Recover a failed database

When a cluster fails or a database is corrupted, you must:

1. **Restore the cluster configuration** from the CCS files
2. Recover the databases with their previous configuration and data

To restore the data that was in the databases to databases in the new cluster you must restore the database persistence files (backup, AOF, or snapshot files) to the databases. These files are stored in the *persistence storage location*.

The database recovery process includes:

1. If the cluster failed, **recover the cluster**.
2. Identify recoverable databases.
3. Restore the database data.
4. Verify that the databases are active.

**Prerequisites**

- Before you start database recovery, make sure that the cluster that hosts the database is healthy. In the case of a cluster failure, you must **recover the cluster** before you recover the databases.
- We recommend that you allocate new persistent storage drives for the new cluster nodes. If you use the original storage drives, make sure that you backup all files on the old persistent storage drives to another location.

**Recovering the databases**

After you prepare the cluster that hosts the database, you can run the recovery process from the **rladmin** command-line interface (CLI).

To recover the database:

1. Mount the persistent storage drives with the recovery files to the new nodes. These drives must contain the cluster configuration backup files and database persistence files.

   **Note:** Make sure that the user *redislabs* has permissions to access the storage location of the configuration and persistence files on each of the nodes.

   If you use local persistent storage, place all of the recovery files on each of the cluster nodes.

2. To see which databases are recoverable, run:

   ```
   rladmin recover list
   ```

   The status for each database can be either ready for recovery or missing files. An indication of missing files in any of the databases can result from:

   - The storage location is not found - Make sure that on all of the nodes in the cluster the recovery path is set correctly.
• Files are not found in the storage location - Move the files to the storage location.
• No permission to read the files - Change the file permissions so that redislabs:redislabs has 640 permissions.
• Files are corrupted - Locate copies of the files that are not corrupted.

If you cannot resolve the issues, contact Redis support.

3. Recover the database, either:

• Recover the databases all at once from the persistence files located in the persistent storage drives: rladmin recover all

• Recover a single database from the persistence files located in the persistent storage drives: rladmin recover db <database_id|name>

• Recover only the database configuration for a single database (without the data): recover db only_configuration <db_name>

Note:
• If persistence was not configure for the database, the database is restored empty.
• For Active-Active databases that still have live instances, we recommend that you recover the configuration for the failed instances and let the data update from the other instances.
• For Active-Active databases that all instances need to be recovered, we recommend that you recover one instance with the data and only recover the configuration for the other instances. The empty instances then update from the recovered data.
• If the persistence files of the databases from the old cluster are not stored in the persistent storage location of the new node, you must first map the recovery path of each node to the location of the old persistence files. To do this, run the node <id> recovery_path set command in rladmin. The persistence files for each database are located in the persistent storage path of the nodes from the old cluster, usually under /var/opt/redislabs/persist/redis.

4. To verify that the recovered databases are now active, run: rladmin status

After the databases are recovered, make sure that your redis clients can successfully connect to the databases.

Updated: April 14, 2022