The 10-Layer Kubernetes Monitoring Checklist

The exact framework we use when auditing monitoring setups for clients running Kubernetes in production.

By Amjad Syed | Founder & CEO, Tasrie IT Services

1 System & Infrastructure

Is the underlying infrastructure healthy?

- [ ] CPU usage and load average
- [ ] Memory usage and available memory
- [ ] Disk usage and disk I/O
- [ ] Network I/O
- [ ] Node up/down status
- [ ] Pod up/down status & restart counts
- [ ] CrashLoopBackOff alerts
- [ ] OOMKilled alerts
- [ ] ImagePullBackOff alerts
- [ ] Pending/Evicted pod alerts

Tools: Prometheus + Node Exporter, kube-state-metrics

2 Application Performance

Is the code behaving correctly?

- [ ] Response times per endpoint (p50, p95, p99)
- [ ] Error rates (4xx, 5xx)
- [ ] Transaction traces
- [ ] Slow database queries
- [ ] Slow external API calls

Tools: New Relic (free tier), Datadog APM, SigNoz (open source), Jaeger | **Instrumentation:** OpenTelemetry

3 HTTP, API & Real User Monitoring

Can users actually reach and use the application?

- [ ] Health check endpoint probes
- [ ] Critical user flow probes (login, checkout)
- [ ] Multi-region synthetic monitoring
- [ ] API response schema validation
- [ ] Core Web Vitals (LCP, FID, CLS)
- [ ] Page load times by geography
- [ ] JavaScript errors in production

Synthetic: Blackbox Exporter, Checkly | **API:** Runscope, Postman Monitors | **RUM:** Datadog RUM, LogRocket, Sentry

4 Database

Is the database healthy and performing?

- [ ] Active connections vs pool size
- [ ] Query latency (p50, p95, p99)
- [ ] Slow query logging (> 1s)
- [ ] Replication lag
- [ ] Lock waits and deadlocks
- [ ] Disk and memory usage

Metric	Warning	Critical
Connection pool usage	70%	90%
Replication lag	10s	60s
Query latency p95	500ms	2s

Tools: PostgreSQL Exporter, MySQL Exporter, PgHero, PMM

5 Cache

Is the cache working effectively?

- [ ] Hit/miss ratio (alert if < 80%)
- [ ] Memory usage
- [ ] Eviction rate
- [ ] Connection count
- [ ] Cache up/down status

Tools: Redis Exporter, Memcached Exporter, CloudWatch (ElastiCache)

6 Message Queues

Is async work getting processed?

- [ ] Queue depth
- [ ] Consumer lag (alert if > 1000 messages)
- [ ] Messages per second (in/out)
- [ ] Dead letter queue size (alert on any growth)
- [ ] Queue up/down status

Tools: Kafka Exporter, Burrow, RabbitMQ Prometheus Plugin, SQS Exporter

7 Tracing Infrastructure

Is your observability infrastructure healthy?

- [ ] Collector health and up/down status
- [ ] Span ingestion rate
- [ ] Storage backend health
- [ ] Dropped spans (data loss indicator)

Tools: Built-in Jaeger/Tempo metrics, Prometheus

8 SSL & Certificates

Will certificates expire and cause an outage?

- [ ] Certificate expiry monitoring
- [ ] Alert at 30 days (Slack notification)
- [ ] Alert at 14 days (Slack + ticket)
- [ ] Alert at 7 days (Page on-call)
- [ ] TLS version monitoring

Tools: Blackbox Exporter (probe_ssl_earliest_cert_expiry), cert-manager

9 External Dependencies

Are third-party services working?

- [ ] Response times from external APIs
- [ ] Error rates from external calls
- [ ] Third-party status page monitoring
- [ ] Payment provider health (Stripe, PayPal)
- [ ] Auth service health (Auth0, Okta)
- [ ] CDN health (Cloudflare, Fastly)

Tools: StatusGator, Instatus, Hyperping, your own probes

10 Log Patterns & Errors

What specific errors are happening?

- [ ] Sudden spike in 5xx errors
- [ ] Unusual increase in 4xx errors
- [ ] "timeout" pattern alerts
- [ ] "connection refused" pattern alerts
- [ ] "deadlock" pattern alerts
- [ ] "out of memory" pattern alerts
- [ ] "connection pool exhausted" alerts
- [ ] "circuit breaker open" alerts

Tools: Loki, Elasticsearch, CloudWatch Logs, Datadog Logs

Alerting Philosophy

Rule #1: Alert on symptoms, not causes. High CPU isn't always a problem. Users getting errors is always a problem.

Page Someone

Users affected NOW. 5xx errors, service down, data loss risk.
Channel: PagerDuty

Slack Notification

Needs attention today. Cert expiring in 14 days, disk at 80%.
Channel: Slack

Just Log It

Interesting but not urgent. High CPU without user impact.
Channel: Dashboard

Key principle: If an alert fires and you do nothing about it, delete the alert. Alert fatigue is real.

Common Mistakes to Avoid

- ✕ **Only watching pod metrics** - Node can be dying while pods look fine (disk full, network issues)
- ✕ **No multi-region probes** - App works from cluster but unreachable from the internet
- ✕ **Missing RUM** - 50ms API response, 4s page load. Users frustrated, you'd never know.
- ✕ **Status 200 = healthy** - API returns 200 with empty or wrong data. Add data assertions.
- ✕ **No external dependency monitoring** - Blame your app when Stripe is actually down

Quick Reference: Tool Stacks

Budget-Friendly (Open Source)

- Prometheus + Grafana (metrics)
- Loki (logs)
- Jaeger or Tempo (traces)

Enterprise Stack

- Datadog or New Relic (all-in-one)
- PagerDuty (alerting & on-call)
- Checkly (synthetic monitoring)

- Blackbox Exporter (synthetic)
- Alertmanager → Slack

- StatusGator (external deps)
- LogRocket (session replay)

Need Help Implementing This?

We've implemented this exact framework for clients with 400+ servers.
Let us audit your monitoring setup and show you what you're missing.

Book a Free Monitoring Audit | Read the Full Guide