Executive Summary

“The earlier you see and deal with a problem, the smaller it is. The later you see and fix a problem, the bigger it is.”

This fundamental precept applies to everything you do. With respect to cloud computing, you can use this precept to save time, save money and achieve greater uptime and availability. Managing your company’s IT resources to ensure both the lowest possible expense and the best possible customer experience is often a difficult balancing act. Renting IT services from a cloud vendor (such as Amazon) can be helpful, but you have to constantly monitor cloud services for unexpected outages, software errors, and performance issues.

Your disaster recovery plan - which you should always have at hand, ready to use - also needs to have a Plan B, an alternative that you can turn to when Amazon Web Services stop, slow down or misbehave. Quickly finding out that you have a problem with your cloud data storage, cloud Web hosting or cloud remote computing is the first step toward solving that problem.

Amazon cloud-based facilities are, unfortunately, out-of-sight and out-of-reach. However, you can easily run monitoring software that closely watches your cloud

Pitfall 1: AWS Outages

AWS has suffered many outages over the past few years.

A cloud is just someone else's data center that you rent time and space from. It's not magic, and it certainly isn't perfect. Amazon Web Services (AWS), like all other clouds (and all other data centers, for that matter), is subject to a wide range of problems – connectivity failures, software errors, and hardware malfunctions. AWS experiences all of these from time to time.

On June 5, 2016, a storm-related outage denied network access and computing services to Australian AWS customers for more than six hours. The unavailable AWS services included Elastic Compute Cloud (EC2), Relational Database Service (RDS), Database Migration Service, Elastic Beanstalk, and several others.

On September 20, 2015, a five-hour outage disrupted access to several popular online sites, such as Netflix, Tinder, Airbnb, Reddit and IMDb. The outage also halted a number of Amazon's own services, including Prime Instant Video. Amazon later admitted that a software error at its US-EAST-1 data center in northern Virginia had caused the outage. The software error triggered faults in 24 AWS services, ten of which crashed and died.

Note that Netflix was able to rapidly thwart the effects of the five-hour September 20, 2015 outage. Netflix used a monitoring tool to quickly detect the problem, and it implemented a disaster recovery plan that switched its customers to other data centers. Netflix had replicated itself across multiple data centers in anticipation of just such an outage.

A hardware malfunction on April 20, 2011 brought several AWS services to a standstill. A disk array controlled by Elastic Block Store (EBS) stopped responding to read/write commands. The outage lasted more than two days.

A storm of epic proportions (which local people would later refer to as “the inland hurricane”) disrupted access for many hours to the northern Virginia AWS data center on June 29, 2012.

Software issues caused major AWS outages on October 22, 2012 and on December 24, 2012. The October 22 outage, which affected sites such as Reddit, Foursquare, Pinterest, and others, was the result of a memory leak programming error. A different programming error, in the Elastic Load Balancing service, caused the December 24 outage.

Another long-duration northern Virginia AWS data center outage, on August 10, 2015, interrupted access to many popular Web sites. The affected AWS services

included Elastic Compute Cloud (EC2), Elastic Beanstalk and Simple Storage Service (S3). Amazon reported that “a configuration error in one of the systems that Amazon S3 uses to manage request traffic” caused the outage.

Other recent AWS outages of varying durations occurred on the following dates:

30 August 2016	22 August 2016	16 August 2016
29 July 2016	19 July 2016	20 June 2016
9 June 2016	2 June 2016	21 April 2016
30 March 2016	10 March 2016	2 February 2016
25 November 2015	19 November 2015	18 November 2015
15 October 2015	14 October 2015	

Does AWS host your company's Web site(s)? Store your company's data? Run software to update or analyze your company's information? Your disaster recovery plan needs to explain, in detail, the steps you take to switch temporarily to an alternative. The alternative might be slower than you'd like, or it might be function-limited, but having that alternative can save your company a great deal of money and goodwill.

First and foremost, however, you must run a monitoring tool to get early detection of AWS outages.

Pitfall 2: AWS Performance Problems

A good monitoring tool identifies bottlenecks, measures resource utilization and reveals what's happening inside the otherwise opaque AWS cloud.

Performance issues in the cloud are notoriously difficult to troubleshoot and solve. The fault might be related to a hardware malfunction, a programming error, a network change, a system configuration error, a lack of resources or, sometimes, just high traffic levels.

You may suspect that AWS itself has slowed down when in fact the problem is in the tangled web of network connections between your site and the AWS site. For example, a Tier 1 Internet backbone provider (such as AT&T, CenturyLink, Cogent Communications, Level 3 Communications or NTT Communications) might re-route traffic around a broken cable (backhoe equipment operators sometimes dig holes in the wrong places). A misconfigured router in any tier of the Internet can cause slowdowns. And a hacker can launch Distributed Denial of Service (DDoS) attacks that flood parts of the Internet with traffic or that specifically target Internet resources essential to your cloud connection (such as a DNS server).

Amazon rents a variety of computing resources to its AWS customers. Prices naturally increase as you choose faster CPUs, more memory, faster disk access and

greater bandwidth. But you always have to keep in mind that you're sharing the cloud computer(s) with other Amazon customers – the responsiveness you might see early Sunday morning, when the computers are not very busy, will be distinctly different from the sluggishness you might experience in the middle of the afternoon on a Thursday.

Specifically, Amazon uses hypervisors to create virtual machine (VM) instances in which your software runs. You rent one or more VM instances from Amazon. Note that hypervisors themselves can become quite busy creating, managing and destroying VM instances. On a physical machine running many VMs, hypervisor overhead can sometimes impede VM instances from using the CPU or from accessing necessary resources.

Software running inside AWS can suffer unforeseen slowdowns if CPU utilization, disk I/O, memory usage or network traffic/latency exceeds what you've allocated. All too often, you're faced with a combination of these situations.

CPU utilization problems can sometimes be solved by reprogramming (optimizing the code), by splitting up the workload and running multiple instances of the software (perhaps with the help of AWS' load balancer service) or by allocating more AWS "Elastic Compute Units" (ECUs). In simplest terms, Amazon says one ECU is equal

to the computing speed of a 1.0 GHz to 1.2 GHz Intel CPU. Use your monitoring tool to alert you when CPU utilization is high.

AWS Elastic Block Store (EBS) is a popular service for storing files on disk. EBS gives you large capacity disks on network-connected block storage devices. Be aware that, from a performance viewpoint, EBS slowdowns occur when the rate of I/O requests is greater than the storage devices can accommodate, when network traffic between the computers and the storage devices is high (other AWS customers may be competing with you for disk I/O on the same devices) or when the “chunks” of data to be stored or retrieved are greater than 16 kb in size. EBS is optimized to handle 16 kb data blocks.

An application crashes and dies when it completely exhausts its AWS-allocated memory. You can add swap volumes to your AWS instance to sometimes cure an out-of-memory condition. However, because the operating system uses the swap volumes to page memory blocks to and from disk, performance suffers. Disk I/O is orders-of-magnitude slower than memory access.

If you have a transaction-oriented application that runs out of memory in high-traffic situations, you might consider having AWS run additional instances of the application, via Elastic Load Balancing (ELB), to handle the occasional high workload. Such scaling is quite common on AWS.

Use your monitoring tool to track memory usage and take steps to avoid application memory exhaustion. Note that if ELB scales your application based on traffic levels, you should also track latency for ELB instances. Load balancing operations can cause considerable network activity (and thus latency), sometimes in a cascading fashion.

You can also track such statistics as:

- The number of requests that could not be properly load-balanced (caused sometimes by a lack of healthy servers to which ELB can route extra traffic)

- Web requests per minute

- The number of healthy Web servers in the load balancer pool

Pitfall 3: AWS Database Issues and Disparities

AWS RDS database services (such as DynamoDB, ElastiCache, Elastic MapReduce, Redshift and the cloud versions of the relational databases MySQL, Oracle, SQL Server, and PostgreSQL) are subject to issues and troubles beyond the typical ones that you see when you install and use one of these database products in your own data center.

Each RDS instance is a version of the database product running as an AWS EC2 virtual machine instance on an EC2 platform. It uses AWS EBS volumes for data storage. You have no access to the underlying EC2 instance, and you do not get access to S3 in order to "see" or otherwise process your stored database snapshots. On the other hand, AWS creates on-demand database snapshots for you (extra storage charges apply), and AWS Automatic Backup promises point-in-time data recovery that's no more than five minutes old.

You also cannot install software running alongside the database product. This restriction precludes you from using an agent-based monitoring tool to keep watch over your database operations. Make sure the monitoring tool you buy doesn't need to install an agent on the database server.

AWS does not allow you direct access to database configuration files, but rather exposes an API that you can use to configure the database.

Some of the facilities of the database product, such as replication, are not available in the AWS cloud. Furthermore, you have no access to transaction logs or the MySQL binary log.

AWS does not allow you to act as Supervisor or Administrator of your database. This means, for example, that you cannot manually shutdown the database (as you would be able to if the database server were located in your own data center).

Network latency can dramatically affect your database updates, retrievals, and queries. For any database operations you perform over the Internet from your site, the cloud-based AWS database is by definition going to be less responsive than a database server that sits in your own data center.

Pitfall 4: AWS Fault Lines

AWS has a number of other idiosyncrasies and deficiencies you'll want to be aware of.

DNS Security – A Domain Name Service (DNS) cyber-attack can replace the correct IP address for a URL with a substitute IP address that redirects people to a fake server. The spoofed Internet address is often a Web server whose Web pages look like those of a legitimate bank or a popular online merchant. The IETF's Domain Name System Security Extensions (DNSSEC) protocol thwarts such attacks by using digital certificates and private keys to establish DNS name server trusted sources. Note that AWS supports DNSSEC for domain registration but not for regular, ongoing DNS name resolution. If you want to use DNSSEC for a domain registered with AWS Route 53, you must use a different DNS service provider.

Latency-based Routing – AWS Route 53 doesn't support ecdn-client-subnet DNS extensions, a protocol that's especially useful for geographically-dispersed content delivery. These extensions forward the higher part of the client IP address to the authoritative DNS server for the specific purpose of latency-based routing.

IP Address Assignment – Avoid assigning (or even needing to know) IP addresses. You can then use AWS load balancing to automatically scale up (or down) as appropriate, and you can disperse your application across multiple AWS availability zones.

AWS Service Limits – Amazon enforces a variety of limits on customer activities. Be aware of these by reviewing AWS service limits before you begin using AWS. If

AWS notifies you that you've exceeded one of these limits, you can email a request to AWS technical support asking that the limit be increased, but you don't want your AWS processes to sit in an idle state while you wait for AWS support to act.

Monitor your AWS usage and make your service limit increase requests before you reach a limit. Note that some of the limits are AWS-global, while others are region-specific.

Pitfall 5: "Amazon Will Do It All"

Like all new technologies, AWS is seen as a panacea, as a cure all. While AWS does provide many benefits and introduces useful new features and facilities, it is still just technology, with all the limitations that implies.

Chief among these limitations is the wishful thinking that Amazon addresses all the AWS issues. While it is true that Amazon has done a superb job with AWS, in the final analysis, it is the customer who must implement a complete and comprehensive monitoring and automation facility that is completely independent of AWS itself.

Argent for AWS is a uniquely powerful solution, engineered as a completely new product, from the ground up, not simply a tarted-up lash up.

Argent for AWS has many unique features, some of which include:

- Ability To Monitor All Aspects Exposed By AWS Console And AWS SDK
- Integration With Argent Console, The World's Leading Alerting Console
- Integration With Argent Predictor For Long-Term Trend Analysis (Not Just 14 Days)
- Capable Of Native Monitoring of Windows And Linux Applications
- Comprehensive S3 Monitoring
- Complete Log Monitoring On S3, EBS Volume Or EC2 Instance Store Volume

For a free 30-day evaluation of Argent for AWS, please email Sales@Argent.com

About the Author

Barry Nance is a networking expert, magazine columnist, book author and application architect. He has more than 29 years of experience with IT technologies, methodologies and products.

Over the past dozen years, working on behalf of Network Testing Labs, he has evaluated thousands of hardware and software products for ComputerWorld, BYTE Magazine, Government Computer News, PC Magazine, Network Computing, Network World and many other publications. He's authored thousands of magazine articles as well as popular books such as Introduction to Networking (4th Edition), Network Programming in C and Client/Server LAN Programming.

He's also designed successful e-commerce Web-based applications, created database and network benchmark tools, written a variety of network diagnostic software utilities and developed a number of special-purpose networking protocols.

About Network Testing Labs

Network Testing Labs performs independent technology research and product evaluations. Its network laboratory connects myriads of types of computers and virtually every kind of network device in an ever-changing variety of ways. Its authors are networking experts who write clearly and plainly about complex technologies and products.

Network Testing Labs' experts have written hardware and software product reviews, state-of-the-art analyses, feature articles, in-depth technology workshops, cover stories, buyer’s guides and in-depth technology outlooks. Its experts have spoken on a number of topics at Comdex, PC Expo, and other venues. In addition, they've created industry-standard network benchmark software, database benchmark software, and network diagnostic utilities.