FREE RESOURCE

MCP Stack Readiness Checklist

Choose, review, and deploy your MCP stack with confidence.

An approval framework for MCP servers headed to shared or production use. Every item includes what a **healthy** answer looks like and the **red flag** that should pause the review. Reviewing a server just for yourself? Use the companion one-pager, *MCP Server Vetting in 10 Minutes*.

---

01 Purpose and ownership

Confirm why the server is needed and who remains accountable after deployment.

COMMON FAILURE A server is added for a one-off demo. The demo ships. Eighteen months later, nobody can say who owns the credential it still holds.

- The server solves a defined workflow problem that cannot be handled more safely or simply another way.

  HEALTHY The problem statement fits in two sentences and names the workflow, the data touched, and the expected frequency of use.

  RED FLAG The justification is "the team wanted to try it," or an existing approved server already covers the job.

  NOTES . . . . . .

- A named technical owner is responsible for configuration, updates, access, and incident response.

  HEALTHY One person (not a team alias) is named, with a deputy for absence. Ownership survives reorgs via the review date in section 08.

  RED FLAG Ownership is assigned to "platform team" with no individual, or the owner does not know the server is theirs.

  NOTES . . . . . .

- Expected users, connected systems, and permitted actions are documented before rollout.

  HEALTHY A one-page scope note lists user groups, systems reached, and the top 5 to 10 permitted tool actions.

  RED FLAG The permitted-action list is discovered by reading the server's source code after deployment.

  NOTES . . . . . .

- A removal or rollback plan exists if the server becomes unreliable or unnecessary.

  HEALTHY Removal is a config change plus credential revocation, tested once, documented in under half a page.

  RED FLAG Removing the server would break a workflow that nobody has mapped.

  NOTES . . . . . .

02 Maintenance and provenance

Verify that the software is active, attributable, and supportable.

COMMON FAILURE A lookalike package with a near-identical name is installed from a registry instead of the official publisher's repository. Typosquatting is an active, documented attack vector for MCP servers.

- The repository, publisher, package name, and installation source are verified end-to-end.

  HEALTHY Publisher identity matches across the registry listing, the repo, and the vendor's own docs. The install command is copied from the vendor, not a blog.

  RED FLAG The package is a fork, a mirror, or has a name one character away from a popular server.

  NOTES . . . . . .

- Release history and recent maintenance activity are acceptable for the intended risk level.

  HEALTHY For production: a release or meaningful commit within the last 90 days, and an issue tracker where maintainers respond.

  RED FLAG No release in 6+ months for a server touching production data, or the sole maintainer has publicly stepped away.

  NOTES . . . . . .

- Open security issues, abandoned dependencies, and unresolved critical bugs have been reviewed.

  HEALTHY Open security issues are zero, or triaged with a written accept/deny decision. A dependency audit runs clean.

  RED FLAG Security issues sit open with no maintainer response for 30+ days.

  NOTES . . . . . .

- Version pinning and a controlled update process are in place.

  HEALTHY The deployed version is pinned (exact version or digest). Updates go through the same review as the original approval.

  RED FLAG The config pulls "latest" and every upstream release silently becomes your production code.

  NOTES . . . . . .

03 Authentication and permissions

Apply the smallest practical access scope.

COMMON FAILURE A server that only reads calendar events is granted a full-workspace admin token because that was the token someone already had.

- The authentication method and secret-storage location are documented.

  HEALTHY OAuth with short-lived tokens where the server supports it. Secrets live in a managed secret store, referenced by environment variable.

  RED FLAG A long-lived personal API key is pasted directly into a JSON config file.

  NOTES . . . . . .

- Tokens and service accounts use least-privilege, read-only-by-default permissions.

  HEALTHY Scopes map 1:1 to the permitted actions in section 01. Each write scope is individually justified in the notes.

  RED FLAG A broad or admin scope is used "to be safe." That is the opposite of safe.

  NOTES . . . . . .

- Production credentials are absent from source control, shared drives, and chat history.

  HEALTHY A secret-scanning hook or scheduled scan covers the repos and config paths this deployment uses.

  RED FLAG The setup guide tells users to commit a config file that contains the token.

  NOTES . . . . . .

- Credential rotation, revocation, expiry, and offboarding procedures are defined.

  HEALTHY Rotation is scheduled (90 days is a common ceiling for static keys). Revocation on offboarding is part of the leaver checklist.

  RED FLAG Nobody can answer "what do we revoke if this laptop is stolen tonight?" within one working hour.

  NOTES . . . . . .

04 Security boundary

Understand what the server can read, write, execute, and transmit.

COMMON FAILURE A filesystem server is scoped to the drive root instead of the project directory. The model can now read SSH keys, browser profiles, and every document on the machine.

- Accessible files, databases, APIs, tools, and network destinations are explicitly listed.

  HEALTHY Filesystem access is scoped to named directories. Network egress is limited to named hosts.

  RED FLAG The honest answer to "what can it reach?" is "everything the host user can."

  NOTES . . . . . .

- Write, delete, execute, payment, messaging, and administrative actions require human confirmation or approval controls.

  HEALTHY Destructive and outbound actions sit behind per-call confirmation, an allowlist, or a dry-run mode. This was tested, not assumed.

  RED FLAG An agent can send external email, move money, or delete records with no human step in between.

  NOTES . . . . . .

- Prompt-injection and untrusted-content risks are assessed for every connected data source.

  HEALTHY The review names which sources carry third-party content (inboxes, tickets, web pages, shared docs) and pairs each with a mitigation.

  RED FLAG A server reads untrusted content *and* holds write or send permissions in the same session with no mediation. This is the classic exfiltration setup.

  NOTES . . . . . .

- Logging captures activity without exposing secrets, personal information, or confidential prompts.

  HEALTHY Logs record tool name, caller, timestamp, and outcome. Payloads are redacted or sampled, with an explicit retention period.

  RED FLAG Full prompts and responses are retained indefinitely in a log store that more people can read than the source system.

  NOTES . . . . . .

05 Client compatibility

Test the real client and transport combination you will deploy, not the one from the vendor's demo video.

COMMON FAILURE The server works in the vendor's demo on a Mac over stdio, then fails in the client your organisation actually uses. The transport, OS path handling, and auth flow were never tested together.

- The server is tested with the intended client, operating system, and transport mode.

  HEALTHY A written test note names client + version, OS, transport (stdio or streamable HTTP), and auth flow, all exercised together.

  RED FLAG "It supports MCP" is treated as proof it works in *your* client.

  NOTES . . . . . .

- Tool discovery, authentication, reconnects, timeouts, and error behavior work as expected.

  HEALTHY A deliberate failure test exists: kill the server mid-call, expire a token, disconnect the network. The client degrades legibly.

  RED FLAG The first reconnect test happens in production.

  NOTES . . . . . .

- Known tool-count, context, configuration, and transport limitations are documented for this client.

  HEALTHY The deployment note records the client's practical tool ceiling and any MCP features (resources, prompts, sampling) the client does not support.

  RED FLAG Users file bugs for behavior that is actually a documented client limitation nobody wrote down.

  NOTES . . . . . .

- A tested, working configuration example is stored with the deployment documentation.

  HEALTHY A copy-pasteable config (secrets referenced, not included) sits next to the approval record and is updated on every change.

  RED FLAG The only working config lives on the laptop of the person who set it up.

  NOTES . . . . . .

06 Context and token cost

Measure the operating cost before scaling usage. Tool definitions load into the model's context every session. They are a permanent tax, not a one-off.

COMMON FAILURE Twelve servers are enabled "just in case," consuming a five-figure token overhead per session before the user types a word, and degrading tool-selection accuracy at the same time.

- Tool definitions and response payloads have been measured for representative workflows.

  HEALTHY A well-scoped server adds roughly 1 to 5K tokens of definitions. Per-server overhead and typical response sizes are written down.

  RED FLAG A single server injects 20K+ tokens of definitions, or one tool response routinely exceeds 10K tokens unpaginated.

  NOTES . . . . . .

- Large schemas, verbose outputs, and unnecessary tools are filtered or scoped where possible.

  HEALTHY Unused tools are disabled at the client or gateway. Total enabled tools across all servers stay within the client's practical ceiling (commonly 40 to 80).

  RED FLAG Every tool from every server is enabled by default. Nobody has ever pruned the list.

  NOTES . . . . . .

- Context-window impact is acceptable for the target model and client.

  HEALTHY Combined tool definitions consume under 10 to 15% of the context window your real workflows need.

  RED FLAG Long sessions hit context limits, or quality loss disappears when servers are disabled.

  NOTES . . . . . .

- Usage, latency, and paid API costs have an owner and an agreed budget.

  HEALTHY A monthly cost line exists per server, with an owner and an alert threshold.

  RED FLAG The first budget conversation is triggered by the invoice.

  NOTES . . . . . .

07 Reliability and failure modes

Plan for the server, dependency, or network being unavailable.

COMMON FAILURE A server times out silently. The model reports the task as done. A week passes before anyone notices the records were never written.

- Timeout, retry, rate-limit, and partial-failure behavior has been tested deliberately.

  HEALTHY Each was induced at least once in staging. Observed behavior is recorded in the deployment note.

  RED FLAG Failure behavior is inferred from the README rather than observed.

  NOTES . . . . . .

- Users can recognize failed or incomplete actions rather than assuming success.

  HEALTHY Errors surface to the user in plain language. Agents are instructed to report failures instead of retrying silently.

  RED FLAG The model's summary is the only confirmation that an action happened.

  NOTES . . . . . .

- Critical workflows have a fallback or manual recovery path.

  HEALTHY For each workflow the server touches, a written "if the server is down, do X" line exists and has been walked through once.

  RED FLAG The fallback is "wait for the server to come back."

  NOTES . . . . . .

- Monitoring covers availability, authentication failures, error rates, and unusual activity.

  HEALTHY Auth-failure spikes and off-hours activity alert a human. Availability is checked from the client's side, not just the server's.

  RED FLAG The monitoring plan is users complaining.

  NOTES . . . . . .

08 Production approval

Record the decision and the conditions attached to it.

- Security, engineering, data, and compliance reviewers have approved the relevant scope.

  HEALTHY Each reviewer signs off on the sections they own. "Approved with conditions" lists the conditions and their deadlines.

  RED FLAG Approval is a thumbs-up emoji in a chat thread.

  NOTES . . . . . .

- The approved version, configuration, permissions, and environments are documented in one place.

  HEALTHY The approval record alone is enough to rebuild the deployment from scratch.

  NOTES . . . . . .

- A review date is scheduled for maintenance, compatibility, and security re-verification.

  HEALTHY 6 months for standard servers. 3 months for servers with write access to production data or external communication.

  RED FLAG The review date is "when something breaks."

  NOTES . . . . . .

- The final decision is recorded as approved, approved with conditions, or rejected, with reasons.

  HEALTHY Rejections record the blocking sections so a future re-review starts from evidence, not memory.

  NOTES . . . . . .

FINAL DECISION

- Approved
- Approved with conditions
- Rejected

OWNER . . . . . .

REVIEW DATE . . . . . .

SERVER / VERSION . . . . . .

CONDITIONS / DECISION NOTES

. . . . . .

. . . . . .

Continue your evaluation at mcpverdict.com