

# Restoring a Backup

Restoring KeyDB backups is critical for disaster recovery, staging environment replication, or rolling back to a known good state. Elestio supports restoration via its web dashboard and manual methods using Docker Compose and command-line tools. This guide covers how to restore KeyDB backups from RDB or AOF files, for both full and partial restore scenarios, and includes fixes for common errors during the process.

## Restoring from a Backup via Terminal

This method assumes you already have a backup file such as `dump.rdb` or `appendonly.aof`. Restoring involves stopping the container, replacing the data file(s), and restarting KeyDB so it can load the new data at boot time.

### Stop the KeyDB Container

Cleanly stop the container to prevent data corruption:

```
docker-compose down
```

### Replace the Backup File

Move the desired backup file into the volume directory that maps to the KeyDB container's `/data`.

Example for RDB:

```
cp ./backup_2025_06_24.rdb /opt/app/data/dump.rdb
```

“ Ensure your `docker-compose.yml` contains the correct volume mapping:

```
volumes:  
  - ./data:/data
```

For AOF-based persistence:

```
cp ./appendonly_2025_06_24.aof /opt/app/data/appendonly.aof
```

## Restart KeyDB

Bring the container back up:

```
docker-compose up -d
```

KeyDB will automatically load dump.rdb or appendonly.aof depending on its configuration in keydb.conf or Docker entrypoint.

# Restoring via Docker Compose Terminal

If you prefer working inside the container environment, you can directly inject the backup file into the KeyDB container using Docker commands.

## Copy the Backup File into the Container

For RDB:

```
docker cp ./backup_2025_06_24.rdb $(docker-compose ps -q keydb):/data/dump.rdb
```

For AOF:

```
docker cp ./appendonly_2025_06_24.aof $(docker-compose ps -q keydb):/data/appendonly.aof
```

## Restart the KeyDB Container

```
docker-compose restart keydb
```

KeyDB will now reload the updated data file(s) during startup.

# Partial Restores in KeyDB

KeyDB, like Redis, does not support partial data restoration out of the box. However, workarounds exist to selectively restore key-value pairs or subsets of data.

# Restore Selected Keys via CLI

If you've exported a list of keys and their values, you can restore them using a script with `keydb-cli`:

```
cat keys_to_restore.txt | while read key; do
  value=$(cat dump.json | jq -r ".\"$key\"")
  keydb-cli SET "$key" "$value"
done
```

This method is useful when working with pre-filtered exports in JSON, CSV, or key dumps

## Restore from a Partial AOF

If your backup is a trimmed-down AOF file (e.g., created by filtering certain operations), KeyDB will replay it entirely at startup:

1. Replace the existing `appendonly.aof` file.
2. Restart the container.
3. KeyDB will process only the included operations, effectively performing a partial restore.

# Common Errors & How to Fix Them

Restoring KeyDB may occasionally fail due to configuration mismatches, permission issues, or corrupted backup files. Below are common errors and their solutions.

## 1. NOAUTH Authentication Required

### Error:

```
(error) NOAUTH Authentication required.
```

**Cause:** The KeyDB instance requires authentication for any CLI interaction.

### Fix:

```
keydb-cli -a yourpassword
```

In scripts:

```
keydb-cli -a "$KEYDB_PASSWORD" < restore_script.txt
```

## 2. KeyDB Fails to Start After Restore

### Error:

```
Fatal error loading the DB: Invalid RDB format
```

**Cause:** The backup file is corrupted or incompatible with the KeyDB version.

### Fix:

- Make sure the backup was created with the same or compatible KeyDB version.
- If necessary, downgrade or upgrade the container image to match the backup version.

## 3. Data Not Restored

**Cause:** KeyDB is configured to use a different persistence method than the one you restored.

### Fix:

Check your persistence mode in keydb.conf or Docker entry:

```
appendonly yes    # for AOF
appendonly no     # for RDB
```

Ensure the right file (dump.rdb or appendonly.aof) exists at /data.

## 4. Permission Denied When Copying Files

### Error:

```
cp: cannot create regular file '/opt/app/data/dump.rdb': Permission denied
```

### Fix:

Use sudo if your shell user doesn't have write access:

```
sudo cp ./backup_2025_06_24.rdb /opt/app/data/dump.rdb
```

Or adjust directory permissions:

```
sudo chown $USER:$USER /opt/app/data
```

Updated 2025-06-25 09:03:07 UTC