the Availability Digest

www.availabilitydigest.com
@availabilitydig

Amazon S3 Storage Taken Down by Fat Finger

March 2017

Amazon Simple Storage Service (Amazon S3) stores objects via a simple web interface. S3 is designed to scale past trillions of objects worldwide. Once data is stored in S3, it can be automatically tiered into lower cost, longer-term cloud storage. Many web sites depend upon S3 to store their data.

The S3 Service Disruption

Amazon's S3 storage has been unbelievably available and reliable. Amazon claims eleven 9s when running redundant copies in multiple regions. However, on the morning of February 28th, 2017, S3 storage failed while an Amazon team was debugging an issue that was causing the S3 billing system to run slowly in the Northern Virginia (US-EAST-1) Region.

At 9:37 AM PST, a team member executed a command to remove a small number of servers from one of the S3 subsystems used by the billing process. Unfortunately, the team member entered the command incorrectly; and a larger set of servers than was intended was removed.

One of the servers removed was the Index Subsystem. The Index Subsystem manages the location information of all S3 objects in its Region. The Index Subsystem is necessary to process all GET, LIST, PUT, and DELETE requests.

Another server that was removed was the Placement Subsystem. The Placement Subsystem manages the updating of S3 storage. In order to function, the Placement Subsystem uses PUT requests to allocate storage for new objects. Without the Index Subsystem being available, PUT requests would not function.

Without S3, much of the Internet ground to a halt. Both the Index Subsystem and the Placement Subsystem required a full restart. During the recovery, S3 was unable to process requests. Other AWS services in the region that rely on S3 for storage also were unavailable during the restart.

Amazon had not restarted either the Index Subsystem or the Placement Subsystem for several years. S3 has experienced massive growth over the last several years, and the process of restarting the subsystems took longer than expected.

It took four hours to recover the Index Subsystem. The Placement Subsystem and the PUT API took another hour to recover. Other services that relied on S3 had accumulated a backlog of work while S3 was down. The outages of some of these services lasted up to eleven hours.

Website Outages

The S3 outage took down dozens of major websites, including Apple's website. However, Amazon's Service Health Dashboard (SHD) reported that everything was operating normally. The problem with the

SHD was that it could not be updated because it relied on the Northern Virginia S3 to receive updated status information.

With no SHD, Amazon turned to Twitter to communicate the status of S3. Recognizing that the SHD could not be updated from a single-region S3 that was currently down, Amazon modified the SHD so that it ran across multiple regions. This allowed the SHD to be updated with the Northern Virginia Region S3 fault status from the S3 in another region.

Amazon Upgrades Its Removal Tool

To prevent a recurrence of team members removing subsystems too rapidly, Amazon modified its removal tool to force it to remove S3 capacity more slowly. The tool will now prevent removing capacity from any subsystem that would take the subsystem below its minimum required capacity.

The Need for Cloud Redundancy

How can you use the world's biggest cloud provider and still avoid downtime when the provider has a major outage? The answer is to deploy redundant storage.

The S3 outage demonstrates the need to have backup storage. No matter how reliable S3 storage has proven to be, it can still fail. Such backup storage can be provided by another S3 region, or it may be provided via another storage service altogether.

The agility and flexibility of cloud computing combined with lower startup costs help organizations spend less time and money on infrastructure and computing resources. The focus of expenditures moves from capital expenses to higher operating costs. Therefore, redundant storage can perhaps best be built into a redundant cloud infrastructure.

You can keep multiple copies of objects and virtual machines in multiple clouds located in different regions to prevent loss due to a common catastrophe. The multiple regions could be provided by the same provider. Alternatively, a more ideal arrangement would be to have the clouds run by multiple cloud providers such as Amazon Web Services, Microsoft Azure, or the Google Cloud Platform. Add your own data centers to the mix to really enhance the redundancy of your data and services.

Since these redundancies span geographical regions, WAN replication[1] is probably going to have to be employed to keep all the data copies synchronized.

Amazon's recommendation is to deploy a virtual private cloud into at least two regions. The regions could be in the same Amazon Availability Zone, or they could be resident in multiple (geographically dispersed) Availability Zones in different Amazon regions.

Summary

Amazon S3 storage went down because of a human fat-finger error. As we have said many times in the *Digest*, humans need to be redundant also. If a critical command is about to be entered, do it with two people – one to enter it and one to check it before it gets executed. If Amazon had done this, they would have averted an embarrassing outage.

[1] See the HPE Shadowbase data replication engine.
http://shadowbasesoftware.com/

Acknowledgements

Thanks to our subscribers Jim Keller and Terry Critchley for pointing us to this incident.

Information for this article was taken from the following sources:

Summary of the Amazon S3 Service Disruption in the Northern Virginia ((US-EAST-1) Region, *Amazon Blog*; undated.
AWS outage proves one cloud isn't enough, *InfoWorld*; March 2, 2017.
AWS Outage Demonstrates Need For Redundancy Even In The Cloud, *Silicon*; March 2, 2017.
How a typo took down S3, the backbone of the Internet, *The Verge*; March 2, 2017.
Analysis: Rethinking cloud architecture after the outage of Amazon Web Services, *Geek Wire*; March 5, 2017.
How to Survive a Cloud Meltdown, *Data Center Knowledge*; March 7, 2017.