# Floodgate tasks

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

- Plan a Floodgate username-prefix change without splitting player identity
- Check player progress before enabling Floodgate global account linking
- Allowlist a Floodgate player using the identity the server actually sees
- Give a Bedrock player a rank without relying on a display name
- Replace a leaked Floodgate key across the trusted proxy path
- Diagnose missing Bedrock skins on one backend server

# Plan a Floodgate username-prefix change without splitting player identity

A Floodgate name prefix helps distinguish Bedrock names from Java names. Changing it on an established server can confuse plugins that store names, permission targets or display labels. It is an identity migration, even if the player-facing change looks cosmetic.

## Record the starting point

Record the present prefix and pick one consenting Bedrock tester with known homes, claims and permissions. Note the tester's UUID from a trusted server record. Back up the world and identity-dependent plugin data, including their databases where applicable.

## Work through the change

1. Inspect which installed plugins use UUIDs and which rely on player names or commands containing names. List the rank, claim, economy and teleport integrations the tester actually uses.
2. Trial the prefix change on a private server copy using the installed Floodgate release's documented configuration. Do not change authentication type or account-linking policy during the same trial.
3. Join with the same Bedrock account and compare its UUID, inventory, balance, homes and claim membership. Rejoin with a Java account whose display name could be confused with the new format.
4. Update only the name-based integration entries that the documented migration requires. Keep a mapping from old visible names to verified UUIDs so staff do not create replacement accounts by guesswork.

## Check the result

The same person should retain the same intended data and permissions, while staff can still distinguish identities. Check console commands and moderation searches as well as chat appearance. Record any plugin that needs a separate migration before scheduling the live change.

# If the result differs

Missing data does not prove it was deleted. Stop the trial and inspect the UUID and account-linking state before creating a new record or copying inventories. An online-mode change can introduce a second identity problem that hides the original prefix issue.

Return to the saved prefix and configuration if any identity-dependent feature fails. Restore affected plugin data only from a matching checkpoint if the trial wrote incompatible records; do not merge player folders or database rows without a documented migration.

# References and related help

- Floodgate maintainer reference 1
- Floodgate maintainer reference 2
- Set up Geyser and Floodgate for Java and Bedrock crossplay
- Back up game files before making changes

# Check player progress before enabling Floodgate global account linking

Floodgate global linking can let a player use their Java identity when joining from Bedrock. That changes which existing data they see. A player who links first and checks inventories later can mistake their old Bedrock progress for a deletion.

## Record the starting point

Ask the player to identify both accounts and the progress they need to retain. Keep a recovery checkpoint of the server and relevant plugin databases. Do not ask them to send passwords, linking codes or Microsoft authentication tokens in a public ticket.

## Work through the change

1. Read the current official linking instructions together. Explain that linked sessions use the Java account data described there, while the separate Bedrock data does not simply merge into it.
2. Before linking, have the player record and arrange the items or ownership they need to transfer using your normal supervised server rules. Check ender storage, equipped items, claims and economy records.
3. Let the account owner complete the official linking flow themselves. Staff should observe the resulting identity through normal server records rather than impersonating either account.
4. After linking, join from Bedrock and verify the intended Java identity, permissions and progress. Repeat the test after reconnecting, and confirm the player understands the effect of unlinking.

## Check the result

The player should see the expected linked identity and a documented account of what moved through supervised gameplay or plugin tools. A link is not a generic inventory-merging utility. Keep records of any staff-assisted transfer to prevent later duplicate-item disputes.

# If the result differs

If progress appears missing, compare the active UUID and linking status before restoring anything. The official documentation explains how unlinking can make the separate Bedrock data accessible again. Avoid importing both data sets into one record without a tested per-plugin method.

Pause further account changes while the owner and staff review the checkpoint. Use the official unlinking route when appropriate, then reassess the identities. Restore live data only after confirming which account and timestamp the recovery must target.

# References and related help

- Floodgate maintainer reference 1
- Set up Geyser and Floodgate for Java and Bedrock crossplay
- Recover a Minecraft player inventory from a backup

# Allowlist a Floodgate player using the identity the server actually sees

The name typed into a Bedrock client is not enough to identify every server-side account. Prefixes, linked accounts and plugin allowlists can affect the name or UUID the backend uses. Resolve the actual identity before adding access.

## Record the starting point

Keep an existing staff session or console route available so the allowlist cannot lock you out. Ask the player for the account name they use, then verify it against a trusted connection attempt or an existing record. Keep private identifiers out of public screenshots.

## Work through the change

1. Identify whether the server uses the built-in allowlist or a separate admission plugin. Read that tool's support for Floodgate and UUID entries before choosing its command.
2. Check whether the player has linked a Java account. Record the prefixed Bedrock identity or linked Java identity that this server sees instead of guessing from an old chat nickname.
3. Add the single verified identity through the documented admission tool. Leave authentication enabled and keep unrelated allowlist entries unchanged during the test.
4. Ask the player to reconnect from outside your network. Also verify that a different unlisted test account remains unable to join; a successful allowed join alone does not prove admission is restricted.

## Check the result

The intended account should pass admission and retain its existing inventory and rank. An unlisted account should still fail at the admission step. Keep the record tied to the verified UUID when the admission tool supports it, particularly if visible names may change.

# If the result differs

If the allowed player still fails, read the exact rejection and compare the identity at the proxy and backend. A proxy allowlist and a backend allowlist can enforce separate checks. Do not disable both permanently to hide an identity mismatch.

Remove only the trial entry if it targeted the wrong identity, then restore the recorded admission policy. If the player appears as a new account, investigate linking or forwarding before transferring data or granting administrator access.

# References and related help

- Floodgate maintainer reference 1
- Floodgate maintainer reference 2
- Set up Geyser and Floodgate for Java and Bedrock crossplay
- Manage Minecraft operators, permissions and the whitelist

# Give a Bedrock player a rank without relying on a display name

A rank can appear correct in chat while permission checks target a different account record. Floodgate prefixes, nicknames and global linking make name-only staff workflows fragile. Use the identity your permission plugin records for the actual login.

## Record the starting point

Choose one ordinary Bedrock tester and one harmless permission to check. Record the existing primary group, UUID and linking state. Keep a copy or export of the permission data before changing a live network role.

## Work through the change

1. Have the player join through the normal public route, then find their account in the permission plugin using its documented lookup or editor. Compare the UUID with trusted login records.
2. Assign the intended group to that account only. Do not add every spelling of their name or grant a wildcard permission in an attempt to cover all possible identities.
3. Test the selected command as the Bedrock player and check one command the rank should still deny. If linking is enabled, compare the Java account's expected role before changing it.
4. Reconnect and, on a network, transfer between two intended backend servers. Confirm the same permission record applies under the configured forwarding and storage arrangement.

## Check the result

The player should gain the intended command and remain unable to perform unrelated staff actions. Check permission output rather than relying on a prefix in chat. Record whether the group belongs to a standalone Bedrock identity or the linked Java identity.

# If the result differs

A different result after switching servers suggests inconsistent permission storage, context or identity forwarding. Compare account UUIDs first. Changing the Floodgate prefix to match a permission record can create additional name-based problems without fixing the underlying identity.

Remove the trial group from the single target record if the test selected the wrong account. Use the saved permissions checkpoint only when a wider change needs recovery; keep normal player progress and unrelated groups untouched.

# References and related help

- Floodgate maintainer reference 1
- Floodgate maintainer reference 2
- Set up Geyser and Floodgate for Java and Bedrock crossplay
- Create Minecraft ranks and inheritance with LuckPerms

# Replace a leaked Floodgate key across the trusted proxy path

Floodgate's key is an authentication credential. If it appears in a public download, screenshot or repository, removing the public file alone does not make the old key trustworthy. Plan a coordinated replacement for every component that must share it.

## Record the starting point

Identify the proxy, Geyser deployment and backend Floodgate installations that belong to this network. Confirm which components need the key under the official setup for their platforms. Keep the existing configuration and a complete recovery checkpoint in restricted storage.

## Work through the change

1. Close the relevant maintenance window to player logins and stop the affected components through their normal controls. Avoid a rolling half-update that leaves some servers trusting the old key.
2. Follow the current Floodgate release instructions to produce the replacement key and distribute it only to the required trusted components. Use private file transfer, never a public paste or repository.
3. Check the documented forwarding settings on the proxy and backend. The key does not replace Velocity or Bungee forwarding configuration, and backend admission still needs the protected proxy path.
4. Start the network in its planned order, then test a Bedrock login and a Java login. Confirm that a normal player retains their identity and can transfer to each intended backend.

## Check the result

The new shared credential should work on the trusted path, while the exposed credential no longer remains in active configuration. Check every backend, not just the first lobby. Record the rotation date without recording the secret in a public change log.

# If the result differs

A failure on only one backend often means that component retained the old key, used the wrong folder or did not restart. Inspect redacted errors and configured file locations. Do not solve the mismatch by enabling unauthenticated direct backend access.

If a coordinated restart fails, keep admission closed while correcting the new deployment. Reusing the exposed key is not a sound long-term rollback. Preserve player data; a credential rotation does not require deleting accounts, worlds or inventories.

> CloudExa game panel file manager
> Use the file manager for the specific configuration file described in this guide; the screenshot shows the panel, not a plugin interface.

# References and related help

- Floodgate maintainer reference 1
- Set up Geyser and Floodgate for Java and Bedrock crossplay
- Set up a Velocity network with protected backend servers

# Diagnose missing Bedrock skins on one backend server

A Bedrock player may authenticate through the proxy while a backend lacks the Floodgate support needed by skin-aware integrations. A missing skin on one backend therefore does not prove the player's Microsoft account or stored inventory is broken.

## Record the starting point

Record which backend shows the problem, the Geyser and Floodgate versions and any skin plugin involved. Use the same Bedrock tester across servers. Save the affected backend and proxy configuration before changing the trusted forwarding path.

## Work through the change

1. Check whether the official Floodgate setup for your proxy calls for backend installation to expose the API or display Bedrock skins. Compare the affected server with a working backend rather than installing unrelated skin plugins.
2. Verify the required backend Floodgate file and the documented proxy data-forwarding setting. Check that its credential belongs to this trusted network, without posting the key in a support ticket.
3. Restart the changed component during a quiet window and transfer the tester through the normal proxy route. Do not use a direct backend login as a substitute for testing forwarded identity.
4. Check the skin, UUID and a harmless permission on both backends. If another skin plugin overrides appearance, trial the conflict on a private copy instead of changing account authentication.

## Check the result

The same Bedrock account should retain the same identity while the supported backend shows its expected skin. Reconnect once to distinguish a temporary client cache from a repeatable server

difference. Record the exact backend versions that pass.

# If the result differs

If skins remain inconsistent but UUIDs and login are correct, inspect integration compatibility and skin override rules. If identity also differs, stop the trial and investigate forwarding first. Appearance changes should not cause staff to merge or rename player data.

Restore the saved configuration for the trial component if it introduces a new login problem. Keep the protected proxy route and authentication policy intact. Escalate with redacted logs and the working-versus-failing backend comparison rather than replacing every plugin at once.

# References and related help

- Floodgate maintainer reference 1
- Floodgate maintainer reference 2
- Set up Geyser and Floodgate for Java and Bedrock crossplay
- Configure SkinsRestorer without changing account authentication