Manual ClickHouse Migration Using clickhouse-backup

Manual migrations using ClickHouse's native tools, such as clickhouse-client, clickhouse-backup, and SQL dump files, are ideal for users who require full control over data export and import, particularly during transitions between providers, ClickHouse version upgrades, or importing existing self-managed ClickHouse datasets into Elestio's managed environment. This guide walks through the process of performing a manual migration to and from Elestio ClickHouse services using command-line tools, ensuring data portability, consistency, and transparency at every step.

When to Use Manual Migration

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

It also provides a reliable approach for performing version upgrades. Because ClickHouse allows full schema and data exports via SQL or compressed binary backups, it can restore into newer versions with minimal compatibility issues. When Elestio's built-in migration tools are not applicable such as migrations from isolated environments or partial database exports manual migration becomes the preferred option. It also supports offline backup and archiving, enabling users to store, transport, and restore datasets independent of platform-specific tools.

Performing the Migration

Prepare the Environments

Before starting the migration, ensure that ClickHouse is properly installed on both the source system and your Elestio service. The source ClickHouse server must allow access (if remote) and have a user with sufficient privileges to export databases, tables, and relevant partitions.

On the Elestio side, provision a ClickHouse service through the dashboard. Once active, retrieve the connection credentials from the Database Info section, which includes host, port (typically 9000 for TCP or 8123 for HTTP), username, and password. Confirm that your public IP is permitted under **Cluster Overview > Security > Limit access per IP** to ensure the ClickHouse port is reachable.

Create a Backup Using ClickHouse Native Tools

There are two primary methods to export a dataset from a ClickHouse instance:

Option 1: SQL Dump

To generate a schema and data dump, run:

```
clickhouse-client --host <source_host> --query="SHOW CREATE TABLE <db>.<table>" > schema.sql
clickhouse-client --host <source_host> --query="SELECT * FROM <db>.<table> FORMAT Native" >
data.native
```

Repeat this process for all required tables.

Option 2: Use clickhouse-backup

Alternatively, use the clickhouse-backup tool to create compressed backups that include metadata
and data:

```
clickhouse-backup create migration_snapshot
clickhouse-backup upload migration_snapshot
```

This tool can also store backups locally or push them to S3-compatible storage.

Transfer the Backup to the Target

Use a secure file transfer utility such as SCP to move exported files to the system that will connect
to Elestio:

```
scp -r /path/to/backup user@host:/path/to/restore-system/
```

If using clickhouse-backup, copy the backup directory or the downloaded archive. These files will
be restored into the Elestio-managed ClickHouse instance using the same tools or SQL replay.

Restore the Dataset to Elestio

To restore using SQL:

1. Recreate the schema:

```
clickhouse-client --host <elestio_host> --port 9000 --user <username> --password <password> <
schema.sql
```

2. Import the data:

```
clickhouse-client --host <elestio_host> --port 9000 --user <username> --password <password> --
query="INSERT INTO <db>.<table> FORMAT Native" < data.native
```

If using clickhouse-backup, download the backup onto a local or remote machine with access to Elestio. Then:

```
clickhouse-backup restore migration_snapshot
```

Ensure the schema is created before restoring data, and verify that all necessary tables and partitions are populated.

Validate the Migration

After the migration, verify that your Elestio ClickHouse instance contains all expected data and performs correctly:

- **Check Row Count**

```
clickhouse-client --host <elestio_host> --port 9000 --user <username> --password <password> --
query="SELECT count() FROM <db>.<table>"
```

- **List Tables**

```
clickhouse-client --host <elestio_host> --port 9000 --user <username> --password <password> --
query="SHOW TABLES FROM <db>"
```

- **Query Sample Data**
  Run queries to validate critical business metrics or analytical functions. Check that partitioning, primary keys, and indexes are preserved.

Finally, ensure that application connection strings have been updated to point to the new Elestio-hosted ClickHouse service and that dashboards, ingestion pipelines, or integrations function correctly.

Benefits of Manual Migration

Manual ClickHouse migration using native tools and backup utilities offers several important advantages:

- **Portability and Compatibility**: Native ClickHouse formats (SQL, Native, backups) are open and can be restored into any compatible instance across VMs, containers, or cloud providers.
- **Version Flexibility**: Easily move between ClickHouse versions using exports that do not rely on replication or binary compatibility.
- **Offline Storage**: Backup files can be archived, versioned, and stored offline to support disaster recovery, compliance, and long-term retention.
- **Platform Independence**: Elestio supports open standards and does not enforce vendor lock-in. Migrations using native tools provide full control over schema design, data

ownership, and performance tuning.