Creating Manual Backups

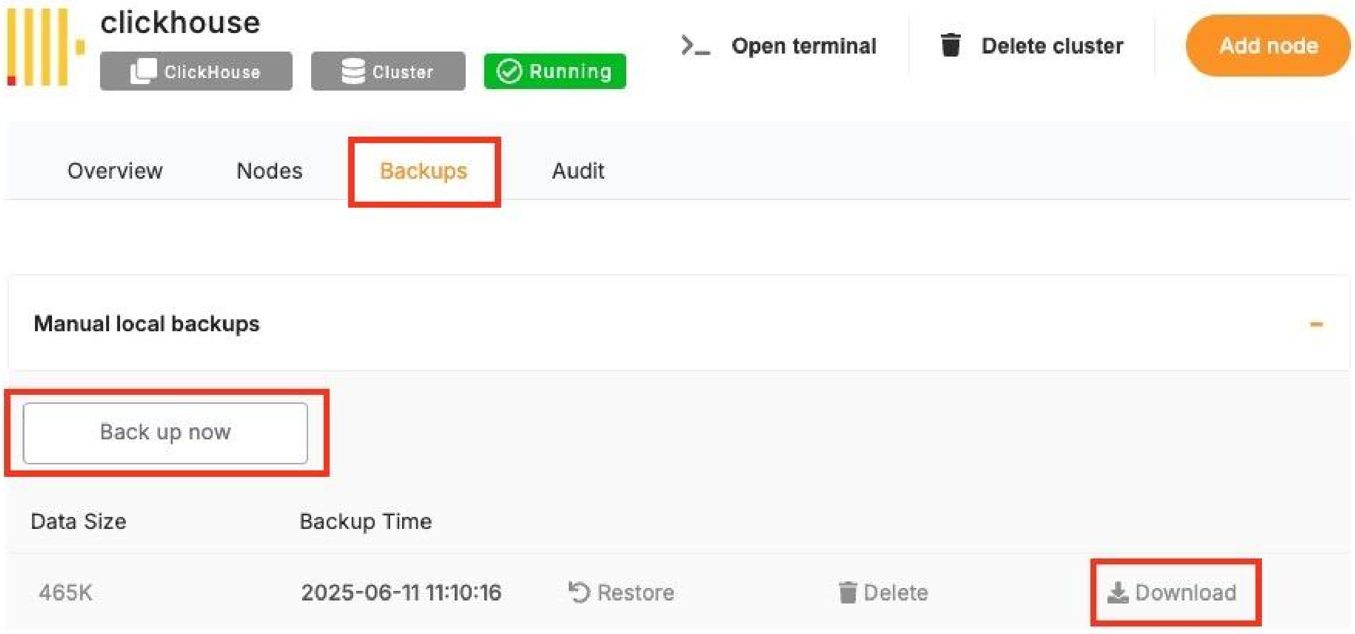
Regular backups are essential when running a ClickHouse deployment, especially if you're using it for persistent analytics or time-series data. While Elestio handles automated backups by default, you may want to create manual backups before configuration changes, retain a local archive, or test backup automation. This guide walks through multiple methods for creating ClickHouse backups on Elestio, including dashboard snapshots, command-line approaches, and Docker Compose-based setups. It also explains backup storage, retention, and automation using scheduled jobs.

Manual Service Backups on Elestio

If you're using Elestio's managed ClickHouse service, the simplest way to perform a full backup is directly through the Elestio dashboard. This creates a snapshot of your current ClickHouse dataset and stores it in Elestio's infrastructure. These snapshots can be restored later from the same interface, which is helpful when making critical changes or testing recovery workflows.

To trigger a manual ClickHouse backup on Elestio:

- Log in to the Elestio dashboard.
- Navigate to your ClickHouse service or cluster.
- Click the **Backups** tab in the service menu.
- Choose **Back up now** to generate a manual snapshot.



Data Size	Backup Time
465K	2025-06-11 11:10:16

This method is recommended for quick, reliable backups without needing to use the command line.

Manual Backups Using Docker Compose

If your ClickHouse instance is deployed via Docker Compose (as is common on Elestio-hosted environments), you can manually back up ClickHouse by either copying its internal storage files or using the native BACKUP SQL command (available in ClickHouse v21.12+). These approaches allow you to maintain control over backup logic and frequency.

Access Elestio Terminal

Go to your deployed ClickHouse service in the Elestio dashboard, navigate to **Tools > Terminal**, and log in using the credentials provided.

Locate the ClickHouse Container Directory

Navigate to your app directory:

```
cd /opt/app/
```

This is the working directory of your Docker Compose project, which contains the docker-compose.yml file.

Trigger a Backup (Using SQL)

If you’re using ClickHouse with backup support enabled, you can execute:

```
docker-compose exec clickhouse clickhouse-client --query="BACKUP DATABASE default TO
Disk('/backups/backup_$(date +%F)')"
```

This creates a full backup of the default database inside the container at /backups.

Copy Backup Files from the Container

Use docker cp to move the backup directory to your host system:

```
docker cp $(docker-compose ps -q clickhouse):/backups/backup_$(date +%F)
./clickhouse_backup_$(date +%F)
```

This gives you a restorable backup snapshot for storage or future recovery.

Backup Storage & Retention Best Practices

After creating backups, it's important to store them securely and manage retention properly. ClickHouse backups can grow large depending on the volume and compression of your data.

Guidelines to Follow:

- Use clear naming: clickhouse_backup_2025_06_09
- Store off-site or on cloud storage (e.g. AWS S3, Backblaze, encrypted storage)
- Retain: 7 daily backups, 4 weekly backups, and 3–6 monthly backups
- Automate old file cleanup with cron jobs or retention scripts
- Optionally compress backups with tar, gzip, or xz to reduce space

Automating ClickHouse Backups (cron)

Manual backup commands can be scheduled using tools like cron on Linux-based systems. This allows you to regularly back up your database without needing to run commands manually. Automating the process also reduces the risk of forgetting backups and ensures more consistent retention.

Example: Daily Backup at 3 AM

Edit the crontab:

```
crontab -e
```

Add a job like:

```
0 3 * * * docker-compose -f /opt/app/docker-compose.yml exec clickhouse \
clickhouse-client --query="BACKUP DATABASE default TO Disk('/backups/backup_$(date +\%F)')" &&
\
docker cp $(docker-compose -f /opt/app/docker-compose.yml ps -q
clickhouse):/backups/backup_$(date +\%F) /backups/clickhouse_backup_$(date +\%F)
```

Make sure /backups/ exists and is writable by the cron user.

You can also compress the file or upload to cloud storage in the same script:

```
tar -czf /backups/clickhouse_backup_$(date +\%F).tar.gz /backups/clickhouse_backup_$(date +\%F)
rclone copy /backups/clickhouse_backup_$(date +\%F).tar.gz remote:clickhouse-backups
```

Backup Format and Restore Notes

Format	Description	Restore Method
`/backups/backup_<date>`	SQL-based backup using BACKUP command	Use RESTORE DATABASE from the same Disk location
`.tar.gz` or `.tar` archive	Filesystem snapshot of `/var/lib/clickhouse`	Stop ClickHouse, extract data back into the directory, then restart

To restore from a backup:

- **Stop ClickHouse**:

```
docker-compose down
```

- **Restore via SQL**:

```
docker-compose exec clickhouse clickhouse-client --query="RESTORE DATABASE default FROM Disk('/backups/backup_2025-06-09')"
```

- **Or restore from file-based archive**:

```
tar -xzf clickhouse_backup_2025-06-09.tar.gz -C /opt/app/data/clickhouse/
docker-compose up -d
```

Revision #1
Created 11 June 2025 06:51:59 by kaiwalya
Updated 11 June 2025 07:05:39 by kaiwalya