

POSTMORTEM REPORT

Incident: Web Stack Outage

Date: February 17, 2024

Duration: 2hrs 15 Min - 09:30 PM - 11:45 PM (EAT)

Issue Summary

February 17, 2024, 09:30 PM - 11:45 PM (EAT) Impact: Web Application Unavailability, 25% of Users Affected Root Cause: Database Connection Pool Exhaustion

Timeline

- 09:30 PM: Issue detected as monitoring system triggered alerts for increased latency.
- 09:32 PM: Engineers began investigating the alert, suspecting a database-related problem.
- 09:40 PM: Initial assumption: Database server load was high due to a recent code release.
- 09:50 PM: Further investigation revealed no significant changes in codebase or database queries.
- 10:00 PM: Initial investigation indicated a possible network issue between the web servers and the database.
- 10:15 PM: Network analysis showed no anomalies; suspicion shifted back to application code.
- 10:30 PM: Incident escalated to senior database administrators as well as the network team.
- 10:45 PM: Load testing conducted to simulate traffic spike, but the issue didn't reproduce.
- 11:00 PM: Engineers identified abnormally high open database connections.
- 11:15 PM: Realized that database connection pooling was misconfigured, causing exhaustion.
- 11:30 PM: Connection pool settings adjusted to accommodate higher traffic.
- 11:45 PM: Web application fully restored, services resumed normal operation.

Root Cause and Resolution

Root Cause

The root cause of the outage was an incorrect configuration of the database connection pooling settings. The connection pool was set to a lower value than required by the increasing traffic, causing the available connections to be exhausted quickly. As a result, new incoming requests were delayed or rejected.

Resolution

To address the issue, the connection pool settings were reconfigured to increase the maximum allowed connections, effectively accommodating the current traffic load. Additionally, connection timeout settings were optimized to ensure timely release of unused connections back to the pool. These adjustments prevented connection exhaustion and improved overall application performance.

Corrective and Preventative Measures: Improvements/Fixes:

1. Implement automated load testing during continuous integration to catch resource-related regressions early.
2. Enhance monitoring and alerts for abnormal database connection usage patterns.
3. Establish a comprehensive incident response plan involving database administrators and network experts.

Tasks to Address the Issue:

1. Update connection pool settings to ensure a sufficient number of connections are available during peak load.
2. Deploy a database connection monitoring tool to track connection usage in real-time.
3. Develop a playbook for diagnosing and addressing connection-related issues, including clear steps for adjusting settings.
4. Conduct a thorough review of the application's resource consumption patterns to identify any other potential bottlenecks.
5. Enhance documentation and training for engineers on best practices for configuring and managing connection pools.

Lessons Learned

This incident highlighted the critical importance of proper resource configuration in maintaining application stability. It also emphasized the need for collaborative cross-team communication during troubleshooting. By implementing proactive monitoring and refining incident response processes, we aim to mitigate similar issues in the future and enhance the overall reliability of our web application.

Conclusion

The outage on February 17, 2024, served as a reminder that seemingly subtle misconfigurations can lead to substantial service disruptions. Through swift collaboration, diligent investigation, and the implementation of corrective measures, we were able to quickly resolve the issue and restore normal operation. We remain committed to continuous improvement, enhancing our monitoring, and fortifying our system against potential vulnerabilities.