

# Preventing Full Disk Issues

Running out of disk space in a ClickHouse environment can cause query failures, part merge errors, and even full service downtime. ClickHouse is highly dependent on disk for storing columnar data, part files, metadata, temporary sort buffers, and backups. On platforms like Elestio, infrastructure is managed, but users are still responsible for monitoring storage, managing data retention, and optimizing resource usage. This guide explains how to monitor and clean up disk usage, configure safe retention policies, and implement long-term strategies to prevent full disk scenarios in ClickHouse when running under Docker Compose

## Monitoring Disk Usage

### Inspect the host system storage

Run this on the host machine to check which mount point is filling up:

```
df -h
```

This shows usage across all mounted volumes. Look for the mount used by your ClickHouse volume—usually mapped to something like `/var/lib/docker/volumes/clickhouse_data/_data`.

### Check disk usage from inside the container

Enter the ClickHouse container shell:

```
docker-compose exec clickhouse bash
```

Inside, check total ClickHouse disk usage:

```
du -sh /var/lib/clickhouse
```

To inspect usage of specific folders like `data/`, `tmp/`, or `store/`:

```
ls -lh /var/lib/clickhouse
```

## Configuring Alerts and Cleaning Up Storage

## Inspect Docker's storage usage

On the host, check space used by containers, images, volumes:

```
docker system df
```

## Identify and remove unused Docker volumes

List all Docker volumes:

```
docker volume ls
```

Remove unused volumes (only if you're sure they're not needed):

```
docker volume rm <volume-name>
```

“ **Warning:** Never delete your active ClickHouse data volume unless you've backed it up.

## Drop data manually using SQL

To free space by removing outdated partitions or tables:

```
ALTER TABLE logs DROP PARTITION '2024-01';  
TRUNCATE TABLE temp_events;
```

## Clean up local backups

If you're storing backups under `/var/lib/clickhouse/backup`, list and delete old ones:

```
ls -lh /var/lib/clickhouse/backup  
rm -rf /var/lib/clickhouse/backup/backup-  
<timestamp>
```

Ensure important backups are offloaded before removing.

# Managing Temporary Files

## Monitor temporary file usage

Check the temp directory inside the container:

```
du -sh /var/lib/clickhouse/tmp
```

Old files may remain if queries or merges crashed. Clean up when the system is idle.

## Redirect temporary paths to persistent storage

Modify the `tmp_path` in `config.xml` to use a volume-backed directory:

```
<tmp_path>/var/lib/clickhouse/tmp/</tmp_path>
```

Restart the container after applying changes.

# Best Practices for Disk Space Management

- **Avoid storing binary blobs:** Do not store large files like PDFs or images in ClickHouse. Use external object storage and only store references.
- **Use TTL to expire old data:** Automatically delete old data based on timestamps:

```
ALTER TABLE logs MODIFY TTL created_at + INTERVAL 90 DAY;
```

- **Drop old partitions regularly:** If partitioned by month/day, remove outdated partitions:

```
ALTER TABLE logs DROP PARTITION '2023-12';
```

- **Enable efficient compression:** Use ZSTD for better compression and lower disk usage:

```
CREATE TABLE logs (...) ENGINE = MergeTree() SETTINGS compression = 'ZSTD';
```

- **Split large inserts into smaller batches:** Avoid memory and disk spikes during large ingest operations.
- **Optimize background merge load:** Tune merge concurrency and thresholds using:

```
<background_pool_size>8</background_pool_size>
```

- **Limit disk spill during queries:** Prevent massive temp usage during large operations:

```
<max_bytes_before_external_sort>500000000</max_bytes_before_external_sort>
```

- **Rotate Docker logs:** Prevent logs from filling up your disk using log rotation:

```
logging:  
  driver: "json-file"  
  options:  
    max-size: "10m"  
    max-file: "3"
```

- **Monitor disk usage from ClickHouse itself:** Track table-level disk usage using system tables:

```
SELECT table, sum(bytes_on_disk) AS size FROM system.parts GROUP BY table ORDER BY size DESC;
```

- **Offload backups to remote storage:** Backups inside containers should be copied off-host. Use Elestio's backup tool or mount a backup volume:

volumes:

```
- /mnt/backups:/backups
```

---

Revision #1

Created 11 June 2025 10:04:24 by kaiwalya

Updated 11 June 2025 11:12:29 by kaiwalya