Lands tasks

Task guides and troubleshooting for Lands. Check the installed release before changing configuration or data.

- Create a Lands subarea with a deliberate height boundary
- Allow Lands bank deposits while restricting withdrawals
- Check Lands taxes before changing the economy provider
- Preview Lands inactive-owner expiration without removing claims
- Test a Lands rental extension and maximum rental time
- Check a Lands claim refusal without raising every player limit
- Prevent duplicate Lands economy collection on a Redis network
- Verify a Lands database backup on an isolated server

Create a Lands subarea with a deliberate height boundary

A room-sized subarea is different from a normal chunk claim. Define its vertical range deliberately so a tenant’s access does not extend into storage above or below the room.

Before you start

Lands separates land membership, areas and role flags. Select the intended land before editing it, and verify with an ordinary player rather than an administrative bypass. Use the documentation matching your release when menus or permission settings differ.

Work through the check

1. Stand in the intended land and use the current release’s `/lands edit here` control, then record the existing area list and room corners.
2. Use `/lands selection` and select the two corners, including the intended floor and ceiling heights. Do not expand to full height unless that is the policy you need.
3. Assign the selection through `/lands area <area> assign`, replacing the placeholder with a new distinctive area name, and inspect the area in the menu.
4. Trust a normal test player in that area only, then test the intended interaction inside the room and just above, below and outside its measured boundary.

Confirm the result

The test player should receive only the area’s intended access within the selected bounds. Adjacent rooms and floors should retain their original role rules.

If the result differs

Older releases use different command targeting. If the selection affects another land, stop and verify the edit land and area name; existing area names can cause assignment to reshape an area rather than create a separate one.

Recover safely

Restore the recorded original bounds if an existing area was reshaped. Remove only the temporary area membership after testing; do not unclaim the surrounding chunks or delete the land.

Documentation and related help

Check the Lands maintainer documentation and Lands maintainer reference against your installed release before applying changes.

Related CloudExa help: Create a Lands claim and control member roles and Find Minecraft logs and send useful troubleshooting details.

Allow Lands bank deposits while restricting withdrawals

Members may need to fund upkeep without being able to empty the land bank. Check deposit and withdrawal role flags independently of general membership or building access.

Before you start

Lands separates land membership, areas and role flags. Select the intended land before editing it, and verify with an ordinary player rather than an administrative bypass. Use the documentation matching your release when menus or permission settings differ.

Work through the check

1. Select the intended land and record its bank balance, the test member’s role and the role’s existing bank permissions.
2. Confirm land banks are enabled in the installed configuration and identify the documented BALANCE_DEPOSIT and BALANCE_WITHDRAW flags for your release.
3. Allow only the intended deposit behavior for the test role, keeping withdrawal restricted, then use `/lands bank deposit <amount>` with a small affordable amount.
4. Check `/lands bank balance` and the player’s wallet, then attempt a small withdrawal as that normal member and verify that the restriction holds.

Confirm the result

The member should be able to contribute the recorded amount without gaining withdrawal access. The land bank credit and personal debit should agree for the deposit.

If the result differs

A different selected land or administrative bypass can make a permission test misleading. Check effective role priority and inherited permissions before changing every member role.

Recover safely

Restore the original role flag if the test grants excess access. An authorized bank manager can reverse the documented test deposit where appropriate; do not edit bank balances directly in the database.

Documentation and related help

Check the Lands maintainer documentation and Lands maintainer reference against your installed release before applying changes.

Related CloudExa help: Create a Lands claim and control member roles and Find Minecraft logs and send useful troubleshooting details.

Check Lands taxes before changing the economy provider

Lands supports more than one economy mode, but taxes have provider requirements. Verify the active provider before enabling collections or interpreting a missing tax payment as player debt.

Before you start

Lands separates land membership, areas and role flags. Select the intended land before editing it, and verify with an ordinary player rather than an administrative bypass. Use the documentation matching your release when menus or permission settings differ.

Work through the check

1. Record the active integration.economy configuration and the provider selected at startup, including any fallback to item or experience currency.
2. Read the current Lands economy documentation: taxes require Vault economy, so keep tax collection disabled when using item or experience economy.
3. On an isolated test server with a supported Vault provider, record a member’s money, the target area’s tax and the collection schedule.
4. Check the member’s displayed tax information and simulate or observe a documented test collection only in that isolated environment, noting debits and bank credits.

Confirm the result

A supported tax configuration should collect the expected amount through the intended money provider. Item or experience economy should not be treated as an interchangeable tax backend.

If the result differs

A displayed currency symbol does not prove the correct provider is active. Check the hook and current balances before changing collection amounts to compensate for a failed integration.

Recover safely

Restore the previous provider configuration and keep collection disabled if the test fails. A provider migration needs a separate balance plan; do not convert every account while investigating one tax message.

Documentation and related help

Check the Lands maintainer documentation and Lands maintainer reference against your installed release before applying changes.

Related CloudExa help: Create a Lands claim and control member roles and Find Minecraft logs and send useful troubleshooting details.

Preview Lands inactive-owner expiration without removing claims

Expiration can transfer, sell or delete land. Use the documented preview mode to review the affected owners before enabling any real cleanup policy.

Before you start

Lands separates land membership, areas and role flags. Select the intended land before editing it, and verify with an ordinary player rather than an administrative bypass. Use the documentation matching your release when menus or permission settings differ.

Work through the check

1. Take a database backup and record the installed Lands release, existing expiration settings and any scheduled collection or cleanup jobs.
2. For releases with the documented land.expiration status system, keep status set to preview and verify that value in the active configuration before running a check.
3. Use `/lands admin expiration list` and the documented preview run control, recording the proposed owner outcomes and member removals without switching to active.
4. Review unknown login data, network activity and playtime tiers against a few known active and inactive owners, then revise the policy on the test copy.

Confirm the result

The check should report proposed outcomes while leaving ownership, trust and land protection unchanged. The review must account for activity on other servers sharing the Lands database.

If the result differs

If the release does not provide the documented preview system, stop and consult its matching documentation. Never assume a similarly named cleanup command is a dry run.

Recover safely

Leave expiration off or in verified preview while investigating unexpected candidates. Keep the database checkpoint; deleted land cannot be reconstructed reliably from a screenshot or a configuration file alone.

Documentation and related help

Check the Lands maintainer documentation and Lands maintainer reference against your installed release before applying changes.

Related CloudExa help: Create a Lands claim and control member roles and Find Minecraft logs and send useful troubleshooting details.

Test a Lands rental extension and maximum rental time

A rental payment adds a configured period, but an owner can limit how far ahead tenants pay. Verify the duration and cap before advertising a long-term room rental.

Before you start

Lands separates land membership, areas and role flags. Select the intended land before editing it, and verify with an ordinary player rather than an administrative bypass. Use the documentation matching your release when menus or permission settings differ.

Work through the check

1. Select a small test subarea and record its rental offer, price, duration, maximum time and current tenant status.
2. Read `/lands rental info` while standing in that area and confirm the intended land is selected before editing the offer through the documented menu.
3. Rent with an ordinary test account using a low price, then extend through the supported sign or hologram interaction and compare remaining time and balance.
4. Try an extension at the configured limit and confirm the refusal does not debit money or affect unrelated rented areas.

Confirm the result

The tenant should gain only the paid period up to the permitted maximum. The owner and tenant should see consistent remaining time for the same area.

If the result differs

Time without an explicit unit may be interpreted differently from what the owner intended. Review the documented offer format and current saved values instead of repeatedly clicking and assuming every payment succeeded.

Recover safely

Remove a temporary offer or cancel the test rental through the documented rental controls when finished. Check compensation rules before removing a real tenant, and preserve the area and its existing blocks.

Documentation and related help

Check the Lands maintainer documentation and Lands maintainer reference against your installed release before applying changes.

Related CloudExa help: Create a Lands claim and control member roles and Find Minecraft logs and send useful troubleshooting details.

Check a Lands claim refusal without raising every player limit

A failed claim can come from world policy, capacity, distance or cost. Read the precise refusal and selected land before changing global limits that affect the entire server.

Before you start

Lands separates land membership, areas and role flags. Select the intended land before editing it, and verify with an ordinary player rather than an administrative bypass. Use the documentation matching your release when menus or permission settings differ.

Work through the check

1. Record the world, chunk location, selected land, current claim count and exact error shown to the normal player.
2. Compare the installed claim and world settings with the player’s effective claim-limit permissions and any documented rank modifiers.
3. Check the bank and personal balance against the quoted cost, including configured cost increases and fallback-personal behavior where enabled.
4. Repeat with a deliberately chosen empty nearby chunk on an isolated copy, changing only the specific documented setting identified by the refusal.

Confirm the result

The controlled claim should succeed only when all intended conditions are met, with the expected owner and cost. A rejected attempt should not change existing land boundaries.

If the result differs

Administrative bypasses, free-chunk allowances or another selected land can hide the original restriction. Test as the affected player group rather than making the account an operator.

Recover safely

Restore the specific test limit or world rule if the result is too broad. Leave existing claims intact; raising every permission limit or unclaiming a player’s base is unnecessary to diagnose one rejected chunk.

Documentation and related help

Check the Lands maintainer documentation and Lands maintainer reference against your installed release before applying changes.

Related CloudExa help: Create a Lands claim and control member roles and Find Minecraft logs and send useful troubleshooting details.

Prevent duplicate Lands economy collection on a Redis network

A shared Lands network needs one configured collection authority. Check the documented Redis master role before interpreting repeated upkeep or tax charges as an economy balance bug.

Before you start

Lands separates land membership, areas and role flags. Select the intended land before editing it, and verify with an ordinary player rather than an administrative bypass. Use the documentation matching your release when menus or permission settings differ.

Work through the check

1. List the Lands instances that share the same database and Redis setup, recording versions, timezone and each server’s collection settings.
2. Read the current network economy guidance and identify which instance has database.redis.master configured for collection authority.
3. Compare collection timestamps and bank changes across two instances without manually triggering another production collection.
4. On an isolated network copy, configure the intended single master according to the release documentation and observe one scheduled collection while checking both instances’ logs.

Confirm the result

One intended collection should affect shared balances once, while other instances continue to show consistent data. The test should not create competing jobs for the same tax or upkeep period.

If the result differs

A network with separate databases is not the same as shared Redis synchronization. Confirm the actual storage connections before changing master settings or assuming similarly named lands share balances.

Recover safely

Restore the recorded network settings if synchronization stops on the test copy. Keep production collections paused only through a planned supported procedure while investigating duplicates, and reconcile confirmed charges from logs rather than resetting banks.

Documentation and related help

Check the Lands maintainer documentation and Lands maintainer reference against your installed release before applying changes.

Related CloudExa help: Create a Lands claim and control member roles and Find Minecraft logs and send useful troubleshooting details.

Verify a Lands database backup on an isolated server

A successful backup message is useful, but a restore test proves whether claims and bank data can be recovered. Keep that test isolated because the documented restore replaces the current dataset.

Before you start

Lands separates land membership, areas and role flags. Select the intended land before editing it, and verify with an ordinary player rather than an administrative bypass. Use the documentation matching your release when menus or permission settings differ.

Work through the check

1. Check the installed release supports `/lands admin database backup`, then create a checkpoint and locate the resulting backup under the documented plugin backup directory.
2. Save the plugin configuration, release information and world references alongside the archive, keeping database passwords private and exporting a protected copy off the game node.
3. Prepare an isolated server with its own storage connection and use the documented restore flow there only, reading its restart and overwrite warning before confirmation.
4. After the test restart, compare several known claims, owners, subareas and bank balances with the checkpoint record, then test a harmless ordinary-player access rule.

Confirm the result

The isolated restore should contain the expected protection and account state from that checkpoint. Neither its startup nor its test actions should write into production storage.

If the result differs

A missing backup filename or backend mismatch needs investigation before confirmation. Do not use an administrative restore command on the live server as a quick way to inspect an archive.

Recover safely

Keep the original backup immutable and rebuild the isolated test when repeating recovery. Production rollback needs a separate maintenance plan to avoid replacing newer claims and transactions.

> CloudExa game-panel controls for creating a backup.
> The game-panel backup controls. Confirm separately whether the chosen backup includes an external plugin database; this screenshot does not show a plugin-specific backup tool.

Documentation and related help

Check the Lands maintainer documentation and Lands maintainer reference against your installed release before applying changes.

Related CloudExa help: Create a Lands claim and control member roles and Back up game files before making changes.