

Database Failover Procedure

INTERNAL — Helix Engineering / CS Operations

OVERVIEW

This runbook describes how to manually fail over the primary Postgres database to its replica. This is a high-impact procedure; only execute under direction of an incident commander.

PRECONDITIONS

Verify that the replica is healthy and replication lag is under 5 seconds. Check the replication monitoring dashboard before proceeding.

STEP 1 — Stop writes

From the bastion: ``kubectl scale deployment/api-server --replicas=0``. This stops new writes from reaching the primary. Workflow runs in flight will fail; this is acceptable for the duration of failover.

STEP 2 — Wait for replication to catch up

Run ``psql -c 'SELECT pg_last_wal_receive_lsn(), pg_last_wal_replay_lsn();'`` on the replica until both values match. This confirms all writes have replicated.

STEP 3 — Promote replica

Run ``pg_ctl promote -D /var/lib/postgresql/data`` on the replica. The replica is now a primary.

STEP 4 — Update DNS

Update the ``db-primary.helix.io`` DNS record to point at the new primary's IP. TTL is 60 seconds; allow 90 seconds for propagation.

STEP 5 — Resume writes

Scale API server back up: ``kubectl scale deployment/api-server --replicas=20``. Monitor error rates closely for the next 15 minutes.

STEP 6 — Provision new replica

The promoted replica is now the primary; you have no replica. Provision a new replica from the latest snapshot. Until the new replica is in sync, you have no failover capacity.

ROLLBACK

If the failover fails or causes more impact, fail back: stop writes again, point DNS back to the original primary if it is still healthy. If the original primary is unhealthy, restore from the most recent backup

(RPO is 5 minutes).