Save a spark profile locally before sharing it

A performance report can disclose server software, plugin names, paths or other operational details. Spark supports local output for supported reporting commands. Saving locally lets staff inspect a capture before deciding which information should leave the service.

Record the starting point

Check the command help for the installed spark build and identify where the service can store diagnostic files. Record the symptom and test window. Ensure there is enough free space for the capture and that the destination is not a publicly downloadable web folder.

Work through the change

1. Start a bounded profile for the actual symptom and record the triggering action. Avoid collecting a long idle session when a short reproduction is available.
2. Use the documented local save option when stopping the capture, such as `/spark profiler stop --save-to-file`. Read the returned location rather than guessing a filename.
3. Download the saved file through the service's authorised file access. Review its metadata and follow the viewer's supported local-file workflow if you need to inspect the profile without publishing it.
4. Share only with the intended support recipient and state which details were removed or withheld. If an upload is necessary, review its visibility and retention before sending the link.

Check the result

You should retain a useful capture with a known private location and an explicit sharing decision. Confirm that the file exists and can be opened through the supported viewer process. A console message alone is not evidence that a downloaded archive is complete.

If the result differs

If the command reports an unsupported flag, check the installed version's help instead of pasting options repeatedly. If saving fails, inspect free space and permissions for that service. Do not make an entire plugin directory publicly readable to retrieve one report.

Cancel the capture if local saving cannot be completed safely, preserving the relevant logs. Remove temporary diagnostic copies only after the investigation's retention needs are understood. Rotate any exposed secret if a report was shared before review.

> 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

- 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