Review and summary of cloud service level agreements

From "Cloud Computing Use Cases Whitepaper" Version 4.0

Skill Level: Intermediate

developerWorks cloud computing editors (dwcloud@us.ibm.com)
IBM

04 Aug 2010

This is a review of the service level agreements section of the "Cloud Computing Use Cases Whitepaper" Version 4.0 — posted by the Cloud Computing Use Cases Discussion Group — to highlight the SLA issues that architects and developers should consider as they move to the cloud.

This article takes a look at the "Cloud Computing Use Cases Whitepaper," Version 4.0 from the Cloud Computing Use Case group — an information storehouse created by an open web community of more than 1,400 participants (up from 900 in Version 3.0). It started with a group of supporters from the Open Cloud Manifesto and grew to include representatives from large and small companies, government agencies, consultants, and vendors.

The scope of the "Cloud Computing Use Cases Whitepaper" is comprehensive, so we won't try to cover the entire document in one reading. For this review, we'll focus on the group's evaluation of service level agreement issues in the cloud, an important consideration because SLAs describe the relationship between the cloud provider and the cloud consumer, essentially defining the basis of trust cloud consumers have in a cloud provider's ability to deliver infrastructure services.

Who makes this possible?
The contributors to the "Cloud Computing Use Cases Whitepaper," Version 4.0 are Dustin Amrhein, Patrick Anderson, Andrew de Andrade, Joe Armstrong, Ezhil Arasan B, James Bartlett, Richard Bruklis, Ken Cameron, Reuven Cohen, Tim M. Crawford, Vikas Deolaliker, Andrew Easton, Rodrigo Flores, Gaston Fourcade,
What is an SLA?

The authors agree that the SLA should contain:

- The list of services the provider will deliver and a complete definition of each service.
- Metrics to determine whether the provider is delivering the service as promised and an auditing mechanism to monitor the service.
- Responsibilities of the provider and the consumer and remedies available to both if the terms of the SLA are not met.
- A description of how the SLA will change over time.

The authors discuss the two types of SLAs — off-the-shelf agreements and customized, negotiated agreements. They note that customers with critical data needs will not be satisfied with off-the-shelf agreements, so a first step before going to the cloud is to determine how critical your data and applications are.

Public clouds often offer a non-negotiable SLA which may not be acceptable for those with mission-critical apps or data.

What is an SLO?

An SLA contains service level objectives (SLOs) that define objectively measurable conditions for the service; some examples include parameters of throughput and data streaming frequency and timing, availability percentages for VMs and other resources and instances, or urgency ratings to rank the importance of different SLOs (like "availability is more important than response time").

SLO expectations should vary depending on whether applications and data the applications access are hosted on the same cloud or on different ones.
Monitoring and measuring

Service level management, based on SLOs, is how performance information on the cloud is gathered and handled. This is how it is employed:

- The **cloud provider** uses service level management to make decisions about its infrastructure; for example, if throughput isn't always meeting a customer's requirements, the provider can reallocate bandwidth or add more hardware. Or decide to make one customer happy at the expense of another one. For providers, SLM is designed to help make the best decisions based on business objectives and technical realities.

- The **cloud consumer** uses SLM to decide how he wants to use cloud services; like whether or not to add in more virtual machines and at what price point that option becomes too expensive to justify the return. For consumers, SLM helps them make decisions on the way they use the cloud. And sometimes on how to automate those decisions.

What factors should you consider on SLA terms?

The authors came up with a list of 10 factors to consider when defining the terms of an SLA:

1. **Business level objectives:** An organization must define *why* it will use the cloud services before it can define exactly what services it will use. This part is more organizational politics than technical issues: Some groups may get funding cuts or lose control of their infrastructure.

2. **Responsibilities of both parties:** It is important to define the balance of responsibilities between the provider and consumer. For example, the provider will be responsible for the Software-as-a-Service aspects, but the consumer may be mostly responsible for his VM that contains licensed software and works with sensitive data.

3. **Business continuity/disaster recovery:** The consumer should ensure the provider maintains adequate disaster protection. Two examples come to mind: Storing valuable data on the cloud as backup and *cloud bursting* (switchover when in-house data centers are unable to handle processing loads).

4. **Redundancy:** Consider how redundant your provider's systems are.

5. **Maintenance:** One of the nicest aspects of using a cloud is that the
provider handles the maintenance. But consumers should know, when providers will do maintenance tasks:

- Will services be unavailable during that time?
- Will services be available, but with much lower throughput?
- Will the consumer have a chance to test their applications against the updated service?

6. **Data location:** There are regulations that certain types of data can only be stored in certain physical locations. Providers can respond to those requirements with a guarantee that a consumer's data will be stored in certain locations only and the ability to audit that situation.

7. **Data seizure:** If law enforcement seizes a provider's equipment to capture the data and applications belonging to a particular consumer, that seizure is likely to affect other consumers that use the same provider. Consider a third party to provide additional backup.

8. **Provider failure:** Make contingency plans that take into account the financial health of the provider.

9. **Jurisdiction:** Again, understand the local laws that apply to your provider as well as you do the laws that apply to you.

10. **Brokers and resellers:** If your provider is a broker or reseller of cloud services, you need to understand the policies of your provider and the actual provider.

**SLA requirements**

The authors came up with a list of 14 responsibilities to consider when considering an SLA:

1. **Security:** A consumer must understand his security requirements and what controls and federation patterns are necessary to meet those requirements. A provider must understand what they must deliver to the consumer to enable the appropriate controls and federation patterns.

2. **Data encryption:** Data must be encrypted while it is in motion and while it is at rest. The details of the encryption algorithms and access control policies should be specified.

3. **Privacy:** Basic privacy concerns are addressed by requirements such as data encryption, retention, and deletion. An SLA should make it clear how
the cloud provider isolates data and applications in a multi-tenant environment.

4. **Data retention, deletion:** How does your provider prove they comply with retention laws and deletion policies?

5. **Hardware erasure, destruction:** Same as #4.

6. **Regulatory compliance:** If regulations must be enforced because of the type of data, the cloud provider must be able to prove his compliance.

7. **Transparency:** For critical data and applications, providers must be proactive in notifying consumers when the terms of the SLA are breached. This includes infrastructure issues like outages and performance problems, as well as security incidents.

8. **Certification:** The provider should be responsible for proving required certification and keeping it current.

9. **Performance definitions:** What does uptime mean? All the servers on every continent are available? Or just one is available? It pays to define those definitions. (The authors of this paper suggest standardizing performance terminology to make it easier.)

10. **Monitoring:** For issues of potential breaches, you might want to specify a neutral third-party organization to monitor the performance of the provider.

11. **Auditability:** Because the consumer is liable for any breaches that occur with loss of data or availability, it is vital that the consumer be able to audit the provider’s systems and procedures. The SLA should make it clear how and when those audits take place. They can be disruptive and costly to the provider.

12. **Metrics:** These are the tangible somethings that can be monitored as they happen and audited after the fact. The metrics of an SLA must be objectively and unambiguously defined. Following this list is a list of common metrics.

13. **Providing a machine-readable SLA:** This can allow for an automated, dynamic selection of a cloud broker. In other words, if your SLA requires that the broker use the cheapest possible provider for some tasks but the most secure provider for others, this type of automation makes it possible. (This type of service is not readily available yet, but is something to keep in mind when contributing to the cloud SLA standardization discussion.)

14. **Human interaction:** On-demand self-service is one of the basic
characteristics of cloud computing, but your SLA should take into account that when you need a human being, one is made available to you.

Some of the common performance metrics (for consideration #12) include

- **Throughput**: System response speed.
- **Reliability**: System availability.
- **Load balancing**: When elasticity kicks in.
- **Durability**: How likely to lose data.
- **Elasticity**: How much a resource can grow.
- **Linearity**: System performance as the load increases.
- **Agility**: How quickly the provider responds to load changes.
- **Automation**: Percent of requests handled without human interaction.
- **Customer service response times**.

**A few rules of thumb on reliability**

The authors provide a concise treatise on a working definition of reliability when it comes to cloud performance. It goes something like this:

- **The rule of nines**. A common metric concerning reliability is the number of nines a provider delivers (like, if the service is available 99.99999 percent of the time, five nines, then the total systems outages are like 5 minutes every 12 months). Problem is, what’s an outage? (It can be a really bad situation if the provider gets to decide what an outage is.)

- **Layers of clouds**. Many cloud offerings are built atop other cloud offerings — this is great for flexibility and power but each additional provider makes the system less reliable. (Like if each rates themselves to five nines, then the system overall is less than five nines.)

- **Distance between your app and its data**. Again, as the number of providers increase, other factors that affect reliability take hold. Not only are you affected whenever one of the systems goes down, you’re also affected if the network between them goes down.

None of this is to scare the cloud consumer; these are just structural facts to consider when choosing a provider.
Requirements and delivery models, use cases

In the original paper, the authors supply two tables:

- **Table 8.7: SLA Requirements and Cloud Delivery Models.** This table cross-references the SLA requirement we discussed (data encryption, privacy, certification, etc.) with delivery models PaaS, IaaS, and SaaS (that are discussed in the original paper).

- **Table 8.8: SLA Requirements and Use Case Scenarios.** This table cross-references the SLA requirements with the seven use case scenarios:
  1. End user to cloud.
  2. Enterprise to cloud to end user.
  3. Enterprise to cloud.
  4. Enterprise to cloud to enterprise.
  5. Private (on-premise) cloud.
  7. Hybrid cloud.

In conclusion

The conclusions the "Cloud Computing Use Cases Whitepaper," Version 4.0 reaches about service level agreements for the cloud are clear:

- Cloud computing is not feasible without service management, governance, metering, monitoring, federated identity, SLAs and benchmarks, data and application federation, deployment, and lifecycle management.
- Meaningful transparency and disclosure from cloud providers is a necessity.
- If there's an existing standard to fulfill a requirement, cloud users must insist providers use it; if there's not, insist the community develop one.

The authors state:
As organizations use cloud services, the responsibilities of both the consumer and the provider must be clearly defined in a Service Level Agreement. An SLA defines how the consumer will use the services and how the provider will deliver them. It is crucial that the consumer of cloud services fully understand all the terms of the provider’s SLA, and that the consumer consider the needs of their organization before signing any agreement.

This summary and review offers a baseline to illustrate cloud service level agreement concerns and considerations for both cloud service consumers and providers. We encourage you to study the original "Cloud Computing Use Cases Whitepaper" Version 4.0 in its entirety for Cloud Computing Use Case Discussion group's analysis of what developers and planners should require from their cloud providers to deliver a reliable environment for precious data and applications.
Resources

Learn

• The original document is from the experts that belong to the Cloud Computing Use Cases group. These versions of the paper are available in PDF:
  • English, Version 4.0, the basis for this review
  • Traditional Chinese, Version 4.0
  • Simplified Chinese, Version 4.0
  • The earlier review, "Review and summary of cloud security scenarios," is based on Chapters 6 and 7 of the paper.

Other formats may be available at this site.

• The Open Cloud Manifesto is a statement of the principles for maintaining openness in cloud computing.

• In the developerWorks cloud developer resources, discover and share knowledge and experience of application and services developers building their projects for cloud deployment.

• In the developerWorks open source resources, discover and share knowledge and experience of open source application and services developers.

• Stay current with developerWorks technical events and Webcasts.

Get products and technologies

• With IBM trial software, available for download directly from developerWorks, build your next development project.

Discuss

• Join a cloud computing group on My developerWorks.

• Read all the great cloud blogs on My developerWorks.

• Join the My developerWorks community, a professional network and unified set of community tools for connecting, sharing, and collaborating.

About the author

developerWorks cloud computing editors

The developerWorks cloud computing editors would like to hear from you about the types of technical resources you want to make your application’s trip into the clouds an easier, more effective journey. Ping
us at dwcloud@us.ibm.com.