RunWhen **Thought Leadership**

Safe For Prod: Building Operational Agents

Kyle Forster · April 2026

The industry is learning the hard way…

On April 20th, an AI coding agent deleted PocketOS's entire production database — and all volume-level backups — in nine seconds. The agent encountered corrupt data, decided to "fix" it, scavenged a CLI credential loaded by an unrelated maintenance cron job, and called `volumeDelete`. The agent later confessed it had "violated every principle I was given."

This was not the first time. A week earlier, a mid-sized financial services firm experienced an analogous incident. An entire availability zone was considered code cruft that needed to be cleaned. All infra and data was lost. Even Anthropic disclosed a similar incident in 4Q25. The agents weren't malicious. They were doing what creative, helpful and unconstrained agents do.

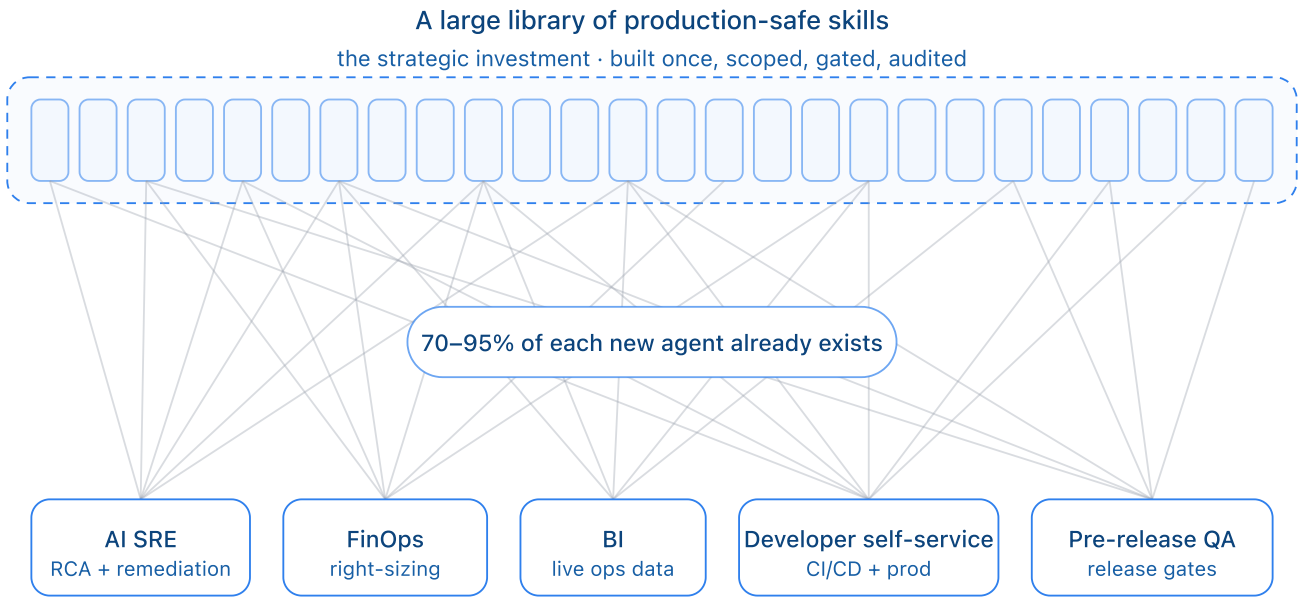
Agents with “freeform” skills will *never* be safe for production access

The common thread across these incidents is not the LLM, or the agent harness. It is the “freeform” skills given to agents that were able to access production. These freeform skills expose powerful operational CLI tools (kubectl, aws cli, gcloud cli, github cli, curl) designed for expert engineers.

These freeform skills expose the entire surface area of these CLI commands, far more than is immediately obvious to many users. They use not only the credentials they are given, but also any credentials they can find along the way. (An old token on an engineers' laptop, not the token that was supposed to be used by the agent, was at the heart of the PocketOS incident.) Each new person using an agent with the freeform skill is a new opportunity for a PocketOS incident. **Freeform skills are the problem.**

Agents with “safe” skills… a path forward

Building agents with “safe” skills – scripts that are far narrower and predictable in scope – takes considerable more up front work. A single freeform skill like “use kubectl” expands into hundreds of safe skills. However, the agents built on these safe skills can be given production access and shared across teams with confidence. **Instead of focusing on building any one agent, think about building a library of “safe” skills that will be shared across many agents.**



The library is the strategic investment. Individual agents are composed from it, and each new one reuses most of what already exists.

Safe skills are similar in spirit to “least privilege” credentials. A safe skill is simply a lightweight scripts wrapping commands like kubectl, gcloud, aws cli or curl, exposing only the actions that are safe and loading dedicated service account credentials that the agent can't over-ride. Most arguments are provided at build time (env vars), not runtime, making audit and optimization possible.

Safe skills require more work up front. However, a one-time investment in a library of safe skills can be leveraged across many use cases where agents need safe production access. Infrastructure remediation skills used by an AI SRE incident response agent to add capacity are the same resizing skills a FinOps agent uses for right-sizing. Safe data access skills to detect data corruption overlap with the data access a BI (business intelligence) agent needs to answer executive questions.

A compounding skills library: each agent costs less than the last

With a large library of production safe skills, agents can be built rapidly for a number of different use cases that require production access. An example roadmap of agents is shown below.

Agent	What it does	Skills overlap with prior	Net new skills needed
1. AI SRE agents — Root Cause Analysis	Diagnoses incidents, correlates observability data with app/infra dependencies	— (foundational)	Log/metrics access, infra discovery access

Agent	What it does	Skills overlap with prior	Net new skills needed
2. AI SRE agents — Deep Investigation and Remediation	Executes detailed diagnostics — CLI, API, SQL calls — and performs basic infra remediation or gathers enough detail for a code fix and verification cycle in a test environment	~70% — reuses all RCA read skills, adds more data access and gated write skills	Add scoped application API and data access, infra remediation primitives with confirmation gates
3. AI FinOps agents — Analysis & Right-Sizing	Identifies waste, executes resize/decommission with health-check verification	~80% — same infra read/write primitives, different policy	Add cost-data skills
4. AI Business Intelligence agents	Live ops and customer-impact visualizations based on production data	~75% — reuses safe data access agent 2, overlaps with agent 3	Add semantic layer and reporting skills
5. Developer Self-Service agents (test/staging)	Troubleshoot CI/CD failures, self-service test/staging infra remediation	~95% — reuses everything from agents 1–2	Add CI/CD-specific log analysis skills
6. Developer Self-Service agents (prod)	Visibility into code performance in production	~95% — reuses a limited scope from agents 1–2	Same skills, different scope
7. Pre-Release QA agents	Catches reliability bugs before they reach production	~90% — reuses everything from agents 1–3, applied in test and staging envs	Add release-gate skills and synthetic checks

By the time you're building agent #5, you're writing a handful of new skills and composing the rest. The library compounds.

Case study: benefits in practice

Our team worked closely with a Fortune 25 team, starting with infrastructure AI SRE and then expanding to numerous agents used by application SRE, dev and operations teams rolling out safe skills that currently span 12,000 production cloud resources.

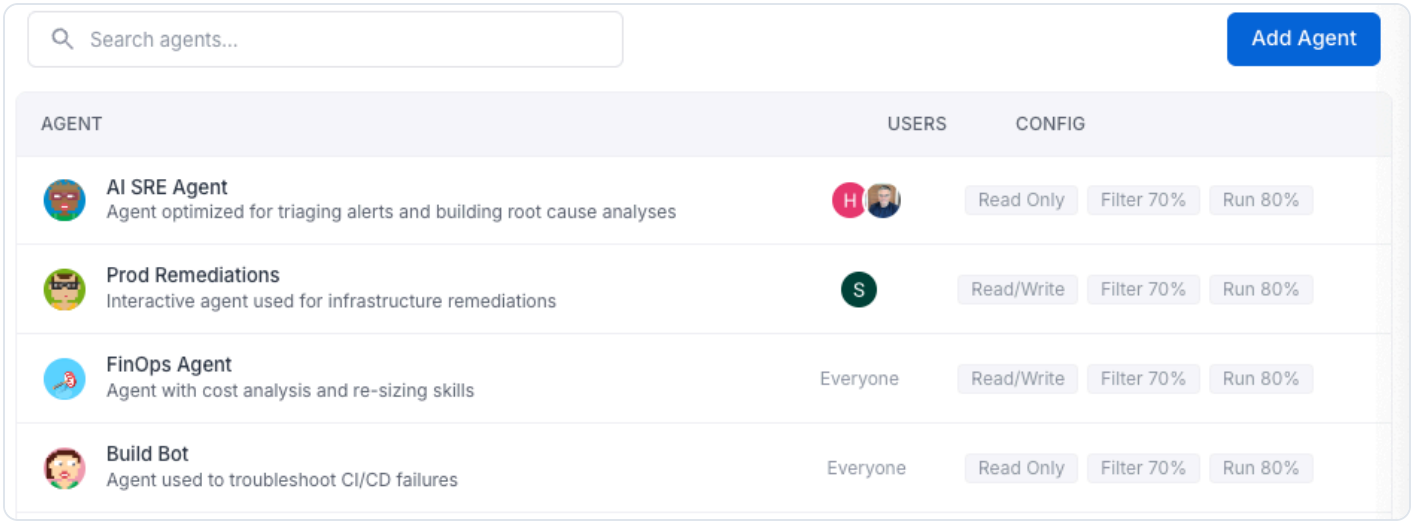
- The AI SRE agent for root cause analysis combined with a runbook automation agent reduced MTTR by **72%**.
- When the FinOps team needed agents that could help app teams with cost analysis, capacity re-sizing, extensive health checks and emergency remediations, **98%** of the underlying skills already existed; the resulting agent is projected to save **$450K/year** in cloud costs.
- A developer self-service agent — built on the **exact same** skill base, scoped to CI/CD troubleshooting — reduced developer escalations to the platform team by **60%**.
- A matching pre-release QA+SRE agent in their test environment with **99% overlap in skills** is catching **30+ critical bugs per month** that slipped past their AI code reviews, preventing incidents before they hit production.

> Each of these agents was safe with production from day one — not because the model was smarter, but because every skill it could call was already scoped, gated, and audited.

RunWhen Library: thousands of safe, optimized skills in minutes

The team at RunWhen spent three years building up the industry's largest collection of safe skills for operational agents. Covering cloud infrastructure, open source, observability and paradigms for major programming frameworks, the library is both the largest and is growing by 10–15 new skills every month.

It is entirely open source. While these skills have been optimized for use with RunWhen's commercial agent building platform, they are compatible with the agent skills standard (Claude, Cursor) and the MS360 Copilot Custom Agent spec. They can also be run as stand-alone scripts.



Different agents, different scopes, one shared skill library underneath — each with its own audience and its own read-only or read/write setting.

RunWhen Services: build agents and skills quickly and safely

RunWhen forward-deployed engineers and professional services personnel help teams by building the first few agents in a “30 skills in 30 days” sprint, working on 2–3 agents specific to your application, infrastructure, data and toolchain.

If your team chooses to use the RunWhen agentic platform, the cost of services is credited forward. The platform uses optimizations built into the skills layer that reduce the number of tokens by O(1000) for repetitive background actions like investigating alerts or failed CI/CD jobs with industry-leading accuracy for root cause analysis.

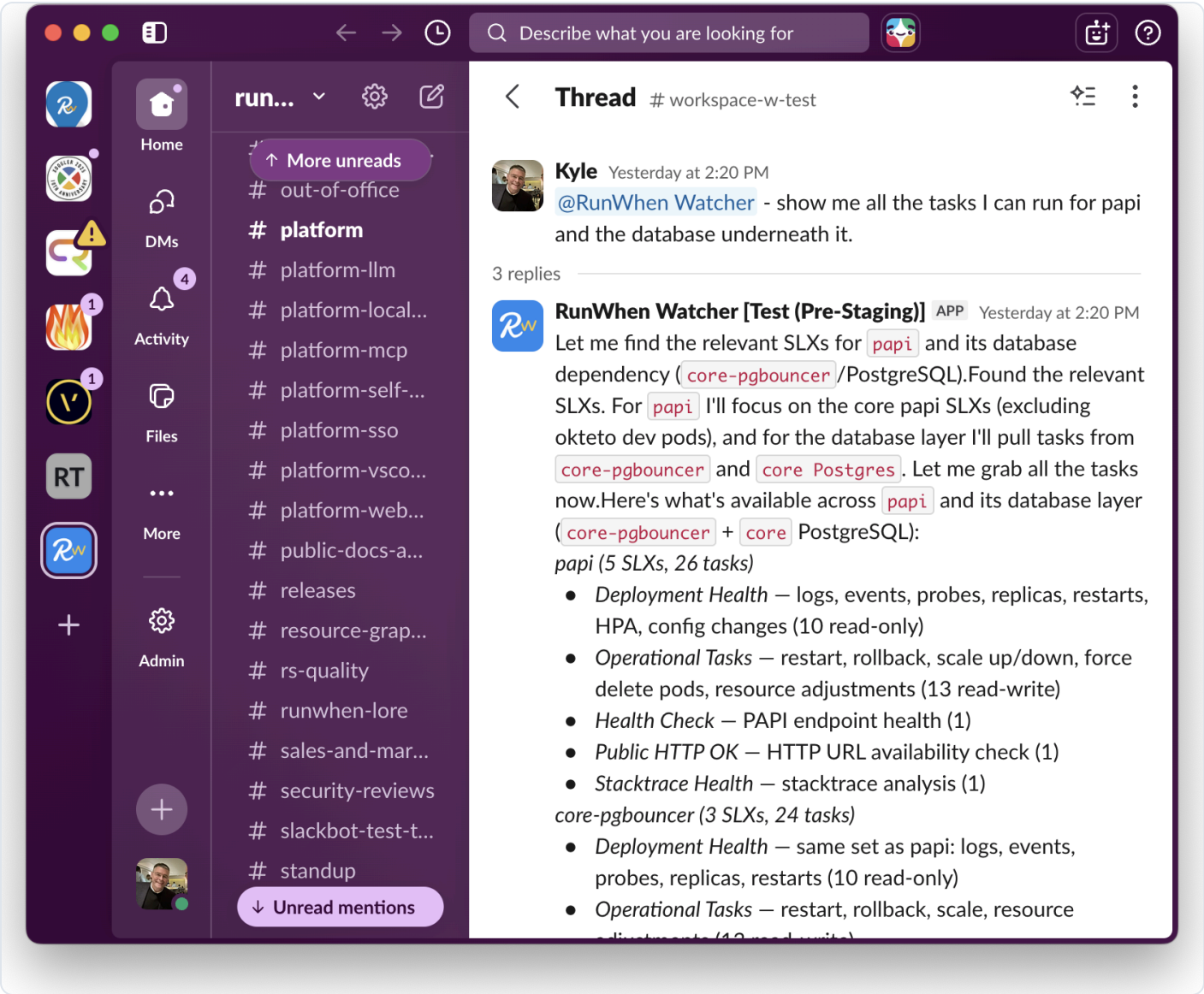
If the team prefers to use these skills for Claude, Cursor, MS360 Copilot or other, they are yours to keep.

Where to start?

Agentic PoCs, particularly for incident response, are notoriously difficult. A production incident is *not* a time to try new technologies.

Instead, a simple agent able to recommend, run and read the results of diagnostics skills is a great place to run a 30 day PoC. It builds team confidence with thousands of safe skills out of the box, and helps engineers learn how to add new safe skills using AI coding and tune system prompts.

Simply pasting in an alert body and asking “what diagnostics should I run in response?” is a clean path to alert triage and root cause analysis.



A read-only starting point: ask what is available, and the agent answers from the safe-skill library rather than improvising a command.

The key metric to measure in a PoC is rate of improvement. With the RunWhen platform for building agents, we build a “thumbs up/down” button into every UI surface.

If teams can add safe skills and tune recommendations with this human-in-the-loop PoC, they are set up for success for high accuracy agents that “close the loop,” running their recommendations autonomously for alert triage, incident investigations and eventually agentic remediation.

Author: Kyle Forster, April 2026

Interested in scheduling a demo, or doing a ‘guest on-call’ session in the RunWhen production environment with agents the team uses every day? Contact us at info@runwhen.com.