Export a MariaDB application database without exposing the backup

A file backup of an application directory may omit the database that stores player accounts, orders or permissions. Export that database as a separate part of the same recovery point. This guide covers an application database you administer on a Linux VPS. For a panel-managed database, use the export tools available to that service or ask support.

Choose the scope and a quiet window

Record the database name, MariaDB version and the matching application version. Identify the application's database account without posting its password in a ticket. Check whether tables use InnoDB or another engine, and whether the application also depends on files that change with database writes. Put the application into its documented maintenance or read-only mode when you need both copies to match.

Use an authorised account with the privileges needed for the objects you export. Ask your database administrator if views, routines or events require additional permissions. Do not grant broad global access or expose port 3306 just to make a backup. Check free space using our Linux disk-space guide before writing a potentially large SQL file.

Create a private destination

Run the following as the Linux user that will own the export. The directory sits outside your web root. The example name application_db is a placeholder for your actual database:

```
umask 077
mkdir -p "$HOME/private-db-backups"
```

```
chmod 700 "$HOME/private-db-backups"
cd "$HOME/private-db-backups"
```

Use a new filename for each attempt so a failed command cannot replace your last known recovery point. Keep passwords out of shell commands and screenshots. The -p option below prompts for the database password without placing it in command history.

Run and check the export

```
mariadb-dump -u backup_user -p --single-transaction --quick \
  --routines --events --triggers application_db > application-2026-11-09.sql
```

Replace both names and choose your own date. Run echo $? immediately after the command; a non-zero result means you must investigate the error. Do not accept a non-empty output file as proof of completion. Preserve the error privately and mark an incomplete export as failed.

MariaDB documents the consistency limits of single-transaction: it suits transactional tables such as InnoDB, while non-transactional data needs a different plan. Avoid schema changes during the export. The routines and events options include objects that a basic table export can miss. This command exports the named application database, not the server's user accounts or global grants.

Protect and test the recovery copy

Confirm the file permissions are 600, record its byte size and checksum, and store an encrypted copy outside the VPS. Keep its decryption key somewhere you can reach during an outage. The SQL may contain personal data and secrets; do not serve it through a public download URL.

Ask an administrator to import the export into a fresh isolated test database with compatible MariaDB tools. Recent dump files can require a client that understands MariaDB's sandbox header. Keep the test application away from live payments, mail and webhooks, then check representative records and application reads. Only an authorised recovery plan should import data into production.

Creating this export does not change the live database, so rollback means stopping the backup attempt and resuming the application's normal mode once checks finish. Never delete the working database to test an export. Keep the matching file backup and SQL together, and record how long the rehearsal took.

Revision #1
Created 2026-11-09 09:49:46 UTC by CloudExa Hosting Admin
Updated 2026-11-09 09:49:46 UTC by CloudExa Hosting Admin