Separate Redis failures from database failures in HuskSync

HuskSync has separate database and Redis configuration. A connection failure in one service does not prove the other is unavailable, and repairing one password does not automatically repair both. Compare status and the first relevant log message to identify the failed dependency.

Record the starting point

Record the failing backend, transfer timestamp and installed HuskSync release. Keep private copies of the database and Redis endpoint configuration. Redact passwords and usernames before sharing logs, and avoid changing several credentials at the same time.

Work through the change

1. Run authorised `/husksync status` and inspect startup or transfer-time errors. Identify whether the failing connection names the database, Redis, authentication or a timeout.
2. Compare the relevant configured host and port with the intended service. Localhost inside a game container refers to that container, not automatically to a separate database or Redis machine.
3. Ask support to check the specific source-to-destination path when a connection times out. Keep management services private; do not expose an unauthenticated Redis listener to every public address.
4. Correct the identified endpoint, access rule or credential through the documented configuration. Restart the affected component normally and repeat a controlled cross-backend transfer.

Check the result

Status and logs should show healthy dependencies, and the test account should transfer with the intended inventory state. Confirm a return transfer and a new saved snapshot where appropriate. A TCP connection alone proves reachability, not successful authentication or application operation.

If the result differs

An authentication error points to a different check from a timeout. A healthy database with failed messaging still warrants investigation of Redis; do not rewrite the database schema because the transfer stalls. Compare a working backend's redacted endpoint arrangement for useful differences.

Restore any ineffective trial endpoint change and keep the protected network boundary. Pause affected transfers if data consistency remains uncertain. Do not disable player locks, delete snapshots or reset shared services simply to suppress a connection error.

References and related help

- HuskSync maintainer reference 1
- HuskSync maintainer reference 2
- Test cross-server inventory synchronisation with HuskSync
- Find Minecraft logs and send useful troubleshooting details

---

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