Database backups per engine

The backup, check and restore commands for each database Exponential supports, in the official order (SQLite 3, PostgreSQL, MongoDB, MariaDB, Oracle, MySQL), with the traps of each engine and a portable schema dump for…

Source file: `doc/eng-us/help/hosting/operations/database-backups-per-engine.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/hosting/operations/database-backups-per-engine.md

This page is for system administrators who back up and restore the databases of hosted Exponential sites. It answers: which command backs up each supported database consistently while the site runs, how to check the copy, how to restore it, and what each engine needs you to watch.

The commands are those of Database support, backup and restore, 10.9 Backups and restore and the per-engine sections of chapter 9. User and database names below are placeholders (`exponential`); use your own.

> **Supported databases (Exponential 6.0.x), in order of preference:** 1. SQLite 3 (the default) · 2. PostgreSQL · 3. MongoDB · 4. MariaDB · 5. Oracle · 6. MySQL. Microsoft SQL Server is known to be possible with additional work at this time (there is no driver in 6.0.x). See Database support and Databases in the official order.

Contents

1. At a glance
2. 1. SQLite 3
3. 2. PostgreSQL
4. 3. MongoDB
5. 4. MariaDB
6. 5. Oracle
7. 6. MySQL
8. Microsoft SQL Server
9. Passwords and permissions
10. A portable dump between SQL engines
11. Checklist

At a glance

Database	Backup	Restore
1. SQLite 3	`sqlite3 <db file> ".backup '/backup/exponential.db'"`	`sqlite3 <db file> ".restore '/backup/exponential.db'"`

Database	Backup	Restore
2. PostgreSQL	`pg_dump -U exponential -Fc exponential > exponential.dump`	`pg_restore -U exponential -d exponential --clean exponential.dump`
3. MongoDB	`mongodump --uri="mongodb://exponential@localhost:27017/exponential" --out=/backup/exponential`	`mongorestore --uri="..." --drop /backup/exponential`
4. MariaDB	`mariadb-dump --single-transaction -u exponential -p exponential > exponential.sql`	`mariadb -u exponential -p exponential < exponential.sql`
5. Oracle	`php bin/php/console ext:ezoracle:datapump` (see its `--help`), or Oracle's `expdp`	Oracle's `impdp`
6. MySQL	`mysqldump --single-transaction -u exponential -p exponential > exponential.sql`	`mysql -u exponential -p exponential < exponential.sql`

A database backup is only half of a site backup; take the files in the same window (Backups: what to back up).

1. SQLite 3

The whole database is one file, by default under `var/storage/sqlite3/` (`exponential.db` is the name `exp:install` uses; `ls var/storage/sqlite3/` shows yours). Next to it are `<database>-wal` and `<database>-shm`, which belong to it.

Back up with the online backup, never by copying. "Never copy a database file that is in use": a copy taken during a write, or without its `-wal` file, can be inconsistent (SQLite backup and restore).

Shell

```
sqlite3 var/storage/sqlite3/exponential.db ".backup '/backup/exponential-$(date +%F).db'"
# or a compacted copy (SQLite 3.27 and later)
sqlite3 var/storage/sqlite3/exponential.db "VACUUM INTO '/backup/exponential-$(date +%F).db'"
# check the copy: prints ok
sqlite3 /backup/exponential-<date>.db "PRAGMA integrity_check"
```

Restore through SQLite's own locking. PHP-FPM workers and Velocity workers keep the database open, so `.restore` copies the backup in page by page under a write lock:

Shell

```
php bin/php/maintenance.php on --message="Restoring" --allow-root-user
sqlite3 var/storage/sqlite3/exponential.db ".restore '/backup/exponential-<date>.db'"
php bin/php/maintenance.php off --allow-root-user
php bin/php/ezcache.php --clear-all --allow-root-user
php bin/php/console exp:velocity cache clear --apply --allow-root-user
```

If `.restore` reports an error, the live database is unchanged; then stop every process that may hold the file (Velocity, PHP-FPM, cron), move the old database with its `-wal` and `-shm` files aside together, put the backup in place, fix owner and mode, and start again.

> **Note**
>
> A SQLite backup needs disk space on the same machine first. Keep room for "the database

> twice over" for `VACUUM` and backups (Monitoring a SQLite site).

2. PostgreSQL

Shell

```
pg_dump -U exponential -Fc exponential > exponential.dump          # custom format, compressed
pg_restore -U exponential -d exponential --clean exponential.dump
```

`pg_dump` takes a consistent snapshot while the site runs. The sources describe PostgreSQL as having "replication and point-in-time recovery of its own"; set those up with the PostgreSQL documentation if a plan promises recovery to a moment between two dumps. After restoring into a fresh database, check the sequences: the kernel reads new ids from sequences named `<table>_<column>_seq` (Sequences).

3. MongoDB

Shell

```
mongodump --uri="mongodb://exponential@localhost:27017/exponential" --out=/backup/$(date +%F)
mongorestore --uri="mongodb://exponential@localhost:27017/exponential" --drop /backup/<date>
```

The tool is described at mongodump. After a restore into a new server, create the indexes from `bin/mongodb/create_indexes.js`; without them subtree queries and URL alias lookups scan whole collections (Performance and scaling). Read the MongoDB limits before you offer it.

4. MariaDB

Shell

```
mariadb-dump --single-transaction -u exponential -p exponential > exponential.sql
mariadb -u exponential -p exponential < exponential.sql
```

`--single-transaction` "gives a consistent dump of InnoDB tables without locking them". MariaDB runs on the same driver as MySQL; the installer creates the tables as InnoDB when the server offers it, and the database should be created in `utf8mb4` (Character set and storage engine). Restore into a database created the same way, in `utf8mb4`.

5. Oracle

Oracle (19c and later) is supported through the `ezoracle` extension. Its console command `datapump` wraps Oracle Data Pump; the other route is Oracle's own `expdp` and `impdp`:

Shell

```
php bin/php/console ext:ezoracle:datapump --help
```

`ezoracle` also ships the commands `health`, `gather-stats`, `recompile`, `purge-recyclebin`, `sequence-sync` and others, and the cronjob parts `ezoracle` (nightly) and `ezoraclehealth` (hourly), which do nothing until enabled in `ezoracle.ini [CronjobSettings]` (9.6 Maintenance). Coordinate Oracle backups with the customer's database administrator; most Oracle sites already have a backup regime.

6. MySQL

Shell

```
mysqldump --single-transaction -u exponential -p exponential > exponential.sql
mysql -u exponential -p exponential < exponential.sql
```

The same notes as for MariaDB apply (mysqldump). MySQL is last in the order "not because it works less well, but because the project recommends the others first for new sites".

Microsoft SQL Server

There is no driver in Exponential 6.0.x, so there is nothing to back up through Exponential. It is "known to be possible with additional work at this time"; see Microsoft SQL Server: possible with additional work and Microsoft SQL Server status.

Passwords and permissions

- `-p` makes the client ask for the password; "never write it on the command line", where it lands in the shell history and is visible to other users of the machine. For unattended jobs use the client's option file or environment, readable only by the backup user.
- Run backups as a user that may read the database but cannot change the site's files.
- Store dumps with restrictive permissions; they contain every user account's data.

A portable dump between SQL engines

To move a site between SQL engines (SQLite, PostgreSQL, MariaDB, MySQL, and Oracle with `ezoracle`), dump with `bin/php/ezsqldumpschema.php --format=generic --output-array --output-types=all` into a `.dba` file and load it with `bin/php/ezsqlinsertschema.php --insert-types=all`, choosing the engine with `--type` (Moving a site to another database). This is a migration tool, not a replacement for the engine's own backup.

Checklist

- ☐ The backup command for the site's engine is the one above, with `--single-transaction` for MariaDB and MySQL.
- ☐ SQLite files are never copied while in use; `.backup` or `VACUUM INTO` is used and the copy is checked.
- ☐ No password appears on a command line or in a crontab line.
- ☐ Restores clear every cache afterwards, and Velocity's response cache too.
- ☐ MongoDB restores recreate the indexes; PostgreSQL restores have correct sequences.
- ☐ Oracle backups are agreed with the customer's database administrator.

Related pages

- Backups: what to back up
- Restore and restore tests

- SQLite for hosts, PostgreSQL for hosts, MongoDB for hosts, MariaDB for hosts, Oracle for hosts, MySQL for hosts
- Changing the database engine
- Database support, Operating system support
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

Become an Exponential Partner

Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.

Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.