COMPLIANCE REPORT · MINDSHINE CORTEX

SOC 2 Audit Findings Report Platform Infrastructure Tier

Findings, testing, and remediation for the Kubernetes platform on which Cortex and its deployed applications run — assessed against the SOC 2 Trust Services Criteria.

REPORT DATE	ASSESSMENT WINDOW	AUDITS OF RECORD	SYSTEM IN SCOPE
11 August 2026	10 – 11 August 2026	#131 → #136 (commit-pinned)	EKS platform `flow-demo-2` (dev)

TSC CATEGORIES	REPORT TYPE
Security (CC6, CC7, CC8)	Internal self-assessment

01 Scope & method

This report covers the **platform infrastructure tier** — the Amazon EKS cluster and the pods, service accounts, and RBAC objects that host the Cortex product and every standalone application it deploys. It is the sequel to the repository-tier report (audit #126, 0 open findings) and addresses the layer beneath the application code.

Cortex audits its own platform with the same continuous-audit engine it offers customers. Infrastructure evidence is produced by **Prowler** (CIS Kubernetes / Prowler ThreatScore benchmarks) run against the live cluster; each scan is ingested as a commit-pinned audit run, and every failing control becomes a tracked issue with a full, append-only lifecycle. The figures below are queried from that system of record, not curated by hand.

TESTING PROCEDURE	WHAT IT EXAMINED	FREQUENCY / INSTANCE
Tracked Prowler scan	CIS/ThreatScore controls across the in-boundary namespaces (`cortex-demo-2-dev` + `app-cortex-*`)	Weekly CronJob; ad-hoc run reconciled as audit #136

TESTING PROCEDURE	WHAT IT EXAMINED	FREQUENCY / INSTANCE
Independent unscoped scan	Full-cluster baseline for regression comparison	Runs `20260811092657` & `20260811101704`
Admission-policy inspection	Live `--dry-run=server` RoleBinding attempts as the exporter service account	Per remediated control
Application health verification	Headed browser round-trip against the platform and a deployed app under the hardened runtime	`tests/e2e/platform-hardening-health.spec.mjs`
Independent adversarial review	Clean-context reviewer instructed to refute each remediation claim	Post-remediation, pre-report

Boundary. The scan boundary is the dev platform: namespace `cortex-demo-2-dev` and the `app-cortex-*` namespaces the platform stamps for deployed applications. AWS-managed `kube-system` add-ons fall under the AWS shared-responsibility model and their own SOC 2 coverage; the production cluster is hardened on its own scheduled change window and is explicitly out of scope for this report.

02 Results at a glance

Two remediation rounds and one independent verification pass were completed in this window. The in-boundary platform now carries **zero open findings**; every residual failing control maps to a documented named acceptance or a verified transient batch job.

0
Open findings in-boundary (audit #136)

66
Platform issues tracked across their lifecycle

50
Remediated & re-detected as absent

16
Formally dispositioned (12 accepted risk · 4 false positive)

ENTERING THE WINDOW

33 / 23 fail

- Writable root filesystems across dynamic & app pods
- Missing seccomp profiles, resource limits, image pins
- Exporter RBAC cluster-admin-equivalent in effect

→

AT REPORT DATE — FRESH UNSCOPED SCAN

3 / 5 fail

- All below the prior baseline; no new failing check families
- Every residual = named acceptance or transient Job pod
- Exporter escalation path closed by admission policy

Counts are per-namespace FAILs for the in-boundary dev and application namespaces; the "before" figures are the pre-remediation baseline recorded on scan `20260811092657`.

03 Tests of controls & results

The platform-tier remediations delivered this window, each mapped to its Trust Services Criterion, the change that closed it, and the test that confirmed closure. Status REMEDIATED means the control was re-detected as passing by a subsequent commit-pinned scan — closure is mechanical, not asserted.

CRITERION	FINDING	SEV.	REMEDIATION & VERIFYING TEST	STATUS
CC6.8 Integrity	Engine-generated pods ran with a writable root filesystem (broad "dynamic/agent" class)	MEDIUM	Read-only root filesystem enforced **at the template level** — every pod the engine stamps (dynamic services, exported-app components, portals, data stores) is born `readOnlyRootFilesystem: true` with scoped writable `emptyDir` mounts. Verified live per pod after deploy `7a8e03ad`.	REMEDIATED
CC6.8	Missing seccomp profiles and CPU/memory requests & limits on dynamic and build pods; a data-store image on a mutable `:latest` tag	MEDIUM	`RuntimeDefault` seccomp and resource requests/limits applied to dynamic-service Deployments, kaniko build jobs, and exported-app components; `dolt-db` image digest-pinned. Confirmed by re-scan.	REMEDIATED

CRITERION	FINDING	SEV.	REMEDIATION & VERIFYING TEST	STATUS
CC6.3 Least privilege	Deployment exporter service account held cluster-wide create/update on secrets, pods, and RBAC — cluster-admin-equivalent in effect	HIGH	Exporter migrated to a minimal `flow-exporter-core` ClusterRole; per-application privileges granted via a namespace-local RoleBinding the exporter stamps inside each `app-cortex-*` namespace it creates.	REMEDIATED
CC6.3	**Residual escalation path** (found by independent review): the unavoidable cluster-wide `rolebindings: create` verb let the exporter token bind a powerful role to itself in any namespace	HIGH	Contained by a **ValidatingAdmissionPolicy** (`flow-exporter-rolebinding-guard`): the exporter may create RoleBindings *only* in `app-cortex-*` namespaces and *only* to `ClusterRole/flow-exporter-app`. Proven by live deny in `kube-system` and production and allow in an application namespace.	REMEDIATED
CC6.8	Service-manager init containers (`wait-for-mongo`, `wait-for-redis`) lacked a read-only root filesystem	LOW	Read-only root filesystem set on both init containers (they only poll a TCP port). Deployment rolled clean, 0 restarts, live spec confirmed.	REMEDIATED
CC7.1 Detection	The scanner violated the standards it enforces (unpinned base image, tautological probes)	MEDIUM	Base image digest-pinned, `prowler==5.37.1` version-pinned, fake liveness/readiness probes removed rather than gamed. Documented in the access matrix.	REMEDIATED
CC7.2 Monitoring	Audit records: disposition justifications not persisted; finding evidence truncated to 5 resource samples	LOW	Justification notes backfilled on all 16 dispositioned issues; ingestion evidence cap raised to 25 samples / 1,200 characters so a disposition is auditable from the issue record alone.	REMEDIATED

Verification basis. Every remediation above was confirmed by re-running the tracked scan (audits pinned to the git commit evaluated) and, for the admission policy, by attempting live RoleBinding creates as the exporter service account. Because `kubectl auth can-i` does not evaluate admission policies, containment was tested by `--dry-run=server` create attempts — not by permission queries.

04 Findings identified by independent review

After the two remediation rounds, an independent reviewer with no prior context was instructed to challenge every claim and search for an escalation path the fixes might leave open. All six remediation claims were **confirmed** under live adversarial testing. The review itself surfaced the following findings, both since resolved.

FR-01 · GAP-1 HIGH RESOLVED

Least-privilege RBAC left a residual self-escalation path

CONDITION The round-2 scoping still required the cluster-wide `rolebindings: create` verb (RBAC cannot scope a create by namespace or roleRef). Combined with bind rights on the powerful `flow-exporter-app` role, a compromised exporter token could have bound that role to itself in `kube-system` or production and read every secret.

RESPONSE A ValidatingAdmissionPolicy now rejects any exporter-authored RoleBinding outside `app-cortex-*` namespaces or referencing any role other than `flow-exporter-app`. The reviewer re-proved allow/deny live and confirmed no alternative escalation path (no ClusterRoleBinding, token-mint, impersonation, or policy-mutation rights).

DISPOSITION **Remediated on dev.** The same policy, keyed to the production service-account identity, is part of the gated production migration.

FR-02 · GAP-2 LOW RESOLVED

Service-manager init containers omitted read-only root filesystem

CONDITION The main container was hardened but its two wait-for-dependency init containers were not.

RESPONSE Read-only root filesystem set on both; Deployment rolled clean with zero restarts; live spec confirmed by the reviewer.

DISPOSITION **Remediated on dev.**

FR-03 · N-1 / N-2 LOW RESOLVED

Audit-record evidence quality

CONDITION Dispositions were recorded without persisted justification (an API field-name mismatch silently dropped the note), and ingestion stored only 5 of each control's affected resources — a disposition could not be fully re-derived from the issue record alone.

RESPONSE Justification notes backfilled on all 16 dispositioned issues; evidence retention raised to 25 samples / 1,200 characters at ingestion (takes effect on the next scanner image roll).

DISPOSITION **Remediated.** An evidence-quality control, not a security exposure.

05 Accepted risks & dispositions

Sixteen residual findings are formally dispositioned rather than remediated. Each carries a recorded rationale on the issue itself; none is silent.

CONTROL	WHAT IS ACCEPTED	RATIONALE	DISPOSITION
`rbac_minimize_*` (9)	Cluster-wide RBAC on built-in and platform roles (secrets, pod-create, SA-token, wildcard, CSR, webhook, PV)	Kubernetes / EKS built-ins and AWS-managed identities that ship with the platform and cannot be removed, plus the cortex exporter roles — the exporter's binding-create verb is contained by the admission policy in §3/§4. Human `cluster-admin` is reachable only via the jumpbox.	ACCEPTED RISK
`core_readonly_root_filesystem_enabled` (3)	Writable root filesystem on named workloads	Named acceptances only — headless agent runtimes (`cortex-agents`, `ai-assistant`, `agent-chat`) that materialize workspaces at runtime, and the transcriber model cache. Every other pod is born read-only.	ACCEPTED RISK
`core_liveness_probe_configured` **`core_readiness_probe_configured`** (4)	Probes absent on batch Job pods (scanner, seed-data, image builds)	Run-to-completion Jobs take no traffic and do not restart — probes are inapplicable; adding fake probes would game the check. Owner references verified as `Job`.	FALSE POSITIVE

Open item carried forward. The production exporter still holds the pre-remediation cluster-wide role and is *not* constrained by the dev admission policy. Retiring that role, standing up the production-scoped admission policy, and applying the platform hardening to production are the subject of a **gated production change window** and are out of scope for this report. Until that window lands, the production escalation path documented as FR-01 remains open on the production cluster.

06 Control environment

The platform's design carries much of the control burden by construction:

- **Change management (CC8.1).** All infrastructure change flows through versioned overlay files and `apply`; images are built only by CI behind a verify gate (unit tests, dependency audit at the critical level, tracked-secret scan). Ad-hoc cluster edits are prohibited by policy.
- **Tamper-evident audit trail.** Every scan is an immutable, commit-pinned audit run; each finding is a tracked issue whose lifecycle events are append-only. Issue closure is mechanical — the scanner auto-resolves only when the control re-detects as passing.
- **Least privilege by admission.** Where RBAC cannot express a constraint (create verbs are inherently cluster-wide), a built-in Kubernetes ValidatingAdmissionPolicy enforces the narrower intent at the API server, keyed to the acting identity.
- **Defense verified, not asserted.** Remediations are re-detected by scan and, for admission controls, tested by live create attempts as the constrained identity — the standard failure mode (trusting `can-i`, which ignores admission policies) is explicitly avoided.

Next steps

ITEM	STATUS
Production hardening — port the dev overlay patches, stand up a production-scoped admission policy, retire the cluster-wide exporter role	Gated on an announced change window
Production scan CronJob — bring the production cluster under continuous tracking	Scheduled with the production window
Scanner image roll — activate the widened evidence-retention change (FR-03)	Next build

ITEM	STATUS
Independent SOC 2 Type I examination by a licensed firm	Planned; this engine is the evidence base

07 Management assertion & basis of report

Management asserts that, for the platform infrastructure tier within the scope defined in §1 and over the assessment window stated above, the controls described in this report were implemented and — where marked REMEDIATED — were re-detected as operating by commit-pinned re-scan. Residual findings are dispositioned with recorded rationale in §5, and open items are disclosed in §5 and §6 rather than omitted.

Prepared by
Cortex continuous-audit engine
(Prowler CIS/ThreatScore ingestion)

Independently reviewed by
Clean-context adversarial reviewer
(challenge-and-refute pass, 11 Aug 2026)

Asserted by
Mindshine platform operations
11 August 2026

Important — nature of this report. This is a rigorous, evidence-linked **internal self-assessment** produced by Cortex's own audit engine and an independent adversarial review. It is **not** an examination or attestation performed by a licensed CPA firm, and it does not constitute a SOC 2 report under AICPA attestation standards. A SOC 2 Type I examination by an independent firm is the scheduled next step; the immutable audit trail behind this report is intended to serve as its evidence base.

Source of record. Cortex SOC 2 audit engine, platform project — audit #136 (11 Aug 2026, 11 fail / 7 manual reconciled to 0 open in-boundary). Baseline scan `20260811092657`; verification scan `20260811101704`. Remediation commits on branch `dev`: `4c8b55e` · `93ccf42` · `3f2cb22` · `c706fe5` · `0edac56` · `c4cebb8`. Supporting evidence versioned in `docs/soc2/`: access matrix, platform hand-off records, risk & vendor registers, data map, retention/disposal and resilience documentation.

Confidential — prepared for internal governance and prospective independent examination. Companion to the repository-tier SOC 2 Audit Findings Report (audit #126).