Create an isolated FiveM staging copy with a separate database

A staging server is useful only when a test cannot damage the live community. A copied resource folder that still connects to production MySQL is not isolated. Use this checklist before testing framework migrations, paid resources or administrator tools. You need a separate allocated service and authority to copy the relevant data.

Define what may leave production

List the files, database tables and integrations required for the test. Use synthetic accounts where possible. If realistic data is necessary, use an authorised consistent export and minimise personal information. Do not copy raw player records to a developer's public repository or expose the test server as an unrestricted download source.

Record the production artifact, framework and resource versions. Capture a baseline backup before making the copy. Request separate connection allocations and a separate database with its own restricted user. Keep the production server online or stopped according to the documented consistency requirements of your export process, not according to convenience.

Remove live connections before first start

1. Upload the copied files while staging is stopped. Keep its startup configuration separate and confirm you are editing the test service.
2. Replace every database connection with the isolated database address and credentials. Review included configuration files and environment settings as well as the main configuration.

3. Replace or disable payment, reward, email, Discord and other webhook integrations. A test action should not deliver real purchases or notify every member of your community.
4. Give staging separate administrator credentials and appropriately authorised Cfx settings.

   Keep private values server-side; the convar reference explains why replicated or public information convars are unsuitable for secrets.
5. Confirm the txAdmin data path belongs to staging. Current txAdmin environment settings differ from older configuration methods, so have support verify host-managed paths instead of overriding them blindly.

Prove isolation

Before starting the framework, have the database administrator confirm the test account can access only the intended test database. Start staging and inspect its connection messages without printing passwords. Make a harmless test record through the application, then verify it appears only in staging. Do not test isolation by attempting a destructive production query.

Restrict joining with your supported access controls and give staff the test address directly. Do not rely on a friendly “TEST” name as access control. Keep a written check that no production-facing integration fired during the test. If an integration cannot be safely disabled, postpone that feature's test until an appropriate sandbox is available.

Run and reverse one change

Take a staging checkpoint, apply one update and follow its acceptance checklist. Record both failures and successful persistence checks. Restoring staging should use a matching code, configuration and database checkpoint together.

If isolation fails, stop staging immediately, preserve its logs and assess which external actions occurred before restarting it. Rotate credentials if copies escaped their intended boundary. Never promote the staging database back over production after players have continued playing; production rollout requires a fresh backup and the verified migration procedure applied to current data.

Related help

- Connect FiveM resources to a private database with oxmysql
- Install a FiveM resource and check its manifest dependencies
- Schedule game server backups and check retention

Revision #1
Created 2026-11-09 10:42:42 UTC by CloudExa Hosting Admin
Updated 2026-11-09 10:42:42 UTC by CloudExa Hosting Admin