Observability

Monitoring stack, logging pipeline, alert routing, dashboards, health checks

- Chapter 12 — Observability

Chapter 12 — Observability

12.1 Current State and Scope

Prometheus + Grafana 13.1.0 + nginx/TLS on `monitor-servers`, serving `monitor.sbsdash.com`; node_exporter on all four droplets (prod/dev private-bind at 10.100.0.3/.2:9100, wiki public-bind with ufw scope — the §3.3 interim, monitor local at 127.0.0.1); scrape job `droplets` with per-server labels; dashboards "Node Exporter Full" (ID 1860) and custom "SBS Servers Overview" (uid `sbs-overview`, provisioned as home dashboard); 30-second kiosk playlist. Verified 2026-07-17: 5 targets up, `promtool` clean, Grafana healthy.

What exists is **metrics + visualization**. This chapter completes the design: logging pipeline (absent), alerting (absent — dashboards without alerts is observability that only works when someone is looking), blackbox/health probes (absent), and the MON-family feeds every prior chapter has been promising (MON-01 unified audit trail inputs, MON-02 anomaly detection). Grafana local-admin → OIDC migration is already item 52 (Ch. 9); Prometheus job additions below fold into the Ansible `monitoring` role (§10.3) rather than remaining hand-edits.

12.2 Metrics (Prometheus)

Retained as deployed — apt-installed system service on `monitor-servers`, 127.0.0.1-bound behind nginx (§6.6 rationale: don't containerize the watcher). Extensions, all IaC-managed:

Job	Targets	Purpose
`droplets` (live)	node_exporter ×4 (+`prod-vault`, +rebuilt wiki private bind at rebuild wave)	Host CPU/RAM/disk/net
`blackbox-external` (new)	Probes of all five public URLs (§4.3) from the monitor via blackbox_exporter	Surface up/down, TLS validity, **cert expiry** (§5.2 feed), HTTP status, redirect correctness
`nginx` (new)	stub_status per edge droplet, private bind	Request rates, connections — MON-02 baseline input
`containers` (new, at Compartment build)	cAdvisor on prod/dev, private bind	Per-Compartment container CPU/RAM — §6.4 resource-limit pressure visibility
`integrity` (new)	textfile-collector metrics: nftables ruleset hash, compose-vs-running match, backup age, drift-job status	§10.5 fast-tamper indicators; §14 backup freshness
`vault` (new, at Ch. 8 build)	Vault telemetry endpoint, VPC-private	Seal status, token/lease counts, audit-device health

Job	Targets	Purpose
`postgres` / `mongo` / `redis` exporters (new, at Ch. 7 build)	Private binds per engine	Engine health, connections, replication/persistence state

Retention: 30 days local TSDB (sized fine on 120 GB); longer-horizon capacity trending is served by the 30-day window plus quarterly-review screenshots — remote-write long-term storage is a Chapter 16 growth item, not Day-1. Scrape interval stays 15 s. Every new exporter binds VPC-private or localhost — the §3.7 public-listener gate applies to observability itself.

12.3 Logging Pipeline

Grafana Loki + promtail/alloy agents — chosen over an ELK stack deliberately: one vendor surface with Grafana (logs and metrics correlate in the same UI), index-light architecture that fits a 4 GB monitor droplet, and LogQL's label model matches the Prometheus labels already in place.

- **Loki** on `monitor-servers`, 127.0.0.1-bound, filesystem storage on a dedicated DO Volume; retention 90 days hot. **Audit-relevant streams are additionally shipped to the Ch. 13 hash-chained archive — Loki is the operational search tier, not the evidentiary store** (different integrity guarantees, different retention: 90 d vs 7 y).
- **Agents** on every droplet (Ansible role), shipping over the VPC (wiki joins post-rebuild; until then its logs ship over TLS to the monitor's public interface — same interim-acceptance logic as its scrape path, §3.3):

Stream	Source	Notes
System	journald	sshd (bastion session evidence, §3.3/§9.6), sudo, unattended-upgrades
Edge	nginx access/error, structured format (§5.7)	All five surfaces; access logs are the MON-02 baseline for request anomalies
Auth	oauth2-proxy, Grafana, BookStack auth events	IAM-01/02 evidence; failed-auth series feeds §12.4 alerts
Vault	audit device (§8.4)	Secret-access event stream — MON-01's highest-value feed
Containers	Docker json-file via agent, per-Compartment labels	Compartment-scoped log isolation: Grafana folder permissions (§9.5 group mapping) keep c03 logs invisible to c08 personnel — the log layer of IAM-04
Egress denials	nftables log target (§3.4)	AI-02/LLM01 exfiltration-attempt indicator
Drift/CI	§10.5 job outputs	Codified-change audit trail

12.4 Alert Routing

Alertmanager on `monitor-servers` (the changelog's own "pendiente" item, now normative), 127.0.0.1-bound:

- **Channels:** ops Slack channel (webhook from Vault `secret/platform/alerting/`) for everything; **email to named on-call for `severity=critical`** as the Slack-independent second path. The per-Compartment Slack channel fabric (platform-level alerting, §5.2 of the SOW-side design) is application-layer and separate — this section is *infrastructure* alerting; the two fabrics share the Slack workspace but not routes.
- **Severity model:** `critical` = platform-down/data-risk (surface down ≥ 2 min, cert ≤ 7 d with renewal failing, disk ≥ 90 %, Vault sealed, backup age > 26 h, integrity-hash mismatch, target down ≥ 5 min); `warning` = degradation (disk ≥ 80 %, cert ≤ 14 d, CPU/RAM sustained ≥ 85 % 15 min, drift detected, Trivy nightly CRITICAL finding, egress-denial burst, failed-auth burst ≥ 10/5 min/IP); `info` = notable (deploys, rotation confirmations, weekly Ansible enforcement diff).
- Routing: group by alertname+instance, 4 h repeat for unresolved critical, inhibition (droplet-down inhibits its per-service alerts). **Dead-man's switch:** a permanently-firing `Watchdog` alert routes to a heartbeat receiver — if the watchdog goes *silent*, the monitoring stack itself is down; this is the alert that covers the monitor (§12.6 covers it structurally).
- Alert rules live in the repo (`monitoring` role), promtool-validated in CI (§10.4) — alert changes are PRs, not console edits.

12.5 Dashboards

Provisioned-by-file discipline (as already practiced — `sbs-overview` and 1860 are file-provisioned) becomes the rule: **no console-authored production dashboards**; Grafana UI is for exploration, dashboards graduate to the repo. Dashboard set:

Dashboard	Audience (via §9.5 groups)	Content
SBS Servers Overview (live)	`dash-monitor-view`	Fleet UP/DOWN, CPU/RAM/disk bars, history, uptime — home dashboard, kiosk playlist retained
Node Exporter Full (live)	`dash-monitor-view`	Per-host deep dive
Edge & Surfaces (new)	monitor-view	Blackbox probe status, cert days-remaining, request rates, 4xx/5xx, auth failures
Compartments (new, at build)	monitor-view; per-Compartment panels visible per group	cAdvisor per-Compartment resources, container restarts, egress denials by Compartment
Data Services (new, at Ch. 7 build)	monitor-admin	Engine health, connections, volume capacity, backup age
Security & Integrity (new)	monitor-admin + security routing	Vault seal/audit status, drift jobs, ruleset hashes, gitleaks/Trivy findings, bastion session count

12.6 Monitoring the Monitor

The stack's structural blind spot, addressed three ways: the dead-man's switch (§12.4); an **external uptime check** on `https://monitor.sbsdash.com` from outside DO (UptimeRobot free tier or equivalent — the one approved external service, alerting to the same email path); and the monitor droplet's own metrics riding the same `droplets` job with its alerts routed identically. Loki/Alertmanager/blackbox process health via systemd unit-state metrics on the integrity job. The bastion role (§3.3) already makes this host priority-patched; observability adds priority-*watched*.

12.7 Verification

Gates (LH-SBS-INST-001): all §12.2 jobs green with private binds confirmed (`ss -tlnp` sweep per droplet); blackbox probes cover all five URLs incl. cert-expiry metric present; induced-failure drill — stop a service, alert received on both channels inside SLA, inhibition correct; dead-man silence drill — stop Alertmanager, heartbeat receiver fires; Loki streams present per table incl. Vault audit and egress denials (LogQL sample per stream); Compartment log visibility probe — c03 group cannot read c08 stream (mirrors §7.7/§9.7); dashboard set file-provisioned, console-save on prod folders denied; alert rules pass promtool in CI, console-edit reverted by drift run; external uptime check firing path tested; retention conf 30 d TSDB / 90 d Loki confirmed; wiki agent path migrates to VPC at rebuild (carried gate).