Configure LiteBans moderation and preserve punishment history

LiteBans manages moderation records and enforcement. Begin with a clear policy for warnings, mutes and bans, then give staff the commands required for their role. An installation that accepts a ban command still needs testing for scope, expiry and appeals.

Choose the installation and data scope

Obtain a legitimate copy through the developer's official distribution and check its supported platforms. A proxy installation and an individual backend installation have different enforcement reach. Decide which instance should own commands and where the punishment data belongs before adding it across a network.

Stop affected instances and back up existing plugin files. If LiteBans already uses an external database, take a matching database export while writes are paused. Keep connection details private. A file backup alone cannot recover moderation history stored somewhere else.

Configure and verify storage

1. Install the compatible JAR in the selected instance and start once to generate configuration. Inspect Console for loading errors.
2. Stop before editing the generated configuration. Follow the installed version's comments for its selected storage driver, database and server identity.
3. Use a dedicated database account with the permissions the plugin documents. Do not reuse a panel administrator or database root account.
4. Start and confirm the plugin connects. Check the configuration points to the intended database rather than an empty test database.

5. For shared storage, test the intended local or network scope before enabling moderation for all staff.

Give moderators bounded access

Read the current permission reference rather than granting `litebans.*`. Separate routine warnings and history inspection from permanent bans, IP restrictions and administrative history changes. Where supported, apply duration limits to the appropriate roles and verify them with a non-operator test moderator.

For a read-only first check, run `/history TestPlayer` and `/checkban TestPlayer` in chat, replacing the name with your own test account. Console commands omit the slash. A command-name collision can route a request to another moderation plugin; identify the command owner before assuming the wrong system stored a punishment.

Test reversibility and expiry

During a private test window, use the installed release's documented syntax to apply a short temporary punishment to an account you control. Confirm its reason, duration and server scope, then remove that specific test punishment. Check the account can join or chat again and that the audit history remains understandable.

Do not use bulk history pruning or staff rollback as a routine setup test. IP matches can include shared households or networks; investigate them rather than automatically treating every associated account as the same person. Keep identifying information and appeal details out of public screenshots.

Before migrating storage, test the documented migration process against a copy and retain the original database. See plugin database connections and staff permission groups. Further reading: commands, permissions and official configuration reference.

Looking for another plugin? Browse the Minecraft plugin guide directory.

Revision #2
Created 2026-11-09 08:57:17 UTC by CloudExa Hosting Admin
Updated 2026-11-09 08:57:18 UTC by CloudExa Hosting Admin