**Cloud and Big Data Business Case**

**By - Workneh Yilma Ayele**

**Acronym**

**NIST - (National Institute of Standards and Technology) USA**

**IaaS – Infrastructure as a Service**

**PaaS – Platform as a Service**

**SaaS – Software as a Service**

**SLA – Service Level Agreements**

**CAPEX – Capital Expenses**

**OPEX – Operational Expenses**

**Table of Contents**

[Cloud computing 2](#_Toc500802050)

[Definition by Rosenberg and Mateos 2](#_Toc500802051)

[Definition by NIST (Mell and Grance) 2](#_Toc500802052)

[Reasons for “Going to the Cloud” 8](#_Toc500802053)

[Scalable Software Architectures 9](#_Toc500802054)

[Filesystems 11](#_Toc500802055)

[Databases 12](#_Toc500802056)

[Security in the Cloud 13](#_Toc500802057)

[Security benefits of cloud computing. 13](#_Toc500802058)

[Security liabilities of cloud computing 13](#_Toc500802059)

Introduction

This document has six sections, in the first section the definition of cloud computing and the basic principles of cloud computing according to Rosenberg and Mateos are presented. The second section follows by presenting reasons for going to the cloud, and then scalable software architectures, file system, databases, and security in the cloud are presented.

1. Cloud computing **-** What is cloud computing?

## Definition by Rosenberg and Mateos

Rosenberg and Mateos argue that the five principles of cloud computing presented in their book define cloud computing. These principles are pooled resources, virtualization, elasticity, automation, and metered billing. Therefore, they argue that there is a consensus that is gaining a growing acceptance among cloud vendors, analysts, and users to consider cloud computing as a computing service provided by the third party and is available when needed with dynamic scaling capability in response to fluctuating demands. Also, Rosenberg and Mateos presented classifying layers of cloud computing such as IaaS, PaaS, SaaS, and private clouds as precursors of public clouds (Rosenberg and Mateos 2011).

## Definition by NIST (Mell and Grance)

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.” (Mell and Grance, 2011)

There are many definitions of cloud computing in literature. However, the definition of cloud computing by NIST serves as a reference among the numerous definitions for Cloud Computing as well as a starting point for the discourse of **what cloud is** and **how to best use** it. Additionally, the NIST definition is also meant to serve as a reference for comprehensive comparisons of deployment strategies and hosted services (Kulkarni and Agrawal 2014).

I present the comparison of the two definitions of cloud computing, see table below. The comparison indicated that the definition by Rosenberg and Matos covers most of the key concepts of cloud computing but seems to overlook cloud deployment models. The definition by NIST covers different viewpoints of cloud computing. See below Table 1, for the comparison of the key concepts of cloud computing.

|  |  |
| --- | --- |
| ***NIST*** | ***Rosenberg and Mateos*** |
| Essential characteristics of cloud computing   * On-demand self service * Broad network access * Resource pooling * Rapid elasticity and scaling * Measured service | The five principles of cloud computing   * Pooled Computing Resources * Virtualized Computing Resources * Elastic Scaling up or down * Automatic creation and deletion of new VM’s * Resource usage billed only as used |
| Cloud service models   * SaaS * PaaS * IaaS | classifying layers of cloud computing   * SaaS * IaaS * PaaS, and private clouds as predecessors  of public clouds |
| Cloud deployment models   * Public clouds * Private clouds * Community clouds * Hybrid clouds | *private clouds as precursors of public clouds* |

## Rosenberg and Mateos cloud computing

Rosenberg discuss five principles of cloud computing, i.e.:

* Pooled Computing Resources
* Virtualised Computing Resources
* Elastic Scaling up or down
* Automatic creation and deletion of new VM’s
* Resource usage billed only as used

In this section for readability answers are presented following questions.

1. Briefly discuss each of these principles. Are they necessary? What do they mean for a cloud service? What do they mean for a service using the cloud?

Each of these five principles is necessary. For a cloud service pooled, computing resources are resources that are accessible by cloud services when needed. Virtualized computing resources for a cloud service means that services are running on virtual servers rather than physical computing infrastructures directly. For a cloud service, elastic scaling means resource capacity is scaled dynamically based on the need for the service. Automatic creation and deletion of new VM’s is the ability to provide services through virtual instances. Finally, resource usage billed only as used refers to the billing based usage of services.

These five principles from the perspectives of services using cloud applications or services are presented in this paragraph. Pooled computing resources are resources that are accessible to anyone with credit cards, but these resources are not dedicated for a certain user. Virtualization of computing resources enables services using the cloud to deploy any services such as any flavor of an operating system and load balancer for example. Services using the cloud also have access to the capability of the cloud to scale based on demand. Auto creation and deletion of new VM’s enables services utilizing the cloud to create or destroy virtual environments as needed dynamically. Finally, metered billing enables services utilizing the cloud to monitor and manage incurred costs.

1. Is this a complete set? Are there other [fundamental] principles that you expect for cloud computing?

It is not a complete set compared to the definition of cloud computing by NIST. Cloud deployment models are not explicitly described by the authors.

1. Some of these principles are based in technology, others are geared towards the cloud ownership business, and some are actually relevant for a cloud user. Moreover, some of them *enables* the other principles. Please discuss these relationships, and what this means for a cloud user.

Pooled resources and scalability or elasticity are related since elasticity is only possible on existing pooled resources. Virtualization and auto-creation and deletion of virtual machines are also related concepts. Virtualization is the creation of virtual environments or infrastructures for virtual machines to run on. Metered billing is also related to elasticity and pooled resources, if there are no pooled resources, then there will be no elasticity or scaling dynamically. For a cloud user metered billing dictates how she/he is billed based on what virtual resources are used.

1. Elasic Scaling is one of the key principles. Please reason about the business implications of this.

Elastic scaling, dynamic scaling, enables applications running in a cloud to expand or contract smoothly. This means, elasticity enables the release of resources when needed and the to have capacity when needed (Rosenberg and Mateos 2011).

* + Give examples of applications where a quick (within minutes) elastic scaling might be necessary.

Banks, social media, e-commerce sites applications and the like needs quick elastic scaling in emergency situations for example.

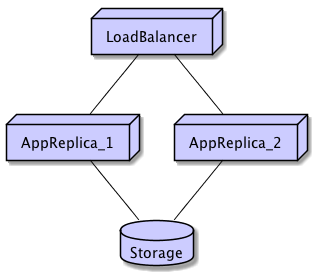
* + Give examples of applications where a slower scaling would be more relevant.

Corporate applications with cloud-based ERP systems whose applications are used by a predictable number of users. Cloud-based government information systems that are only used by the government is also likely to scale slowly.

* + If your application grows at a steady state, what does this mean for your scaling needs? Is “going to the cloud” still the best option?

It means my scaling need is more or less predictable, and on-premises IT infrastructure might suffice for such needs. Besides, scaling may also lead us to deploy to colocation before the move to the cloud.

1. Metered Billing (“Resource usage billed only as used”) is another one of the key principles.
   * Let’s say you have an application running on three machines plus a load balancer as illustrated in Figure [1](https://mickesv.github.io/cloudbd/BusinessCase/BusinessCase-Assignment.html#fig:SimpleApp). What would be the cost per hour, day, and month for this setup if you ran it yourself (Remember that network access, routers, and cooling etc. also cost money)?



|  |  |  |  |
| --- | --- | --- | --- |
| Item | Specification | Product Link | Price |
| AppReplica-Storage | 24 HDs 2.5", rackmount, 2Us, 12Gb HD-Mini to HD-Mini SAS cable, 0.5m, 1.6TB Solid State Drive SAS Mix Use 12Gbps 512e 2.5in Hot-plug Drive, PM1635a, 2.4TB 10K RPM Self-Encrypting SAS 12Gbps 2.5in Hot-plug Hard Drive, Hard Drive 300GB 10K RPM SAS 12Gbps 2.5in Hot-plug Hard Drive | Dell Storage MD1420[[1]](#footnote-1) | $13,431 |
| AppReplica\_1 | * Intel® Xeon® Bronze 3104 1.7G, 6C/6T, 9.6GT/s 2UPI, 8M Cache, No Turbo, No HT (85W) DDR4-2133 * No Operating System * 8GB RDIMM, 2666MT/s, Single Rank * 120GB SSD SATA Boot 6Gbps 512n 2.5in Hot-plug Drive, 1 DWPD, 219 TBW | New PowerEdge FC640 Server[[2]](#footnote-2) | $5,138 |
| AppReplica\_2 | * Intel® Xeon® Bronze 3104 1.7G, 6C/6T, 9.6GT/s 2UPI, 8M Cache, No Turbo, No HT (85W) DDR4-2133 * No Operating System * 8GB RDIMM, 2666MT/s, Single Rank * 120GB SSD SATA Boot 6Gbps 512n 2.5in Hot-plug Drive, 1 DWPD, 219 TBW | New PowerEdge FC640 Server | $5,138 |
| LoadBalancer | KEMP LoadMaster 4000 Load Balancer - load balancing device | KEMP[[3]](#footnote-3) | $13,645 |
| Router-switch | Cisco ASA 5500 Edition Bundle ASA5512-K9 ASA 5512-X with SW, 6GE Data, 1GE Mgmt, AC, 3DES/AES | CISCO[[4]](#footnote-4) | $2,000 |
| Rack | Normal rack | 27u 4 Post[[5]](#footnote-5) | $157 |
| ISP | 1 Gbps - $100 per month | Telia | $3,600 |
| Cooling | Designer Inspired Ultra Quiet Fan Panel - Black | UQFP-4DRA[[6]](#footnote-6) | $486.75 |
| **Total cost** | | | **$43595.75** |

According to Rosenberg and Mateos depreciation period is 36 months or three years (Rosenberg and Mateos 2011). Therefore the price: per year 43595.75/3= **$14,532/year**, per month is 43595.75/36 = **1,210.99 $ / month**, 1,210.99$/30 = **$40.37/day**, 40.37/24=**$1.68/hour.** Also, considering the future expansion the hardware specification here reflects the capapcity of the servers to be open for upgrade.

* + Explore the different licensing models on Amazon EC2. What would the same setup cost to run as a T2.large or M3.large setup?

According to EC2Instances.infor

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name of instance | API Name | Memory | vCPUs | Instance Storage | Network Performance | Linux On Demand (presented for comparison) | Linux reserved cost in $ per hour | Cost in $ Per Day = 4XServer price\*24 | Cost in $ Per Month = 4XServer price\*24\*30 |
| T2 Large (four servers) | t2.large | 8.0 GiB | 2 vCPUs for a 7h 12m burst | EBS only | Low to Moderate | 0.0928 | 0.058 | 5.568 | 167,04 |
| M3 General Purpose Large (four servers) | m3.large | 7.5 GiB | 2 vCPUs | 32 GiB SSD | Moderate | 0.133 | 0.095 | 9.12 | 273.6 |

The costs are calculated for reserved instance because reserved instances are cheaper as illustrated in the table above in greyed sells. The prices compared to the prices of traditional IT is not even comparable.

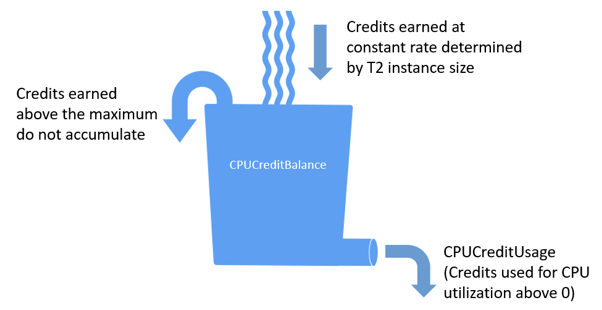
(0.1 /CPU-hour – load balancer, 0.4 /CPU-hour two application servers, 0.8 /CPU-hour two DB Server Reese, G. (2009)

* + What are the differences between Amazon’s “on-demand” and the “Reserved Instance”?

According to Amazon, on-demand instances do not require upfront payment, or in other words, the customer is not subject to have a long-term subscription. Customers pay per hour or per second depending on the service they are using. On-demand instances are suitable for short-term or unpredictable workloads and for applications testing on Amazon EC2. On the other hand, reserved instances have higher discounts than on-demand instances and can be assigned to specific availability zone which makes them desirable to have an ability to launch instances whenever needed. Reserved instances are suitable for customers with over 1 or 3 years term for reducing total computing costs. Also, reserved instances are suitable for steady-state usage applications and for applications that require reserved capacity.

* + Since there is loadbalancer, we may guess that this application probably uses the CPU extensively. What is Amazon’s policies regarding CPU utilisation? Please spend some time trying to find an answer to this before “cheating” by reading this footnote: [1](https://mickesv.github.io/cloudbd/BusinessCase/BusinessCase-Assignment.html" \l "fn.1)

The Amazon uses the model you pay per CPU-hour (Reese, G. 2009). Amazons policies regarding CPU utilization, dictates that traditional Amazon EC2 instances offer fixed performance. However, T2 instances offer a baseline level performance with the ability to burst based on CPU credits. A CPU credit is equal to 100% utilization of vCPU for one minute. One CPU credit is also equal to two vCPUs running at 50% capacity. Earning of CPU credits is based on utilization based on the rate of CPU credits per hour. Earned CPU credits will expire if your instance stops and it will not expire if the instance is still running. The maximum number of CPU credit per hour is determined by the size of the instance and the credit earned per hour can be expressed as a percentage of CPU utilization. The diagram below is taken from Amazon credit, and baseline performance page illustrates the process of CPU credit balance accumulation.



* + What are the consequences of Amazon’s CPU policy on a running application?

Running applications will lose CPU credits if they expire, there is also a limit to maximum CPU credits earned based on baseline performance of CPU utilization. As CPU utilization bursts CPU credits are credited.

* + What can you do about it?

Architects should design their cloud system in such a way that CPU burst is minimized or is less likely by for example utilizing load balancer. Even if burst occurs earned credits will compensate if it is within a reasonable limit.

1. Different cloud providers offer different services (x As A Service).

* Briefly describe each of Software As A Service, Framework As A Service, Platform As A Service, and Infrastructure As A Service.
* Give examples of applications that use each of these.

The description below is based on NIST (Mell and Grance 2011) and (Rosenberg and Mateos 2011).

**Software as a Service (SaaS)** – provides a packaged software application, provides services or applications on demand.

* Example – Salesforce.com – is a customer relationship management system

NIST- the provision of an application on the cloud infrastructure and consumers or customers do not have the control to the underlying infrastructure such as operating systems, network, storage, servers, and the capabilities of the applications but they have limited configuration possibilities related to user-specific applications settings (Mell and Grance 2011).

**Framework as a Service (FaaS)** – is an environment to building a module for an ERP system. It is also useful for enhancing and augmenting the capabilities of the base SaaS system (Rosenberg and Mateos 2011).

* Example – Force.com – which is used to structure or develop salesforce apps

**Platform as a Service (PaaS)** – this provides an environment for building a managed application with an IDE. PaaS requires less interaction with the infrastructure so that users can focus more on application development (Rosenberg and Mateos 2011).

* Examples – Google AppEngine (Google Compute Engine), Microsoft Azure

NIST- the customer does not manage or control the underlying infrastructure but has access and control to one's application. Therefore, customers create applications and have access to control the deployed applications (Mell and Grance 2011).

**Infrastructure as a Service (IaaS)** – environment for building a native application. The user of IaaS uses the system at the lowest level of granularity; IaaS provider offers virtual machines with operating systems. Users can tailor these virtual machine images to run applications. IaaS provides greater autonomy to cloud resources (Rosenberg and Mateos 2011).

* Example - Amazon Computing Engine (EC2)

NIST – In this model, customers are provisioned processing, storage, network and other computing resources. Customers can run and deploy operating systems and applications, and they have limited control of selecting networking components, e.g. host firewalls. (Mell and Grance 2011).

1. Reasons for “Going to the Cloud”

Adopting a cloud solution is often touted for economic reasons – the metered billing aspect is particularly attractive for many companies[[2](https://mickesv.github.io/cloudbd/BusinessCase/BusinessCase-Assignment.html" \l "fn.2).

From a technical perspective, the reasons are more geared towards quality attributes. ISO/IEC 25010:2011 (the successor of ISO 9126) lists the following quality attributes:

* Functional Suitability
  + Functional Completeness
  + Functional Correctness
  + Functional Appropriateness
* Performance Efficiency
  + Time-Behaviour
  + Resource Utilisation
  + Capacity
* Compatibility
  + Co-existence
  + Interoperability
* Usability
  + Appropriateness
  + Recognisability
  + Learnability
  + Operability
  + User error protection
  + User interface aesthetics
  + Accessibilty
* Reliability
  + Maturity
  + Availability
  + Fault tolerance
  + Recoverability
* Security
  + Confidentiality
  + Integrity
  + Non-repudiation
  + Accountability
  + Authenticity
* Maintainability
  + Modularity
  + Reusability
  + Analysability
  + Modifiability
  + Testability
* Portability
  + Adaptability
  + Installability
  + Replaceability

1. Which of these may be a reason for moving towards a cloud solution (Pick the top five)? Please motivate your answers.

According to Reese SLA is an important document indicating key metrics such as availability, relativity, and performance that are expected by customers under reasonable circumstances. Availability – how often a server is used, reliability – how well one can trust the system, and performance –is about designing for best performance such as clustering vs. a single node system, segmenting databases, and forking the capabilities of your underlying platforms (Reese 2009). Therefore, I chose reliability as the first, performance efficiency as the second since the cost of running is a key aspect of choosing the cloud and security as the third. Finally, I would choose maintainability to be the fourth and usability to be the fifth reasons for moving towards a cloud solution.

1. Elastic Scale (and Scale in general) is one of the previously discussed key principles, but scale with respect to what?

Elasticity is to dynamically scale resources as needed; it can mean scaling with respect to the need for capital expenses (CAPEX).

1. In what way does a cloud solution address usability?

According to Rosenberg and Mateos usability deals with how real users interact with the application's interface and functionality (Rosenberg and Mateos 2011). A cloud solution can address usability by providing operability, validation of user inputs and learnable interface with ease of accessibility and relevant service.

1. In what way can a cloud solution address security and all its sub-attributes?

* Integrity – can be achieved by using checksums to make sure that files are not changed in transit. Confidentiality – can be achieved by using encryption techniques as for example health related records.
* Non-repudiation – can be achieved by integrity mechanisms
* Accountability – can be achieved through auditability by the use of certification such as SAS 70 to ensure auditing capability of providers.
* Authenticity – can be achieved by providing access control mechanisms to make sure only authorized users can access resources as needed.

1. Scalable Software Architectures

A suggestion of software architecture based on the requirements described in grey boxes below is presented in this section:

**System Type: Computing Jobs** This system works on the principle that a large chunk of data is made available and a number of computing-intensive tasks should be performed on the data. These computations can be dependent on each other (First, do *A*, then do *B*), or independent. Moreover, the computations may require access to the entire dataset, or they may be able to work on smaller portions of the data.

* important quality attributes – reliability, availability, fault-tolerance, usability such as error protection, performance efficiency, and functional suitability or efficiency are required
* The system can contain:
  + **load balancers** to handle incoming request from users
  + **App server** to process the data
  + Execution orchestrator is a **scheduler** that orchestrates execution of tasks based on dependency of computations
  + **Cash server** to provide access to the whole or part of the dataset
  + **Replicated database** NoSQL servers for reduced access loads when entire dataset or portion of dataset is accessed,
* Loose coupling can ensure that both dependent and independent computations can be carried out as B depends on A since A must be done before B, and also the computations may also be independent.

**System Type: Computing Stream** This system is similar to the “Computing Jobs” system, except that data arrives continuously instead of as large chunks. Obviously, this means that computations work on individual portions of the data. It may also mean that the computations are run in sequence A -> B -> C for each chunk of data. *However*, it may also mean that you have a large number of streams to process in parallel.

* important quality attributes – reliability, availability, fault-tolerance, usability such as error protection, performance efficiency, and functional suitability or efficiency are required, and functional suitability
* The system can contain:
  + **load balancers** to handle incoming request from users
  + **App server** to process the data
  + Execution orchestrator is a **scheduler** that orchestrates execution of tasks based on dependency of computations
  + **Streaming server** to maintain streaming data on the replicated database read write operations
  + **Replicated database** NoSQL database,

**System Type: User Silos** This system is a system where end-users interact with the system to perform a task. This may be a session-based task that consists of more than several steps, and the system must appear responsive for each individual user. However, there is little interaction between different users; each user experience the system as if they were currently alone on it.

* important quality attributes – reliability such as availability, functional suitability, usability such as user interface and accessibility
* The system can contain users silos with:
  + **load balancers** to handle incoming request from users silos
  + **App servers** to ensure session control, responsiveness.
  + **Cash server** to handle session active sessions consistently
  + **Database servers** for storing data,

**System Type: User Silos with Post-Processing** Similar to “User Silos”, but during one session a user may kick off one or several tasks involving considerable amounts of computations or data processing. The key here is that these tasks need not be interactive in nature – the results can be reported back to the users asynchronously (indeed, it may not even be the user that is the recipient of the tasks’ results).

* important quality attributes – reliability such as availability, functional suitability, usability such as user interface and accessibility, fault- tolerance, recoverability and functional suitability
* The system can contain users silos with:
  + **load balancers** to handle incoming request from users silos
  + **App servers** to ensure session control, responsiveness.
  + **Cash server** to handle session active sessions consistently
  + **Process** **scheduler** that orchestrates execution of tasks based on dependency of computations
  + **Clustered database** servers for reduced access loads and input and output read loads

**System Type: Networks of Users** In this type of system, the user interacts with a network of other users in realtime. When one user makes some information available, it should be made available to all other users concurrently. To “make information available” may require processing by the system (for example parsing the input format and translating it to desired formats for viewing). Each individual processing task may not be very intensive, but with a large number of concurrent users the amount of required processing power adds up. In this type of system a balance need to be struck between pushing processing back to the users’ devices and processing server-side. Pushing processing back to the clients may mean larger network transfers, and increased energy consumption on the client devices (which may be mobile devices).

The requirement of this system is similar to the social media application **Flicker** as illustrated by Rosenberg and Mateos.

Important quality attributes can be reliability (availability, fault-tolerance, recoverability), performance efficiency, usability, non-repudiation, and functional suitability.

* The system can contain users with:
  + **load balancers** to handle incoming request from users
  + **App server** to keep the accessed or shared information consistent by parsing input formats and translating it desired formats
  + **Net app** storage to make mass information available concurrently, server with higher RAM capacity
  + **Cash server** (Squid caches) to generate contents to the user HTML contents viewable by users
  + **Clustered database** servers for reduced access loads and input and output read loads

1. Filesystems

Consider again the simple application in Figure [1](https://mickesv.github.io/cloudbd/BusinessCase/BusinessCase-Assignment.html#fig:SimpleApp). In this example there is a single database that serves both instances of the application. But what about other data that does not fit into a usual database, such as binary files, or even whole filesystems? For this, you essentially have three options:

1. A self-administered networked filesystem. This requires that one of your hosts is designated as filesystem server.
2. A cloud-based networked filesystem. Some cloud providers also offer online storage facilities.
3. A self-administered distributed filesystem that is synchronized across your hosts.

**Questions:**

1. Briefly describe the advantages and disadvantages of each of these options

**Self-administered networked filesystem** – is any file system that enables access to files from several hosts through a shared computer network ex GlusterFS.

Advantages

* It can aggregate several servers and connect them through a shared network.

Disadvantages

* Since it uses known protocols such as TCP/IP – security vulnerabilities might be inevitable

**Cloud-based networked filesystem** - these are file systems provisioned by cloud-providers such as Google Cloud Storage.

Advantages

* Setting up might be easier

Disadvantages

* Speed of access and security are major concerns see next section

**Self-administered distributed filesystem** – a file system that is configurable by the user and that can run on a server configured on the cloud infrastructure opening up opportunities of users to administer the file system

Advantages

* Since such file systems are designed and administered by users therefore,

Disadvantages

* When run on the cloud it can have latency

1. Briefly describe each of the storage solutions below and list benefits and liabilities:

**Hadoop HDFS** [[7]](#footnote-7)– is a distributed file system which is designed to run on commodity hardware. HDFS is falut-tolerant and is designed to be deployed on low-cost hardware. This file system is suitable for applications with larger datasets.

Advantage

* Suitable for large datasets
* File can be mounted with file:// URL so it is easier

Disadvantage

* Security provided by Hadoop HDFS is still poor (Weets et al. 2015).

**GlusterFS[[8]](#footnote-8)** – is a network attached scale-out storage file system which aggregates storage servers over a given Ethernet interconnect them into a single large parallel network file system. GlusterFS has a server and a client component connected using TCP/IP protocols.

Advantage

* simple

Disadvantage

* it is educational version

**Amazon S3 (Simple Storage Service)** – is a cloud storage, which runs independently from other Amazon services and which stores persistent data and retrieve it when needed through web service API’s. It is not a file system but it has a two-level namespace with the first level, bucket, and bucket is like a directory, but you cannot create a bucket within another bucket. Also, bucket names should be consistent and shouldn't clash with another bucket elsewhere because bucket namespaces are shared across Amazon. You can store objects ranging from 1 byte to 5GB (Reese 2009).

Advantage

* you can put data into it without worrying about it getting full
* it can be used for short term or long term backup.

Disadvantage

* is relatively slow, it is not suitable for to be used as an operational storage medium
* It is not a file system therefore we need to build web services to access S3, it stores objects and not files
* Is relatively slow
* Objects stored in S3 cannot be larger than 5 GB

**Amazon Glacier[[9]](#footnote-9)-** according to Amazon Glacier is extremely low-cost, secure, durable cloud storage service for long-term backup and archiving. It stores a Gigabyte data for $0.004 per month.

Advantage

* is designed in compliance with stringent regulatory requirement
* allows to run querying-in-place functionality
* cheaper than on-premises solutions
* suitable to handle petabytes of data usually needed by healthcare information systems
* suitable for digital preservation

Disadvantage

* Lack of platforms to upload files but there and it requires third-party tools to store and retrieve[[10]](#footnote-10)

**Google Cloud Storage** – is RESTful file storage service running on Google infrastructure for storing and accessing data. It is suitable for enterprises, it can store objects up to 5 TiB and previously it used to store up to 100 GiB.

Advantage

* Google storage is interoperable with other cloud providers such as Amazon and Eucalyptus
* It has access control list to manage access control to objects and buckets
* Resume-able uploads i.e uploads even after upload operations fails

Disadvantage

* There is maximum request per 100 seconds per user, may be this is a limitations for systems like IoT systems where latency is a major concern

1. What are the security implications of using Amazon S3?

Amazon S3 is stored and accessed independently from other Amazon services, its will be accessed by its access key and private key. Third-party tools are required to mount S3, and I think third-party tools might have security vulnerabilities. The S3 can be made available to the public (Reese 2009). Bleepingcomputer[[11]](#footnote-11) claims that 7% of all Amazon S3 Servers are exposed to data leaks and 7% of buckets have unrestricted access. Also, it is possible for attackers to obtain URLs of buckets by brute-force domains for hidden URLs and MitM attacks.

1. Databases

In this Sprint it is claimed that the choice of database technology is essentially between an SQL database and a NoSQL database, and the NoSQL has received quite some hype in recent years. In this section, I will present the principles of NoSQL databases, the benefits of the NoSQL over the SQL databases and finally the I present examples indicating where the NoSQL database should be implemented.

## Principles of NoSQL databases

The NoSQL databases can achieve massive scale because it avoids joining operations and fixed table schemas in addition to overruling the governing SQL fundamental principles.

NoSQL uses a single data item, which provides eventual consistency and transaction. It is also possible to guarantee atomicity, consistency, isolation, and durability. An example of a NoSQL table can contains items such as car models, colors, car makes, and year along with the corresponding keys.

Several NoSQL implementations on distributed architectures employ hash table with redundant data on distributed servers, which ultimately leads to scalabilty and fault-tolerance. Whereas some NoSQL advocates promote the use of key value pairs or array while others promote the utilization of XQuery standards such as XML databases.

## Benefits of a NoSQL database over an SQL database

Big data (volume) – is not easily implemented by using SQL databases (Rosenberg and Mateos 2011).

The NoSQL breaks the basic relational database SQL principles to reach massive scale. The SQL has principle that ensures consistency of data at the expense of performance (Rosenberg and Mateos 2011).

NoSQL is suitable for the cloud since it reaches massive scale unlike the SQL counterpart (Rosenberg and Mateos 2011).

Examples of data where a NoSQL database is the most suitable storage alternative are:

* + 1. When companies want to build data-driven application such as the Google itself – this is because it is only possible to build economical systems through NoSQL which are scalable and reliable (Rosenberg and Mateos 2011).
    2. When a company wants to build as data-driven application which can provide fast access of petabytes of data, availability – such implementation is feasible through none relational solutions (Rosenberg and Mateos 2011).
    3. When a table or the data contains vastly different data – it is structurally possible to design such table using the key-value and item table (Rosenberg and Mateos 2011).

1. Security in the Cloud

## Security benefits of cloud computing.

Standard security procedure is needed for companies running their data centers even if their business is not to run secured data centers. It is also possible that companies make mistakes that costs them financial loss due to security breaches (Rosenberg and Mateos 2011).

The utilization of encrypted storage, standardized network security tools such as packet filters & fireworks, virtual local networks and so on enables overcome security concerns of users (Rosenberg and Mateos 2011).

It is possible to ensure auditability and hence confidentiality for providing more secure facilities. It is also possible to provide physical security such as security by obscurity and the use of certification such as SAS 70 to ensure auditing capability of providers. Even if security is the main blocking factor for using the cloud, cloud providers devoted considerable resources to build and deploy cloud infrastructures (Rosenberg and Mateos 2011).

There are best practices tested and being implemented by cloud providers to ensure network security such as run only one server, don’t open unnecessary ports, don’t access sensitive data directly, use reverse proxy as in load balancing, and limit access to your resources by only giving access when needed (Reese 2009).

## Security liabilities of cloud computing

Security concerns slows down cloud adoption. For example, fulfilling goals of service lever agreement is a difficult task to do and corporate fear to move to the cloud (Rosenberg and Mateos 2011).

Physical security is up to the provider while access control is a joint responsibility of providers and users (Rosenberg and Mateos 2011).

Users of cloud computing lose access to the physical location of data and they might even may not know where their data is stored. Also, legal and standard compliance issues are articulated mainly for non-virtual IT infrastrucures. Besides, from cloud security perspectives you will face three kinds of issues (Reese 2009), such as:

* The ***How*** issue – the standards employed governing how data is operated are designed for specific problems (Reese 2009).
* The ***Where*** issue – the EU for example has regulations governing where data can be stored (Reese 2009).
* The ***What*** issue – standards dictate what components should be present in the cloud infrastructure, this can for example be a specification regarding the necessity of antivirus software on each servers (Reese 2009).
* Vulnerabilities of virtualized technologies can attract attackers (Reese 2009).

# Reference

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.

Kulkarni, S., & Agrawal, P. (2014). Introduction. In *Analysis of TCP Performance in Data Center Networks* (pp. 1-15). Springer, New York, NY.

Reese, G. (2009). *Cloud application architectures: building applications and infrastructure in the cloud*. " O'Reilly Media, Inc.".

Rosenberg, J., & Mateos, A. (2010). *The cloud at your service*. Manning Publications Co..

Weets, J. F., Kakhani, M. K., & Kumar, A. (2015, October). Limitations and Challenges of HDFS and MapReduce. In *Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on* (pp. 545-549). IEEE.

http://www.dell.com/en-us/work/shop/povw/dell-compellent-sc4020

http://pilot.search.dell.com/application%20server%20price

http://www.dell.com/en-us/shop/accessories/apd/a8631055?ref=p13n\_ena\_pdp\_vv&c=us&cs=04&l=en&s=bsd

http://www.router-switch.com/asa5512-k9-p-4613.html

https://www.ebay.com/p/27u-4-Post-Open-Frame-Server-Rack-Enclosure-19-Adjustable-Depth/1661238516?iid=151584295835

https://www.cableorganizer.com/middle-atlantic/designer-inspired-ultra-quiet-fan-panel.htm

https://aws.amazon.com/ec2/pricing/

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/t2-credits-baseline-concepts.html

https://hadoop.apache.org/docs/r1.2.1/hdfs\_design.html

https://en.wikipedia.org/wiki/Gluster

https://aws.amazon.com/glacier/

http://www.technologyguide.com/howto/how-to-back-up-your-data-with-amazon-glacier/

https://www.bleepingcomputer.com/news/security/7-percent-of-all-amazon-s3-servers-are-exposed-explaining-recent-surge-of-data-leaks/

1. <http://www.dell.com/en-us/work/shop/povw/dell-compellent-sc4020> [↑](#footnote-ref-1)
2. <http://pilot.search.dell.com/application%20server%20price> [↑](#footnote-ref-2)
3. <http://www.dell.com/en-us/shop/accessories/apd/a8631055?ref=p13n_ena_pdp_vv&c=us&cs=04&l=en&s=bsd> [↑](#footnote-ref-3)
4. <http://www.router-switch.com/asa5512-k9-p-4613.html> [↑](#footnote-ref-4)
5. <https://www.ebay.com/p/27u-4-Post-Open-Frame-Server-Rack-Enclosure-19-Adjustable-Depth/1661238516?iid=151584295835> [↑](#footnote-ref-5)
6. <https://www.cableorganizer.com/middle-atlantic/designer-inspired-ultra-quiet-fan-panel.htm> [↑](#footnote-ref-6)
7. <https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html> [↑](#footnote-ref-7)
8. <https://en.wikipedia.org/wiki/Gluster> [↑](#footnote-ref-8)
9. <https://aws.amazon.com/glacier/> [↑](#footnote-ref-9)
10. <http://www.technologyguide.com/howto/how-to-back-up-your-data-with-amazon-glacier/> [↑](#footnote-ref-10)
11. <https://www.bleepingcomputer.com/news/security/7-percent-of-all-amazon-s3-servers-are-exposed-explaining-recent-surge-of-data-leaks/> [↑](#footnote-ref-11)