**OnPrem - Observability Solution Approach**

# Table of Contents

## 1. OnPrem Strategy

1.0 Introduction

2.0 Guiding Principles

3.0 Key Areas of Focus

4.0 Tooling & Ownership

5.0 Implementation Steps

6.0 Critical Infrastructure Resources

7.0 Process & Outcomes

|  |  |  |
| --- | --- | --- |
| **Domain** | **Monitoring Targets** | **Tools** |
| Infrastructure | CPU, memory, disk I/O, storage saturation | Netdata |
| Application | HTTP 4xx/5xx rates, JVM pauses, request latency | Splunk |
| Database | Slow queries, connection pool saturation | Splunk |
| Network | Packet loss, TCP resets, latency spikes | Netdata + Splunk |
| Critical Logs | System errors, authentication failures | Splunk |

# OnPrem Monitoring and Observability Strategy

## 1.0 Introduction

Unified monitoring for ’s **OnPrem** infrastructure and applications using Splunk (logs) and Netdata (metrics) to ensure rapid issue detection, diagnosis, and resolution.

## 2.0 Guiding Principles

**Pareto Principle**: Prioritize CPU/memory, HTTP errors, and disk failures (20% of metrics detecting 80% of issues).

**Knowns/Unknowns Framework**: Convert unanalyzed data (e.g., disk error logs) into actionable insights.

**3.0 Key Areas of Focus**

## 4.0 Tooling & Ownership

|  |  |  |
| --- | --- | --- |
| **Tool** | **Function** | **Owner** |
| Splunk | Log dashboards, alerting, root-cause analysis | Operations Team |
| Netdata | Real-time infrastructure metrics (1s granularity) | Infrastructure Team |

## 5.0 Implementation Steps

### Phase 1: Instrumentation (Weeks 1-4)

1. Deploy Netdata agents to all Linux/Windows servers.
2. Configure Splunk Universal Forwarders for log ingestion from applications and databases.
3. Build dashboards:

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource Type** | **OnPrem Infra**  **Resources** | **Monitoring Coverage** | **Deployed**  **Services** |
| Web Servers | Apache Tomcat, Nginx | CPU, request latency, error rates |  |
| App Servers | JBoss, .NET | JVM heap, thread deadlocks |  |
| Database  Servers | PostgreSQL, SQL  Server | Slow queries, replication lag |  |

Infrastructure: CPU >90%, disk space <10%.

Application: HTTP error rate >1%, latency >500ms.

### Phase 2: Alerting & Baselining (Weeks 5-8)

1. Define critical alerts:

Netdata: CPU >90% for 5min, disk I/O >80%.

Splunk: ERROR logs >10/min, slow SQL queries (>200ms).

1. Establish performance baselines using historical data.

### Phase 3: Automation & Review (Ongoing)

1. Integrate Splunk alerts with ServiceNow for auto-ticket generation.
2. Weekly audits of Netdata/Splunk to surface "unknown-knowns" (e.g., ignored disk warnings).

## 6.0 Critical Infrastructure Resources

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource Type** | **OnPrem Infra**  **Resources** | **Monitoring Coverage** | **Deployed**  **Services** |
| Storage | NAS/SAN devices | Disk I/O, capacity saturation |  |
| Network  Devices | Routers, switches | Packet loss, interface errors |  |

## 7.0 Process & Outcomes

**Daily**: Splunk dashboard reviews for critical errors.

**Weekly**: RCA sessions to expose "unknown-unknowns" (e.g., hidden network congestion).

**Outcomes**:

MTTR reduction by 40% via automated alerts.

Quarterly reduction of "unknowns" by 15%.

# Appendix: Cross-Environment Alignment

|  |  |  |
| --- | --- | --- |
| **Principle** | **OnPrem** | **AWS Cloud** |
| **Unified Logging** | Splunk for all logs | Splunk ingests CloudWatch logs |
| **Pareto Compliance** | Top 20%: CPU, HTTP errors, disk I/O | Top 20%: Lambda errors, RDS  CPU |
| **Knowns/Unknowns** | Weekly Splunk audits | X-Ray + chaos testing |
| **Blind Spot**  **Reduction** | 15% reduction per quarter | 25% reduction per quarter |