tej stronie (1)

Skrót	Co znaczy
B2B	business to business: sprzedaż i systemy skierowane do innych firm, nie do klientów indywidualnych

Fiszka

TYLKO ODCZYT

Uruchomienie powłoki bezpośrednio na hoście

```
mongosh
```

Łączymy się z domyślnym serwerem lokalnym na `localhost:27017` bez uwierzytelniania. Po połączeniu pojawia się znak zachęty, w który wpisujemy kolejne komendy.

Przykład:

```
Current Mongosh Log ID: 66f1a2b3c4d5e6f7a8b9c0d1
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true
Using MongoDB:          7.0.14
Using Mongosh:          2.3.1
```

```
test>
```

mongosh = Mongo shell: klient MongoDB

TYLKO ODCZYT

Uruchomienie powłoki wewnątrz kontenera

```
sudo docker exec -it NazwaKontenera mongosh
```

Uruchamiamy powłokę wewnątrz działającego kontenera zamiast na hoście, gdy serwer MongoDB działa w Dockerze. Zasada przekładu jest stała: dowolną komendę `mongosh ...` z tej strony można uruchomić w kontenerze, poprzedzając ją `docker exec -it NazwaKontenera` w trybie interaktywnym albo samym `docker exec NazwaKontenera` przy jednolinijkowcach z `--eval`.

docker = narzędzie do kontenerów

TYLKO ODCZYT

Połączenie z uwierzytelnieniem

```
mongosh -u NazwaUsera -p --authenticationDatabase admin
```

Podajemy nazwę użytkownika flagą `-u`, a flaga `-p` bez wartości wymusza interaktywny monit o hasło zamiast wpisywania go jawnie w linii poleceń. Flaga `--authenticationDatabase` wskazuje bazę, w której zdefiniowano to konto — konta administracyjne żyją zwykle w bazie `admin`, nawet jeśli mają uprawnienia do zupełnie innych baz danych.

mongosh = Mongo shell: klient MongoDB

TYLKO ODCZYT

Wykonanie pojedynczej komendy bez wchodzenia do powłoki

```
mongosh --quiet --eval 'db.version()'
```

Wykonujemy jeden fragment kodu i kończymy działanie, bez wchodzenia w tryb interaktywny — przydatne w skryptach powłoki i zadaniach cyklicznych. Flaga `--quiet` wycisza baner powitalny, który inaczej trafiłby razem z wynikiem do przetwarzanego dalej tekstu.

Przykład:

```
7.0.14
```

mongosh = Mongo shell: klient MongoDB

TYLKO ODCZYT

Wylistowanie wszystkich baz danych: show dbs

```
show dbs
```

Wyświetlamy listę wszystkich baz danych na serwerze wraz z ich rozmiarem na dysku. Bazy zupełnie puste, bez żadnych zapisanych dotąd danych, mogą się na tej liście nie pojawić.

Przykład:

```
test> show dbs
admin    180.00 KiB
appdb    24.61 MiB
config    60.00 KiB
local     72.00 KiB
```

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

TYLKO ODCZYT

Wylistowanie wszystkich baz z poziomu kontenera

```
sudo docker exec -it NazwaKontenera mongosh --eval
"db.getMongo().getDBNames().forEach(function(db) { print(db) });"
```

Wypisujemy same nazwy baz danych, jedna pod drugą, bez wchodzenia do interaktywnej powłoki wewnątrz kontenera — wygodne do wykorzystania dalej w skrypcie powłoki hosta, na przykład jako lista do przeiterowania kolejnymi backupami.

Przykład:

```
admin
appdb
config
local
```

docker = narzędzie do kontenerów

TYLKO ODCZYT

Przełączenie na inną bazę danych: use

```
use appdb
```

Przełączamy kontekst na wskazaną bazę. Od tego momentu zmienna `db` w powłoce odnosi się właśnie do niej. Jeśli baza jeszcze nie istnieje, zostanie faktycznie utworzona dopiero przy pierwszym zapisie do niej jakiegokolwiek dokumentu.

TYLKO ODCZYT

Wylistowanie kolekcji bieżącej bazy: show collections

```
show collections
```

Wyświetlamy wszystkie kolekcje istniejące w bazie, na którą jesteśmy aktualnie przełączeni poleceniem `use`.

Przykład:

```
appdb> show collections
klienci
zamowienia
produkty
```

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

TYLKO ODCZYT

Sprawdzenie statystyk bieżącej bazy: db.stats()

```
db.stats()
```

Zwracamy statystyki bieżącej bazy danych — liczbę kolekcji, liczbę dokumentów łącznie i zajętość miejsca na dysku. Pomocne przy ocenie rozmiaru bazy przed wykonaniem kopii zapasowej opisanej na stronie Backup i odtwarzanie: mongodump i mongorestore.

Przykład:

```
{
  db: 'appdb',
  collections: 3,
  views: 0,
  objects: 148213,
  avgObjSize: 512.4,
  dataSize: 75923456,
  storageSize: 40894464,
  indexes: 5,
  indexSize: 3096576,
  totalSize: 43991040,
  ok: 1
}
```

TYLKO ODCZYT

Sprawdzenie statystyk pojedynczej kolekcji: db.collection.stats()

```
db.zamowienia.stats()
```

Zwracamy te same statystyki, ale zawężone do jednej wskazanej kolekcji, wraz z rozmiarem każdego indeksu założonego na niej z osobna.

Przykład:

```
{
  ns: 'appdb.zamowienia',
  size: 51203840,
  count: 98211,
  avgObjSize: 521,
  storageSize: 28901376,
  nindexes: 2,
  indexSizes: { _id_: 1638400, klient_id_1: 921600 },
  ok: 1
}
```

🔒 TYLKO ODCZYT

Sprawdzenie statusu całego serwera: db.serverStatus()

```
db.serverStatus()
```

Zwracamy obszerny dokument z metrykami całego serwera — liczbą aktywnych połączeń, zużyciem pamięci, licznikami operacji i stanem replikacji. Wynik jest zwykle bardzo duży, dlatego w praktyce częściej odczytujemy pojedyncze pola z niego, na przykład `db.serverStatus().connections`.

Przykład:

```
appdb> db.serverStatus().connections
{
  current: 14,
  available: 51186,
  totalCreated: 302,
  active: 3
}
```

🔒 TYLKO ODCZYT

Sprawdzenie aktualnie trwających operacji: db.currentOp()

```
db.currentOp()
```

Zwracamy listę wszystkich operacji aktualnie wykonywanych przez serwer — jej numer (`opid`), czas trwania w sekundach i treść zapytania. Sięgamy po to, gdy podejrzewamy zapytanie, które wisi dłużej, niż powinno.

Przykład:

```
{
  inprog: [
    {
      opid: 4821059,
      secs_running: 42,
      op: 'query',
      ns: 'appdb.zamowienia',
```

```
      command: { find: 'zamowienia', filter: { status: 'oplacone' } }
    }
  ]
}
```

TYLKO ODCZYT

Zakończenie długo trwającej operacji: db.killOp()

```
db.killOp(4821059)
```

Przerywamy operację o wskazanym numerze `opid`, odczytanym wcześniej z `db.currentOp()`. Używamy tego, gdy zapytanie blokuje inne operacje albo trwa wyraźnie dłużej, niż zakładaliśmy.

TYLKO ODCZYT

Sprawdzenie stanu replica setu: rs.status()

```
rs.status()
```

Zwracamy stan całego replica setu — który węzeł jest aktualnie `PRIMARY`, które są `SECONDARY`, jak dawno każdy z nich potwierdził swoją obecność (`heartbeat`) i jak duże jest opóźnienie replikacji między nimi.

Przykład:

```
{
  set: 'rsAppdb',
  members: [
    { _id: 0, name: 'mongo01:27017', stateStr: 'PRIMARY', health: 1 },
    { _id: 1, name: 'mongo02:27017', stateStr: 'SECONDARY', health: 1 }
  ],
  ok: 1
}
```

Z życia wzięte

Z ŻYCIA WZIĘTE

O dziesiątej rano Kasia zgłasza, że strona kategorii w sklepie B2B ładuje się kilkanaście sekund zamiast ułamka sekundy. Tomek loguje się mongoshem na mongo01 i sprawdza bieżące operacje, znajdując zapytanie do kolekcji produkty, które trwa już ponad minutę i skanuje całą kolekcję zamiast korzystać z indeksu. Zabija tę operację, żeby zwolnić serwer, a dopiero potem szuka w kodzie aplikacji, skąd wzięło się zapytanie bez odpowiedniego filtra po polu z indeksem.

Po co to istnieje

Pojęcie: mongosh. Powłoka `mongosh` zastąpiła starszą `mongo`, wycofaną w MongoDB 6.0, i pozostaje głównym narzędziem administracyjnym do pracy z serwerem zarówno z poziomu skryptów, jak i sesji interaktywnej. Sposób wejścia do powłoki zależy od tego, gdzie faktycznie działa serwer — bezpośrednio na hoście czy wewnątrz kontenera — natomiast wszystkie komendy wykonywane już wewnątrz powłoki (zaczynające się od `db.` albo od `show`) wyglądają identycznie w obu przypadkach.

Częste błędy

Podanie hasła wprost w linii poleceń. `mongosh -u NazwaUsera -p HasłoWprost` zostawia hasło w historii powłoki i w wyniku `ps aux` widocznym dla innych kont na tym samym hoście — flaga `-p` bez wartości, wymuszająca monit interaktywny, jest bezpieczniejsza.

Odczyt `db.serverStatus()` w całości zamiast pojedynczego pola. Wynik bywa na tyle obszerny, że przewijanie go w poszukiwaniu jednej metryki jest wolniejsze niż odwołanie się od razu do konkretnego pola dokumentu.

Powiązane: Architektura MongoDB: dokument, kolekcja, baza, replica set · Operacje na dokumentach · Backup i odtwarzanie: mongodump i mongorestore

Źródła: notatki własne administratora MongoDB · dokumentacja MongoDB, rozdział „mongosh" i „Database Commands"

Revision #43
Created 2026-06-09 09:14:54 UTC by Tomasz Konieczny
Updated 2026-10-26 14:09:56 UTC by Tomasz Konieczny