LOOPX ONEOPS

Intelligent Operations Console

Concept brief

A unified command centre for service health, incidents, changes and controlled operational automation.

Purpose

LoopX OneOps brings fragmented operational information into one clear workspace. It connects existing monitoring, cloud, deployment and ticketing tools so teams can understand service impact, coordinate response and execute approved actions without replacing every current platform.

Business need

- Operations data is distributed across monitoring, logging, CI/CD, cloud and ticketing systems.
- Alert volume makes it difficult to identify the event that matters most.
- Incident response depends on manual evidence collection and individual experience.
- Operational actions are often performed through separate scripts with limited governance.

Product outcomes

01	02	03
Faster response	**Less noise**	**Safer action**
Reduce time spent gathering context and coordinating investigation.	Group related signals and prioritise incidents by service impact.	Use approved runbooks, permissions and complete audit records.

Product scope

The console acts as an orchestration and decision layer above the organisation's existing operational toolset.

Unified overview

Service health, availability, active incidents, infrastructure status and recent changes.

Service catalogue

Ownership, dependencies, environments, criticality, support contacts and SLOs.

Alert and incident workspace

Signal ingestion, deduplication, correlation, assignment, timeline and evidence.

Application and infrastructure view

Metrics, logs, traces, cloud resources, Kubernetes, databases and network status.

Deployment intelligence

Release history, change-to-incident correlation, risk indication and rollback links.

Controlled automation

Approved runbooks, human approval, execution status, verification and rollback controls.

Security and governance

Identity risks, configuration findings, certificate expiry, access control and audit history.

Reporting

Operational health, availability, incident trends, response time and automation effectiveness.

Primary users

SRE and DevOps teams | Cloud and infrastructure operations | Application support | Security operations | Technology leadership

Typical operating flow

1	2	3	4	5
Collect	**Correlate**	**Assess**	**Act**	**Review**
Ingest health, alerts and changes.	Group related events by service and time.	Determine impact, ownership and urgency.	Run an approved response or assign work.	Verify recovery and record the outcome.

Control model

- Automatic - limited, low-risk actions under predefined policy.
- Approval required - an authorised user reviews the action before execution.
- Manual only - the console provides context and guidance but does not execute.

Conceptual architecture

Experience layer

Web console, incident workspace, dashboards, reporting and role-based views.

Operations layer

Service catalogue, alert correlation, incident workflow, change context and runbook orchestration.

Platform layer

API services, event processing, PostgreSQL, cache, search/indexing, identity and audit logging.

Integration layer

Datadog, cloud platforms, Kubernetes, GitLab/GitHub/Jenkins, Jira/ServiceNow, Teams/Slack and automation tools.

MVP definition

Included

- Unified dashboard for a selected production service portfolio.
- Service catalogue with ownership, criticality and dependencies.
- Alert ingestion, deduplication and basic correlation.
- Incident workspace with timeline, evidence and assignments.
- Deployment correlation from one CI/CD platform.
- Two to five approved remediation workflows.
- Notifications and a complete operational audit log.

Not included

- Fully autonomous production remediation.
- Full replacement of monitoring, ITSM or security platforms.
- Enterprise-wide predictive forecasting in the first release.
- Complete cloud cost-management capability.
- Multi-tenant commercial billing and marketplace functionality.

Delivery approach

Phase 1 - Discover

Confirm pilot services, users, integrations, baselines and operational workflows.

Phase 2 - Pilot

Deliver the core dashboard, incident workspace, integrations and controlled runbooks.

Phase 3 - Expand

Improve correlation, reporting, governance and coverage based on measured pilot results.

Pilot success measures

Mean time to acknowledge | Mean time to resolve | Alert reduction ratio | Runbook success rate | Availability impact | User adoption

Recommended next step: select one operational team and a limited set of production services for an eight-to-twelve-week pilot.