Configure DriveBackupV2 and verify an off-server recovery copy

DriveBackupV2 can send server file archives to supported storage services. It is a separate backup system from the CloudExa panel's Backups section: its schedules, retention, credentials and failures need their own checks. A remote account being linked is not proof that a usable archive exists.

Choose the recovery scope first

List the worlds, plugin data and configuration needed to recover the server. External MySQL databases require a separate consistent export; including a plugin's configuration file does not include its remote database. Plan a quiet window when a consistent recovery point matters. DriveBackupV2 runs inside the Minecraft process: its commands need that process running. For a fully offline backup, stop Minecraft and use the CloudExa panel backup system or another offline tool instead; do not try to run a plugin command after stopping the process.

Check the plugin's supported server release and obtain the official build. Back up the current installation before adding it. Start once to generate configuration, inspect Console, then stop before editing settings.

Prevent archives from including themselves

Review the generated `backup-list`, `local-save-directory`, remote destination and retention settings. Choose explicit inputs rather than blindly archiving the whole server root. Keep local output and old archives outside every included input tree. If you include the plugins directory, check that no plugin stores its backup output underneath that same directory.

A backup that contains earlier backups grows rapidly and can fill the game server before the upload finishes. Leave storage room for temporary archives as well as normal world saves. Multiple backup plugins or panel schedules can compete for that space, so avoid overlapping jobs until you have measured them.

Link one storage destination securely

1. Choose a supported provider and follow the plugin's current setup guide for that provider. Use an account or bucket dedicated to server recovery where practical.
2. Keep authorisation codes, tokens, access keys and configuration files private. Use the provider's supported encrypted connection method.
3. Enable only the intended destination. Set bounded remote retention and review local retention independently.
4. Start the server and run the installed version's documented provider test. Read the response and the provider's own storage view.
5. Run `drivebackup backup` in Console for a supervised first job. Check `drivebackup status` and wait for successful completion.

Prove that the remote copy can be recovered

Find the completed archive at the destination, download it to a separate location and inspect its contents. Confirm the expected world and plugin data are present and old archives are absent. Keep the backup private: it may contain passwords or service tokens. Provider storage encryption does not make a publicly shared download safe.

Restore a sample onto an isolated test server, including matching database exports when needed. Confirm the world opens and representative player data works. Do not extract over the active production world merely to check whether an archive is readable.

Schedule only after the first test passes

Check the next backup time, timezone and any requirement for players to be online. Review failure messages and remaining storage regularly. A scheduled request can fail after starting, and fixed delays do not establish consistency.

See panel backup and restore checks and full game-server storage. Further reading: provider setup guides, commands and 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