Restoring a Backup

Restoring backups is essential for recovery, environment duplication, or rollback scenarios. Elestio supports restoring backups both through its built-in dashboard and via command-line tools like mysql and mysqldump. You can also restore from inside Docker Compose environments. This guide provides detailed steps for full and partial restores using each method and explains how to address common errors that occur during restoration.

Restoring from a Backup via Terminal

This method is used when you've created a `.sql` dump file using mysqldump. You can restore it using the mysql command-line client. This approach is useful for restoring backups to new environments, during version upgrades, or testing data locally.

Create the target database if it does not exist

If the database you're restoring into doesn't already exist, you must create it first:

```
mysql -u <username> -p -h <host> -P <port> -e "CREATE DATABASE <database_name>;"
```

You'll be prompted to enter the password after running the command.

Run MySQL to import the backup

This command restores the full contents of the .sql file into the specified database:

```
mysql -u <username> -p -h <host> -P <port> <database_name> < <backup_file>.sql
```

You'll again be prompted for the password. This command restores everything from the dump file, including schema and data.

Restoring via Docker Compose

If your MySQL service is deployed using Docker Compose, you can restore the database inside the container environment. This is useful when MySQL runs in an isolated Docker setup, and you want to handle all backup and restore processes inside that environment.

Copy the backup into the container

Use docker cp to move the .sql file from your host machine to the MySQL container:

```
docker cp ./manual_backup.sql $(docker-compose ps -q mysql):/tmp/manual_backup.sql
```

Run the restore inside the container

Use the mysql CLI tool from within the container to restore the file:

```
docker-compose exec mysql \
  bash -c "mysql -u \$MYSQL_USER -p\"\$MYSQL_PASSWORD\" \$MYSQL_DATABASE <
/tmp/manual_backup.sql"
```

Make sure your environment variables in the Docker Compose file (MYSQL_USER, MYSQL_PASSWORD, MYSQL_DATABASE) match the values used here.

Partial Restores

MySQL supports partial restores when the dump file is created with selective options in mysqldump . For example, you can restore just a specific table or only schema definitions.

Restore a specific table

If you created a dump for a specific table using mysqldump -t, you can restore it independently:

```
mysql -u <username> -p -h <host> -P <port> <database_name> < <table_dump_file>.sql
```

Restore schema only (no data)

To restore only the schema (no table contents), ensure that your dump file was created using:

```
mysqldump -u <username> -p -h <host> -P <port> --no-data <database_name> > schema_only.sql
```

Then restore it like this:

```
mysql -u <username> -p -h <host> -P <port> <database_name> < schema_only.sql
```

Partial restores work best when the original backup was generated with the appropriate level of granularity.

Common Errors & How to Fix Them

Errors during restore are often caused by permission issues, incorrect formats, or existing conflicting objects. Understanding the error messages and their causes will help you recover faster and avoid data loss.

1. Access denied for user

```
ERROR 1045 (28000): Access denied for user 'user'@'host'
```

Ensure you are using the correct username/password and that the user has privileges to access the target database.

2. Table already exists

```
ERROR 1050 (42S01): Table 'my_table' already exists
```

Either drop the target database before restoring:

```
mysql -u <username> -p -h <host> -P <port> -e "DROP DATABASE <database_name>;"
mysql -u <username> -p -h <host> -P <port> -e "CREATE DATABASE <database_name>;"
```

Or manually drop the conflicting tables before restore.

3. ERROR 1064 (Syntax Error)

```
ERROR 1064 (42000): You have an error in your SQL syntax...
```

Check if you’re trying to import a binary or incorrectly formatted file. Ensure you’re using .sql text dump files with the mysql command and not raw .ibd or .frm files.

4. ERROR 1049 (Unknown Database)

```
ERROR 1049 (42000): Unknown database 'mydatabase'
```

The specified database doesn’t exist. Create it manually before restoring.

```
mysql -u <username> -p -h <host> -P <port> -e "CREATE DATABASE mydatabase;"
```

Revision #1
Created 2025-04-30 07:19:51 UTC by kaiwalya
Updated 2025-04-30 08:01:14 UTC by kaiwalya