Separate allocation pressure from a Minecraft memory leak

A high allocation rate does not necessarily mean memory is leaking. Java can create many short-lived objects and reclaim them, while a retention problem keeps more live data over time. Compare the measurements before increasing heap or deleting plugin data.

Record the starting point

Record the configured heap, player count, uptime and the time of the reported pressure. Check the service's memory limit separately from Java heap usage. A container limit, native memory and the Java heap are not identical numbers.

Work through the change

1. Collect the relevant spark memory and garbage-collection metrics during ordinary play. Note whether usage falls after collection or whether the post-collection baseline keeps rising.
2. If allocation work is the question, check your installed spark commands and take a short allocation profile using the documented `--alloc` option. Keep it labelled as allocations rather than retained objects.
3. Compare the capture with a similar period after a single suspected plugin change on a test copy. Keep player count and activity comparable.
4. Escalate suspected retained-heap analysis to support with timestamps and measurements. Treat heap dumps as sensitive and potentially large; do not upload one publicly without reviewing its contents and handling requirements.

Check the result

The evidence should show whether pressure follows temporary churn, a growing retained baseline or the service limit. A candidate plugin appearing in an allocation profile indicates object creation, not proof that it retains those objects forever. Keep the distinction clear in a support report.

If the result differs

If the process is killed before Java can report an error, include the panel memory graph and host-side evidence available from support. A flat heap chart does not rule out native or host pressure. Avoid assuming every abrupt exit is a Java leak.

Stop the diagnostic recording and undo an ineffective trial change. Keep ordinary backups before any later data migration, but do not delete worlds or caches solely because they are large. Adjust RAM only after identifying which limit is actually being reached.

References and related help

- spark maintainer reference 1
- spark maintainer reference 2
- Diagnose Minecraft lag with TPS, MSPT and spark
- Find Minecraft logs and send useful troubleshooting details

---

Revision #1
Created 2026-11-09 16:36:26 UTC by CloudExa Hosting Admin
Updated 2026-11-09 16:36:26 UTC by CloudExa Hosting Admin