# Vendor evaluation checklist.

Use the questions below when evaluating any AI tool that requests access to your codebase. **A vendor that can answer all of them clearly, with documentation to back up the answers, takes security seriously.**

## Code usage and model training

- [ ] Does the tool use my code to train the underlying LLM? If so, under what terms, and how do those terms change?
- [ ] What does the vendor do with my code beyond delivering the service? Is that disclosed clearly and completely?
- [ ] If the vendor trains models on customer data, is that model shared across customers or dedicated to your organization?

## Data isolation

- [ ] How is my data isolated from other customers at the database level?
- [ ] Does the LLM maintain context across requests? If so, how is cross-customer contamination prevented?
- [ ] Is fine-tuning or any similar technique used? If so, on what data, and how?
- [ ] What happens to my code, derived insights, and analysis after I stop using the product? What is the deletion process and timeline for each?

## Agentic behavior and autonomous actions

- [ ] If the tool operates in agentic mode, what actions can it take autonomously, and what requires explicit human approval?
- [ ] Are destructive or irreversible actions blocked at the infrastructure level, or only at the instruction level?
- [ ] What credentials does the tool request, and are those credentials scoped to the minimum permissions required?
- [ ] Can the tool's credentials be used for operations beyond its stated purpose?
- [ ] When the agent encounters ambiguity or uncertainty, does it halt and ask for clarification, or does it make a decision and proceed?

## Access and governance

- [ ] Under what circumstances can vendor staff access my code? Is that access logged?
- [ ] Is production infrastructure access structurally controlled, or policy-based?
- [ ] Is infrastructure defined as code, version-controlled, and auditable?

# Operations and ongoing security.

## Ongoing security

- [ ] Does the vendor have a formal, documented SDLC? Is SAST automated as part of every deployment pipeline?
- [ ] Does the vendor conduct independent security assessments? How often, and against what frameworks?
- [ ] Does the vendor test their incident response plan, and how? Tabletop exercises, simulations, and live drills offer demonstrable readiness that a plan on a shelf cannot.
- [ ] Does the vendor have a vulnerability disclosure program or a clear process for reporting security concerns?
- [ ] How is security embedded in the vendor's own development process? Is it enforced at the code level, or left to individual judgment?
- [ ] Has the vendor passed enterprise security reviews? In what industries?
- [ ] What is the vendor's path to formal certification (SOC2, ISO 27001)?
- [ ] Is the vendor's security documentation public, versioned, and kept current? When was it last updated?

## Contractual protections

- [ ] Does the vendor provide contractual confidentiality protections or a generic data processing agreement?
- [ ] Are confidentiality protections formalized in a bilateral agreement that requires mutual written consent to modify, or can terms change unilaterally?
- [ ] Are there specific notification commitments in the event of a breach? Can those commitments be negotiated?

## Supply chain and third-party risk

- [ ] Which third parties have access to your code or derived data, and under what circumstances?
- [ ] Does the vendor apply contractual data governance requirements to third parties that touch customer data?
- [ ] Is supply chain security included in the vendor's incident response and assessment programs?

### How to use this checklist

Treat the checkboxes as a working document. Mark the questions a vendor has answered clearly in writing. Leave questions unmarked where the answer is vague, verbal-only, or absent. The pattern of remaining gaps is itself the signal.