

Preventing Full Disk Issues

Running out of disk space in a Redis environment can lead to failed writes, snapshot errors, and service unavailability. Redis relies on disk storage for persistence (RDB and AOF files), temporary dumps, and logs especially when persistence is enabled. On platforms like Elestio, while the infrastructure is managed, users are responsible for monitoring disk usage, configuring retention policies, and managing backups. This guide covers how to monitor disk consumption, configure alerts, remove unused data, and follow best practices to prevent full disk scenarios in a Redis setup using Docker Compose.

Monitoring Disk Usage

Disk usage monitoring is essential for spotting unusual growth before it leads to failures. In Docker Compose setups, you'll need both host-level and container-level visibility.

Inspect the host system storage

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

```
df -h
```

This shows available and used space for each volume. Identify the mount point used by your Redis volume—usually mapped to something like `/var/lib/docker/volumes/redis_data/_data`.

Check disk usage from inside the container

Open a shell inside the Redis container:

```
docker-compose exec redis sh
```

Inside the container, check the data directory size:

```
du -sh /data
```

This reveals total usage by persistence files (appendonly.aof, dump.rdb, temporary files). You can inspect individual file sizes with:

```
ls -lh /data
```

Configuring Alerts and Cleaning Up Storage

Monitoring alone isn't enough—automated alerts and safe cleanup prevent downtime. You can inspect disk usage across Docker resources on the host with:

```
docker system df
```

Identify unused Docker volumes

```
docker volume ls
```

To remove a specific unused volume:

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

“**Warning:** Never delete the volume mapped to your Redis data unless you've backed up its contents and confirmed it is not in use.

Trigger AOF file compaction

If AOF persistence is enabled, the append-only file can grow large over time. You can manually trigger a rewrite to compact the file:

```
docker-compose exec redis redis-cli BGREWRITEAOF
```

This creates a smaller AOF file containing the same dataset.

Clean up old snapshots

If you are using RDB snapshots, they're stored in /data within the container (mapped to a host volume). To clean up, list them first:

```
docker-compose exec redis ls -lh /data
```

Remove unnecessary .rdb files with:

```
docker-compose exec redis rm /data/dump-<timestamp>.rdb
```

Managing & Optimizing Temporary Files

Redis creates temporary files during fork operations for AOF rewrites and RDB saves. These are stored in the container's /tmp directory.

Monitor temporary file usage:

```
docker-compose exec redis du -sh /tmp
```

If /tmp fills up, writes and forks may fail. You can change the temporary directory by modifying the dir directive in redis.conf to point to /data, which is volume-backed:

```
dir /data
```

Restart the container to apply changes.

Best Practices for Disk Space Management

Long-term disk space health in Redis requires proactive design and ongoing management.

- **Avoid storing binary blobs:** Store large files (images, PDFs, etc.) outside Redis and use Redis only for keys/metadata. Use object storage for large content.
- **Disable persistence if not needed:** For ephemeral cache use cases, you can disable persistence entirely to reduce disk usage:

```
appendonly no  
save ""
```

- **Limit AOF growth:** Fine-tune AOF rewrite behavior in redis.conf:

```
auto-aof-rewrite-percentage 100  
auto-aof-rewrite-min-size 64mb
```

- **Rotate logs in containers:** If logging to file (e.g., /var/log/redis/redis-server.log), configure logrotate on the host or use Docker log rotation options via docker-compose.yml :

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

- **Evict old keys with TTLs:** Set expiration on cache keys to prevent unbounded growth:

```
SET session:<id> "data" EX 3600
```

- **Monitor data size:** Use INFO persistence and INFO memory to track memory usage and AOF file size:

```
docker-compose exec redis redis-cli INFO memory
docker-compose exec redis redis-cli INFO persistence
```

- **Offload backups:** Backups stored in /data should be moved off the container host. Use Elestio backup tools or mount a remote backup volume in your docker-compose.yml.
-

Revision #1

Created 20 May 2025 10:51:39 by kaiwalya

Updated 20 May 2025 11:12:46 by kaiwalya