the Availability Digest

www.availabilitydigest.com
@availabilitydig

# Can You Trust Your Public Cloud?

June 2014

Joyent is the latest cloud to bite the dust. It follows a long series of public-cloud failures by even the largest cloud providers – Amazon, Google, Microsoft, Salesforce, and Rackspace, to name a few. Joyent's failure emphasizes the need to be prepared for your public-cloud services to suddenly disappear for hours and sometimes for days. Even worse, your data might disappear.

In this article, we review many of the public-cloud failures and see what we can learn about trusting cloud services and about protecting our applications and data from their faults.

Joyent Cloud is Joyent's hosting service. It is designed to complete with Amazon's EC2 cloud, providing Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) for large enterprises. Though Joyent is aimed mainly at the online social-network game market, it hosted Twitter in Twitter's early days.

A Joyent system administrator was updating software on some new servers when he accidentally typed the wrong command. Instead of causing just those servers to reboot, he commanded all of the servers in the data center to reboot. Everything went down. Due to the heavy loading imposed upon the boot infrastructure and the control plane, it took hours for the servers to reboot.

The fundamental problem was that the tool that the administrator was using did not have enough input validation to prevent this from happening. There was no "Are you sure you want to reboot all servers" query.

Joyent did not fire the administrator – they pointed out that his mortification was punishment enough. However, they have initiated a major project to enhance all administrative tools so that something like this scenario will never happen again.[1]

Shades of Y2K! Microsoft's Windows Azure Cloud went down for over a day on Wednesday, February 29, 2012. Starting around midnight as the clock ticked over to Leap Day, various subsystems of the Azure Cloud started to fail one-by-one. Soon, applications for many customers became unresponsive.

The outages resulted from a date-related problem in the virtual machine (VM) security certificates that are used in the Azure Cloud to authenticate VMs to their physical hosts. Unless a VM has a valid certificate, the host system will not allow it to run.

[1] Fat-fingered admin downs entire Joyent data center, *Availability Digest*; June 2014. http://www.availabilitydigest.com/public_articles/0906/joyent.pdf

The Azure Cloud VM certificates are good for one year. The certificate-creation program simply incremented by one the year of the current date to get the certificate's expiration date. This meant that a certificate created on Leap Day, February 29, 2012, had an expiration date of February 29, 2013. Unfortunately, there was no such date; and the certificate was deemed invalid.

VMs carrying this invalid certificate could not connect with their hosts. This led the Azure Cloud to determine that these physical host servers were faulty, and they were taken out of service. All of their VMs were moved to other hosts, which suffered the same fate and were taken out of service. The failure of host servers cascaded throughout the Azure Cloud, ultimately taking it down.

It was not until 8 AM Thursday morning, thirty-two hours later, that Microsoft reported the completion of recovery efforts.[2]

What should have been a ten-minute outage at a major Google data center hosting the Google App Engine service turned into a two-and-a-half hour ordeal simply because of erroneous failover documentation.

Following a power outage, Google's backup power kicked in. However, due to a faulty power switch, about 25% of the servers in the data center did not receive backup power and subsequently went down. Their applications failed over to surviving servers in the data center, which were not configured to handle that much additional load; and they failed. The server failures cascaded throughout the data center.

The decision was made to fail over to the backup data center, which had been newly reconfigured. The documented procedures were followed but led to an unsuccessful failover.

It then looked like the primary data center had returned to operation, so processing was returned to it. This turned out to be an erroneous observation since the primary data center with only 75% of its servers operating still could not handle the full load, and it once again failed.

Finally, knowledgeable technical personnel were reached; and the backup data center was brought successfully online. A post mortem discovered that parts of the documentation of the new failover procedures incorrectly referred to the old data-center configuration rather than to the upgraded configuration. Clearly, the newly documented failover procedures had not been tested; nor had the staff been trained. Otherwise, these errors would have been found.[3]

Amazon is one of the largest cloud-service providers in the world. Its Amazon Web Services (AWS) comprises Elastic Compute Cloud (EC2) and the Simple Storage Service (S3). S3 is an online storage service that a customer can use to store and retrieve an unlimited number of data objects through a simple web-services interface, accessing each data object via its unique URL.

Access to S3 is controlled by the AWS Authentication service. Early one morning, system operators started seeing unusually elevated levels of authentication requests at one of their locations that provide AWS Authentication services. This heavy activity ultimately overloaded the S3 infrastructure at that site, and Amazon was unable to process any further requests at that location. Further requests were rerouted to other AWS service sites, but the increased load caused them to fail also. S3 services were down.

It took over three hours to reconfigure enough capacity to handle the increased authorization load and to return S3 services to Amazon's customers. During this time, thousands of web sites that depended upon S3 and its companion EC2 service were down.

[2] Windows Azure Cloud Succumbs to Leap Year, *Availability Digest*; March 2012.
http://www.availabilitydigest.com/public_articles/0703/azure.pdf
[3] Poor Documentation Snags Google, *Availability Digest*; April 2010.
http://www.availabilitydigest.com/public_articles/0504/google_power_out.pdf

An outcry arose from the customer community about Amazon's lack of communication during the outage. Customers complained that they had no way to know whether the problem was due to their own equipment or to Amazon's services. There were neither email notifications nor updates on Amazon's AWS blog. In response to these customer complaints, Amazon developed a service-health dashboard indicating the status of its various services.[4]

Once again for Amazon. Amazon has gone to great lengths to ensure the availability of its cloud services. It has broken its cloud infrastructure into multiple regional, geographically-separated data centers; and within each region it provides independent Availability Zones. A customer may run a critical application in multiple Availability Zones within a region to ensure availability.

However, a maintenance error took down an entire Availability Zone. The outage began with an upgrade to increase the capacity of the primary network of one of the Availability Zones. The first step was to shift all traffic off of the primary network to one of the other high-speed primary links. Unfortunately, a technician error caused all traffic to be rerouted instead to a much slower secondary network. The secondary network could not handle the traffic. Consequently, the compute nodes in the Availability Zone were effectively isolated from their storage subsystems. So far as the nodes were concerned, their database replicas were down.

The nodes started searching the Availability Zone for other storage subsystems that they could use to remirror their data. However, being isolated from the other storage subsystems, many nodes were left continuously searching the Availability Zone for free storage space.

This *remirroring storm* exposed in the Amazon software an unknown bug that caused a node to fail if it was trying to close a large number of replication requests. This caused more nodes to fail and increased the intensity of the remirroring storm. Many customers who were not running applications redundantly across Availability Zones were down for the four days that it took to restore service to the region.[5]


One evening, companies and individuals around the world began to lose their web sites and email services. An estimated fifteen million web sites and an untold number of email accounts suffered failure and did not recover until six hours later. Most of the web sites were those of individuals or small businesses. Small businesses lost massive amounts of revenue in lost sales.

This catastrophe was caused by an outage incurred by GoDaddy. GoDaddy is a major Internet domain registrar and web-hosting company. It is the largest registrar in the world, having registered more than 45 million domain names, four times more than its nearest competitor.

Initial conjecture was that GoDaddy had been taken down by a DDoS attack. In fact, one attacker took credit for the attack. However, GoDaddy was busy getting to the root cause of the failure and finally announced that the outage was not caused by a DDoS attack at all. Rather, it was an internal networking problem that caused corruption of routing tables directing traffic to its DNS servers.

With no access to its DNS servers, domain URLs managed by GoDaddy could not be converted to IP addresses; and those web sites and email domains therefore could not be reached. Not only were the web sites and email addresses hosted by GoDaddy inaccessible, but so were those hosted by other companies but whose domain names were hosted by GoDaddy on its DNS servers.[6]

---

[4] How Many 9s in Amazon?, *Availability Digest*; July 2008.
http://www.availabilitydigest.com/public_articles/0307/amazon.pdf

[5] Amazon's Cloud Downed by Fat Finger, *Availability Digest*; May 2011.
http://www.availabilitydigest.com/public_articles/0605/amazon_ebs.pdf

[6] GoDaddy Takes Down Millions of Web Sites, *Availability Digest*; September 2012.
http://www.availabilitydigest.com/public_articles/0709/godaddy.pdf

It's not a good idea to test a fire-suppression system by triggering it. But that's what happened to WestHost, a major web-hosting provider. The accidental release of a blast of fire-suppressant gas severely damaged most of its data stores.

The WestHost fire-suppression system works by releasing Inergen gas. Comprising common atmospheric gases, Inergen is environmentally friendly and breathable by people; but it reduces the oxygen content of air to a level that does not support combustion.

The WestHost data center underwent a standard yearly test of its Inergen fire-suppression system. Unfortunately, a test technician did not remove one of the actuators that activated the system. When the system was re-armed following the test, the actuator fired and triggered the release of a large blast of Inergen gas designed to put out a fire.

Hundreds of disk-storage systems were severely damaged. WestHost operations immediately came to a halt, and it was days before full service was restored. Recovery efforts put WestHost's customers out of commission for up to six days. Unfortunately, the backup disks were in the same data center and suffered the same damage. Much of WestHost's customers' data was irretrievably lost.

Subsequent tests by Siemens, the manufacturer of the fire-suppression system, and Tyco, the maker of Inergen gas, showed that it was not the sudden increase in gas pressure but the blast of the fire sirens that damaged the disks. Among their recommendations – aim the sirens away from the disk enclosures.[7]

Salesforce.com is a software on-demand utility providing Customer Relationship Management software services to its customers. Its users depend upon critical customer and sales data held by the Salesforce.com data centers to run their daily businesses. As a utility, the Salesforce.com services are expected to be always available.

Salesforce.com is an Oracle RAC user along with Oracle's TimesTen In-Memory database. However, Salesforce.com was pushing Oracle to its limits. At the time, Salesforce.com's database measured in the multiterabytes; and its systems supported over 350,000 users from 18,700 companies generating millions upon millions of queries and transactions per day.

Therefore, Salesforce.com decided to upgrade to the latest version of Oracle. However, Salesforce.com almost immediately began having problems with its new Oracle RAC cluster. The cluster would crash, or it would not fail over when a server failed. Problems continued for four months, with many outages lasting for hours. Oracle assigned a team of senior engineers to Salesforce.com to resolve the issues.

To make matters worse, except for its premium customers, Salesforce.com did not communicate with its customers the status of each outage. Even worse, it characterized these outages as "minor incidents." This led to an outpouring of anger in the form of several blogs. Salesforce.com has since taken a major step to improve customer communications by establishing a tracking facility called "trust.Salesforce.com."[8]

Rackspace, a major hosting service for thousands of web sites, went down for reasons that would be hard to anticipate – a truck driver's heart attack caused his truck to hit a transformer that powered the Rackspace data center. The transformer exploded, and the data center went black. In spite of triply-redundant power backup, this incident started a sequence of events that resulted in many of the web sites that Rackspace hosted to go down for hours.

[7] Fire Suppression Suppresses WestHost for Days, *Availability Digest*; May 2010. http://www.availabilitydigest.com/public_articles/0505/westhost.pdf

[8] On-Demand Software Utility Hits Availability Bump, *Availability Digest*; October 2007. http://www.availabilitydigest.com/public_articles/0210/salesforce.pdf

As planned, when power first failed, Rackspace's emergency diesel backup generator kicked in. The data center came back to life and continued in operation with but a brief interruption. This allowed Rackspace operators to switch to their secondary power source – a completely separate utility line feeding the building. At this point, the emergency generator had done its job and was shut down.

However, fifteen minutes later, the secondary power source shut down. This time, the emergency personnel requested the power utility to shut down power while they were trying to free the trapped truck driver so as to avoid electrocution of not only the truck driver but also of the emergency workers.

Again, the emergency generator started up and continued to power the data center; and the data center was once again operational with little impact on the hosted web sites. However, the dual outages caused the data-center chillers to recycle, a process that took about half an hour. With temperatures in the data center dangerously rising, management had no choice but to shut down its servers. It took until the following day to get the hosted web sites back online.[9]

Hostway is one of the largest web-site hosting services in the world. When Hostway attempted to move 3,700 servers of the Miami data center of a recently acquired company to its own facilities in Tampa, Florida, the servers suffered multiple failures in transit. The result was that the web sites of 500 customers, many of them small online stores, were down for days. A week later, several customers were still down.

The move was planned to take place over a weekend, and Hostway notified impacted customers that their web sites would be down for 12 to 15 hours.

According to plan, Hostway began the process of powering down, disconnecting, packing, loading, moving, repositioning, reconnecting, and testing the servers prior to returning them to service along with the web sites that they hosted. Unfortunately, it found that many hard drives were destroyed due to physical shock during the moving process.

Hostway lost about 500 servers due to the disk failures. A week later, some servers had yet to be restored.

Most of the customers that were affected were using dedicated hosting services. Each failed server represented one customer. Therefore, about 500 customers experienced multiple-day outages. Many of them were small online stores that were literally out of business during that time.[10]

The Planet is one of the world's largest providers of dedicated servers. Its servers are used by thousands of web-hosting companies supporting millions of web sites.

Early one evening, an explosion took down The Planet's Houston data center. A short circuit in a high-voltage wire conduit set a transformer on fire, which then caused an explosion of battery-acid fumes from the UPS battery-backup system.

The explosion was strong enough to blow down three walls surrounding the electrical equipment room on the first floor of the data center. It blew apart the power-transfer switch that transferred the data center from utility power to backup diesel-generator power, thus knocking out power to the entire data center. Fortunately, no one was injured.

[9] Rackspace – Another Hosting Service Bites the Dust, *Availability Digest*; December 2007. http://www.availabilitydigest.com/public_articles/0212/rackspace.pdf

[10] Hostway's Web Hosting Service Down for Days, *Availability Digest*; September 2007. http://www.availabilitydigest.com/public_articles/0209/hostway.pdf

Though no servers or networking equipment were damaged, 9,000 servers leased by 7,500 customers were brought down due to the power outage. More than one-million retail sites were affected by the explosion, denying service to millions of Internet users.

For safety reasons, the fire department evacuated the building and directed that the backup generators could not be turned on. It wasn't until after 10 pm that staff were allowed back into the building to assess the damage. Four days after the explosion, full service to all customers had yet to be restored.[11]

JournalSpace The blogging service JournalSpace suddenly went out of business when it lost its entire database of blogs that had not been backed up and could not be recovered. Thousands of bloggers lost years of their work.

The database's demise was due to the malicious act of a disgruntled employee – even worse, the IT manager. JournalSpace claims that it had caught the IT manager stealing from the company. They summarily fired him; but he did a slash-and-burn on his way out, overwriting the entire blog database with garbage.

It was the IT manager's responsibility to ensure that a backup copy of the database was periodically taken and preserved. However, though he dutifully backed up the HTML code for the site on a remote server, his backup strategy for the blog database was to use a RAID 2 mirrored disk. If one disk failed, the database was still available on the mirror.

Upper management should have known that this was not a backup strategy at all. True, it protected against a hard-disk failure. But it did not protect against a site disaster – or a malicious act of overwriting the entire database.

In a panic, the JournalSpace management attempted to reconstruct the data on the overwritten disks. Unfortunately, the disks were unrecoverable. JournalSpace closed its doors days later.[12]


Sidekick is a smart phone produced by Danger Incorporated that is offered by T-Mobile. The Sidekick service stored all of its subscribers' data, including address books, calendars, photos, to-do lists, and email messages in its server complex, a large Oracle RAC cluster. Sidekick provided no way for its subscribers to back up their data offline.

Danger and its Sidekick smartphone were acquired by Microsoft. One day, Microsoft set out to upgrade the storage area network used by the RAC cluster. Unfortunately, it decided that the upgrade was so minor that it did not need to back up the database before the upgrade.

Bad decision. The upgrade did not go well. Both the primary and backup databases were corrupted. Sidekick subscribers reported losing all of their stored data – photos, address books, and all.

Microsoft started a major effort to recover the lost data. However, a month later, it was only able to restore some of the data. Most subscriber data was irretrievably lost. Microsoft has since taken steps to improve the reliability of Sidekick storage to ensure that this disaster will not be repeated.[13]

[11] The Planet Blows Up, *Availability Digest*; September 2008.
http://www.availabilitydigest.com/public_articles/0309/planet_explosion.pdf
[12] Why Back Up?, *Availability Digest*; April 2009.
http://www.availabilitydigest.com/public_articles/0404/journalspace.pdf
[13] Sidekick: Your Data is in 'Danger', *Availability Digest*; November 2009.
http://www.availabilitydigest.com/public_articles/0411/sidekick.pdf

## Lessons Learned

Many clouds provide SLAs guaranteeing three 9s of availability or better. But a multi-day outage, as has been experienced by many of the largest clouds, can reduce cloud availability for the year to two 9s. A several-day outage is extremely painful to an organization even for non-critical applications. It is intolerable for critical applications, many of which cannot withstand outages lasting more than several minutes.

The SLAs are often not much help. They typically offer a month's rebate on the charges paid by the customer for their cloud services. For a small online store that has lost hours or days of revenue, this can amount to a paltry few dollars as compensation.

The lessons provided by these cloud-disaster examples are clear:

- Clouds are not yet suitable for critical applications unless they can be run in a redundant mode such as Amazon's Availability Zones. This capability is beginning to be offered by more clouds.

- Another redundant mode suitable for critical applications is to have the capability to move them to in-house servers in the event of a cloud failure. However, for many companies, the advantage of the cloud is that they can decommission their data centers so that there are no in-house servers on which to run their critical applications if necessary.

- In some cases, a subscription to the services of a disaster-recovery service or a Recovery-as-a-Service (RaaS) cloud provider[14] can provide servers-on-demand to recover from a cloud failure. However, recovery times using these services is measured in hours, not minutes. This approach may not be suitable for critical applications requiring minimal downtime.

- Data stored in the cloud must be backed up outside of the cloud in case the cloud loses part or all of a company's data. This is required if critical applications are to be backed up outside of the cloud so that the backup system has access to the application database if the cloud is down.. For non-critical applications, any data that cannot be reconstructed is subject to loss.

- Clouds are subject to being taken offline by Distributed Denial of Service (DDoS) attacks.[15] Though a large cloud has the bandwidth to withstand a large attack, the technology is now publicly available to launch attacks of such unprecedented volume that even large clouds may succumb.

- The bottom line is that no matter how you use a cloud to host your applications, you must have a tested Business Continuity Plan detailing how you will continue application services when they are no longer available via your IT assets. This often will entail manual operations of some sort.

Cloud computing is becoming an important resource for increasing numbers of companies. Unfortunately, cloud utilities do not yet provide the availability offered by electrical utilities and telephone services. Until that time, careful thought and planning must go into any decision to utilize cloud services for your applications, whether they are critical or not.

[14] HP's Cloud Recovery-as-a-Service (Raas), *Availability Digest*; June, 2012.
http://www.availabilitydigest.com/public_articles/0706/raas.pdf
[15] Anatomy of a DDoS Attack, *Availability Digest*; April 2013.
http://www.availabilitydigest.com/public_articles/0804/ddos_anatomy.pdf