

Creating Manual Backups

Regular backups are essential when running a Redis deployment especially if you're using it for persistent 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 Redis 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 Redis service, the simplest way to perform a full backup is directly through the Elestio dashboard. This creates a snapshot of your current Redis 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 Redis backup on Elestio:

1. Log in to the [Elestio dashboard](#).
2. Navigate to your Redis service or cluster.
3. Click the **Backups** tab in the service menu.
4. Choose **Back up now** to generate a manual snapshot.

The screenshot shows the Elestio dashboard for a Redis cluster named 'redis-aiont'. The cluster status is 'Running'. The 'Backups' tab is selected in the service menu. Under 'Manual local backups', there is a 'Back up now' button. Below this, a table lists backup details:

Data Size	Backup Time	Restore	Delete	Download
223	2025-05-20 12:25:04	Restore	Delete	Download

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

Manual Backups Using Docker Compose

If your Redis instance is deployed via Docker Compose (as is common on Elestio-hosted environments), you can manually back up Redis by copying its internal snapshot files. Redis persistence is managed through **RDB (Redis Database)** and optionally **AOF (Append-Only File)** logs, both of which reside in the container's filesystem.

Access Elestio Terminal

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

Locate the Redis 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 an RDB Snapshot (Optional)

Redis typically saves snapshots automatically based on configuration, but you can force one manually:

```
docker-compose exec redis redis-cli SAVE
```

This command triggers an immediate snapshot. The resulting file is usually called dump.rdb.

Copy Backup Files from the Container

Use `docker cp` to copy the snapshot file (and optionally the AOF file, if enabled) from the container to your host system:

```
docker cp $(docker-compose ps -q redis):/data/dump.rdb ./backup_$(date +%F).rdb
```

If AOF persistence is also enabled (via `appendonly yes` in `redis.conf`), back up the AOF log as well:

```
docker cp $(docker-compose ps -q redis):/data/appendonly.aof ./appendonly_$(date +%F).aof
```

This gives you complete Redis data snapshots for storage or future recovery.

Backup Storage & Retention

Best Practices

After creating backups, it's important to store them securely and manage retention properly. Redis backups are binary files and can be quite compact (RDB) or larger and more frequent (AOF), depending on configuration.

Guidelines to Follow:

- Use clear naming: `redis_backup_2025_05_19.rdb`
- 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 gzip or xz to reduce space.

Automating Redis 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

1. Edit the crontab:

```
crontab -e
```

2. Add a job like:

```
0 3 * * * docker-compose -f /opt/app/docker-compose.yml exec redis redis-cli SAVE && \  
docker cp $(docker-compose -f /opt/app/docker-compose.yml ps -q redis):/data/dump.rdb
```

```
/backups/redis_backup_$(date +%F).rdb
```

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:

```
gzip /backups/redis_backup_$(date +%F).rdb  
rclone copy /backups/remote-dir/ remote:redis-backups
```

Backup Format and Restore Notes

Format	Description	Restore Method
<code>dump.rdb</code>	Binary snapshot of full dataset	Stop Redis, replace <code>dump.rdb</code> , then restart Redis
<code>appendonly.aof</code>	Append-only command log	Stop Redis, replace AOF file, then restart Redis

To restore from a backup:

1. Stop Redis (`docker-compose down`)
2. Replace the corresponding file in the volumes or `/data` directory.
3. Restart Redis (`docker-compose up -d`)

Revision #1

Created 2025-05-20 09:12:39 UTC

Updated 2025-05-20 09:18:24 UTC