PORTFOLIO / TECHNICAL CASE STUDY BMS IT MONITORING · 2026

BMS IT Monitoring

Network Operations & Infrastructure Monitoring Platform

A real-time NOC platform that brings device inventory, ICMP/SNMP telemetry, syslog events, topology discovery, incident correlation, SLA reporting, and configuration readiness into one operational console. This document explains the problem, the architecture, the implementation decisions, and the evidence behind them.

ROLE Full-stack engineering · System design · Interface design	**STACK** Node.js · Express · SQLite · InfluxDB v2 · Telegraf · WebSocket
SCOPE 8 views · 43 REST routes · 10 relational tables	**ENVIRONMENT** Private four-node reference deployment

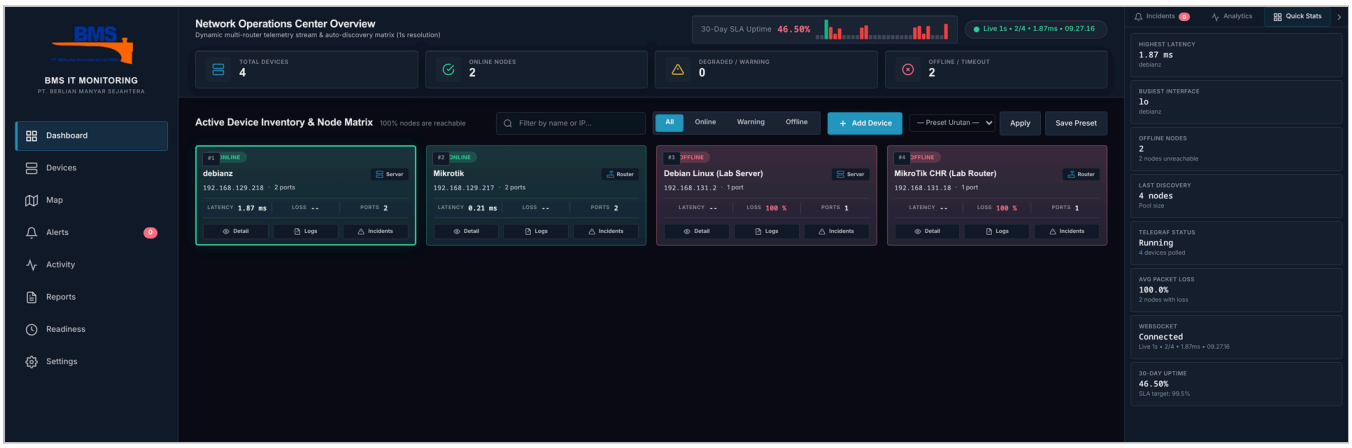


FIGURE 1

The Network Operations Center overview with the Quick Stats rail open. Four nodes were configured in the private reference environment; the capture records the state at that moment, including devices that were offline.

CONTENTS 10 SECTIONS · 14 FIGURES

01	Project overview	02	Problem & solution
03	System architecture	04	Technology stack
05	Key capabilities	06	Interface showcase
07	Engineering highlights	08	Challenges & solutions
09	Skills demonstrated	10	Project summary

01 / Project overview

A single operational reading of a distributed network.

The network has plenty of signals. What it lacks is a single, trustworthy reading.

Monitoring usually fragments across tools: one terminal for ping, one for SNMP counters, one for logs, a spreadsheet for inventory, and a separate mental model for what is connected to what. **BMS IT Monitoring collapses those surfaces into one console.**

Each device is modelled as an inventory record, a telemetry stream, a health state, and a source of evidence. Every event can be traced back to a device. Every incident keeps its root cause, evidence, and status history. Every report is built from stored observations rather than manual notes.

It is not a dashboard that only answers *"is it up?"* — it is an operating surface that answers *"what changed, why did it matter, and what should I do next?"*

AT A GLANCE

OPERATIONAL VIEWS **8**	REST ROUTES **43**	TELEMETRY CADENCE **1s**
RELATIONAL TABLES **10**	DEVICE CATEGORIES **28**	REFERENCE NODES **4**

Implementation facts from the repository. The reference environment is a private four-node lab; no production fleet scale or performance is implied.

How the system turns signals into action

01 **Observe**
Telegraf and direct probes collect ICMP, SNMP interface counters, system inventory, and syslog.

02 **Correlate**
Health rules, alert deduplication, and log rules turn raw signals into incidents and sessions.

03 **Explain**
Root cause, evidence, status history, normalized logs, and raw log fallback stay attached to the record.

04 **Act**
Readiness gaps become copyable fixes; topology provides context; reports make the trend visible.

Scope of this case study

BUILT	A full-stack platform: collection, time-series and relational storage, REST + WebSocket delivery, and a framework-free NOC interface with detail overlays and operational workflows.
VALIDATED	A running four-node reference environment was used for the captures. The screenshots record real states, including offline targets, rather than staged healthy states.
CURRENT BOUNDARY	Authentication/RBAC, hosted multi-tenancy, and public-cloud deployment are outside the audited scope. The project focuses on the monitoring and operations workflow itself.

02 / Problem & solution

The hard part is not collecting a metric. It is earning attention.

Operators do not need another chart. They need the next correct action, with enough context to trust it.

Each product decision maps to an operational failure mode: noise, missing context, stale topology, or configuration drift. The responses below are design decisions visible in the implementation, not claims of a production outcome.

01

Signals live in different tools

RESPONSE One device model ties inventory, telemetry, logs, incidents, and topology to the same record. The console starts from a device card and drills into one intelligence view.

02

A flapping device can page the on-call all night

RESPONSE A state transition requires **two consecutive evaluations**. New alerts of the same type are suppressed for **five minutes**; recovery is confirmed the same way.

03

A missing value is not zero

RESPONSE Nullable telemetry and explicit no-data states keep silence separate from a healthy zero. Reports show no data instead of inventing a perfect day.

04

The map goes stale immediately after discovery

RESPONSE LLDP/CDP/MNDP walks are enriched with ARP, scheduled every five minutes, and every link carries `last_seen` and `stale` status. Unmanaged neighbors stay visible.

05

Configuration drift becomes an outage

RESPONSE Three-layer readiness checks test reachability, SNMP, and syslog forwarding, then generate platform-aware commands and verify that fresh logs actually arrived.

06

History is hard to defend after the fact

RESPONSE Incidents persist root cause, evidence, and status history in SQLite. Archive and restore keep the record available without losing the recovery path.

> *Designed for the moment after the alert, when context matters more than another chart.*

03 / System architecture

Two clocks, one operating picture.

A one-second loop keeps the NOC alive. A durable relational model keeps the story intact.

Hot telemetry and operational state are deliberately separated. Time-series data is written and read at high frequency; inventory, incidents, logs, topology links, and settings live in a relational store where relationships and history matter more than raw write speed.

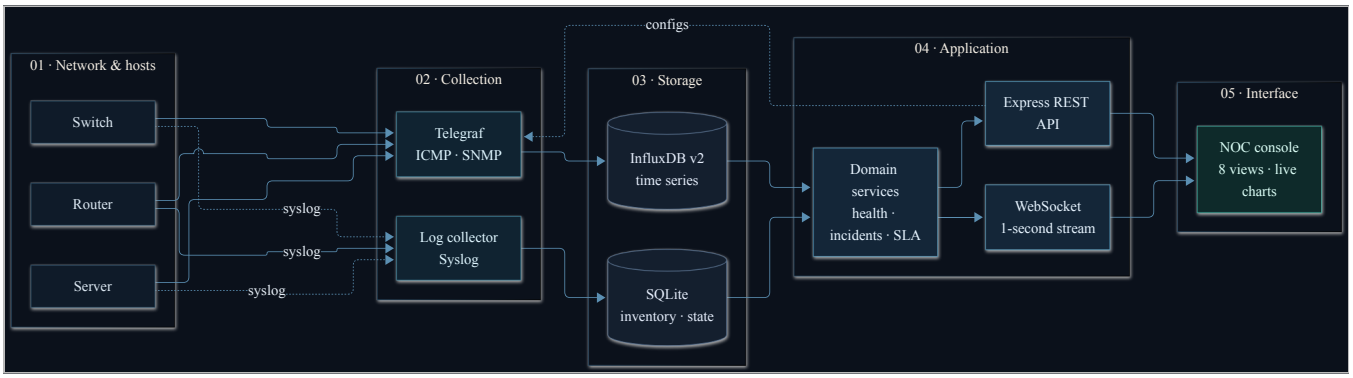


FIGURE 2

Five layers: hosts, collection, storage, application services, and the operator interface. Rendered from Mermaid source; the definitions live in docs/PORTFOLIO.md.

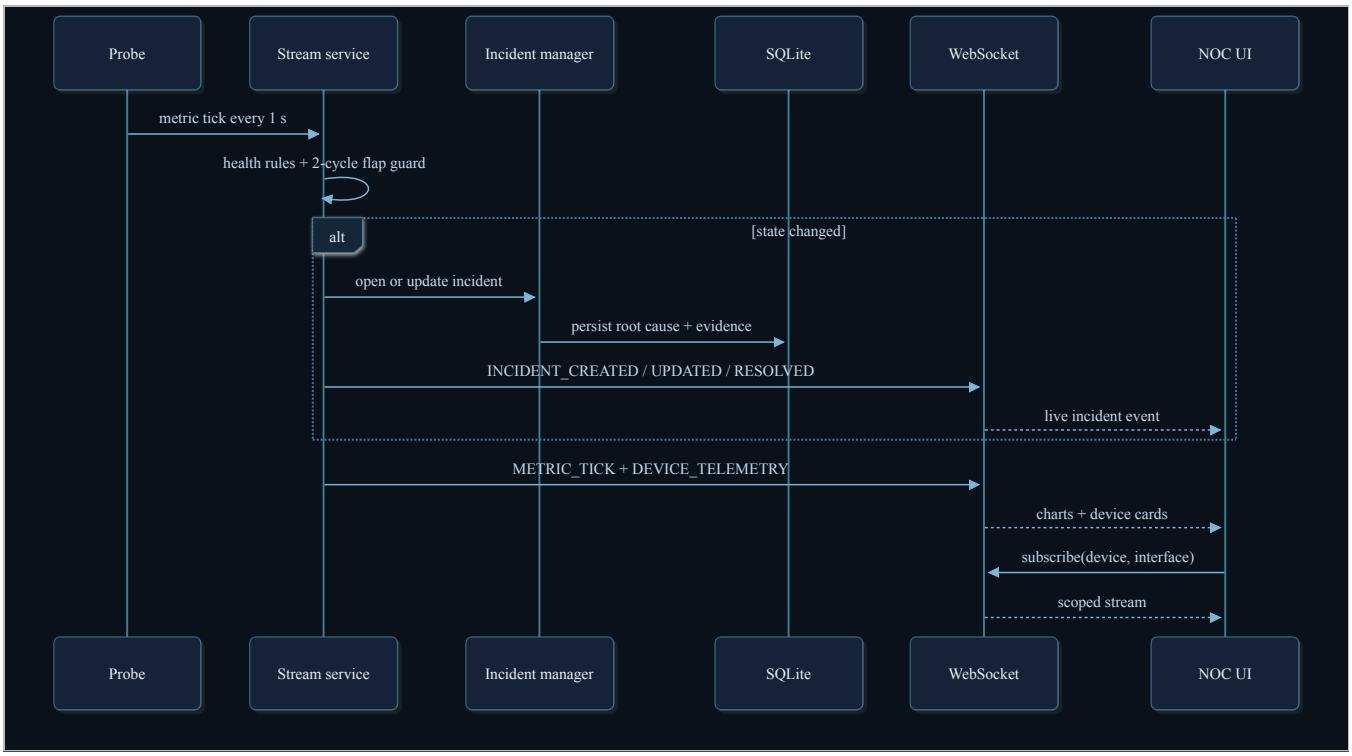


FIGURE 3

A state change becomes a durable incident, then a live event in the console. The one-second telemetry loop and the slower incident workflow are shown in the same path. Rendered from Mermaid source.

Two stores, two jobs

InfluxDB v2

TIME SERIES

High-frequency poller data: latency, packet loss, interface throughput, system resource samples, and the windows behind charts and SLA calculations. **Optimised for time.**

SQLite

OPERATIONAL STATE

Device inventory, interfaces, alerts, events, incidents, normalized logs, topology links, threat sessions, daily uptime, and settings. **Optimised for relationships and history.**

What moves through the system

01 / HOT PATH

One-second stream

The stream service evaluates devices in bounded parallel chunks and pushes a scoped tick to every client.

02 / COLLECTION

Generated agents

Telegraf configs are produced per device; a syslog collector runs alongside for event and threat correlation.

03 / DURABLE STATE

Relational truth

SQLite holds inventory, interface records, events, incidents, logs, links, threat sessions, and settings.

04 / INTERFACE

REST + push

REST handles reads and mutations; WebSocket carries live metrics and incident events into Chart.js and vis-network.

04 / Technology stack

Chosen for the shape of the workload.

Small runtime surface, explicit data boundaries, and browser-native visualisation.

Every component has a job in the monitoring path. The application interface is framework-free; the portfolio itself is a static document and does not import the application stylesheet or call its APIs.

FRONTEND	Framework-free SPA single shell · Chart.js 4 sliding charts · vis-network 9 topology graph · Tailwind 4 + custom CSS design tokens · WebSocket live updates
BACKEND	Node.js 18+ runtime · Express 4 REST API · ws 8 socket server · better-sqlite3 11 relational store · InfluxDB client time-series reads
DATA	SQLite · WAL operational state · InfluxDB v2 metrics bucket · Daily uptime SLA aggregation · 10 tables indexed model · Auto-applied schema zero setup

MONITORING	Telegraf agent · ICMP latency / loss · SNMP v2c interface counters · Syslog UDP/TCP :5514 event intake · SNMP traps optional source
NETWORKING	LLDP-MIB neighbor discovery · CDP / MNDP vendor discovery · ARP / IP-MIB address enrichment · IF-MIB / ifXTable inventory + counters · Link freshness stale / last seen
TOOLING	Playwright UI verification · Tailwind CLI CSS build · systemd Telegraf reload · npm runtime scripts · Mermaid architecture diagrams

The original application uses no frontend framework. This portfolio uses local CSS and a small JavaScript helper only for the print action.

05 / Key capabilities

Eight surfaces. One operating model.

The views are not separate mini-products. They share the same device identity, incident record, telemetry history, and topology graph.

01 Network Operations Dashboard

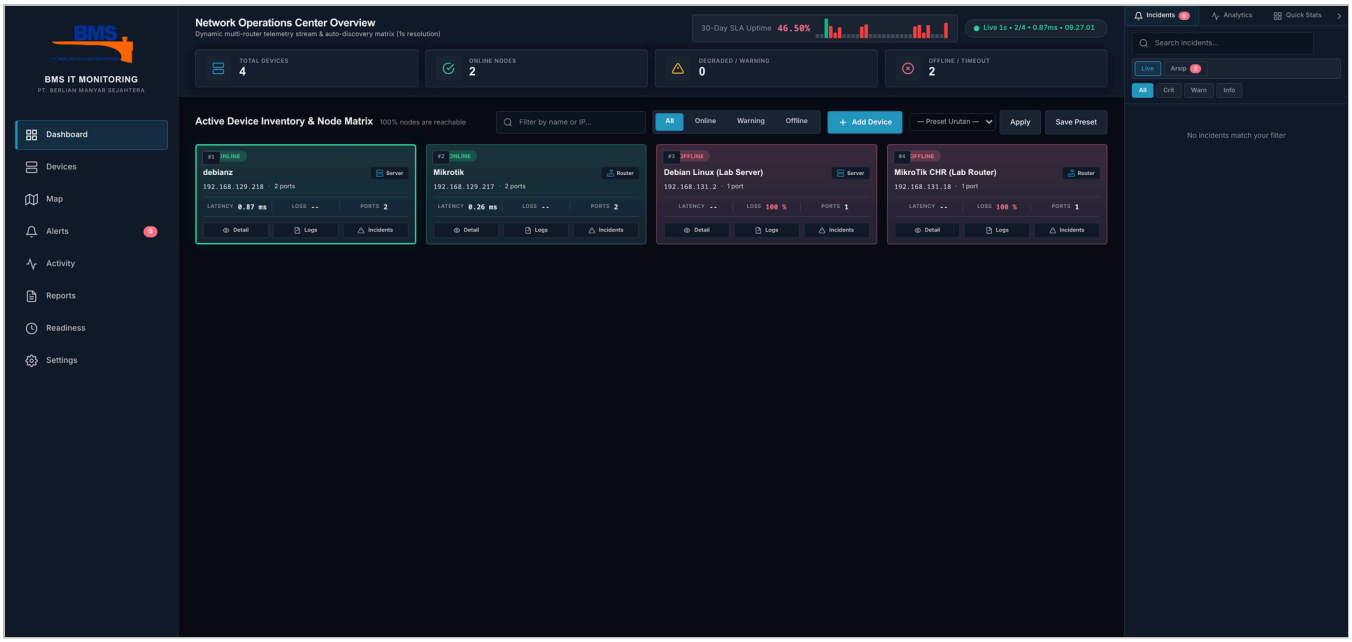


FIGURE 4

Five-zone cockpit: KPI strip, live device grid, incident rail, analytics, and Quick Stats. The analytics area is empty in this capture; no data was staged.

A five-zone cockpit for the state of the network: KPIs, inventory, incidents, analytics, and quick statistics. The dashboard is designed to make the next action obvious before the operator asks.

— One-second WebSocket updates for device cards and charts.

— Severity filters, search, drag order, and saved device presets.

— 60-second sliding throughput window with live in/out series.

WEBSOCKET · CHART.JS · 1-SECOND UPDATES

02 Device Management

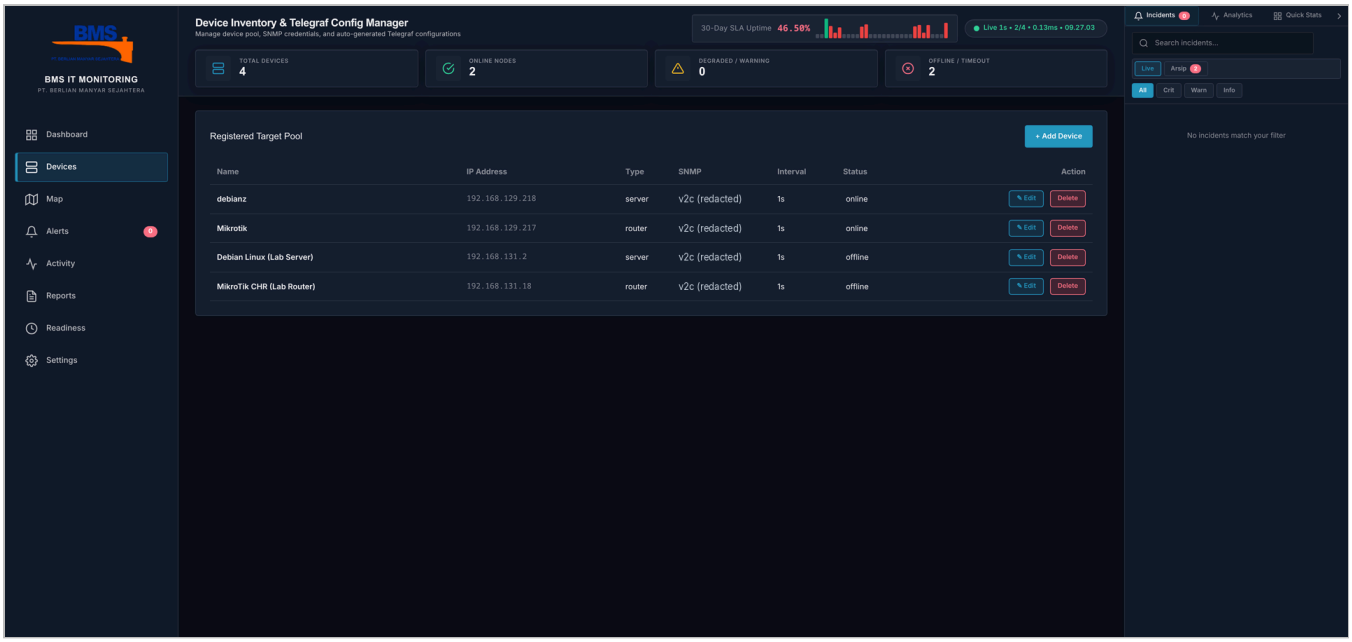


FIGURE 5 Registered Target Pool with per-device status and actions. The SNMP community value is redacted in this portfolio copy.

Onboard a target, prove it is reachable, and let the platform generate its observability configuration. Saving a device is the boundary between inventory and active collection.

- ICMP + SNMP connection test before deploy.
- Add, edit, delete, polling interval, and credentials.
- 28 device categories with per-type iconography.
- Saving a device generates a Telegraf config.

CRUD · TELEGRAF · 28 CATEGORIES

03 Network Topology

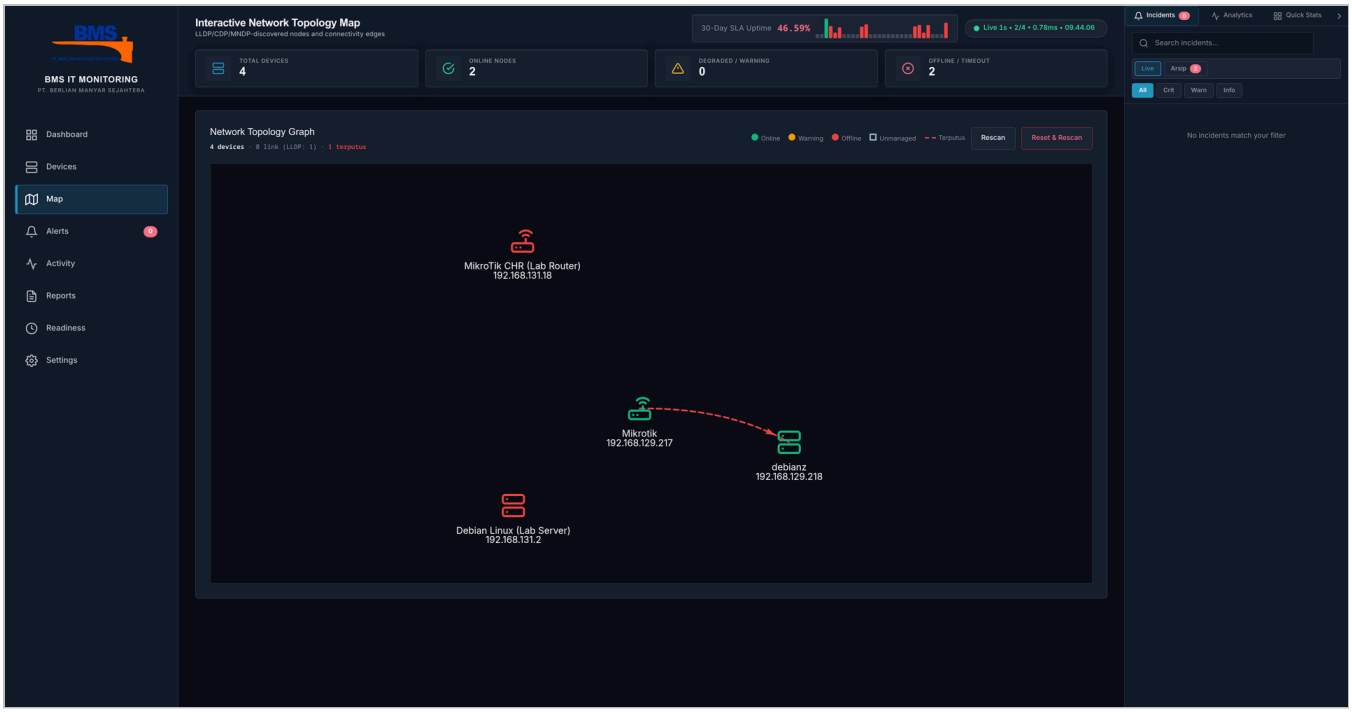


FIGURE 6 Managed devices and discovered links, with status legend and rescan controls.

A discovery-driven map of managed and unmanaged neighbors, with link freshness made visible instead of assumed. The map is allowed to show what discovery does not know.

- LLDP-MIB, Cisco CDP, and MikroTik MNDP walks.
- ARP/IP-MIB enrichment for neighbor addresses.
- Stale links, unmanaged nodes, manual IPv4 override.
- Five-minute discovery cycle and manual rescan.

LLDP · CDP · MNDP · FRESHNESS

04 Alert & Incident Center

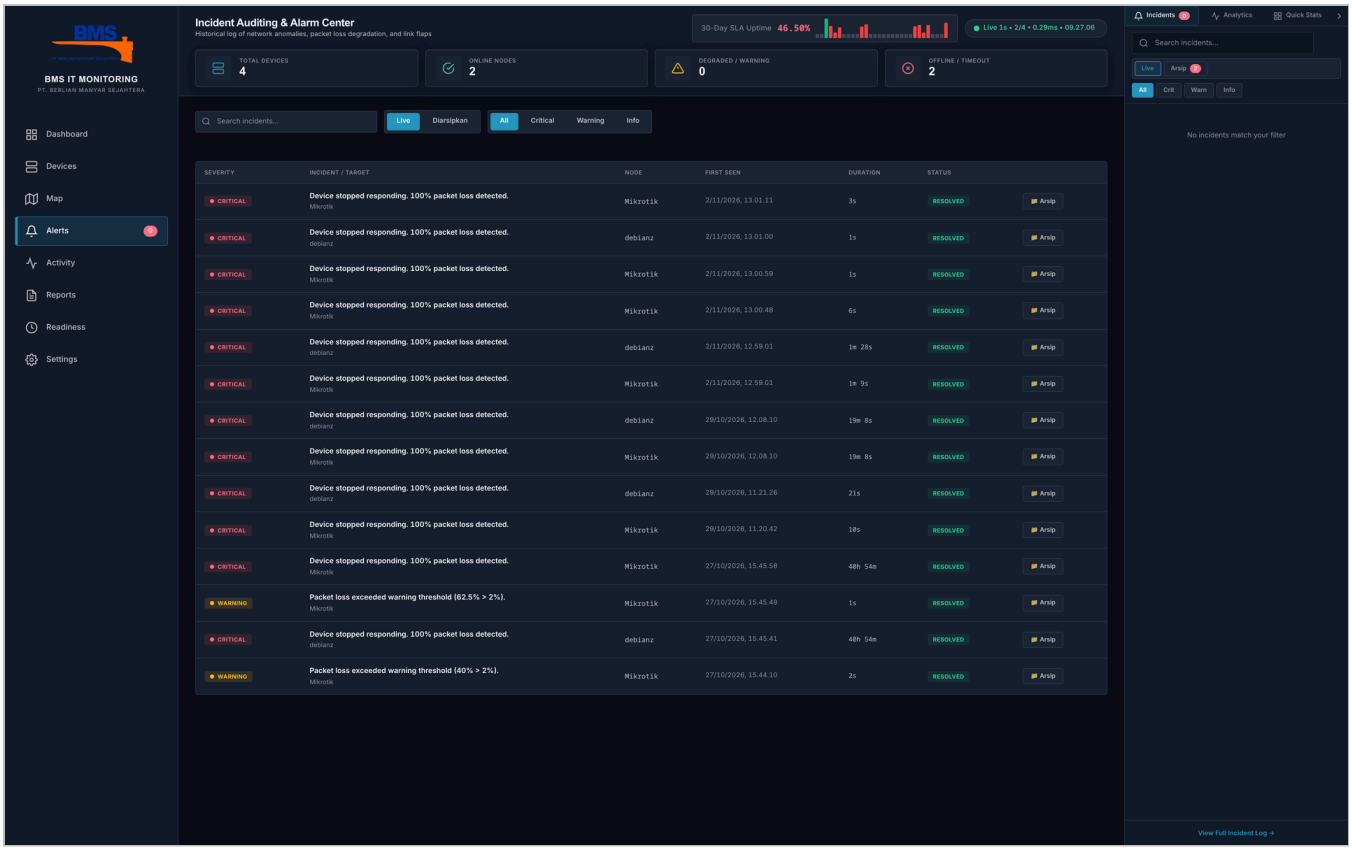


FIGURE 7 Incident ledger with severity, duration, state, and archive action. This capture is from the resolved view.

An incident ledger, not just an alert list. Every row opens the evidence behind the state change, so on-call can read the story without reconstructing it from separate tools.

- Severity/status filters, search, and archived mode.
- Two-cycle confirmation and five-minute duplicate suppression.
- Root cause, evidence, duration, and full status history.
- Threat timeline with acknowledge action.

LIFECYCLE · DEDUPE · EVIDENCE

05 Activity Log

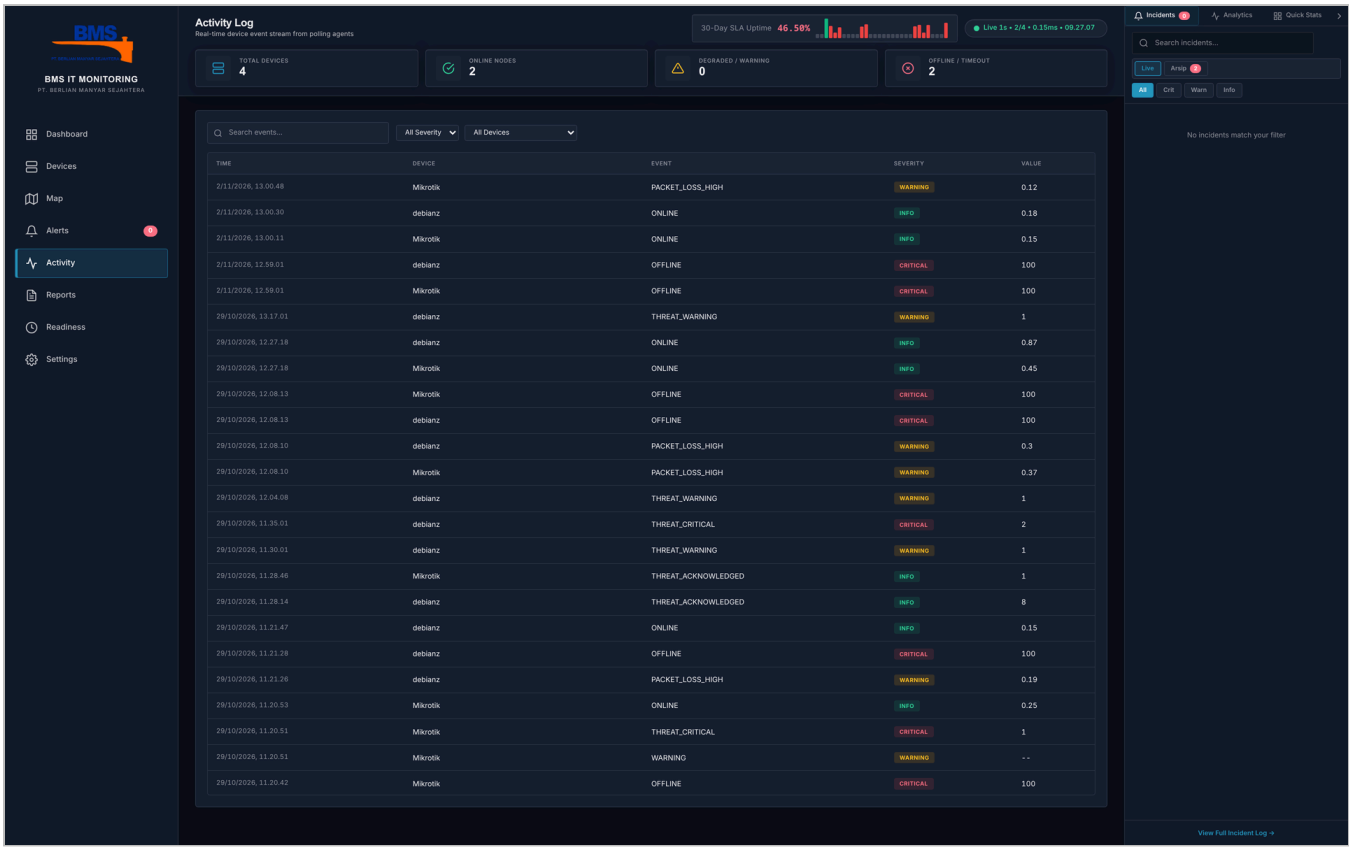


FIGURE 8 A durable event ledger across devices, event types, severities, and values.

A durable event ledger that answers what happened, where, and when without leaving the console. Events are evidence, not an afterthought to the dashboard.

— Device, event type, severity, and value columns.

— Clickable rows jump to the related active incident.

— Server-side device/severity filters plus local search.

SQLITE · EVENTS · AUDIT TRAIL

06 SLA & Performance Reports

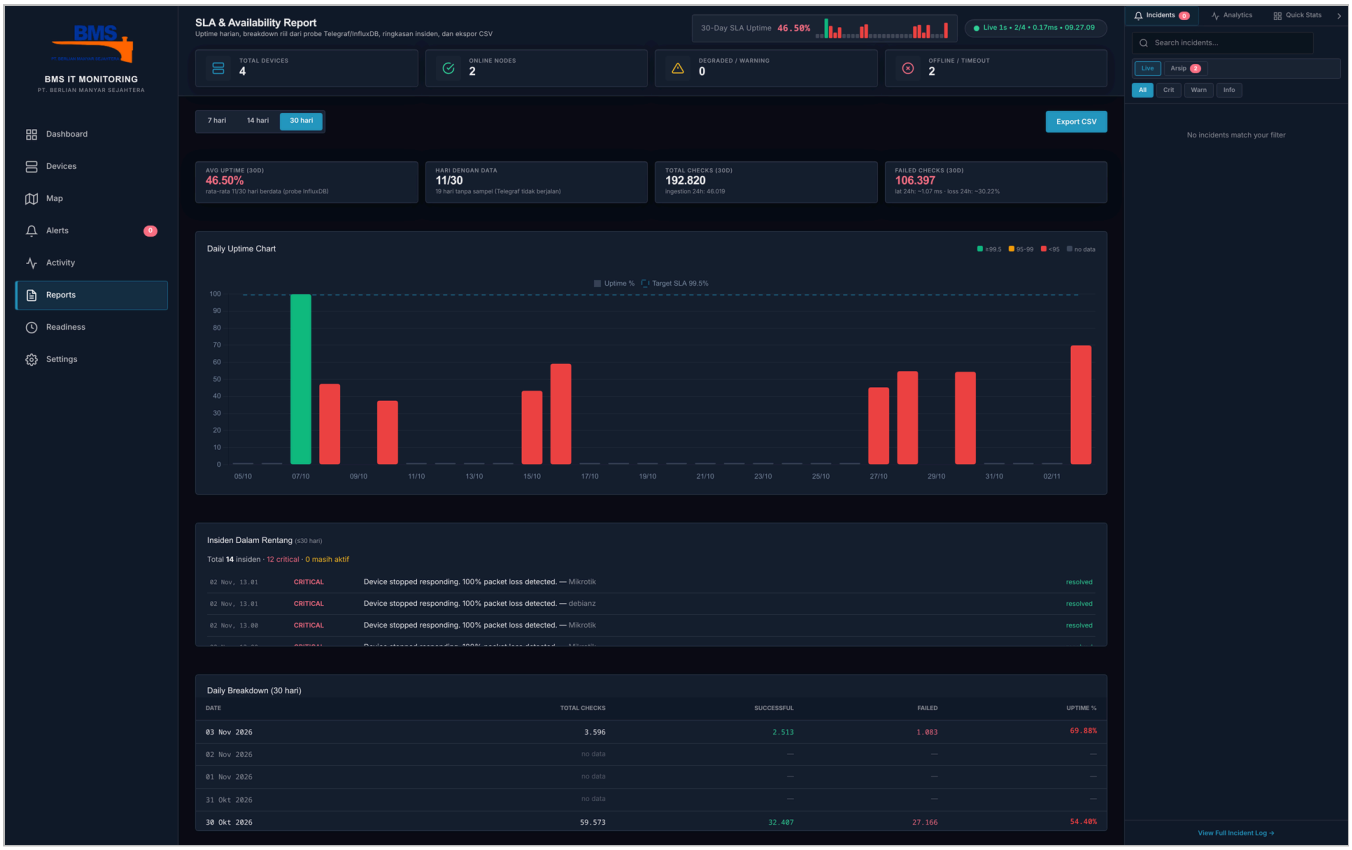


FIGURE 9

30-day uptime, daily breakdown, incident context, and CSV export. The percentages reflect the reference environment, not a production fleet.

Seven-, fourteen-, or thirty-day uptime with daily breakdown, incident context, and an exportable record. The report is built from observations rather than manually maintained notes.

— 99.5% target line over daily probe counts.

— Missing days remain no data, not zero.

— In-range incident summary and CSV export.

SLA 99.5% · CSV · DAILY UPTIME

07 Configuration Readiness

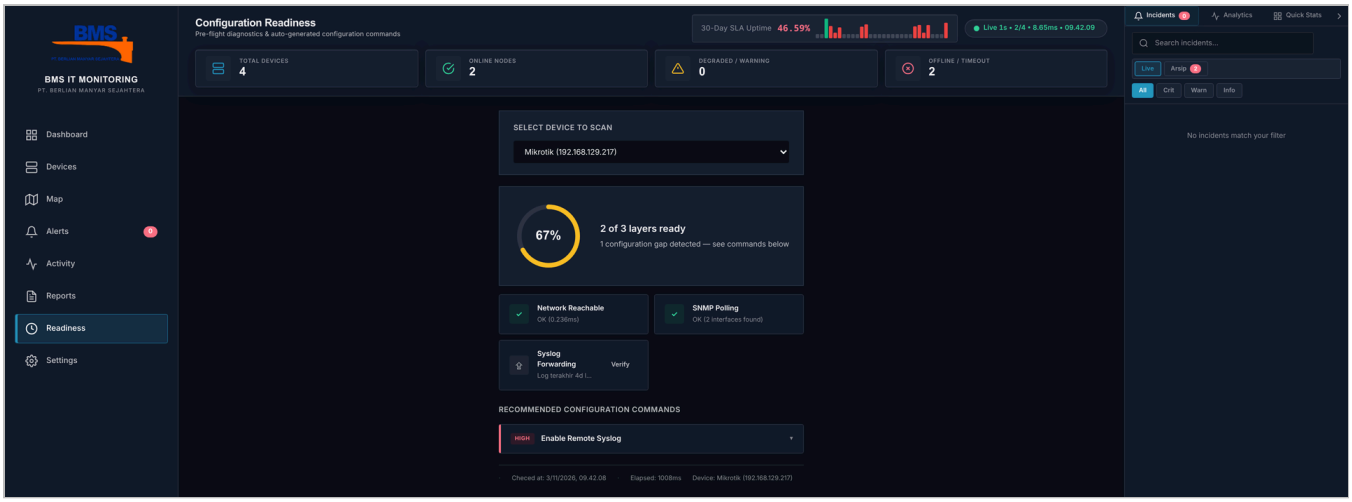


FIGURE 10 A real partial-success scan: 2 of 3 layers ready, with the syslog gap and generated remediation command visible.

A check that asks whether a device is actually observable, not just whether it answers a ping. The gap is paired with an actionable fix.

— Network, SNMP, and syslog checks in one scan.

— Copy-to-clipboard and active syslog verification.

— Gap list with urgency and generated commands.

3 LAYERS · SYSLOG · REMEDIATION

08 Infrastructure Settings

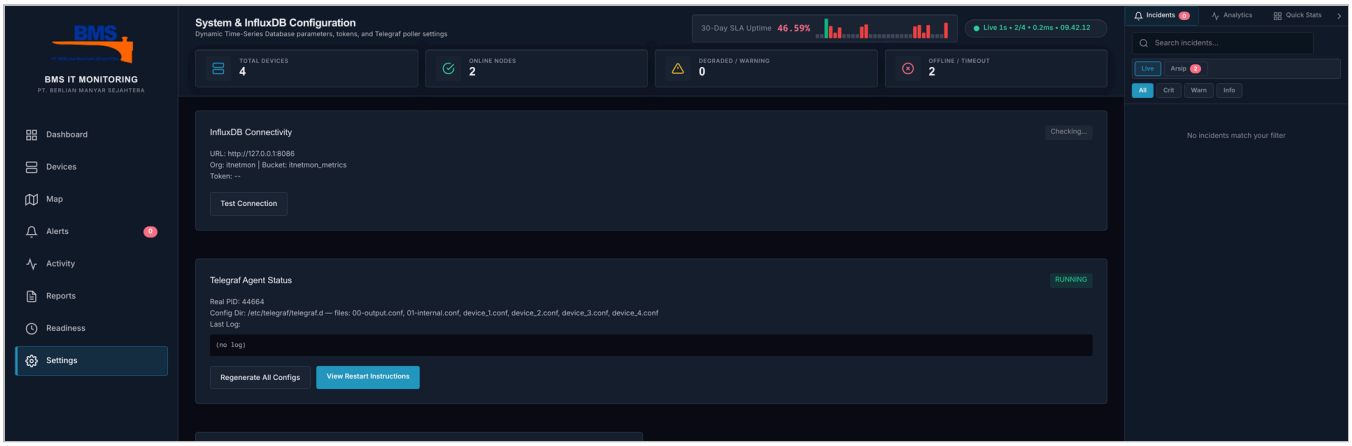


FIGURE 11

InfluxDB connection, Telegraf status, generated config inventory, and runtime settings. The capture shows a connectivity check in progress, not proof of a successful connection.

The operational boundary: time-series connection, agent configuration, and runtime state in one place. Tokens are never echoed back after they are stored.

— InfluxDB URL, org, bucket, and token rotation.

— Telegraf PID, config directory, and generated file inventory.

— Regenerate all configs and test connectivity.

INFLUXDB · TELEGRAF · OPERATIONS

06 / Interface showcase

Detail is the differentiator.

The console is designed for the moment after the alert, when the operator needs context rather than another summary.

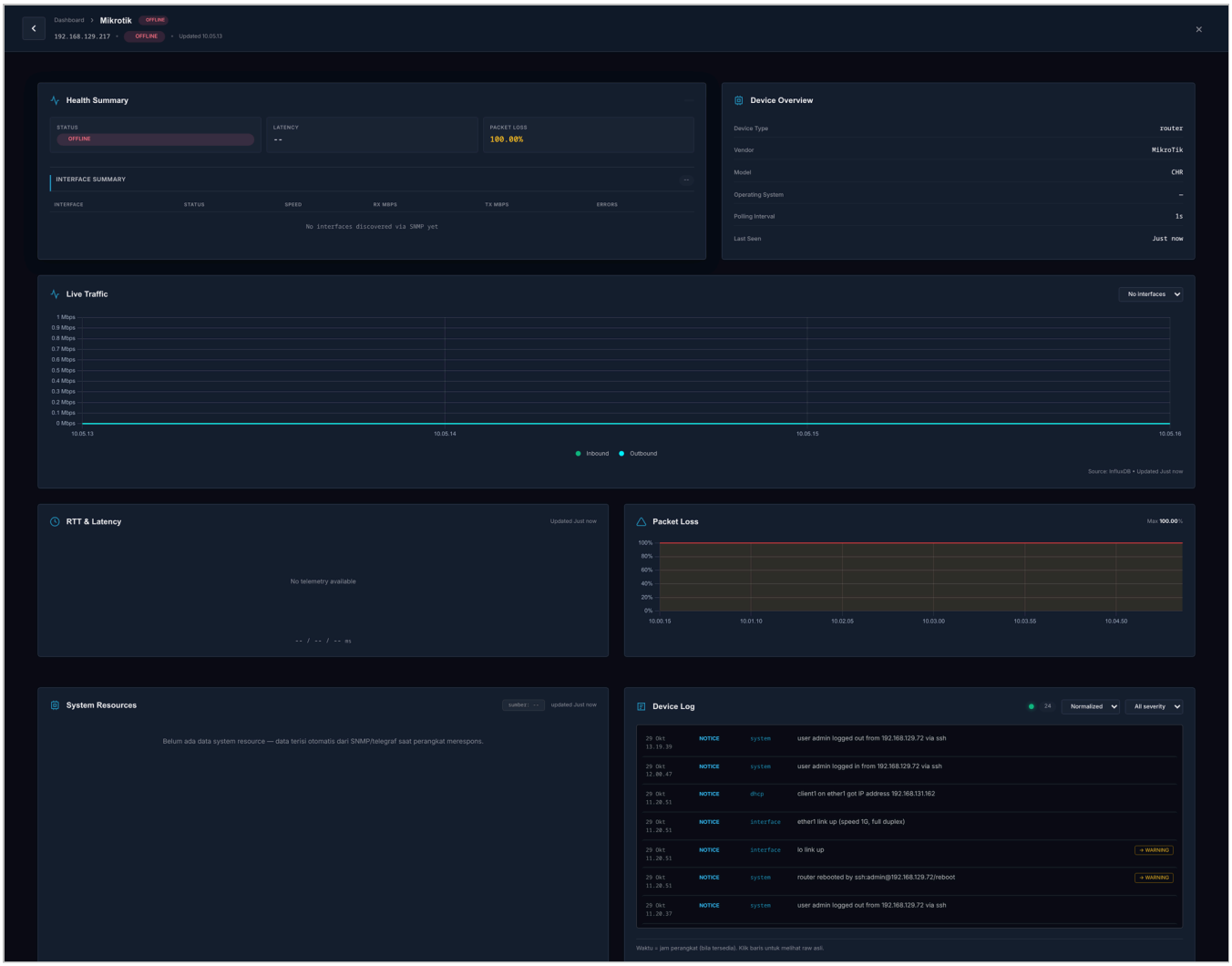


FIGURE 12

Device Intelligence Center, captured with the target offline: health, 100% loss, retained logs, and the surrounding operational record remain in one workspace. This is an outage capture, not a live traffic claim.

One drill-down view joins the live signal to the operational record: health, loss, live traffic, system resources, normalized and raw logs, events, threat sessions, incident history, and per-interface counters.

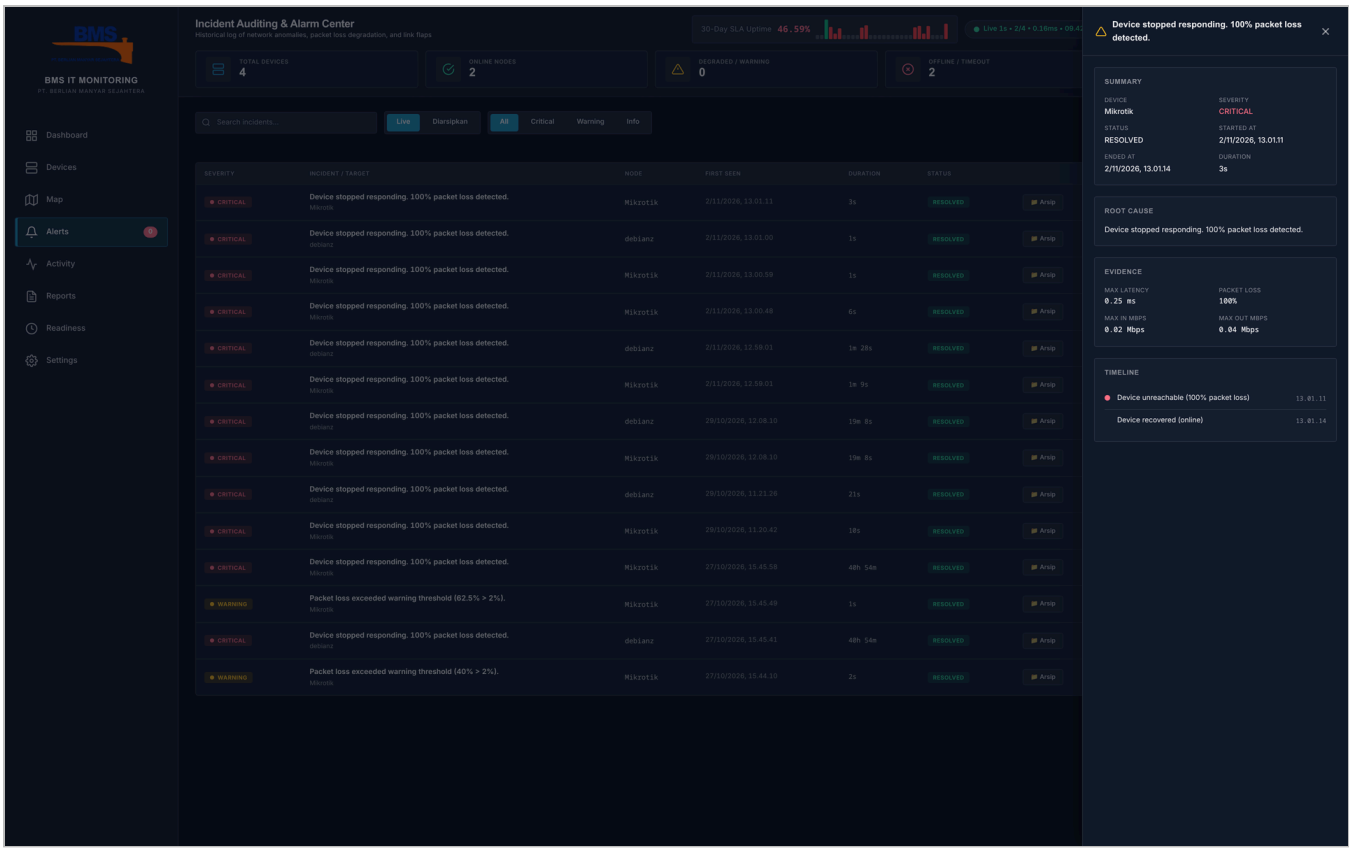


FIGURE 13

Incident drawer: root cause, evidence, duration, and status history remain attached to the record. On-call can read the story without leaving the console.

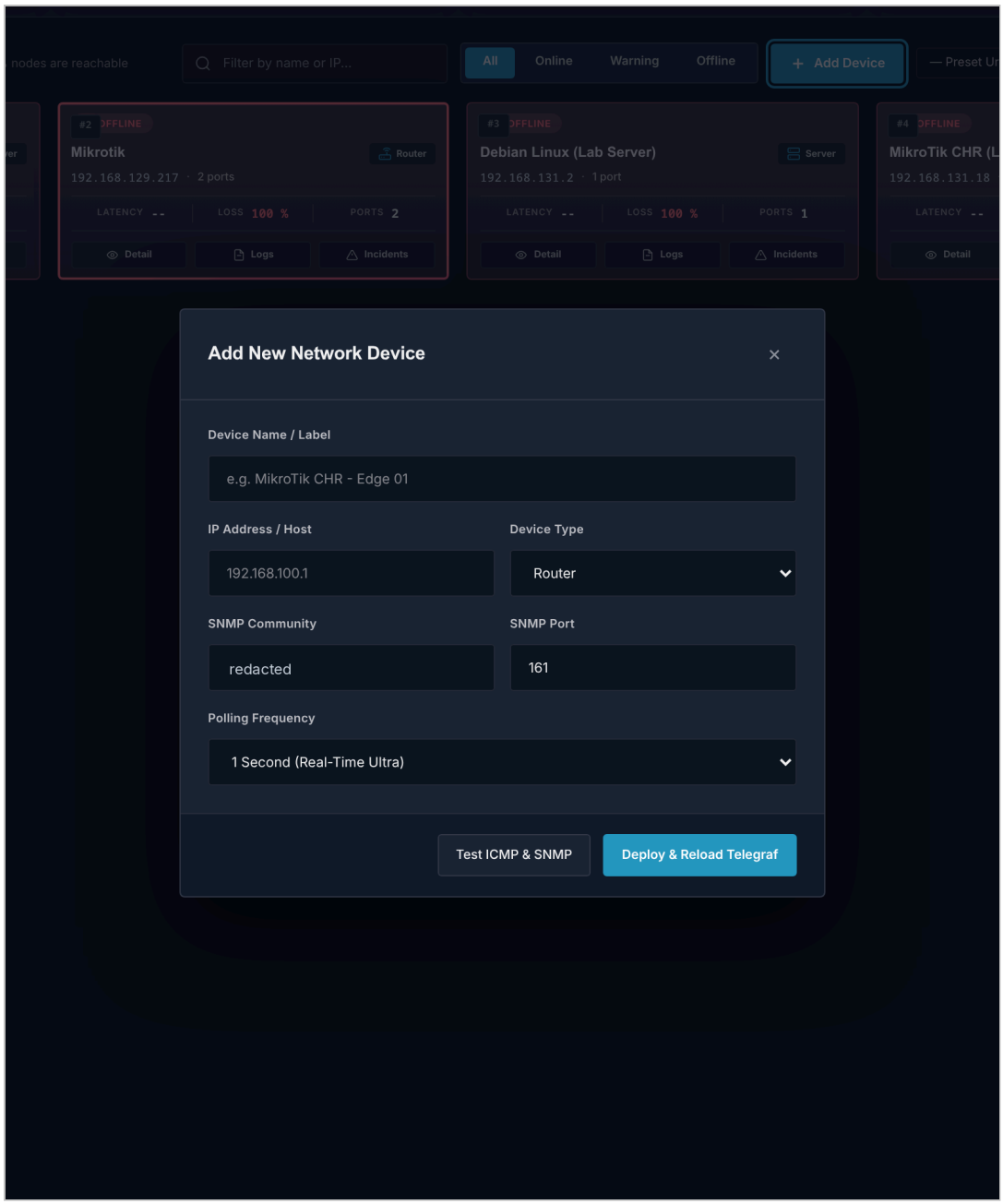


FIGURE 14

Guided onboarding: prove reachability before deploy. A successful save persists the device and generates its Telegraf configuration. The SNMP community value is redacted in this capture.

The twelve interface figures are captures from the running reference environment, not staged mockups. The two architecture figures are rendered from Mermaid source. Private `192.168.x` addressing is retained. No tokens, passwords, public IPs, or credentials are exposed; SNMP community values are redacted. Use the full-resolution links to inspect the interfaces.

---

07 / Engineering highlights

The implementation is where the trade-offs live.

Representative excerpts from the service layer. They show the decisions that keep a live monitoring loop trustworthy under real failure modes.

A one-second loop that is allowed to be slow

The stream service evaluates all devices concurrently in bounded chunks, then reuses interface results inside the same cycle. A device waiting on SNMP cannot serialize the entire NOC.

src/services/streamService.js excerpt

```
// bounded parallelism: never await an entire device list serially
const CHUNK = 5;
for (let i = 0; i < devices.length; i += CHUNK) {
  await Promise.all(devices.slice(i, i + CHUNK).map(async (dev) => {
    const metric = await generateOrFetchMetrics(dev, defaultIfaceFor(dev));
    deviceMetrics.set(dev.id, metric);
  }));
}

// per-cycle cache: one SNMP walk per device + interface, reused by every client
const cacheKey = `${targetDevice.id}:${reqIface}`;
let subMetric = ifaceCycleCache.get(cacheKey);
if (!subMetric) {
  subMetric = await generateOrFetchMetrics(targetDevice, reqIface);
  ifaceCycleCache.set(cacheKey, subMetric);
}
```

BACKPRESSURE · CACHING · 1-SECOND CADENCE

State changes are earned twice

Offline, warning, and recovery transitions require two consecutive evaluations. New alerts of the same type are deduplicated per device over a five-minute window.

health + alert suppression excerpt

```
if (track.failCount >= 2 && track.currentStatus !== 'offline') {
  // confirm DOWN, create or escalate the incident
}

-- alert dedupe: same device + type + active state
AND strftime('%s', created_at) >= strftime('%s', 'now', '-5 minutes')
```

FLAP GUARD · DEDUPE · INCIDENT LIFECYCLE

Telemetry has a fallback path

InfluxDB is the first read path for time-series metrics. When a bucket has no data, the service falls back to real ICMP and SNMP probes instead of drawing zeros.

src/services/streamService.js excerpt

```
const realData = await influxService.getLiveTelemetry(
  device.id, interfaceName
);
if (realData.has_data) {
  return {
    latency: realData.latency_ms,
    packetLoss: realData.packet_loss,
    has_data: true
  };
}

// no time-series data: probe the device directly
const pingResult = await realTimePing(device.ip_address);
```

INFLUXDB · ICMP · SNMP · NO FAKE ZEROS

Observability is generated, not hand-written

Saving a device writes a per-device Telegraf configuration with deliberate intervals: one second for reachability, five seconds for interface counters, and sixty seconds for inventory and system data.

generated per-device config TOML

```
[[inputs.ping]]
  interval = "1s"
  count = 3

[[inputs.snmp]]
  interval = "5s"
  oid = "IF-MIB::ifXTable"

[[inputs.snmp]]
  interval = "60s"
  # interface + system inventory
```

TELEGRAF · AUTOMATION · CONFIG GENERATION

Logs become sessions, not rows

Syslog is normalized across vendor formats, then evaluated by cumulative and sequence rules with time windows. A confirmed rule opens a threat session; unmatched rows stay as raw evidence.

src/services/threatRules.js excerpt

```
{
  windowMs: 5 * 60 * 1000,
  minEvents: 3,
  severity: 'critical',
  // confirmed only inside the live window
}
```

SYSLOG · CORRELATION · SESSIONS

Topology keeps its uncertainty

Discovery marks every known link stale before a sweep, persists fresh links with `last_seen`, and keeps unmanaged neighbors visible. A manual IPv4 override survives resets.

src/services/topologyDiscovery.js excerpt

```
// a new sweep begins: nothing is trusted until rediscovered
UPDATE device_links SET stale = 1 WHERE stale = 0;

// fresh links are upserted with last_seen = now; stale rows remain visible
// unmanaged nodes retain target_device_id = NULL and a discovered IP when available
```

LLDP / CDP / MNDP · GRAPH · FRESHNESS

08 / Challenges & solutions

Built around the failure modes, not the happy path.

The interesting work is in the edge cases: a device that answers late, a metric that is absent, a log that arrives in a vendor-specific format.

01 Flapping devices created alert noise and false urgency

RELIABILITY

PROBLEM

A single missed ICMP reply is not an outage. A naive threshold flips the device state every second, creating alert and incident churn.

DECISION

Gate transitions behind **two consecutive evaluations** and deduplicate alert inserts per device/type over a **five-minute** window.

RESULT

A brief packet loss event no longer becomes a page. Recovery is also confirmed twice, so an incident closes on a stable signal.

02 A one-second loop meeting slow SNMP would stretch the cycle

PERFORMANCE

PROBLEM

The dashboard expects a one-second cadence. A sequential loop would spend the cycle waiting on the slowest agent, while multiple clients could trigger duplicate interface walks.

DECISION

Evaluate devices in **bounded parallel chunks** and cache interface results once per cycle, then broadcast the same snapshot to every subscribed client.

RESULT

The loop remains responsive under a slow or unreachable target, and the stream stays scoped without multiplying probe work.

03 Topology is inherently partial and time-sensitive

NETWORKING

PROBLEM

LLDP, CDP, and MNDP expose different levels of detail. Discovery can miss a link, return an unknown IP, or find a neighbor that is not in inventory.

DECISION

Persist link **last_seen** and **stale** state, model unmanaged neighbors explicitly, enrich addresses through ARP/IP-MIB, and allow a manual IPv4 override.

RESULT

The map communicates what discovery knows and what it does not. A stale edge stays visible in context instead of silently disappearing.

04 Missing telemetry looked like a perfect zero

DATA INTEGRITY

PROBLEM

An empty time-series bucket or an unreachable probe is not the same as 0% packet loss. Rendering missing values as zeros makes an outage look healthy.

DECISION

Keep telemetry **nullable**, expose explicit no-data states in charts and tables, and let reports show “no data” for a day with no observations.

RESULT

Operators can distinguish silence from a healthy zero, which changes the next action: investigate collection, not the network path.

05 Syslog is noisy and vendor-shaped

CORRELATION

PROBLEM

RouterOS topics, Linux facilities, and generic syslog priorities do not share a clean schema. Correlating raw lines directly produces brittle matches and false incidents.

DECISION

Normalize severity, facility, program, topics, and message first. Evaluate cumulative and sequence rules inside a time window, and keep unmatched rows as evidence.

RESULT

Logs become traceable sessions without discarding the raw event. A confirmed rule has a visible window and evidence, not just a keyword match.

06 Configuration drift creates observability gaps

OPERATIONS

PROBLEM

A device can be reachable and still be effectively unmonitored: SNMP may be blocked, or syslog may never arrive. “Ping is green” is not enough to trust the telemetry.

DECISION

Add a **three-layer readiness scan**, expose the missing layer with an urgency level, generate platform-aware commands, and verify that a fresh log actually arrived.

RESULT

Onboarding becomes a closed loop: reachability, data collection, and event delivery are each proven before the device is considered observable.

09 / Skills demonstrated

Engineering breadth, backed by working surfaces.

Each skill is anchored to a concrete part of the repository rather than a keyword list.

System design & architecture	Service boundaries between collection, storage, application logic, and transport; a deliberate split between hot telemetry and durable state. EVIDENCE: ARCHITECTURE · DUAL-STORE BOUNDARY · STREAM SERVICE

Networking & protocols	ICMP reachability, SNMP v2c interface tables, LLDP/CDP/MNDP discovery, ARP enrichment, syslog forwarding, and SNMP traps. EVIDENCE: TOPOLOGY · TELEGRAF CONFIGS · READINESS
Backend & API engineering	Forty-three REST routes covering inventory, telemetry, incidents, logs, topology, reports, settings, and readiness workflows. EVIDENCE: EXPRESS ROUTER · SERVICE ORCHESTRATION · VALIDATION
Data modeling	Ten relational tables with indexes for event, incident, log, and threat queries; WAL-backed SQLite; schema applied at boot. EVIDENCE: SCHEMA.SQL · DAILY UPTIME · INDEXED LOOKUPS
Real-time systems	One-second scheduler, per-client subscriptions, bounded parallelism, cycle caches, and WebSocket incident broadcasts. EVIDENCE: STREAMSERVICE.JS · METRIC_TICK · DEVICE_TELEMETRY
Frontend & visualization	Framework-free SPA, responsive NOC layout, Chart.js sliding windows, vis-network topology, detail overlays, and explicit empty/error states. EVIDENCE: PUBLIC/INDEX.HTML · APP.JS · DETAIL VIEWS
Reliability engineering	Flap guard, alert deduplication, stale link marking, graceful fallback, and data-integrity choices that avoid false healthy states. EVIDENCE: HEALTH TRACKER · DEDUPE SQL · TOPOLOGY FRESHNESS
Security-minded implementation	Target and community validation, subprocesses spawned with argument arrays instead of shell strings, and token values never echoed back. EVIDENCE: VALIDATORS · SPAWN ARGS · SETTINGS API
Product thinking for operators	Incident drawers, copyable remediation, no-data language, archive/recovery paths, and a visual hierarchy built for triage under pressure. EVIDENCE: INCIDENT DRAWER · READINESS · EMPTY STATES

10 / Project summary

A complete vertical slice of network operations software.

A signal becomes an incident with evidence, a device becomes a topology node, and a configuration gap becomes a copyable fix.

BMS IT Monitoring connects collection agents, time-series and relational storage, an API, a real-time transport, and the operator interface that makes the data useful. It is deliberately narrow in deployment scope and deliberately deep in implementation.

The result is a platform where a signal becomes an incident with evidence, a device becomes a topology node, and a configuration gap becomes a copyable fix. That is the difference between a dashboard and an operations tool.

INTERFACE	DELIVERY	TELEMETRY	RELATIONAL MODEL
8 views + overlays	REST + WebSocket	1-second stream	10 tables
DEVICE TAXONOMY	**READINESS**	**STORES**	**PROTOCOLS**
28 categories	3 layers	InfluxDB + SQLite	ICMP · SNMP · LLDP/CDP/MNDP · syslog

NARRATIVE PROFILE PORTFOLIO README APPLICATION README

All screenshots come from the running app in a private four-node reference environment. Private `192.168.x` addressing is retained. No tokens, passwords, public IPs, or credentials are exposed; SNMP community values are redacted. Counts describe the implementation, not production fleet scale. Authentication/RBAC and hosted multi-tenancy are outside the current scope.

BMS IT MONITORING · TECHNICAL CASE STUDY · 2026

PREPARED FROM REPOSITORY EVIDENCE AND THE RUNNING REFERENCE ENVIRONMENT. NO PRODUCTION METRICS ARE CLAIMED.