

AI Production Readiness Checklist

Before you give AI agents production access, walk this checklist. Every item is a statement that should be true of your system, not of your intentions. Items you cannot check honestly are the findings of a readiness audit – cheap to fix on paper, expensive to fix in an incident channel.

How to score: each unchecked box is a finding. Rank findings by blast radius (what an agent could reach if that guard is missing), fix the widest ones first, then re-run the checklist. A pilot is ready when every item in sections 1–4 is checked and sections 5–8 have an owner and a date.

Adapted from the readiness-audit methodology at

adventureonthewave.com/production-ai/readiness-audit – Marcus Patman, Adventure On The Wave. Free to use, no warranty. Version 1.0, October 2026.

1. Scope and inventory

- ☐ We can list, in one place, every AI agent that touches or will touch production.
- ☐ For each agent we can state: its purpose, its tools, its data, and its owner.
- ☐ Data the agents touch is classified (public / internal / sensitive / regulated).
- ☐ Someone is accountable for the system as a whole (not only per-component).
- ☐ We have said out loud what the agents must never do – and written it down.

2. Access and permissions

- ☐ Agents run under least-privilege credentials scoped per tool – not a developer's full access.
- ☐ No production secret is reachable by an agent that does not demonstrably need it.
- ☐ Secrets live in a manager or CI vault, never in repos, prompts, or agent memory.
- ☐ We have checked what a prompt injection could exfiltrate, and bounded it.
- ☐ Agent memory and context stores are access-controlled and can be audited.
- ☐ Credentials are rotatable without redeploying the agent.

3. Review and merge (the code agents write)

- ☐ Nothing reaches main without review – no direct pushes, by anyone, including automation.
- ☐ AI-written code is reviewed by a different model family than the one that wrote it.
- ☐ The review gate is fail-closed: a missing or errored verdict blocks the merge.
- ☐ Only humans merge. Agents may build and review; merge authority stays human.
- ☐ The gate's verdicts are logged in a form we can audit later (who/what/when/verdict).

4. Rollout and rollback

- ☐ Changes land where the blast radius is smallest first; promotion is a decision.
- ☐ Rollback is one step, documented, and rehearsed – not an archaeology project.
- ☐ A model-provider change (deprecation, rate limit, behavior shift) has a documented response.
- ☐ We know which agent actions are reversible and which are not, and treat the irreversible ones accordingly.

5. Observability

- ☐ Every agent run leaves a trace: what it was asked, what it did, what it spent, what it changed.
- ☐ We can answer "what did the agents do last night" from data, in minutes, without archaeology.
- ☐ Agent spend is visible per run and per day – before the invoice arrives.
- ☐ Agent SLOs exist and are monitored: success rate, cost per task, human-intervention rate.
- ☐ Failures your customers would notice page a human before the customers notice.

6. Cost control

- ☐ Each agent has a budget, enforced by the platform – not requested by the prompt.
- ☐ Retry and loop caps exist so a stuck agent cannot run all night.
- ☐ Spend alerts fire at thresholds we chose, with a named responder.
- ☐ The worst 24 hours of agent spend is bounded and known.

7. Failure and incident response

- ☐ A kill switch exists per agent, and everyone who might need it knows where it is.
- ☐ Runbooks cover agent-specific failures: loops, context poisoning, silent regressions, runaway spend.
- ☐ Agent-caused incidents get postmortems in a fixed template, with trace references.
- ☐ There is a path from "agent behaved oddly" to "decision to stop it" that does not require a founder.

8. Team and process

- ☐ One governed workflow for how the team uses agents – not fifty personal setups.
- ☐ Everyone who runs agents has been trained on the permissions and approval boundaries.
- ☐ The governance rules live in version control, not in one engineer's head.
- ☐ There is a re-audit cadence: the checklist is re-run when scope, tools, or models change.

Scoring is deliberately unweighted: an unchecked box in section 2 or 3 (access, review) is the kind of gap that turns into an outage; sections 5–8 unchecked turn outages into

long ones. If more than a handful of boxes resist honest checking, that is what a structured engagement solves faster than willpower:

adventureonthewave.com/consulting/genetic-readiness-audit.