

NEXUSFLOW INC.
POST-INCIDENT REVIEW (PIR) & ROOT CAUSE ANALYSIS (RCA) REPORTS
Compilation of 15 Fictional Incident Reports

CONFIDENTIAL: For Internal Use Only
Generated: February 21, 2024

INTRODUCTION

This document contains a compilation of 15 fictional Root Cause Analysis (RCA) reports for NexusFlow Inc., a global streaming and data analytics company. These reports cover a variety of scenarios across different platforms (AthenaStream, DataNexus, NexusCommerce, Internal Tools) and AWS services to serve as a learning and training tool for the Cloud Engineering team.

All incidents are fictional and created for educational purposes.

INCIDENT #: NFI-2023-0010
DATE: October 26, 2023
SEVERITY: SEV-1
PLATFORM: AthenaStream
SERVICES: Amazon CloudFront, EC2

TITLE: Global Video Buffering Incident

SUMMARY: Users worldwide experienced severe buffering and failed video playback for 45 minutes.

ROOT CAUSE: A deployment of a new "recommendation engine" to an EC2 Auto Scaling Group (ASG) contained a flawed memory cache configuration. The instances consumed all available memory within minutes of launch, causing them to fail health checks. The ASG continuously terminated unhealthy instances and launched new ones, which also failed. This rapid, continuous cycling of every instance in the pool caused a thundering herd problem on the backend database, crippling the entire service.

RESOLUTION: Rolled back the deployment by attaching the previous launch template to the ASG. Implemented a scaling policy to add instances more gradually.

PREVENTION:

1. Implement canary deployments with a slower traffic shift.
2. Add stricter memory-based health checks in pre-production environments.
3. Define and test rollback procedures under load.

INCIDENT #: NFI-2023-0011
DATE: November 5, 2023

SEVERITY: SEV-2
PLATFORM: DataNexus
SERVICES: AWS Glue, S3

TITLE: ETL Pipeline Failure

SUMMARY: Nightly ETL jobs failed, causing dashboards to display stale data.

ROOT CAUSE: A Glue Job's IAM execution role did not have the `s3:ListBucket` permission on a newly created target S3 bucket. The job could write data but failed during its final step to list and verify the output, causing it to error out.

RESOLUTION: Added the missing `s3:ListBucket` permission to the IAM role. Manually triggered the failed jobs.

PREVENTION:

1. Implement a standardized IAM policy template for Glue jobs.
2. Add a pre-flight check in the CI/CD pipeline to validate required permissions against the target environments.

INCIDENT #: NFI-2023-0012
DATE: November 12, 2023
SEVERITY: SEV-2
PLATFORM: NexusCommerce
SERVICES: Amazon RDS (Aurora PostgreSQL), Lambda

TITLE: Checkout Latency

SUMMARY: Checkout process was timing out for 20% of users during a flash sale event.

ROOT CAUSE: The `orders` table in the Aurora PostgreSQL database lacked an index on the `user_id` and `created_at` columns. A critical query executed by a Lambda function during checkout performed a full table scan, which became exponentially slower as the table grew during the sale event.

RESOLUTION: Added a composite index on `(user_id, created_at)` to optimize the query.

PREVENTION:

1. Implement a process for regular database performance review and index analysis.
2. Use AWS Database Advisor and enable Performance Insights on all RDS instances.

INCIDENT #: NFI-2023-0013
DATE: November 18, 2023
SEVERITY: SEV-1

PLATFORM: Internal Tools
SERVICES: Amazon VPC, NACLs

TITLE: CRM Application Outage

SUMMARY: The entire CRM application became unreachable.

ROOT CAUSE: A network engineer attempted to block a specific malicious IP address by modifying the Network ACL (NACL). The new rule to deny the IP was placed *after* the default "allow all" rule. In the process, a typo was made in the rule number for the standard "Allow All" rule, inadvertently deleting it and replacing it with an effective "Deny All" rule.

RESOLUTION: Identified the erroneous NACL change and restored the correct allow rules.

PREVENTION:

1. Implement infrastructure as code (IaC) for all network changes (e.g., via Terraform).
2. Require a peer review for all manual NACL and Security Group changes.

INCIDENT #: NFI-2023-0014
DATE: November 22, 2023
SEVERITY: SEV-2
PLATFORM: AthenaStream
SERVICES: Amazon CloudFront

TITLE: CDN Cache Invalidation Storm

SUMMARY: Origin server load increased by 300%, causing latency for users.

ROOT CAUSE: A script designed to invalidate cached assets for a new movie release used a wildcard (`/*`) instead of a specific path prefix (e.g., `/assets/movie_123/*`). This invalidated the entire CloudFront cache for the distribution, forcing all subsequent user requests to hit the origin servers simultaneously.

RESOLUTION: The issue resolved itself as the cache gradually warmed back up. Monitored origin capacity to ensure it handled the load.

PREVENTION:

1. Implement approval workflows in CI/CD for invalidation requests containing `/*`.
2. Create a standardized script for teams to use that constructs precise invalidation paths.

INCIDENT #: NFI-2023-0015
DATE: December 1, 2023
SEVERITY: SEV-3
PLATFORM: DataNexus

SERVICES: AWS Cost Explorer, S3

TITLE: Cost Overrun Alert

SUMMARY: Unexpected \$5,000 spike in S3 costs.

ROOT CAUSE: A developer's script, which was intended to run once, was incorrectly configured and began running in an infinite loop. Each iteration uploaded several MBs of test data to an S3 bucket. Over a weekend, this resulted in millions of PUT requests and several TB of storage.

RESOLUTION: Identified and terminated the script. Implemented S3 Lifecycle Policies to automatically delete objects in the target bucket after 1 day.

PREVENTION:

1. Implement S3 Cost Allocation tags.
2. Create AWS Budgets alerts with a lower threshold.
3. Use S3 Intelligent-Tiering for unpredictable access patterns.

INCIDENT #: NFI-2023-0016

DATE: December 5, 2023

SEVERITY: SEV-2

PLATFORM: NexusCommerce

SERVICES: Amazon API Gateway, Lambda

TITLE: Failed Payment Processing

SUMMARY: Payment processing API returned 5xx errors for 15 minutes.

ROOT CAUSE: The Lambda function behind the API Gateway endpoint hit the default concurrency limit (1000). A sudden surge in traffic from a marketing campaign caused the function to throttle, rejecting additional invocation requests.

RESOLUTION: Increased the reserved concurrency limit for the critical function and configured automatic scaling.

PREVENTION:

1. Proactively set appropriate concurrency limits for functions expected to receive high traffic.
2. Use AWS Auto Scaling or configure provisioned concurrency for predictable burst traffic.

INCIDENT #: NFI-2023-0017

DATE: December 10, 2023

SEVERITY: SEV-2

PLATFORM: Internal Tools
SERVICES: AWS IAM Identity Center (AWS SSO)

TITLE: SSO Authentication Failure

SUMMARY: Employees could not log in to the AWS Management Console via SSO.

ROOT CAUSE: The permission set attached to the "ReadOnly" group was accidentally modified, removing the `sts:AssumeRole` permission. This broke the ability for users to assume the role that grants them console access.

RESOLUTION: Re-applied the correct managed policy to the permission set.

PREVENTION:

1. Define permission sets as code.
2. Restrict modify permissions on production permission sets to a senior cloud admin group.

INCIDENT #: NFI-2023-0018
DATE: December 15, 2023
SEVERITY: SEV-1
PLATFORM: AthenaStream
SERVICES: AWS Elemental MediaLive

TITLE: Live Stream Failure

SUMMARY: A major live sporting event stream failed to start on time.

ROOT CAUSE: The MediaLive channel was in a "IDLE" state and needed to be manually started. The runbook for live events assumed the channel would be started via an automated Lambda function, but a configuration error in the function's event source (Amazon EventBridge) prevented the execution command from being sent.

RESOLUTION: Manually started the channel, causing a 7-minute delay to the live stream.

PREVENTION:

1. Implement a dual-check process where an engineer must confirm the channel status 1 hour before a major event.
2. Add a CloudWatch alarm for channels that are in "IDLE" state 30 minutes before a scheduled start.

INCIDENT #: NFI-2023-0019
DATE: December 20, 2023
SEVERITY: SEV-1
PLATFORM: DataNexus

SERVICES: Amazon DynamoDB

TITLE: Data Loss Incident

SUMMARY: One week of user analytics data was deleted from a DynamoDB table.

ROOT CAUSE: A developer ran a script with a `DeleteItem` command in a production environment, intended for a pre-production table. The script had a hard-coded table name variable pointing to the production table ARN.

RESOLUTION: Restored the table to a point-in-time backup from 4 hours prior using PITR. The 4 hours of data between the backup and the incident were lost.

PREVENTION:

1. Enforce the use of environment-specific configuration files, never hard-coded ARNs.
2. Implement mandatory infrastructure and tooling reviews for scripts performing destructive operations.

INCIDENT #: NFI-2024-0001

DATE: January 8, 2024

SEVERITY: SEV-2

PLATFORM: All

SERVICES: AWS SDK, us-west-2 Region

TITLE: Regional API Latency

SUMMARY: Applications experienced intermittent timeouts when calling AWS APIs in `us-west-2`.

ROOT CAUSE: An application's retry logic was overly aggressive. During a brief period of network congestion, the application's exponential backoff was disabled, causing it to retry failed calls instantly and repeatedly. This created a retry storm that further degraded performance.

RESOLUTION: The network congestion resolved. The application's flawed retry logic was identified and fixed.

PREVENTION:

1. Mandate the use of the latest AWS SDKs, which have built-in, sane retry mechanisms.
2. Code reviews must include logic for API call retries and backoff.

INCIDENT #: NFI-2024-0002

DATE: January 15, 2024

SEVERITY: SEV-2

PLATFORM: NexusCommerce
SERVICES: Amazon ElastiCache (Redis)

TITLE: Shopping Cart Emptying

SUMMARY: User shopping carts were spontaneously emptying.

ROOT CAUSE: The Redis cluster's `maxmemory-policy` was set to `volatile-lru`. A memory pressure event caused Redis to evict keys with a TTL to make space. However, many cart items were stored without a TTL and should have been persistent. The policy was incorrectly set.

RESOLUTION: Scaled the ElastiCache node type to one with more memory. Corrected the eviction policy.

PREVENTION:

1. Define CloudFormation templates for ElastiCache to ensure consistent configuration.
2. Implement CloudWatch alarms for memory utilization.

INCIDENT #: NFI-2024-0003
DATE: January 22, 2024
SEVERITY: SEV-3
PLATFORM: Internal Tools
SERVICES: AWS Backup

TITLE: Backup Failure

SUMMARY: Scheduled nightly backups for an EC2 instance failed for 5 consecutive days.

ROOT CAUSE: The EC2 instance's IAM role lacked the `aws-backup` tag. The AWS Backup resource selection was configured to automatically include only instances with that specific tag. The instance was never included in the backup plan.

RESOLUTION: Added the required tag to the instance IAM role. Manually initiated a backup to verify success.

PREVENTION:

1. Create a proactive compliance check using AWS Config to flag EC2 instances missing the backup tag.
2. Perform periodic recovery drills.

INCIDENT #: NFI-2024-0004
DATE: January 30, 2024
SEVERITY: SEV-1

PLATFORM: AthenaStream
SERVICES: Amazon Route 53

TITLE: DNS Resolution Failure

SUMMARY: The primary domain `athenastream.com` became unresolvable.

ROOT CAUSE: During a DNS record update, a misconfigured CI/CD pipeline accidentally deleted the entire hosted zone's NS (Name Server) records instead of just updating the A-record alias.

RESOLUTION: Recovered the NS records from a Terraform state file and reapplied them to the hosted zone. Full DNS propagation took ~30 minutes.

PREVENTION:

1. Implement manual approval gates in the deployment pipeline for changes to critical DNS records (especially NS and SOA).
2. Use Terraform `prevent_destroy` lifecycle flags on the hosted zone resource.

INCIDENT #: NFI-2024-0005
DATE: February 5, 2024
SEVERITY: SEV-2
PLATFORM: DataNexus
SERVICES: Amazon Redshift

TITLE: Redshift Query Performance Degradation

SUMMARY: Business intelligence queries were running 10x slower than usual.

ROOT CAUSE: A large, unoptimized `COPY` operation from S3 loaded a massive dataset without sorting it on the intended sort key. This led to massive zone maps being ineffective and required Redshift to scan almost all blocks for subsequent queries.

RESOLUTION: Ran a `VACUUM FULL` and `ANALYZE` on the affected table to re-sort the rows and update statistics.

PREVENTION:

1. Implement data loading best practices: use sort keys, distribute keys, and break large copies into smaller files.
 2. Schedule regular maintenance operations during off-peak hours.
-