Alert Triage Workflow

Cloud security tools generate a high volume of alerts across multiple cloud accounts and services. Security engineers need a way to quickly understand which alerts matter most, investigate their impact, and track resolution without being overwhelmed by noise or forced into complex remediation flows.

The challenge is not only detecting issues, but enabling effective triage, investigation, collaboration, and resolution tracking in a way that reflects real-world security workflows.

Figma Document

User Persona

Primary user: Cloud / Security Engineer

Responsibilities:

- Monitor security alerts across environments
- Assess severity and potential impact
- Investigate alerts and gather context
- Coordinate remediation with service owners or infrastructure teams
- Verify fixes and mark alerts as resolved

Pain points:

- Alert fatigue and poor prioritization
- Lack of context when reviewing alerts
- Difficulty tracking ownership and resolution status
- Unclear audit trail for who handled an alert and when

Design Overview

The proposed design focuses on a two-screen workflow that mirrors how security teams actually work:
a triage queue for prioritization, followed by a detailed view for investigation and resolution tracking.

The system is intentionally designed as a visibility and coordination tool, not a remediation console. Technical fixes are assumed to happen outside the system, with this interface serving as the source of truth for alert status and accountability.

Screen 1: Alert List (Triage View)

The alert list acts as a prioritized work queue.

Key features:

- Alerts grouped by day to support daily on-call and triage routines
- Severity-based visual indicators for quick scanning
- Clear resolution state using a reversible checkbox
- "Resolved by" attribution to support accountability and handoffs

Design rationale:

Grouping alerts by day reflects how engineers typically process alerts in time-based batches, while severity indicators ensure that critical issues are not buried. Resolution is treated as a reversible state, allowing alerts to be reopened if remediation is incomplete or incorrect.

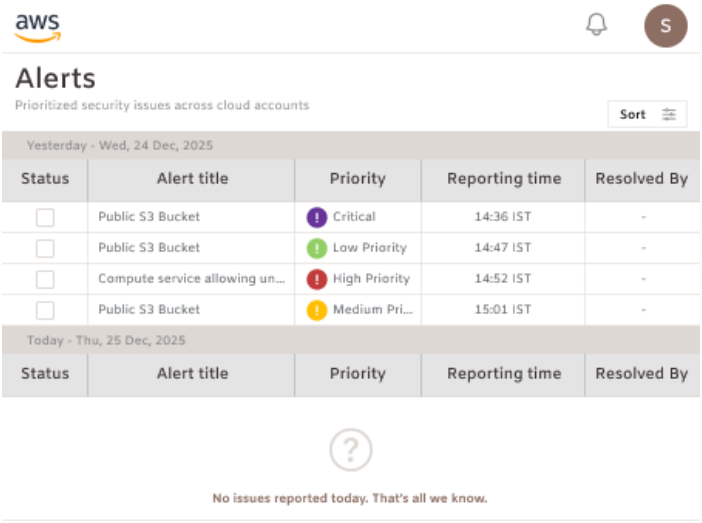
Alert Prioritization Logic:

Alert priority is derived from a combination of severity, resource criticality, and exposure duration, rather than raw alert volume alone. Severity reflects the inherent risk of the issue (e.g., public access, unrestricted network exposure), while resource criticality accounts for whether the affected asset is production-facing, sensitive, or business-critical. Exposure duration acts as a modifier, ensuring that long-standing unresolved risks are surfaced even if their initial severity is lower. This approach ensures

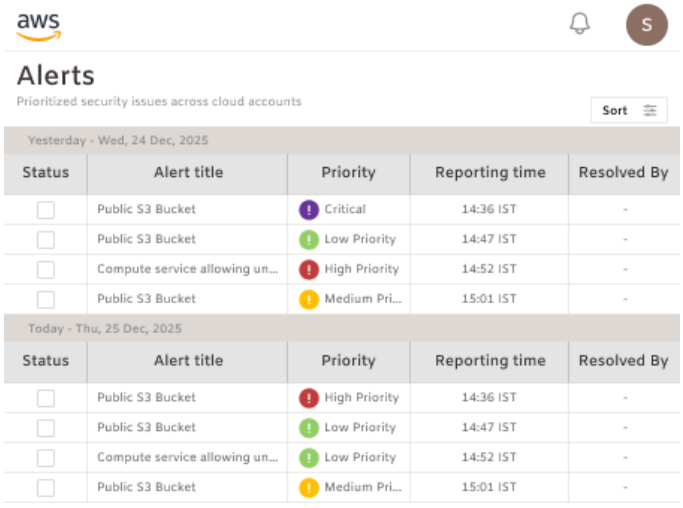
that alerts with the highest potential impact are surfaced first, while still allowing security engineers to quickly scan and batch lower-risk issues.

Designs:

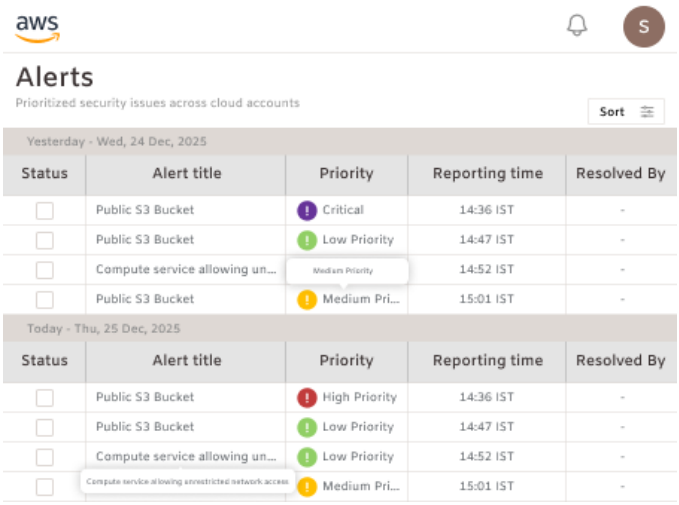
(Please do not rely on screenshots. Reference the figma design links)



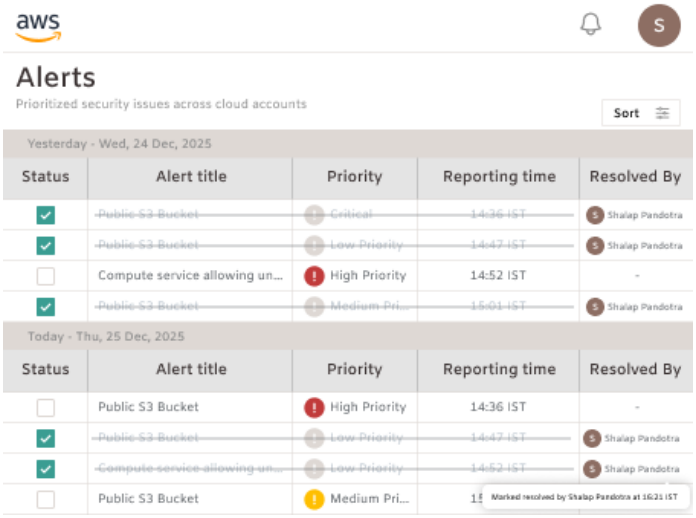
[Zero State]



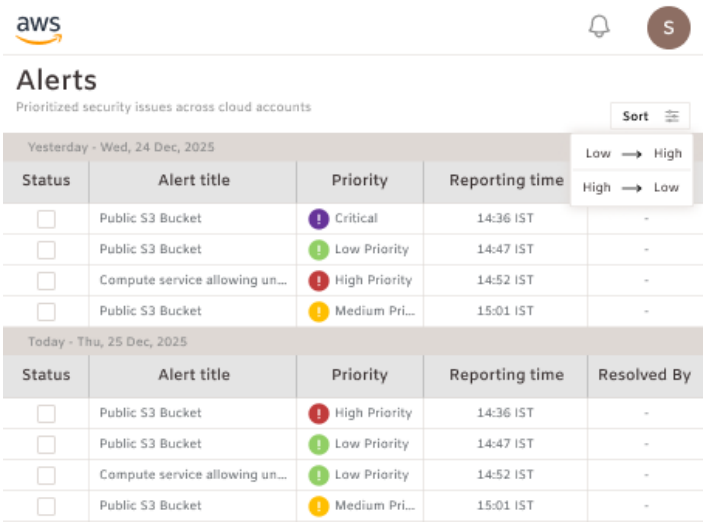
[Triage view]



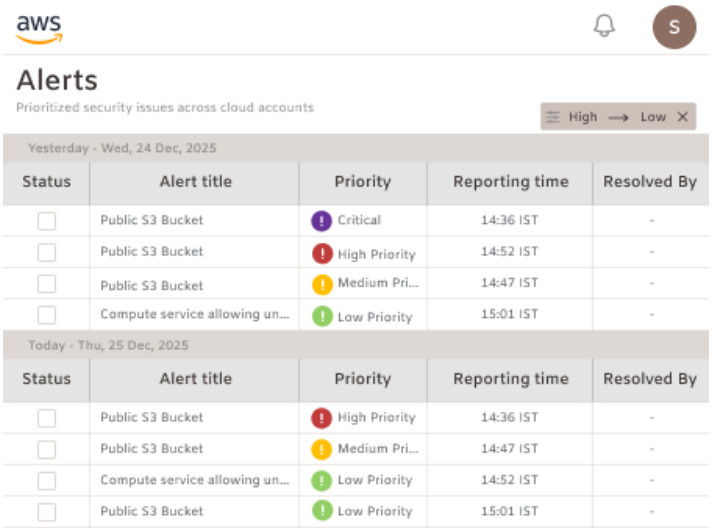
[Hover tooltips]



[Resolved State]



Sorting Options



Sorted State

Screen 2: Alert Details (Investigation & Resolution)

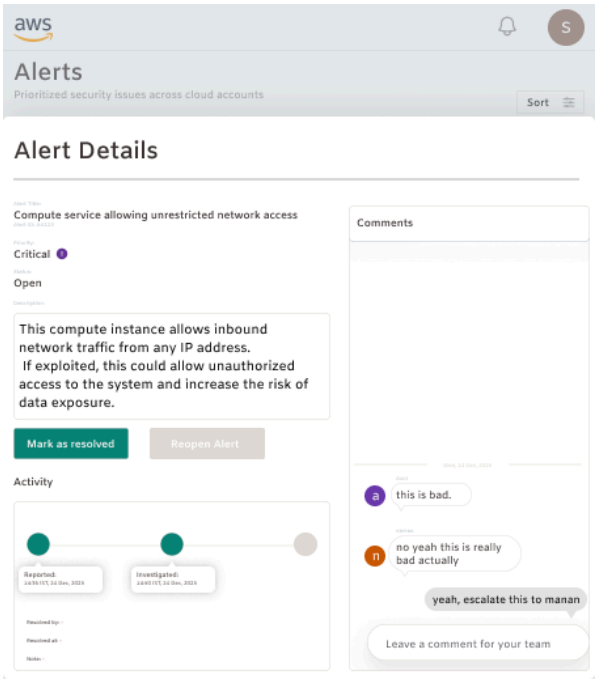
The detail view provides all necessary context to investigate and close an alert.

Key features:

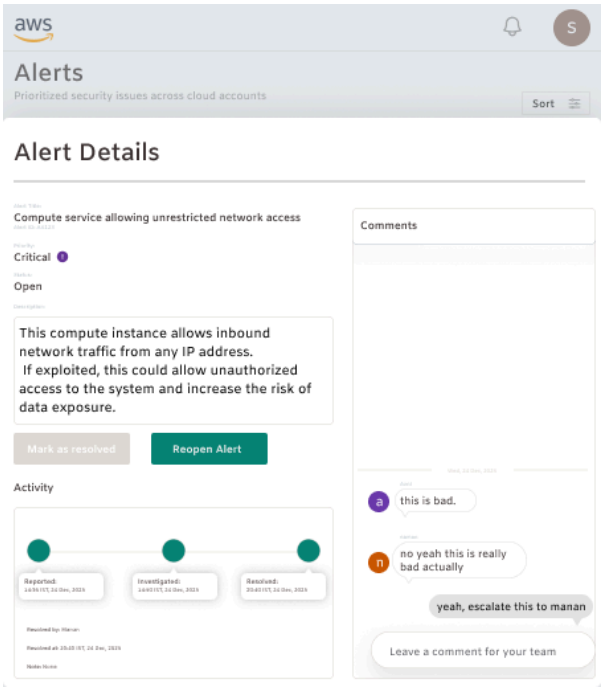
- Plain-English explanation of the issue and its potential impact
- Clear display of affected resources and timestamps
- Internal discussion/comments for collaboration and escalation
- Explicit resolution controls and audit trail (who resolved the alert and when)

Design rationale:
This view focuses on understanding and coordination rather than direct remediation. By surfacing risk context and supporting team discussion, the design reflects real-world workflows where fixes are applied externally and verified before an alert is resolved.

Designs:



[Details View - unresolved]



[Details View - resolved]

Success Metrics

The effectiveness of this design could be measured by:

- Time to first action on newly detected alerts
- Percentage of alerts resolved vs reopened
- Reduction in unresolved critical alerts over time
- Clarity of ownership and resolution accountability

Development Action Items

- Implement an alert state machine (open → resolved → reopened) with reversible actions
- Maintain an immutable audit log for status changes, ownership, and resolution events
- Support priority calculation based on severity, asset criticality, and exposure duration
- Enable alert ownership assignment and reassignment
- Provide a lightweight comment system with timestamps and user attribution
- Expose sorting and filtering APIs to support scalable alert volumes