Bilko Sentinel — Tier-0 Self-Healing Agent 2026-06-10

Status

LIVE and Proveo-verified as of 2026-06-10. MC #103337 (AgentForge implementation) + MC #103337 Proveo independent verification. Parent MC #103328. Dynamic policy discovery added MC #103420 (2026-06-11).

Related: Bilko Observability (GCP-native) 2026-06-10 — **Tier-1 (bounded auto-remediation, SHADOW):** Bilko Sentinel Tier-1 (Shadow-First) 2026-06-11 — the GCP alert layer this agent reads from.

What It Is

Bilko Sentinel is a **read-only ops agent** that runs on ANVIL every 3 minutes. It follows a four-stage pipeline:

1. **Detect** — at cycle start, dynamically discovers all enabled GCP Monitoring alert policies via `gcloud alpha monitoring policies list` (SA `alai-cli-deployer`, quota project). Normalizes each `conditionThreshold` into the evaluator's internal shape, then evaluates the last 6 minutes of time-series data against every condition. The policy set is cached for 5 minutes (`bilko-sentinel-policy-cache.json`) to avoid hammering the API every 180-second cycle. If the fetch fails, falls back to the embedded list and logs a WARN — never crashes, never goes silently blind. Currently evaluates **9 policies (13 conditions)**.
2. **Enrich** — on a breach, fetches recent Cloud Run logs and the current revision/traffic split for the affected service.
3. **Diagnose** — calls FORGE Ollama (`qwen2.5:7b-instruct-q8_0` at `10.0.0.2:11434`) with a structured JSON prompt (temperature 0.1) to produce a root-cause hypothesis and recommended action. Falls back to a deterministic template per cause category if Ollama is unreachable.
4. **Propose** — posts exactly one structured proposal per unique incident to Slack **#ceo** and email **alem@alai.no**. Deduplicates by incident key; does not re-notify the same breach for 24 hours.

It never changes anything. Proveo independently verified: zero mutating verbs, no GCP mutations of any kind (no `run deploy`, no `set-iam-policy`, no SQL writes, no secrets writes). The only HTTP POST in the script goes to the Ollama local inference endpoint, not to googleapis.com. The `gcloud alpha monitoring policies list` call added in MC #103420 is a read-only list operation — forbidden-verb scan still returns 0 matches (verified by AgentForge evidence proof_5).

Infrastructure

Component	Location
Script	`/Users/makinja/system/tools/bilko-sentinel.js`
LaunchAgent plist	`/Users/makinja/Library/LaunchAgents/com.alai.bilko-sentinel.plist`
State file (dedup)	`/Users/makinja/system/state/bilko-sentinel-state.json`
Policy discovery cache	`/Users/makinja/system/state/bilko-sentinel-policy-cache.json` — 5-min TTL
Audit log	`/Users/makinja/system/logs/bilko-sentinel-audit.jsonl`
Run log	`/Users/makinja/system/logs/bilko-sentinel.log`
Host	ANVIL (makinja local Mac)
Schedule	180-second interval, RunAtLoad=true
Node.js path	`/opt/homebrew/bin/node`

Policies Monitored — Dynamic Discovery (9 policies, 13 conditions)

As of MC #103420 (2026-06-11), the Sentinel **dynamically discovers all enabled GCP alert policies** each cycle. The list below reflects the 9 policies currently active. Any policy added to GCP Console or via FlowForge is automatically picked up without a code change.

1. Cloud SQL CPU utilization high (prod + stage)
2. Container restart/crash on prod services
3. HTTP 5xx rate high on bilko-api-demo
4. HTTP 5xx rate high on bilko-web-demo
5. Request latency P95 high on prod services (API + Web — 2 conditions)
6. CIAM — High 429 rate on bilko-api-demo
7. Cloud SQL connections near max on bilko-demo-db
8. Uptime check failed (app.bilko.cloud + app-api.bilko.cloud — 2 conditions)
9. Bilko API Demo — Backend ERROR log rate (`bilko_api_demo_error_count`, policy #2342970117877340710, added MC #103364) — *this policy was missed by the old*

hardcoded list and is what prompted MC #103420

Condition type support: `conditionThreshold` (metric threshold) — fully evaluated; covers all 9 current policies. `conditionAbsent` and other types — logged and skipped, cannot fire false positives.

Severity Scale

Label	Meaning
P1-DOWN	Service is down or uptime check failing
P2-DEGRADED	Elevated error rate or restart loop
P3-WARN	Latency spike, DB pressure, CIAM abuse rate

Notification Format

Every proposal contains:

- Header: `BILKO SENTINEL — PROPOSAL (Tier-0, no action taken)`
- Incident ID, severity, env, resource, condition name
- Metric value vs threshold (exact numbers)
- Root-cause hypothesis (Ollama-generated or deterministic fallback)
- Proposed remediation steps (for human to execute)
- GCP Console link for the alert incident
- Detected timestamp

Dedup key format: `bilko-{policyId[-8:]}-{condId[-8:]}`. Once notified, silent for 24 hours on the same condition.

Proveo Verification Summary

Proveo (MC #103337) independently verified all critical properties:

Property	Method	Result
Read-only guarantee	Exhaustive grep of all spawnSync calls and HTTP methods	CONFIRMED — zero mutating verbs
LaunchAgent loaded + healthy	`launchctl list | grep bilko-sentinel` — LastExitStatus=0	PASS
Detect → Propose → Slack delivery	Independent verifier script with synthetic threshold (2ms vs real 9.5ms P95)	PASS — Slack message confirmed in #ceo at 04:24 UTC

Property	Method	Result
Detect → Propose → Email delivery	Same synthetic test	PASS — Message-ID confirmed in audit DB
Dedup across cycles	Real 2-cycle disk-persistence test (not code inspection only)	PASS — Cycle 2 silent, no second Slack message
Healthy = silent	Normal threshold against real metric value	PASS — zero messages sent
No GCP mutation	Cloud Run revision before/after comparison	PASS — bilko-api-demo-00167-h9v unchanged
Read-only guarantee (MC #103420)	Forbidden-verb grep: `gcloud run deploy`, `set-iam`, `secrets write`, policy create/update/delete — 0 matches	CONFIRMED — `gcloud alpha monitoring policies list` is a read-only list call

Honest gaps noted by AgentForge (now closed by Proveo): email exit-code quirk (fixed in script via stdout check); dedup 2-cycle test (now independently proven); Ollama not re-exercised in Proveo test (builder's synthtest confirmed it live).

Incident-Driven Hardening (MC #103420)

On **2026-06-10**, a 503 burst on `bilko-api-demo` fired alert policy **`bilko_api_demo_error_count`** (policy ID 2342970117877340710, added in MC #103364). The Sentinel did not fire a proposal because that policy was not in the original hardcoded list — it had been added after the Sentinel was built.

MC #103420 replaced the static list with dynamic discovery (`discoverPolicies()`): each cycle the Sentinel fetches all enabled policies from GCP, so any future policy added in GCP Console or by FlowForge is automatically evaluated with zero code changes. The hardcoded `ALERT_POLICIES` array is kept as a fallback only. AgentForge re-verified the read-only guarantee post-fix (forbidden-verb scan: 0 matches). The Tier-0 read-only contract is unchanged.

Runbook

Pause sentinel

```
launchctl unload ~/Library/LaunchAgents/com.alai.bilko-sentinel.plist
```

Resume sentinel

```
launchctl load ~/Library/LaunchAgents/com.alai.bilko-sentinel.plist
```

Check last run status

```
launchctl list | grep bilko-sentinel
# PID="-" = not currently running (between intervals). LastExitStatus=0 = healthy.

tail -20 /Users/makinja/system/logs/bilko-sentinel.log
```

View audit trail

```
tail -f /Users/makinja/system/logs/bilko-sentinel-audit.jsonl
```

View current policy discovery cache

```
cat /Users/makinja/system/state/bilko-sentinel-policy-cache.json
```

Add a new alert policy

Create or enable the alert policy in GCP Console (or via FlowForge). The Sentinel will automatically discover and evaluate it at the next cache refresh (within 5 minutes). No code change needed. To force an immediate pick-up, delete the cache file and wait for the next cycle:

```
rm -f /Users/makinja/system/state/bilko-sentinel-policy-cache.json
```

Tune alert thresholds

Thresholds live in the GCP alert policy definitions, not in the Sentinel script. Update the threshold in GCP Console; the Sentinel picks up the new value at the next cache refresh. To update the **fallback** embedded list (used only when GCP fetch fails), edit `ALERT_POLICIES` in `/Users/makinja/system/tools/bilko-sentinel.js` and reload:

```
launchctl unload ~/Library/LaunchAgents/com.alai.bilko-sentinel.plist
# edit the fallback array in the script
launchctl load ~/Library/LaunchAgents/com.alai.bilko-sentinel.plist
```

Tier Model and Safety Rationale

The tier model was defined after the 2026-06 IAM incident, in which an automated `set-iam-policy` call wiped project IAM. The lesson: any agent that can mutate production infra must earn trust via a demonstrated read-only track record first.

Tier	Capability	Status	Safety gates
Tier 0 — current	Detect + Diagnose + Propose. Read-only. Posts structured proposal to #ceo and alem@alai.no. Zero blast radius.	LIVE	No code path to write to GCP. Proveo-verified. Dynamic discovery is a read-only list call.
Tier 1 — future MC	Bounded auto-remediation: Cloud Run revision rollback, instance scale adjustment, hung service restart. Circuit breaker (max N actions/hour). Full audit trail. Never touches DB schema, IAM, secrets, or financial data. Always announces before acting.	BUILT — SHADOW (MC #103435). Calibration clock started. See Tier-1 reference page.	Explicit CEO approval token (`/tmp/bilko-sentinel-tier1-approved`) required before any mutation. Separate script (`bilko-sentinel-tier1.js`). Only after Tier-0 proves signal quality over weeks.
Tier 2	Broader autonomy.	Probably never for a prod-financial SaaS	N/A

The IAM incident reference is intentional: Tier-1 will be built with a hard whitelist of reversible Cloud Run and scaling operations only. No set-iam-policy, no SQL DDL, no secret rotation — ever.

Revision #3
Created 2026-06-10 02:30:46 UTC by John
Updated 2026-06-11 11:57:53 UTC by John