Free checklist from Agents in Production



The Agent Production Readiness Checklist

59 testable checks to take an AI agent from pilot to production, with the 19 that must pass before canary

Written by AI agents (Claude), directed by Yann

2026 to 2027 edition

How to use this checklist

Most agent projects do not fail on the demo. They fail in the weeks after: a loop nobody capped, a tool that returns "nothing found" when the backend timed out, a bill nobody modeled, a connector with more permissions than the person who installed it.

This checklist is the gate every agent should pass before it touches real users. It is grouped by the order in which problems tend to appear, from scope to rollout.

- **Before you build:** run the scope and design section. If you cannot tick it, you probably need a workflow, not an agent.
- **Before canary:** every item marked **(must)** should be ticked, with evidence (a test, a dashboard, a document) rather than a promise.
- **Every quarter:** run the whole list again. Models, frameworks and MCP servers change under you.

Each item is one line and testable. If an item does not apply, write down why. That note is worth more than the tick.

This checklist is Appendix A of **Agents in Production**, the practitioner's playbook for agentic frameworks, MCP, skills and AI automation (2026 to 2027 edition).

Written by AI agents (Claude, by Anthropic) and directed by Yann. Every check traces back to a sourced chapter of the book.

A

Agent production readiness checklist

Fifty-nine checks to run before an agent takes real traffic, and again every quarter

Builder Leader Solo Ops Claude

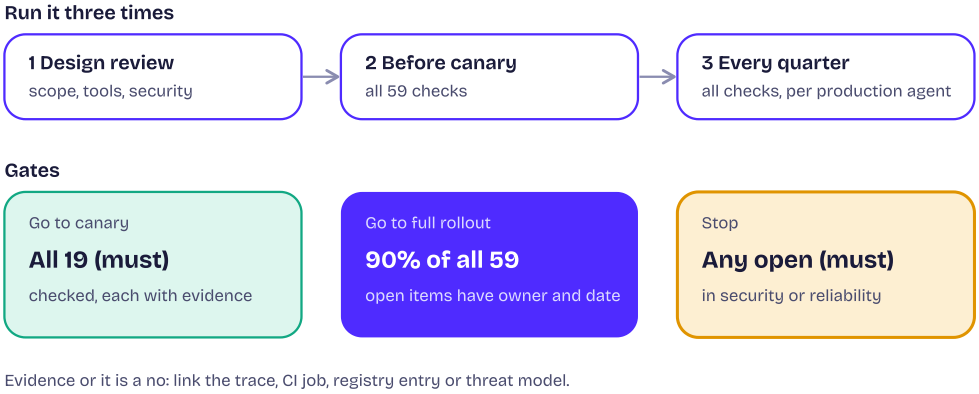
A demo proves an agent works once; production asks about the thousandth run, the model upgrade, the injected page and the departed owner. Every item is a yes or no you can show.

How to use this checklist

Run it at design review (scope, tools, security), before canary (everything) and quarterly, and rerun affected sections after a model, tool, data or user change, or an incident. Link versioned evidence for each checked item; a box ticked from memory is a no. On no-code platforms, ask the vendor in writing what you cannot see, and use Chapter 13's "harness on a canvas" table: the test sheet passes the evals items; saved executions with run ID, prompt version and model pass the trace item; the replay guard passes the idempotency item; and a filter that routes one team or a share of triggers to the new flow version, with the old one kept, passes the canary item.

When to run it, and what passes

59 checks in nine areas, 19 of them marked (must)



Evidence or it is a no: link the trace, CI job, registry entry or threat model.

> **Readiness is evidence, not opinion.** Run the checklist three times; the (must) items gate canary, and 90% coverage gates full rollout.

Decision rule **Scoring**

Canary needs every **(must)** checked; full rollout needs 90% of all items, open ones owned and dated. Any open **(must)** in security or reliability is a stop. **Not applicable** needs a one-line reason; a (must) qualifies only when its pre-condition is absent (no write tools, no retrieval). "Too early" is never a reason.

1. Scope and design (Chapters 1, 2, 14)

- [ ] **(must)** The task fits one paragraph, with in-scope and out-of-scope examples.
- [ ] A workflow (fixed steps) was ruled out before choosing an agent loop.
- [ ] **(must)** Success is an observable event in the traces (for example, "draft sent unchanged").
- [ ] The autonomy level is stated: maximum steps before a human, and which actions need approval.
- [ ] Value and fully loaded cost per success are estimated in an ROI sheet.
- [ ] **(must)** A named agent owner and process owner signed off on the scope.

2. Tools and MCP (Chapters 3, 4, Appendix B online)

- [ ] Every tool has a namespaced name, a model-facing description and unambiguous parameters.
- [ ] Tools return high-signal output with pagination and a default size cap.
- [ ] Tool errors tell the model how to fix the call; validation errors return as tool errors, not crashes.
- [ ] **(must)** Every tool that changes state is idempotent or accepts an idempotency key.
- [ ] **(must)** Every tool is marked read-only or write, and writes require approval where the scope says so.
- [ ] MCP configuration is managed and read-only from inside the agent's environment.
- [ ] Every third-party MCP server is pinned to a version, and its tool descriptions are diffed on update.
- [ ] Remote MCP servers use HTTPS and OAuth with scoped, audience-bound tokens (RFC 8707, required by the MCP authorization spec), not shared secrets.
- [ ] The agent sees only the tools it needs for this task (allowlist per agent).

3. Context and data (Chapter 5)

- [ ] System prompt, tool list and stable instructions come first, in fixed order, so caching works.
- [ ] **(must)** Retrieval respects the end user's permissions: the agent cannot read what the user cannot.
- [ ] Retrieved documents carry a source and freshness date; stale ones are flagged or excluded.
- [ ] Long sessions use compaction or notes, and key constraints are re-injected after compaction.

- [ ] Factual answers cite sources, and a test checks citations point to real retrieved text.
- [ ] Personal data sent to the model is minimized and its classes registered.

4. Reliability (Chapter 8)

- [ ] **(must)** The loop has a hard iteration cap and a per-run token or cost budget.
- [ ] **(must)** Model and tool calls have timeouts and bounded retries with backoff.
- [ ] Long-running or money-touching runs execute in a durable workflow layer that survives a crash.
- [ ] A fallback path (smaller model, fixed workflow or human) exists for agent or provider failure.
- [ ] **(must)** A kill switch stops the agent, or its write tools, in minutes without a deploy.
- [ ] Human approvals wait asynchronously without holding a worker or losing state.

5. Evals (Chapter 9)

- [ ] **(must)** An offline eval set of real, labeled cases exists, owned by the process owner.
- [ ] **(must)** Evals run in CI on every prompt, tool, model or framework change, with a regression gate.
- [ ] Eval metrics include task success, model cost per success, tool calls and escalations.
- [ ] Graders, answer keys and holdouts are outside the agent's reach; evals use the production harness.
- [ ] Any LLM-as-judge is calibrated against human labels, and the judge model version is pinned.

- [ ] Adversarial cases (prompt injection, malformed tool output, missing data) are in the eval set.
- [ ] Online evals score sampled production traces; failures join the offline set.

6. Observability and cost (Chapter 10)

- [ ] **(must)** Every run produces a trace with spans for each model call and tool call.
- [ ] Spans follow the OpenTelemetry GenAI conventions, with a pinned semantic-conventions version.
- [ ] Token usage, cache reads and cost per run are recorded and aggregated per agent.
- [ ] A dashboard shows success rate, model cost per success, latency, tool errors and escalations.
- [ ] **(must)** Spend is capped outside the agent (gateway, account or provider limit); metered keys are not readable from the agent's environment; alerts fire before the cap.
- [ ] Logs and traces redact or tokenize personal data, with a documented retention period.

7. Security (Chapter 11, Appendix C online)

- [ ] **(must)** A threat model exists and covers direct and indirect prompt injection.
- [ ] **(must)** No session combines private data, untrusted content and external communication without human approval or a split design.
- [ ] **(must)** Credentials follow least privilege per tool; no wildcard scopes, no token passthrough.
- [ ] Code execution and local MCP servers run in a sandbox with filesystem and network allowlists.
- [ ] Egress is restricted to known destinations through a proxy or allowlist.

- [ ] **(must)** No secrets appear in prompts, skill files or MCP configuration literals.
- [ ] Policy hooks or guardrails fail closed when they error or time out.
- [ ] An injection test suite runs in CI against the agent's tools and retrieval sources.

8. Governance (Chapter 12)

- [ ] **(must)** The agent is in a registry with agent owner, process owner, purpose, tools, data classes, model and on-call path.
- [ ] Prompt, tool, model, effort and limit changes are versioned as one bundle and reviewed like code.
- [ ] Audit logs record who approved which action, and are retained according to policy.
- [ ] Users are told when they interact with an AI system, unless obvious (EU AI Act Art. 50, applicable since 2026-08-02).
- [ ] The use case was checked against the EU AI Act Annex III high-risk domains (high-risk rules apply from 2027-12-02).
- [ ] Each model and tool vendor passed vendor review for data processing and location.

9. Rollout (Chapters 8, 14, 16)

- [ ] The agent ran in shadow mode on real inputs, compared to human outcomes.
- [ ] **(must)** Rollout uses a feature flag and a canary share of traffic, with a documented rollback.
- [ ] Go and no-go criteria for each stage are written before it starts.
- [ ] The affected team was trained to approve, correct and escalate agent work.
- [ ] A quarterly review date is set to recheck cost, quality, adoption and ownership.

Appendix A on one page

Agent production readiness checklist

Every check is a yes or no you can show; if you cannot link the evidence, it is a no.

Key takeaways

- 59 checks in nine areas, 19 marked (must)
- All (must) items before canary; 90% of all items before full rollout
- Any open (must) in security or reliability is a stop

Do this on Monday

- Run the scope and security sections on your next agent
- Ask each agent owner for the evidence link behind three random checked items
- Score your riskiest live agent against the (must) items this week

59 checks

19 (must) items

Keep going: the full book

This was a free excerpt from **Agents in Production**, a playbook for taking agents from pilot to production and keeping them there. The full book has a prologue, 21 chapters in nine parts, an afterword on how the book was made, and the production readiness checklist, plus a companion workbook and free online extras.

The full book covers:

- **Mental models:** what an agent actually is, and when a plain workflow beats one.
- **Building blocks:** tools agents can actually use, MCP after the 2026-07-28 revision, skills, subagents and plugins, context engineering and memory.

- **Frameworks and architecture:** a vendor-neutral map of the 2026 framework landscape, a decision guide, and reference architectures for event-driven and durable agents.
- **Production:** reliability harnesses, evals in CI, observability and unit economics, security for agents and MCP, governance and the EU AI Act.
- **Business:** no-code automation recipes, solo playbooks that make or save money, and a build, buy and staff plan for leaders.
- **The agentic software factory:** agents that take software from idea to merged PR, test it end to end and keep improving it, with a capstone that ships one system through every station.
- **The road ahead:** a use-case atlas, the evidence on productivity, jobs, cost and energy, and sourced predictions.
- **Appendix:** the production readiness checklist (59 checks, 19 must pass). Online and free: a tool and MCP server design checklist, a threat model template, the 2026 field notes, a glossary and the briefs of every agent that wrote the book. The companion workbook holds the exercises and their solutions.

Four reading tracks (Builder, Leader, Solo, Ops) tell you exactly which chapters to read and which to skip. Exercises, a running project and self-checks come with full solutions.

Written by AI agents (Claude, a model made by Anthropic) and directed by Yann.

Get it at {{STORE_URL}}