此内容的所选语言版本不可用。我们一直在不断努力，以便以所选语言提供我们的内容。感谢您的耐心等待。

Summary of the Amazon SimpleDB Service Disruption

We wanted to share what we've learned from our investigation of the June 13 SimpleDB disruption in the US East Region. The service was unavailable to all API calls (except a fraction of the eventually consistent read calls) from 9:16 PM to 11:16 PM (PDT). From 11:16 PM to 1:30 AM, we continued to have elevated error rates for CreateDomain and DeleteDomain API calls.

SimpleDB is a distributed datastore that replicates customer data across multiple data centers. The service employs servers in various roles. Some servers are responsible for the storage of user data (“storage nodes”), with each customer Domain replicated across a group of storage nodes. Other nodes store metadata about each customer Domain (“metadata nodes”), such as which storage nodes it is located on. SimpleDB uses an internal lock service to determine which set of nodes are responsible for a given Domain. This lock service itself is replicated across multiple data centers. Each node handshakes with the lock service periodically to verify it still has responsibility for the data or metadata it hosts.

In this event, multiple storage nodes became unavailable simultaneously in a single data center (after power was lost to the servers on which these nodes lived). While SimpleDB can handle multiple simultaneous node failures, and has successfully endured larger infrastructure failures in the past without incident, the server failure pattern in this event resulted in a sudden and significant increase in load on the lock service as it rapidly de-registered the failed storage nodes from their respective replication groups. This simultaneous volume resulted in elevated handshake latencies between healthy SimpleDB nodes and the lock service, and the nodes were not able to complete their handshakes prior to exceeding a set “handshake timeout” value. After several handshake retries and subsequent timeouts, SimpleDB storage and metadata nodes removed themselves from the SimpleDB production cluster, and SimpleDB API requests returned error messages (http response code 500 for server-side error). The affected storage nodes were not able to rejoin the SimpleDB cluster and serve API requests until receiving authorization to rejoin from metadata nodes. This process ensures that we do not allow a node with stale data to join the production cluster accidentally and start serving customer requests. However, in this case the metadata nodes were also down due to the same handshake timeout issue, and therefore could not authenticate the storage nodes.

Once the problem was identified, we had to manually increase the handshake timeout values and restart a subset of metadata nodes so that they could authorize the storage nodes. This allowed the affected storage nodes to rejoin the SimpleDB cluster and resume serving customer data requests. At this point (11:16 PM), all APIs but CreateDomain and DeleteDomain were functioning normally. To allow the rest of the metadata nodes to fully recover without risk, we throttled CreateDomain and DeleteDomain API calls (which are served from metadata nodes) until 1:30 AM.

We have identified two significant improvements that can be made to SimpleDB coming out of the event to prevent recurrence of similar issues. First, we will set a longer lock service handshake timeout. The original intent behind the low handshake timeout value we set was to enable rapid detection of replica failure. However, hindsight shows the value was too low. Second, the behavior of nodes removing themselves from the SimpleDB cluster immediately after experiencing multiple handshake timeouts increased the scope of the event and caused SimpleDB API errors. Instead, the nodes should have waited and retried handshake requests later with an increased handshake timeout value. We are addressing these two issues immediately and rolling out fixes to all SimpleDB Regions. We apologize for the impact this issue had on SimpleDB customers.

Sincerely,
The AWS Team

Learn About AWS

What Is AWS?
What Is Cloud Computing?
What Is DevOps?
What Is a Container?
What Is a Data Lake?
AWS Cloud Security
What's New
Blogs
Press Releases

Resources for AWS

Getting Started
Training and Certification
AWS Solutions Portfolio
Architecture Center
Product and Technical FAQs
Analyst Reports
AWS Partner Network

Developers on AWS

Developer Center
SDKs & Tools
.NET on AWS
Python on AWS
Java on AWS
PHP on AWS
Javascript on AWS

Help

Contact Us
AWS Careers
File a Support Ticket
Knowledge Center
AWS Support Overview
Legal

Create an AWS Account

Amazon is an Equal Opportunity Employer: *Minority / Women / Disability / Veteran / Gender Identity / Sexual Orientation / Age.*

Language عربي | Bahasa Indonesia | Deutsch | English | Español | Français | Italiano | Português | Tiếng Việt | Türkçe | Русский | ไทย | 日本語 | 한국어 | 中文 (简体) | 中文 (繁體)

Privacy | Site Terms | Cookie Preferences |