

MySQL / MariaDB backups to a NAS

Dump every database to its own compressed file, keep a few days, and fail loudly when something is wrong.

Requirements

- `mysqldump` or `mariadb-dump`. Newer MariaDB installs may only have `mariadb-dump`, so detect which one exists.
- `gzip`.
- A stored login for root (see *Store MySQL backup credentials safely*).
- A mounted backup share, for example `/mnt/backup`.

The core loop

```
DUMP=$(command -v mysqldump || command -v mariadb-dump) || exit 1
CLI=$(command -v mysql || command -v mariadb) || exit 1
mountpoint -q /mnt/backup || { logger -t mysql-backup "backup share not mounted"; exit 1; }

dbs=$(("$CLI" -N -e 'SHOW DATABASES') || { logger -t mysql-backup "cannot connect"; exit 1; })
for db in $dbs; do
    case $db in information_schema|performance_schema|sys) continue;; esac
    "$DUMP" --single-transaction --routines --triggers --complete-insert "$db" \
        | gzip > "/mnt/backup/${hostname -s}/${date +%Y.%m.%d_%H}.${db}.sql.gz" \
        || logger -t mysql-backup "dump of $db failed"
done
find "/mnt/backup/${hostname -s}" -maxdepth 1 -type f -name '*.sql.gz' -mtime +7 -delete
```

- `--single-transaction` gives a consistent dump of InnoDB tables without locking them.
- Check the `SHOW DATABASES` step. If the login fails and you don't check, the loop silently backs up nothing.
- Add `set -o pipefail` so a failed dump isn't hidden by a successful `gzip`.

Schedule

Run it only on servers that actually run MySQL:

```
0 * * * * root /usr/local/bin/backup_mysql.sh >/dev/null 2>&1
```

Hourly dumps kept for 7 days is 168 copies of each database. Daily, or hourly with fewer days kept, is usually plenty.

Restore

```
gunzip -c 2026.01.01_03.mydb.sql.gz | mysql mydb
```

Revision #1

Created 24 October 2026 13:29:32 by Admin

Updated 24 October 2026 13:29:32 by Admin