

# Manual Migration Using mysqldump and mysql

Manual migrations using MySQL's built-in tools `mysqldump` and `mysql`, are ideal for users who require full control over data export and import, particularly during transitions between providers, database version upgrades, or importing existing self-managed MySQL datasets into Elestio's managed environment. This guide walks through the process of performing a manual migration to and from Elestio MySQL services using command-line tools, ensuring data portability, consistency, and transparency at every step.

## When to Use Manual Migration

Manual migration using `mysqldump` is well-suited for scenarios that demand complete control over the migration process. It is especially useful when transferring databases from a self-hosted MySQL instance, an on-premises server, or another cloud provider into Elestio's managed MySQL service. This method supports one-time imports without requiring ongoing connections between source and destination systems.

It also provides a reliable approach for performing version upgrades. Because `mysqldump` creates logical backups, the resulting SQL files can be restored into newer MySQL versions with minimal compatibility issues. When Elestio's built-in tools are not applicable such as in migrations from isolated environments or in selective schema transfers manual migration becomes the preferred option. It also enables offline backup archiving, providing users with transportable and restorable datasets independent of platform-specific backup formats.

## Performing the Migration

### Prepare the Environments

Before starting the migration, ensure that MySQL is properly installed on both the source system and your Elestio service. The source MySQL server must allow network connections (if remote) and have a user with sufficient privileges to export the database, including read access to all necessary tables, views, stored procedures, and triggers.

On the Elestio side, provision a MySQL service through the dashboard. Once it's active, retrieve the connection credentials from the **Database Info** section this includes host, port, database name, username, and password. Verify that your public IP is allowed under **Cluster Overview** >

**Security > Limit access per IP**, or the MySQL port will not be reachable.

## Create a Dump Using mysqldump

Use mysqldump to export the entire source database into a SQL file. This utility serializes the schema, data, indexes, and routines into a plain-text SQL script.

```
mysqldump -h <source_host> -P <source_port> -u <source_user> -p <source_database> > backup.sql
```

You'll be prompted to enter the source password. Once complete, this produces a portable SQL file named backup.sql. You can also include flags such as `--routines` or `--triggers` if your database uses stored procedures or custom triggers.

To avoid restore-time permission issues, consider adding:

```
--skip-add-drop-table --skip-add-locks --set-gtid-purged=OFF
```

This is especially helpful when importing into a different environment or user setup than the source.

## Transfer the Dump File to the Target

If your local system differs from the Elestio service access point, transfer the dump file using a secure file transfer tool:

```
scp backup.sql your_user@your_workstation:/path/to/local/
```

Ensure that the file is accessible on the system you plan to use to run the restore. The dump does not need to be uploaded to the Elestio server — restores are executed via a remote connection using MySQL's native protocol.

## Create the Target Database

Elestio provisions a default database during setup. If the dump file references a different database name, you may need to create it manually before restore.

Connect to Elestio's MySQL service using the CLI:

```
mysql -h <elestio_host> -P <elestio_port> -u <elestio_user> -p
```

Once inside the MySQL shell, create the target database:

```
CREATE DATABASE target_database CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

This ensures proper character encoding and collation settings for compatibility with modern applications and multilingual content.

## Restore Using MySQL Client

With the dump file in place and the target database created, restore the data:

```
mysql -h <elestio_host> -P <elestio_port> -u <elestio_user> -p target_database <
/path/to/backup.sql
```

This command connects to the Elestio-managed MySQL instance and executes every statement from the dump file, recreating tables, inserting data, and applying schema-level objects.

Ensure the target user has privileges to create tables, insert data, and apply any stored routines or triggers included in the dump file.

## Validate the Migration

After completing the import, verify that the migration was successful by connecting to the Elestio MySQL instance and running checks against key tables and schema components.

Start by listing the tables and checking row counts:

```
SHOW TABLES;
SELECT COUNT(*) FROM your_important_table;
```

Also review any stored procedures, views, or functions if your application depends on them. Confirm the database schema matches the original setup and that your application is able to connect and operate as expected.

If you've updated environment variables or connection strings, make sure these changes are reflected in your deployment or application config. Consider enabling automated backups via Elestio to protect your imported database going forward.

## Benefits of Manual Migration

Manual MySQL migration using mysqldump and mysql offers several important advantages:

- **Portability and Compatibility:** Logical SQL dumps can be restored into any MySQL-compatible instance, whether hosted locally, on another cloud, or in containers.
- **Version Flexibility:** Migrate across MySQL versions without relying on binary replication or platform-specific formats.
- **Offline Storage:** SQL files serve as portable backups that can be stored offline, versioned, or archived for disaster recovery.

- **Platform Independence:** Elestio does not enforce proprietary formats standard MySQL tools give you complete control over the migration and restore process.

This method complements Elestio's automated backup and migration features by enabling custom workflows and one-off imports with full visibility into each stage.

---

Revision #1

Created 2025-04-25 11:11:34 UTC

Updated 2025-04-25 11:41:37 UTC