Monitoring & Observability

Monitoring & Observability

> **Project:** {{PROJECT_NAME}} **Version:** {{VERSION}} **Date:** {{DATE}}
> **Author:** {{AUTHOR}} **Status:** Draft | In Review | Approved **Reviewers:** {{REVIEWERS}}

Document History

Version	Date	Author	Changes
0.1	{{DATE}}	{{AUTHOR}}	Initial draft

---

1. Observability Strategy

Observability Platform: {{OBS_PLATFORM}} **Strategy:** Instrument everything, alert on symptoms (not causes), correlate across pillars

Core Questions We Must Be Able to Answer:

1. Is the system up and serving users correctly?
2. How fast is it responding?
3. What errors are occurring and why?
4. Where is the bottleneck?
5. What changed before this problem started?

---

2. Three Pillars

2.1 Metrics

Infrastructure Metrics

Metric	Source	Alert Threshold	Severity
CPU utilization	Node exporter / CloudWatch	> {{CPU_WARN}}% (warn), > {{CPU_CRIT}}% (critical)	Warning / Critical
Memory utilization	Node exporter / CloudWatch	> {{MEM_WARN}}% (warn), > {{MEM_CRIT}}% (critical)	Warning / Critical
Disk utilization	Node exporter / CloudWatch	> {{DISK_WARN}}% (warn), > {{DISK_CRIT}}% (critical)	Warning / Critical
Network in/out	Node exporter / CloudWatch	> {{NET_LIMIT}}Mbps sustained	Warning
Container restarts	Kubernetes / ECS	> {{RESTART_LIMIT}} in 5min	Critical
Node not ready	Kubernetes	Any	Critical

Application Metrics (RED Method)

Metric	Description	Target	Alert Threshold
Request rate	Requests per second per service	Baseline ± 20%	50% deviation
Error rate	% requests returning 5xx	< {{ERROR_RATE}}%	> {{ERROR_ALERT}}%
P50 latency	Median response time	< {{P50}}ms	> {{P50_ALERT}}ms
P95 latency	95th percentile response time	< {{P95}}ms	> {{P95_ALERT}}ms
P99 latency	99th percentile response time	< {{P99}}ms	> {{P99_ALERT}}ms

Business Metrics

Metric	Description	Collection Method	Dashboard
Active users (DAU/MAU)	Daily/monthly active users	Frontend instrumentation	Business dashboard
{{CONVERSION_METRIC}}	{{CONVERSION_DESC}}	Backend event	Business dashboard
{{REVENUE_METRIC}}	{{REVENUE_DESC}}	Payment events	Finance dashboard
Feature usage	Feature-level engagement	Feature flag SDK	Product dashboard

Custom Metrics Definition

Metric Name	Type	Labels	Description	Unit

`{{APP}}_job_queue_depth`	Gauge	`queue_name`	Number of pending jobs	count
`{{APP}}_job_processing_duration`	Histogram	`queue_name, status`	Job processing time	seconds
`{{APP}}_external_api_calls_total`	Counter	`service, status`	External API call count	count
`{{APP}}_cache_hit_ratio`	Gauge	`cache_type`	Cache hit percentage	ratio

2.2 Logs

Log Levels & Usage Guide

Level	When to Use	Examples
`ERROR`	Unexpected failure requiring attention	Database connection failure, unhandled exception
`WARN`	Unexpected but handled situation	Deprecated API called, retry succeeded
`INFO`	Normal business events	User logged in, order created, job completed
`DEBUG`	Diagnostic detail (dev/staging only)	Function parameters, internal state
`TRACE`	Extremely verbose (local dev only)	SQL queries, HTTP request/response bodies

Production log level: `INFO` and above

Structured Logging Format

```
{
  "timestamp": "2026-01-15T10:30:00.000Z",
  "level": "INFO",
  "service": "{{SERVICE_NAME}}",
  "version": "{{VERSION}}",
  "trace_id": "abc123def456",
  "span_id": "789xyz",
  "user_id": "{{HASHED_OR_OMIT}}",
  "request_id": "req-uuid-here",
  "message": "Order created successfully",
  "order_id": "ord-123",
  "duration_ms": 45
}
```

Required fields: `timestamp`, `level`, `service`, `message`, `trace_id` **Forbidden in logs:** passwords, tokens, credit card numbers, SSN, full email addresses (hash or truncate)

Log Aggregation Pipeline

```
flowchart LR
    APP[Application] -->|stdout/stderr| AGENT[Log Agent<br/>Fluent Bit / Filebeat]
    AGENT -->|structured JSON| STORE[Log Store<br/>Loki / Elasticsearch / CloudWatch]
    STORE --> QUERY[Query Interface<br/>Grafana / Kibana]
    STORE --> ALERT[Alert Engine<br/>AlertManager / PagerDuty]
```

Stage	Tool	Configuration
Application logging	{{LOG_LIB}}	Structured JSON to stdout
Log agent	{{LOG_AGENT}}	Deployed as sidecar / DaemonSet
Transport	{{LOG_TRANSPORT}}	TLS encrypted
Storage	{{LOG_STORE}}	Indexed, compressed
Query	{{LOG_QUERY}}	Access via dashboard

Log Retention Policy

Environment	Retention	Storage Tier
Dev	7 days	Hot
Staging	30 days	Hot
Production	{{PROD_LOG_RETENTION}} days	Hot (30d) → Cold archive
Audit logs	1 year (regulatory)	Hot (90d) → Cold archive

PII in Logs — Masking Strategy

Data Type	Strategy	Example
Email address	Hash + truncate	`user:sha256(email)[:8]`
Phone number	Redact	`[PHONE_REDACTED]`
IP address	Anonymize last octet	`192.168.1.xxx`
Payment data	Never log	Use `[PAYMENT_DATA_OMITTED]`
Auth tokens	Never log	Use `[TOKEN_OMITTED]`
Names	Omit or pseudonymize	Reference by ID only

2.3 Traces

Distributed Tracing Setup

Tracing Framework: {{TRACE_FRAMEWORK}} **Backend:** {{TRACE_BACKEND}} **Auto-instrumentation:** {{AUTO_INSTRUMENT}}

Service	Instrumented	Framework	Notes
{{SERVICE_1}}	Yes	OpenTelemetry	HTTP, DB, Redis
{{SERVICE_2}}	Yes	OpenTelemetry	HTTP, external calls

Trace Sampling Strategy

Environment	Strategy	Rate	Notes
Dev	Always-on	100%	Full visibility
Staging	Always-on	100%	Full visibility
Production	Tail-based	{{SAMPLE_RATE}}% + errors	Error traces always kept

Tail-based sampling rules:

- Always sample: traces with errors, traces > {{SLOW_THRESHOLD}}ms
- Sample rate: {{SAMPLE_RATE}}% of successful, fast traces
- Head-based fallback: {{HEAD_SAMPLE_RATE}}% if tail-based collector unavailable

Span Naming Conventions

Operation Type	Naming Pattern	Example
HTTP handler	`HTTP {{METHOD}} {{ROUTE}}`	`HTTP POST /api/orders`
DB query	`db.{{operation}} {{table}}`	`db.select orders`
Cache	`cache.{{operation}} {{key_pattern}}`	`cache.get user:*`
Queue	`queue.{{operation}} {{queue_name}}`	`queue.publish order-events`
External HTTP	`{{service}} {{METHOD}} {{path}}`	`stripe POST /charges`

Context Propagation

Standard: W3C TraceContext (`traceparent` header) **Baggage:** W3C Baggage (for `user_id`, `tenant_id` propagation) **Async:** Inject context into message queue headers / job metadata

---

3. Alerting

3.1 Alert Rules

Alert Name	Condition	Duration	Severity	Channel	Runbook
`HighErrorRate`	error_rate > {{ERROR_ALERT}}%	2 min	Critical	PagerDuty	[link]
`SlowP99`	p99_latency > {{P99_ALERT}} ms	5 min	Warning	Slack #alerts	[link]
`ServiceDown`	health_check failing	1 min	Critical	PagerDuty	[link]
`HighCPU`	cpu > {{CPU_CRIT}}%	10 min	Warning	Slack #alerts	[link]
`DiskAlmostFull`	disk > {{DISK_CRIT}}%	5 min	Critical	PagerDuty	[link]
`DeploymentFailed`	deployment status = failed	Immediate	Critical	Slack #deployments	[link]
`CertificateExpiringSoon`	cert_expiry < 30 days	—	Warning	Slack #ops	[link]
`BackupFailed`	backup job = failed	—	Critical	PagerDuty	[link]
`SLOBudgetBurning`	error_budget < 10% remaining	—	Critical	PagerDuty	[link]

3.2 Alert Routing & Escalation

```
flowchart TD
    ALERT[Alert fires] --> SEVERITY{Severity?}
    SEVERITY -->|Critical| ONCALL[On-call engineer<br/>PagerDuty / phone]
    SEVERITY -->|Warning| SLACK[Slack #alerts<br/>No immediate response required]
    ONCALL -->|Not acknowledged in 5min| ESCALATE[Escalate to secondary]
    ESCALATE -->|Not acknowledged in 10min| MANAGER[Notify engineering lead]
```

Severity	Response SLA	Channel	Escalation
Critical (P1)	Acknowledge in 5 min, resolve in 1h	PagerDuty + call	Escalate at 5 min
High (P2)	Acknowledge in 30 min, resolve in 4h	PagerDuty	Escalate at 30 min
Warning (P3)	Review within 1 business day	Slack	Manual

Severity	Response SLA	Channel	Escalation
Info	No response required	Slack	None

3.3 On-Call Rotation

Schedule: {{ONCALL_SCHEDULE}} **Calendar:** {{ONCALL_TOOL}} **Primary rotation:** {{ONCALL_MEMBERS}} **Secondary (escalation):** {{ESCALATION_MEMBERS}} **Minimum rotation size:** 3 people (to avoid burnout)

3.4 Alert Fatigue Prevention

- Alert review cadence: Monthly — remove/adjust alerts with < {{ACTIONABLE_RATE}}% actionable rate
- Minimum alert duration: 2+ minutes (no single-spike alerts)
- Deduplication window: {{DEDUP_WINDOW}} minutes
- Business hours suppression: Allowed for non-critical alerts {{SUPPRESSION_HOURS}}
- Post-mortem requirement: Every Critical alert reviewed after incident

---

4. Dashboards

4.1 Dashboard Inventory

Dashboard	Purpose	Link	Audience
System Overview	High-level health of all services	{{LINK}}	Everyone
{{SERVICE_1}}	Service-level detail	{{LINK}}	Dev team
Infrastructure	Host/container metrics	{{LINK}}	DevOps
Business Metrics	KPIs and conversions	{{LINK}}	Leadership, PM
SLO Tracker	Error budget tracking	{{LINK}}	Engineering lead
On-Call	Current incidents, top errors	{{LINK}}	On-call engineer

4.2 Key Dashboard Specs — System Overview

Required panels:

1. Service health matrix (all services, green/red/yellow)

2. Request rate (all services, last 1h)
3. Error rate (all services, last 1h)
4. P99 latency (all services, last 1h)
5. Active incidents count
6. Error budget remaining (all SLOs)
7. Last deployment (service, version, time)
8. Infrastructure health (CPU, memory, disk — aggregate)

---

5. SLOs / SLIs

5.1 SLI Definitions

SLI	Definition	Measurement Method
Availability	% requests returning non-5xx	(total_requests - 5xx_requests) / total_requests
Latency	% requests completing within threshold	histogram_quantile(0.95, ...) < {{LATENCY_SLI}}ms
Error rate	% requests not returning errors	(total_requests - error_requests) / total_requests

5.2 SLO Targets

Service	SLI	Target	Window	Error Budget
{{SERVICE}}	Availability	{{AVAIL_TARGET}}%	30 days	{{BUDGET_MINUTES}} min/month
{{SERVICE}}	Latency (P95 < {{P95}}ms)	{{LATENCY_TARGET}}%	30 days	{{LATENCY_BUDGET_MINUTES}} min/month

5.3 Error Budget Tracking

Service	Monthly Budget	Burned This Month	Remaining	Burn Rate (24h)
{{SERVICE}}	{{BUDGET}}min	TBD	TBD	TBD

Error budget policy:

- Budget > 50% remaining: Move fast, deploy freely
- Budget 10-50% remaining: Slow down, prioritize reliability work
- Budget < 10% remaining: Freeze non-critical deploys, focus on reliability

6. Tooling

Tool	Version	Purpose	Hosted
{{METRICS_TOOL}}	{{VERSION}}	Metrics collection & storage	{{HOSTING}}
{{LOG_TOOL}}	{{VERSION}}	Log aggregation	{{HOSTING}}
{{TRACE_TOOL}}	{{VERSION}}	Distributed tracing	{{HOSTING}}
{{DASHBOARD_TOOL}}	{{VERSION}}	Visualization	{{HOSTING}}
{{ALERT_TOOL}}	{{VERSION}}	Alert routing & on-call	{{HOSTING}}

Related Documents

- Deployment Architecture
- Disaster Recovery Plan
- Incident Report
- Operational Runbook
- SLA Report

Approval

Role	Name	Date	Signature
Author			
Reviewer			
Approver			

Revision #7
Created 2026-02-23 12:05:55 UTC by John
Updated 2026-05-25 07:33:54 UTC by John