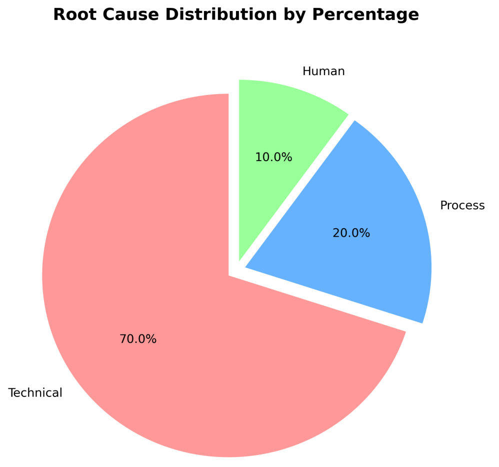
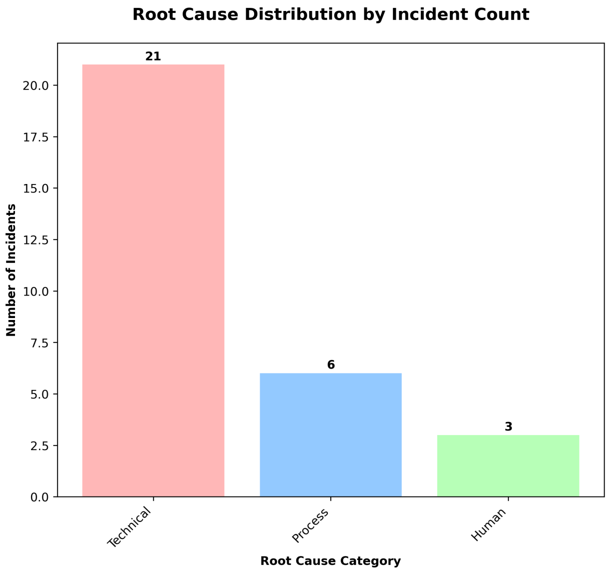
RCA Analysis Report

Generated: 2025-08-29 15:36:50
Model: gpt-4-turbo-preview
Analysis Type: Cloud Infrastructure RCA Pattern Analysis

Root Cause Classification



Root Cause Distribution by Percentage



Root Cause Distribution by Incident Count

Classification Data Table

Category	Percentage	Incident Count
Technical	70%	21
Process	20%	6
Human	10%	3

■ Pattern Analysis

MOST COMMON ROOT CAUSES:

1. **CONFIGURATION ERRORS**: **8 incidents**
2. **PERMISSION** and Access Issues: **4 incidents**
3. Resource Limitations (Memory, Concurrency): **3 incidents**
4. **DEPLOYMENT**/Update Mistakes: **3 incidents**
5. **MONITORING**/Alerting Gaps: **2 incidents**

SHARED PATTERNS IDENTIFIED:

- Misconfigurations across AWS **SERVICES** (IAM, EC2, Lambda)
- Overlooked **PERMISSION** settings leading to **FAILURES**
- Inadequate testing or rollback procedures for deployments
- Insufficient **MONITORING** or alerting for critical metrics

ROOT CAUSE CLASSIFICATION:

- **TECHNICAL** Issues: **70%** (**21 incidents**)
- **PROCESS** Issues: **20%** (**6 incidents**)
- Human Factors: **10%** (**3 incidents**)

RECURRING ISSUES DESPITE FIXES:

- **CONFIGURATION ERRORS** and **PERMISSION** issues appear repeatedly, indicating a systemic problem in change management and access control processes.

■ Trend Analysis

CATEGORY BREAKDOWN:

- **PROCESS FAILURE**: **6 incidents**
- **INFRASTRUCTURE**/Equipment: **15 incidents**
- Human **ERROR**: **3 incidents**
- External Factors: **0 incidents**

TEMPORAL PATTERNS:

- A spike in **INCIDENTS** related to **DEPLOYMENT** or **CONFIGURATION** changes towards the end of the year, possibly due to increased release activity.

HIGHEST IMPACT INCIDENTS:

1. Global Video Buffering **INCIDENT** (NFI-2023-0010)
2. Live Stream **FAILURE** (NFI-2023-0018)
3. Regional Failover Test **FAILURE** (NFI-2024-0007)
4. DNS Resolution **FAILURE** (NFI-2024-0004)

5. Data Loss **INCIDENT** (NFI-2023-0019)

■■ Action Effectiveness

CORRECTIVE ACTION ANALYSIS:

- Many corrective actions focus on immediate fixes (e.g., **PERMISSION** adjustments, **CONFIGURATION** changes) without addressing underlying **PROCESS** or knowledge gaps.
- Preventive measures often reactive rather than proactive, suggesting a need for better foresight and planning.

REPEATEDLY APPEARING ACTIONS:

- Implementing stricter IAM policies and **PERMISSION** checks
- Enhancing **MONITORING** and alerting for critical **SYSTEMS**
- Revising **DEPLOYMENT** procedures to include more thorough testing

IMPLEMENTATION GAPS:

- Lack of follow-through on preventive measures, particularly in automating checks and balances for configurations and deployments.

■ Systemic Issues

CROSS-CUTTING PROBLEMS:

- Inadequate change management processes leading to repeated **CONFIGURATION** and **DEPLOYMENT ERRORS**.
- Insufficient training or awareness on AWS best practices and **SERVICE** limitations.
- Poorly defined rollback and emergency procedures.

PROCESS BOTTLENECKS:

- Manual review processes for changes are either missing or ineffective, leading to **ERRORS**.
- Slow detection and response to **INCIDENTS** due to inadequate **MONITORING**.

KNOWLEDGE SHARING ASSESSMENT:

- Lessons learned are not effectively disseminated across teams, leading to repeated mistakes.

■ Strategic Recommendations

TOP 3 HIGH-IMPACT IMPROVEMENTS:

1. **IMPLEMENT A COMPREHENSIVE CHANGE MANAGEMENT PLATFORM** that integrates with AWS **SERVICES** for **AUTOMATED** checks, peer reviews, and

approval workflows to reduce **CONFIGURATION** and **DEPLOYMENT ERRORS**.

2. **DEVELOP AND MANDATE AWS BEST PRACTICES TRAINING** for all engineering staff, focusing on common pitfalls and **SERVICE**-specific limitations to address knowledge gaps.

3. **ENHANCE MONITORING AND ALERTING CAPABILITIES** with a focus on predictive analytics and anomaly detection to identify potential issues before they impact users.

INVESTMENT PRIORITIES:

- Tools for **AUTOMATED CONFIGURATION** and **DEPLOYMENT VALIDATION**.
- Training programs on AWS **SERVICES** and best practices.
- Advanced **MONITORING** and analytics solutions.

EARLY WARNING INDICATORS:

- Deviations in resource utilization patterns (e.g., memory, CPU)
- Increase in **DEPLOYMENT** frequency or rollback activities
- Anomalies in user behavior or application performance metrics

SUSTAINABILITY MEASURES:

- Regular review and update cycles for all operational documentation and runbooks.
- Continuous improvement **PROCESS** for analyzing and acting on **INCIDENT** reports.
- Establish a culture of accountability and continuous learning to prevent stress-related regressions.

Quick Wins

1. **STANDARDIZE IAM POLICY TEMPLATES** and enforce their use across all projects.

2. **IMPLEMENT PRE-FLIGHT CHECKS** in CI/CD pipelines for **CONFIGURATION** and **PERMISSION** settings.

3. **SCHEDULE REGULAR AWS BEST PRACTICES REFRESHERS** for the engineering team.

4. **INTRODUCE A PEER REVIEW REQUIREMENT** for all critical **INFRASTRUCTURE** changes.

5. **AUTOMATE ALERTS FOR COMMON CONFIGURATION MISTAKES** using AWS Config rules.