

Operations

A monitoring system that is installed and never tended decays into either noise you ignore or silence you trust wrongly. Operating it means two things: keep the alerts meaningful, and make sure the monitor itself is still working.

Tuning out noise

The failure mode that kills a monitoring system is not too few alerts, it is too many. A system that alerts on things that turn out not to matter trains you, surprisingly quickly, to ignore its alerts and an ignored, valid alert is just as bad as no alert. So the ongoing work is keeping the signal-to-noise ratio at a reasonable level.

In practice this means:

- **When an alert fires that did not need to, tune the check that produced it.** A threshold slightly too tight, a window slightly too short, a condition that does not distinguish a transient blip from a sustained problem, each produces false alarms. The health check's distinction between a momentary spike and a sustained condition is the model: most false positives come from not making that distinction somewhere.
- **Resist the urge to monitor more.** Every new check is new potential noise. Add checks deliberately, for categories that genuinely reveal trouble, not just for awareness. You still have the full logs when you just want to see how systems are behaving. A smaller set of trusted alerts beats a larger set you have trained yourself to skim past.
- **Review what has been firing.** Periodically look at what the system has alerted on and ask whether each alert was worth it. The ones that were consistently not worth it are candidates for tuning or removal.

The conservative posture throughout, watching high-value categories, requiring conditions to persist before alerting, keeping thresholds where real problems live rather than where noise does, is all in service of one goal. When an alert arrives, you have confidence in it enough to act.

Watching the watcher

Here is the failure that a monitoring system is uniquely prone to, and it is subtle: **a monitor that has silently stopped looks exactly like a quiet, healthy environment.** No alerts arrive in both cases. If your collector stops receiving logs, if a forwarding rule breaks, if the automation layer stops running the checks, the symptom is silence, and silence is also what "everything is fine"

looks like. You cannot tell the two apart by waiting for alerts, because the broken system will never send one.

So part of operating a monitoring system is monitoring the monitoring:

- **Watch for the absence of expected logs.** Some log streams are steady, the health snapshots arrive every few minutes from every host, for instance. A stream that normally flows steadily and then stops is itself a signal, and a check that notices "I expected snapshots from this host and got none" catches a broken pipeline that would otherwise be invisible.
- **Confirm the checks are actually running.** An automation platform shows run history, a cron can log its executions. Periodically confirm the scheduled checks ran when they should have, rather than assuming that no alert means no problem.
- **Test the path end to end occasionally.** Deliberately produce something that should alert, and confirm the alert arrives. This is the monitoring equivalent of testing a smoke detector. The only way to know it still works is to set it off on purpose now and then.

Adapt this for...

Any monitoring system. The operating disciplines are universal. Keep the noise down so the alerts stay trusted, watch for the silence that means the monitor stopped rather than that all is well, test the path deliberately, and treat the monitoring infrastructure as the sensitive, high-value thing it is. The install was the easy part, maintaining its relevancy is what makes it a system you can rely on a year later.

Revision #1

Created 2026-07-31 22:44:55 UTC by Chris Landis

Updated 2026-07-31 22:52:07 UTC by Chris Landis