### **Postmortem: Outage of Web Application Due to Database Connection Pool Exhaustion**

#### **Issue Summary**

* **Duration of Outage**: 2 hours, from 10:00 AM to 12:00 PM UTC on October 10, 2023.
* **Impact**: The web application experienced a complete outage, rendering it inaccessible to all users. 100% of users were affected, resulting in failed login attempts, transaction timeouts, and inability to access critical services.
* **Root Cause**: The database connection pool was exhausted due to a misconfigured connection timeout setting, leading to a cascading failure where new requests could not be served.

#### **Timeline**

* **10:00 AM**: The issue was detected when the monitoring system triggered an alert indicating a spike in database connection errors.
* **10:05 AM**: Engineers noticed that the application servers were unable to establish new database connections. Initial assumption was a database server overload.
* **10:15 AM**: The database team was engaged to investigate server load and query performance. No anomalies were found in the database itself.
* **10:30 AM**: Application logs were reviewed, revealing a high number of idle connections in the connection pool. Misleadingly, the team suspected a connection leak in the application code.
* **10:45 AM**: The incident was escalated to the platform engineering team to investigate the connection pool configuration.
* **11:15 AM**: The root cause was identified as a misconfigured connection timeout setting in the application’s database connection pool.
* **11:30 AM**: The configuration was updated to reduce the connection timeout and increase the pool size.
* **12:00 PM**: The application was fully restored after restarting the servers to apply the new configuration.

#### **Root Cause and Resolution**

* **Root Cause**: The database connection pool was configured with an excessively high connection timeout (600 seconds), causing connections to remain open longer than necessary. Under high traffic, this led to exhaustion of the connection pool, preventing new connections from being established.
* **Resolution**: The connection timeout was reduced to 60 seconds, and the maximum pool size was increased from 50 to 100. Additionally, the application servers were restarted to clear the existing pool and apply the new settings.

#### **Corrective and Preventative Measures**

* **Improvements/Fixes**:
  1. Review and optimize database connection pool settings across all environments.
  2. Implement better monitoring for connection pool usage and database performance.
  3. Conduct regular load testing to identify bottlenecks before they impact production.
* **Tasks**:
  1. Update the connection pool configuration in all environments to use a timeout of 60 seconds and a maximum pool size of 100.
  2. Add a monitoring dashboard to track connection pool usage and alert on thresholds (e.g., 80% pool utilization).
  3. Schedule load testing for the next release cycle to simulate high traffic scenarios.
  4. Document the incident and share learnings with the engineering team to prevent similar issues in the future.