# Diagnose ARK Survival Evolved startup and joining failures without wiping

Separate an Evolved server that never finishes starting from one that starts successfully but refuses a player. Wiping a world cannot repair a missing network allocation or incompatible client. Keep the failure evidence and current save while finding which stage breaks.

## 1. Capture one attempt

Record the UTC time, client edition and storefront, server build, selected map and exact error. Save the first relevant startup error, not only the final shutdown line. If the process remains running, avoid repeatedly pressing start: loading a large world or completing a mod update can take time, and repeated interventions obscure the original event.

A panel websocket failure is a separate symptom. If the console itself cannot connect, report that alongside the game status instead of assuming a blank console proves the game stopped.

## 2. Check startup before networking

Review the existing startup configuration for a changed map, missing content or a mistyped file path. Confirm that the installation is ARK: Survival Evolved, not Ascended. Compare the current configuration with your last working copy and any recent changes. Do not replace all INI files with a template from another game.

Check available storage and the timestamps of the current configuration. A failed update or inability to save needs investigation before more restarts. Keep the active save directory, player data and mod data together in your recovery copy; do not remove unfamiliar files to make an error disappear.

# 3. Distinguish discovery from gameplay

The Evolved reference setup uses a UDP game port, an associated UDP peer port at game port plus one, and a separate UDP query port for Steam's browser. RCON uses TCP and is not a player's joining endpoint. Compare the actual panel allocations and startup mappings, not just the familiar default numbers.

If the server appears in a browser but joining times out, query visibility has proved only part of the path. If it is absent from the browser but the supplied direct connection method works, focus on discovery, filters and the query mapping. Do not disable the firewall or publish every port as a diagnostic shortcut.

# 4. Compare client and content compatibility

Check that the player is launching the correct ARK title and a compatible build. Record the configured mod IDs and their order before comparing them with the client's required content. A modded community may have additional storefront restrictions; verify those against the exact deployment rather than promising crossplay from a server name.

Use an isolated test service if you need an unmodded comparison. Removing mods from the only production copy and loading its world can affect mod-provided content. A controlled test must use separate save data and a separate assigned endpoint.

# 5. Verify and escalate

After one identified correction, make one new connection attempt and compare its timestamp with the server log. Confirm the expected world loads and that an existing player retains the expected character and ownership. Successful process startup alone is incomplete verification.

Send CloudExa support the service ID, failure stage, client platform, server build and redacted relevant log lines if the cause remains unclear. State which changes preceded the incident. Do not delete the world or reinstall while support is correlating the error with its network and startup configuration.

# Related CloudExa guides

- Protect game files before changing settings
- Follow the game connection checks

# Official references

- ARK Official Community Wiki: dedicated server network requirements
- ARK Official Community Wiki: Evolved and Ascended configuration distinctions

---

Revision #1
Created 2026-11-10 10:57:20 UTC by CloudExa Hosting Admin
Updated 2026-11-10 10:57:20 UTC by CloudExa Hosting Admin