

# 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 `pg_restore` `psql`. 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 `.dump` file using `pg_dump` in custom format. You can restore it using `pg_restore`, which gives you fine-grained control over what gets restored. This 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.

```
PGPASSWORD='<your-password>' createdb \  
-U <username> \  
-h <host> \  
-p <port> \  
<database_name>
```

### Run `pg_restore` to import the backup

This command restores the full contents of the `.dump` file into the specified database.

```
PGPASSWORD='<your-password>' pg_restore \  
-U <username> \  
-h <host> \  
-p <port> \  
-d <database_name> \  
-v <backup_file>.dump
```

You can add `--clean` to drop existing objects before restoring.

# Restoring via Docker Compose

If your TimescaleDB service is deployed using Docker Compose, you can restore the database inside the container environment. This is useful when TimescaleDB 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 `.dump` file from your host machine to the TimescaleDB container.

```
docker cp ./manual_backup.dump $(docker-compose ps -q postgres):/tmp/manual_backup.dump
```

## Run the restore inside the container

Use `pg_restore` from within the container to restore the file to the database.

```
docker-compose exec postgres \  
  bash -c "PGPASSWORD='\${TIMESCALE_PASSWORD}' pg_restore -U \${TIMESCALE_USER} -d \${TIMESCALE_DB} \  
  -Fc -v /tmp/manual_backup.dump"
```

Make sure your environment variables in the Docker Compose file match the values used here.

# Partial Restores

TimescaleDB supports partial restores, allowing you to restore only selected tables, schemas, or schema definitions. This can be useful when recovering a specific part of the database or testing part of the data.

## Restore a specific table

Use the `-t` flag to restore only one table from the `.dump` file.

```
PGPASSWORD='<your-password>' pg_restore \  
  -U <username> \  
  -h <host> \  
  -p <port> \  
  -d <database_name> \  
  -t <table_name>
```

```
-t <table_name> \  
-v <backup_file>.dump
```

## Restore schema only (no data)

This command will restore only the table structures, types, functions, and other schema definitions without inserting any data.

```
pg_restore \  
-U <username> \  
-h <host> \  
-p <port> \  
-d <database_name> \  
--schema-only \  
-v <backup_file>.dump
```

Partial restores work best with custom-format .dump files generated by `pg_dump -Fc`.

# Common Errors & How to Fix Them

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

### 1. Could not connect to database

```
pg_restore: [archiver] could not connect to database
```

This usually happens if the database doesn't exist or the credentials are incorrect. Make sure the database has been created and the connection details are correct.

### 2. Permission denied for schema

```
ERROR: permission denied for schema public
```

This error indicates that the user account used for restore lacks the privileges needed to write into the schema. Use a superuser account or adjust the schema permissions before restoring.

### 3. Input file appears to be a text format dump

```
pg_restore: error: input file appears to be a text format dump
```

This means you are trying to use `pg_restore` a plain SQL file. In this case, you should use `psql` instead:

```
psql -U <username> -h <host> -p <port> -d <database_name> -f backup.sql
```

#### 4. Duplicate key value violates unique constraint

This occurs when the restore process tries to insert rows that already exist in the target database. You can either drop the target database before restoring or use `--clean` it in `pg_restore` to drop existing objects automatically.

---

Revision #1

Created 2025-05-13 09:48:25 UTC

Updated 2025-05-13 09:49:43 UTC