# Coordinate a HuskSync database recovery across all writers

A shared HuskSync database can be written by several backends. Restoring it while those writers remain active can mix old and new state or replace recovered data. Treat recovery as a coordinated network operation rather than a database file copied beneath running servers.

## Record the starting point

Identify every participating backend, its cluster ID and the database being recovered. Select the intended recovery point and keep a fresh checkpoint of the current database and relevant configuration. Decide how player sessions and new progress will be handled before applying old state.

## Work through the change

1. Read the installed HuskSync recovery and snapshot commands along with your database's supported restore procedure. Choose whether a targeted user snapshot or a wider database recovery is actually required.
2. Close the affected route to players and stop the relevant writers through normal controls. Confirm no backend or test copy still points at the production database.
3. Recover the chosen data using the supported database or snapshot procedure. Keep the pre-recovery checkpoint available privately and preserve the configuration mapping to each backend.
4. Start in a controlled order and test a disposable account before reopening admission. Check an intended recovered player record and a return transfer without creating duplicate sessions.

## Check the result

Every active writer should use the same intended recovered state, and new test data should persist normally. Record which recovery point was used and what later progress may need reconciliation. A database service starting successfully does not establish application-level consistency.

# If the result differs

If an old state reappears after a transfer, inspect another writer or cached active session before repeating the restore. If only one player is affected, a targeted snapshot recovery may be more appropriate than another network-wide rollback. Keep logs from the controlled startup.

Keep admission closed if consistency cannot be demonstrated and return to the fresh pre-recovery checkpoint only through the same coordinated process. Do not alternate database restores while players continue trading. Never delete the original checkpoint until the recovery has been reviewed.

> Create a backup from the CloudExa game panel
> A panel backup is separate from the plugin-specific record or remote archive discussed here.

# References and related help

- HuskSync maintainer reference 1
- HuskSync maintainer reference 2
- Test cross-server inventory synchronisation with HuskSync
- Back up game files before making changes

---

Revision #1
Created 2026-11-09 16:36:27 UTC by CloudExa Hosting Admin
Updated 2026-11-09 16:36:27 UTC by CloudExa Hosting Admin