Unit III (Cloud Computing)

1. Service level agreements in Cloud computing

A **Service Level Agreement (SLA)** is the bond for performance negotiated between the cloud services provider and the client. Earlier, in cloud computing all Service Level Agreements were negotiated between a client and the service consumer. Nowadays, with the initiation of large utility-like cloud computing providers, most Service Level Agreements are standardized until a client becomes a large consumer of cloud services. Service level agreements are also defined at **different levels** which are mentioned below:

- Customer-based SLA
- Service-based SLA
- Multilevel SLA

Few Service Level Agreements are enforceable as contracts, but mostly are agreements or contracts which are more along the lines of an Operating Level Agreement (OLA) and may not have the restriction of law. It is fine to have an attorney review the documents before making a major agreement to the cloud service provider. Service Level Agreements usually specify **some parameters** which are mentioned below:

1. Availability of the Service (uptime)
2. Latency or the response time
3. Service components reliability
4. Each party accountability
5. Warranties

In any case, if a cloud service provider fails to meet the stated targets of minimums then the provider has to pay the penalty to the cloud service consumer as per the agreement. So, Service Level Agreements are like insurance policies in which the corporation has to pay as per the agreements if any casualty occurs.
Microsoft publishes the Service Level Agreements linked with the Windows Azure Platform components, which is demonstrative of industry practice for cloud service vendors. Each individual component has its own Service Level Agreements. Below are two **major Service Level Agreements (SLA)** described:

1. **Windows Azure SLA –** Window Azure has different SLA's for compute and storage. For compute, there is a guarantee that when a client deploys two or more role instances in separate fault and upgrade domains, client's internet facing roles will have external connectivity minimum 99.95% of the time. Moreover, all of the role instances of the client are monitored and there is guarantee of detection 99.9% of the time when a role instance's process is not runs and initiates properly.
2. **SQL Azure SLA –** SQL Azure clients will have connectivity between the database and internet

gateway of SQL Azure. SQL Azure will handle a “Monthly Availability” of 99.9% within a month. Monthly Availability Proportion for a particular tenant database is the ratio of the time the database was available to customers to the total time in a month. Time is measured in some intervals of minutes in a 30-day monthly cycle. Availability is always remunerated for a complete month. A portion of time is marked as unavailable if the customer’s attempts to connect to a database are denied by the SQL Azure gateway.

Service Level Agreements are based on the usage model. Frequently, cloud providers charge their pay-as-per-use resources at a premium and deploy standards Service Level Agreements only for that purpose. Clients can also subscribe at different levels that guarantees access to a particular amount of purchased resources. The Service Level Agreements (SLAs) attached to a subscription many times offer various terms and conditions. If client requires access to a particular level of resources, then the client need to subscribe to a service. A usage model may not deliver that level of access under peak load condition.

2. Overview of Cloud Billing concepts

You can configure billing on Google Cloud in a variety of ways to meet different needs. This section introduces the core concepts for your organization and for billing, and discusses how to use them effectively.

Resource Overview

What is a resource?

In the context of Google Cloud, a resource can refer to the service-level resources that are used to process your workloads (VMs, DBs, and so on) as well as to the account-level resources that sit above the services, such as projects, folders, and the organization.

What is resource management?

Resource management is focused on how you should configure and grant access to the various cloud resources for your company/team, specifically the setup and organization of the account-level resources that sit above the service-level resources. Account-level resources are the resources involved in setting up and administering your Google Cloud account.

Resource Hierarchy

Google Cloud resources are organized hierarchically. This hierarchy allows you to map your organization's operational structure to Google Cloud, and to manage access control and permissions for groups of related resources. The resource hierarchy provides logical attach points for access management policies (Cloud Identity and Access Management) and Organization policies.

Both Cloud IAM and Organization policies are inherited through the hierarchy, and the effective policy at each node of the hierarchy is the result of policies directly applied at the node and policies inherited from its ancestors.

The following diagram shows an example resource hierarchy illustrating the core account-level resources involved in administering your Google Cloud account.

Domain

- Your company Domain is the primary identity of your organization and establishes your company's identity with Google services, including Google Cloud.
- You use the domain to manage the users in your organization.
- At the domain level, you define which users should be associated with your organization when using Google Cloud.
- Domain is also where you can universally administer policy for your users and devices (for example, enable 2-factor authentication, reset passwords for any users in your organization).
- The Domain is linked to either a G Suite or Cloud Identity account.
- The G Suite or Cloud Identity account is associated with exactly one Organization.
- You manage the domain-level functionality using the **Google Admin Console (admin.google.com)**.

For more information on the hierarchy of resources, see the Resource Manager documentation.

Organization

- An Organization is the root node of the Google Cloud hierarchy of resources.
- All Google Cloud resources that belong to an Organization are grouped under the Organization node, allowing you to define settings, permissions, and policies for all projects, folders, resources, and Cloud Billing accounts it parents.

- An Organization is associated with exactly one Domain (established with either a G Suite or Cloud Identity account), and is created automatically when you set up your domain in Google Cloud.
- Using an Organization, you can centrally manage your Google Cloud resources and your users' access to those resources. This includes:
- Proactive management: reorganize resources as needed (for example, restructuring or spinning up a new division may require new projects and folders).
- Reactive management: an Organization resource provides a safety net to regain access to lost resources (for example, if one of your team members loses their access or leaves the company).
- The various roles and resources that are related to Google Cloud (including the organization, projects, folders, resources, and Cloud Billing accounts) are managed within the **Google Cloud Console**.

For more information on organizations, see Creating and Managing Organizations.

Folders

- Folders are a grouping mechanism and can contain projects, other folders, or a combination of both.
- To use folders, you must have an Organization node.
- Folders and projects are all mapped under the Organization node.
- Folders can be used to group resources that share common Cloud IAM policies.
- While a folder can contain multiple folders or resources, a given folder or resource can have exactly one parent.

For more details about using folders, see Creating and Managing Folders.

Projects

- Projects are **required** to use service-level resources (such as Compute Engine virtual machines (VMs), Pub/Sub topics, Cloud Storage buckets, and so on).
- All service-level resources are parented by projects, the base-level organizing entity in Google Cloud.
- You can use projects to represent logical projects, teams, environments, or other collections that map to a business function or structure.
- Projects form the basis for enabling services, APIs, and Cloud IAM permissions.
- Any given resource can only exist in one project.

For more details about projects, see Creating and Managing Projects.

Resources

- Google Cloud service-level resources are the fundamental components that make up all Google Cloud services, such as Compute Engine virtual machines (VMs), Pub/Sub topics, Cloud Storage buckets, and so on.
- For billing and access control purposes, resources exist at the lowest level of a hierarchy that also includes projects and an organization.

Labels

- Labels help you categorize your Google Cloud resources (such as Compute Engine instances).
- A label is a key-value pair.
- You can attach labels to each resource, then filter the resources based on their labels.
- Labels are great for cost tracking at a granular-level. Information about labels is forwarded to the billing system, so you can analyze your charges by label.

For more details about using labels, see Creating and Managing Labels.

Cloud Billing account & payments profile

Overview

A **Cloud Billing account** is set up in Google Cloud and is used to define who pays for a given set of Google Cloud resources and Google Maps Platform APIs. Access control to a Cloud Billing account is established by Cloud Identity and Access Management (Cloud IAM) roles. A Cloud Billing account is connected to a **Google payments profile**. Your Google payments profile includes a payment instrument to which costs are charged.

monetization_on Cloud Billing account	payment Payments Profile
A Cloud Billing account: Is a cloud-level resource managed in the Cloud Console. Tracks all of the costs (charges and usage credits) incurred by your Google Cloud usage	A Google Payments Profile: Is a Google-level resource managed at payments.google.com. Connects to *ALL* of your Google services (such as Google Ads, Google Cloud,

A Cloud Billing account can be linked to one or more projects.

Project usage is charged to the linked Cloud Billing account.

Results in a single invoice per Cloud Billing account

Operates in a single currency

Defines who pays for a given set of resources

Is connected to a Google Payments Profile, which includes a payment instrument, defining how you pay for your charges

Has *billing-specific* roles and permissions to control accessing and modifying billing-related functions (established by Cloud Identity and Access Management roles)

and Fi phone service).

Processes payments for *ALL* Google services (not just Google Cloud.

Stores information like name, address, and tax ID (when required legally) of who is responsible for the profile.

Stores your various payment instruments (credit cards, debit cards, bank accounts, and other payment methods you've used to buy through Google in the past.)

Functions as a document center, where you can view invoices, payment history, and so on.

Controls who can view and receive invoices for your various Cloud Billing accounts and products.

Cloud Billing account types

There are two types of Cloud Billing accounts:

- Self-serve (or Online) account
- Payment instrument is a credit or debit card or ACH direct debit, depending on availability in each country or region.
- Costs are charged automatically to the payment instrument connected to Cloud Billing account.
- You can sign up for self-serve accounts online.
- The documents generated for self-serve accounts include statements, payment receipts, and tax invoices, and are accessible in the Cloud Console.
- Invoiced (or Offline) account
- Payment instrument can be check or wire transfer.
- Invoices are sent by mail or electronically.
- Invoices are also accessible in the Cloud Console, as are payment receipts.
- You must be eligible for invoiced billing. Learn more about invoiced billing eligibility.

Payments profile types

When you create your payments profile, you'll be asked to specify the profile type. This information must be accurate for tax and identity verification. **This setting can't be changed**. When you are setting up your payments profile, make sure to choose the type that best fits how you plan to use your profile.

There are two types of payments profiles:

- Individual
- You're using your account for your own personal payments.
- If you register your payments profile as an individual, then only you can manage the profile. You won't be able to add or remove users, or change permissions on the profile.
- Business
- You're paying on behalf of a business, organization, partnership, or educational institution.
- You use Google payments center to pay for Play apps and games, and Google services like Google Ads, Google Cloud, and Fi phone service.
- A business profile allows you to add other users to the Google payments profile you manage, so that more than one person can access or manage a payments profile.
- All users added to a business profile can see the payment information on that profile.

Charging cycle

The charging cycle on your Cloud Billing account determines how and when you pay for your Google Cloud services and your use of Google Maps Platform APIs.

For self-serve Cloud Billing accounts, your Google Cloud costs are charged automatically in one of two ways:

- Monthly billing: Costs are charged on a regular monthly cycle.
- Threshold billing: Costs are charged when your account has accrued a specific amount.

For self-serve Cloud Billing accounts, your charging cycle is automatically assigned when you create the account. You do not get to choose your charging cycle and you cannot change the charging cycle.

For invoiced Cloud Billing accounts, you typically receive one invoice per month and the amount of time you have to pay your invoice (your payment terms) is determined by the agreement you made with Google.

Billing contacts

A Cloud Billing account includes one or more contacts that are defined on the Google Payments profile that is connected to the Cloud Billing account. These contacts are people who are designated to receive billing information specific to the payment instrument on file (for example, when a credit card needs to be updated). To access and manage this list of contacts, you can use the Payments console or you can use the Cloud Console.

Subaccounts

Subaccounts are intended for resellers. If you are a reseller, you can use subaccounts to represent your customers' charges for the purpose of chargebacks.

Cloud Billing subaccounts allow you to group charges from projects together on a separate section of your invoice. A billing subaccount is a Cloud Billing account with a billing linkage to a reseller's master Cloud Billing account on which the charges appear. The master Cloud Billing account must be on invoiced billing.

A subaccount behaves like a Cloud Billing account in most ways: it can have projects linked to it, Cloud Billing data exports can be configured on it, and it can have Cloud IAM roles defined on it. Any charges made to projects linked to the subaccount are grouped and subtotalled on the invoice, and the effect on resource management is that access control policy can be entirely segregated on the subaccount to allow for customer separation and management.

The Cloud Billing Account API provides the ability to create and manage subaccounts. Use the API to connect to your existing systems and provision new customers or chargeback groups programmatically.

Relationships between organizations, projects, Cloud Billing accounts, and payments profiles

Two types of relationships govern the interactions between organizations, Cloud Billing accounts, and projects: ownership and payment linkage.

- **Ownership** refers to Cloud IAM permission inheritance.
- **Payment linkages** define which Cloud Billing account pays for a given project.

The following diagram shows the relationship of ownership and payment linkages for a

sample organization.

In the diagram, the organization has ownership over Projects 1, 2, and 3, meaning that it is the Cloud IAM permissions parent of the three projects.

The Cloud Billing account is linked to Projects 1, 2, and 3, meaning that it pays for costs incurred by the three projects.

The Cloud Billing account is also linked to a Google payments profile, which stores information like name, address, and payment methods.

Note: Although you link Cloud Billing accounts to projects, Cloud Billing accounts are not parents of projects in an Cloud IAM sense, and therefore projects don't inherit permissions from the Cloud Billing account they are linked to.

In this example, any users who are granted Cloud IAM billing roles on the organization also have those roles on the Cloud Billing account or the projects.

For more information on granting Cloud IAM billing roles, see Overview of Cloud Billing access control.

Roles Overview

What are roles?

Roles grant one or more privileges to a user that allow performing a common business function.

How do roles work in Google Cloud?

Google Cloud offers Cloud Identity and Access Management (Cloud IAM) to manage access control to your Google Cloud resources. Cloud IAM lets you control **who (users)** has **what access (roles)** to **which resources** by setting Cloud IAM policies. To assign permissions to a user, you use Cloud IAM policies to grant specific role(s) to a user. Roles have one or more permissions bundled within them, controlling user access to resources.

You can set a Cloud IAM policy (roles) at the organization level, the folder level, the project level, or (in some cases) on the service-level resource.

Policies are inherited through the hierarchy. The effective policy at each node of the hierarchy is the result of policies directly applied at the node and policies inherited from its ancestors. If you set a policy at the Organization level, it is inherited by all its child folders and projects. If you set a policy at the project level, it is inherited by all its child

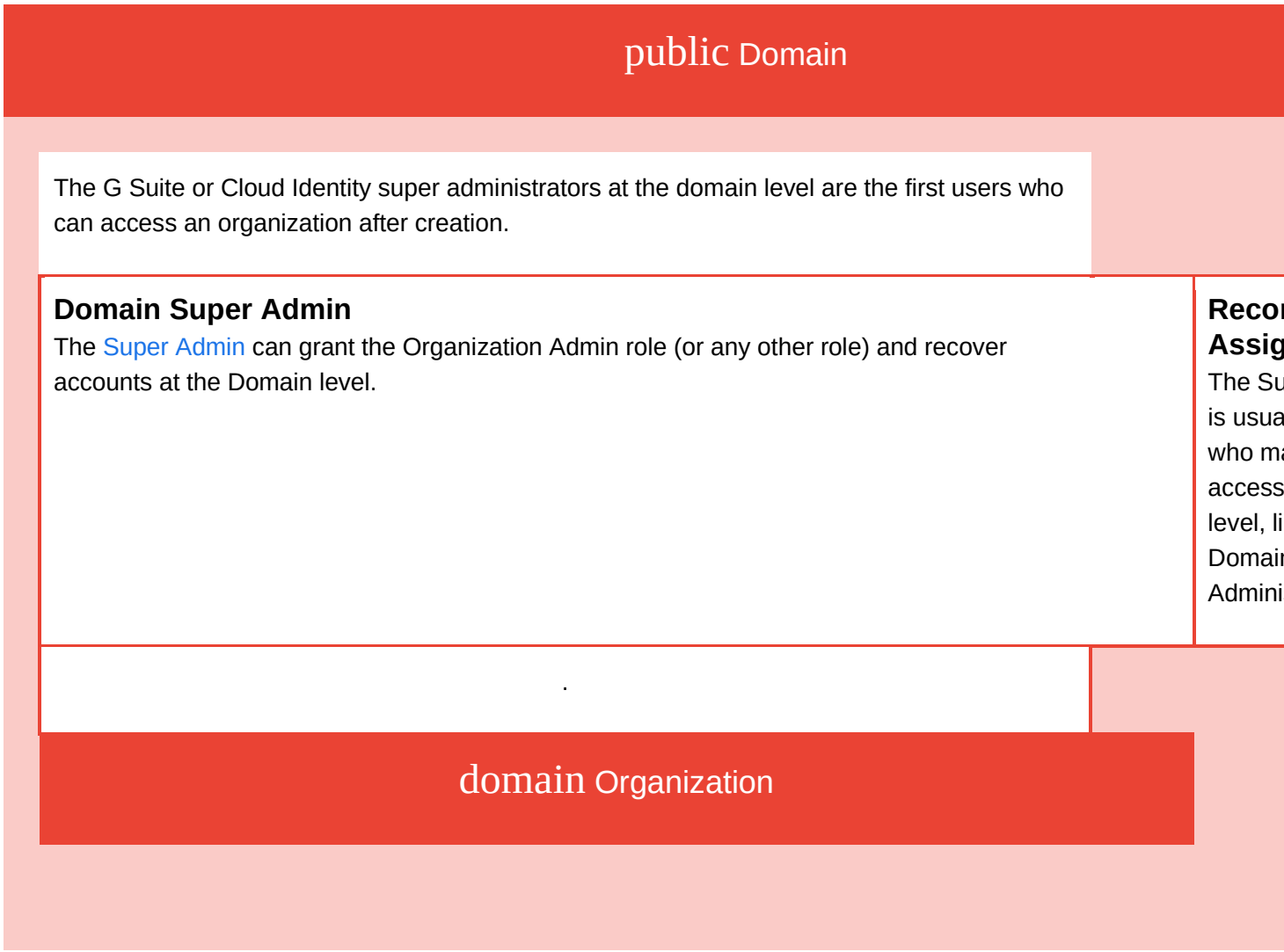
resources. You can enforce granular permissions at different levels in the resource hierarchy to ensure that the right individuals have the ability to spend within Google Cloud.

Best Practices for Roles

- Assign key roles to more than one person (reasonable redundancy)
- Document who your admins are and communicate those names to people in your organization
- Keep role assignments up to date

Important Roles

The following diagram represents the Google Cloud resource hierarchy in complete form, and calls out the important high-access roles at each level:



payment Payments Profile

Payments Profiles are managed outside of your Cloud Organization, in the Google Payments Center, a single loc
where you can manage the ways you pay for all Google products and services, such as Google Ads, Google Clou
and Fi phone service. Payments Profiles are connected to Cloud Billing accounts.

Payments Profile Admin	**Recommended Assignee**
The Payments Profile Admin can view and manage payment methods, make payments, view invoices, and see Payments Accounts.	The Payments Profile Admins in your organization are typically part of your Fin or Accounting teams.

3. Comparing Scaling Hardware:

Cloud Computing vs Traditional IT Infrastructure

Cloud computing is really popular nowadays. More and more companies prefer using cloud infrastructure rather than the traditional one. Really it's much more reasonable to buy a cloud storage subscription instead of investing in physical in-house servers. However, are there any benefits of using the cloud computing instead of traditional one? Let's review the main differences.

The differences between **cloud computing** and **traditional IT** infrastructure

Elasticity and resilience

First of all, you do not need to buy the hardware and maintain it with your own team. The information in the cloud is stored on several servers at the same time. It means that even if 1 or 2

servers are damaged, you will not lose your information. It also helps to provide the high uptime, up to 99.9%.

When we talk about their traditional infrastructure, you will have to buy and maintain the hardware and equipment. If something happens, you can lose the data and spend a lot of time and money to fix the issues.

Scalability and flexibility

The cloud computing is the perfect Choice for those who do not require a high performance constantly but use it time by time. You can get a subscription and use the resources you paid for. Most providers even let pause the subscription if you do not need it. and at the same time, you're able to control everything and get instant help from the support team.

The traditional infrastructure is not so flexible. You have to buy an equipment and maintain it even if you do not use it. In many cases, it's even more expensive because you might need their own technical crew.

Automation

One of the biggest differences between cloud and traditional infrastructure is how they are maintained. Cloud service is served by the provider's support team. They take care of all the necessary aspects including security, updates, hardware, etc.

The traditional infrastructure required the own team to maintain and monitor the system. It requires a lot of time and efforts.

Cost

With cloud computing, you do not need to pay for the services you don't use: the subscription model means you choose the amount of space, processing power, and other components that you really need.

With traditional infrastructure, you are limited to the hardware you have. If your business is growing, you will regularly have to expand your infrastructure. At the same time, you will have to support and maintain it.

Security

Many people are not sure about the security of cloud services. Why can it be not so secure? As the company uses the third party solution to store data, it's reasonable to think that the provider can access the confidential data without permission. However, there are good solutions to avoid the leaks.

As for traditional infrastructure, you and only you are responsible for who will be able to access the stored data. For the companies who operate the confidential information, it's a better solution.

What kind of infrastructure is a good choice for your business? It depends on what your company does and what are your needs. Nevertheless, more and more organisations today prefer cloud infrastructure.

4. Data organization on the cloud

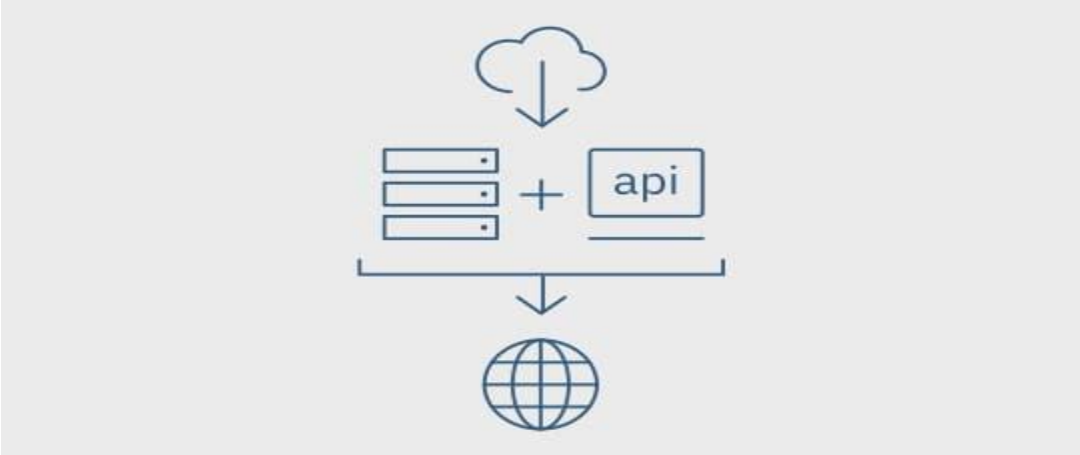
A cloud database is a database service built and accessed through a cloud platform. It serves many of the same functions as a traditional database with the added flexibility of cloud computing. Users install software on a cloud infrastructure to implement the database.

Key features:

- A database service built and accessed through a cloud platform
- Enables enterprise users to host databases without buying dedicated hardware
- Can be managed by the user or offered as a service and managed by a provider

- Can support relational databases (including MySQL and PostgreSQL) and NoSQL databases (including MongoDB and Apache CouchDB)
- Accessed through a web interface or vendor-provided API

Why cloud databases



Ease of access

Users can access cloud databases from virtually anywhere, using a vendor's API or web interface.

Scalability

Cloud databases can expand their storage capacities on run-time to accommodate changing needs. Organizations only pay for what they use.

Disaster recovery

In the event of a natural disaster, equipment failure or power outage, data is kept secure through backups on remote servers.

Considerations for cloud databases

- **Control options**
  Users can opt for a virtual machine image managed like a traditional database or a provider's database as a service (DBaaS).
- **Database technology**
  SQL databases are difficult to scale but very common. NoSQL databases scale more easily but do not work with some applications.
- **Security**
  Most cloud database providers encrypt data and provide other security measures; organizations should research their options.
- **Maintenance**
  When using a virtual machine image, one should ensure that IT staffers can maintain the underlying infrastructure.

Large Scale data processing in cloud computing

Today, the need to process large amount of data has been enhanced in the area of Engineering,Science, Commerce and the Economics of the world. The ability to process huge data frommultiple resources of data remains a critical challenge.Many organizations face difficulties when dealing withalarge amount of data. They are unable tomanage, manipulate, process, share and retrieve large amounts of data by traditional softwaretools due to them being costly and time-consuming for data processing. The term large-scaleprocessing is focused on how to handle the applications with massive datasets.Such applicationsdevote the largest fraction of execution time to movement of data from data storage to thecomputing nodein a computing environment .The main challenges behind such applications are data storage capacity and processor computing power constrains. Developers need hundredsor thousands of processing nodes and large volume of storage devices to process complexapplications with large datasets, such applications process multi-terabyte to petabyte-sizeddatasets and using traditional data processing methods like sequential processing andcentralized data processing are not effective to solve these kinds of application'

sproblems.The question is how to process large amounts of distributed data quickly with good responsetimes and replication at minimum cost? One of the best ways for huge data processing is toperform parallel and distributed computing in a cloud computing environment.Cloud computingas a distributed computing paradigm aimsatlarge datasets to be processed on available computernodes by using a MapReduce framework. MapReduce is a software framework introduced to

theworld by Google in 2004;it runs on a large cluster of machines and is highly scalable . It is ahigh-performance processing technique to solve large-scale dataset problems. MapReducecomputation processes petabyte to terabyte of unit data on thousands of processors. Google usesMapReduce for indexing web pages. Its main aim is to process large amount of data in parallelstored on a distributed clusterof computers. This studypresents a way to solve large-scale datasetprocessingproblems in parallel and distributed mode operating on a large cluster of machines byusing MapReduce framework. It is a basis to take advantage of cloud computing paradigm asanew realistic computation industry standard.The first contribution of this work is topropose aframeworkfor running MapReduce system in acloud environmentbased on the captured requirements and to present its implementationonAmazon Web Services.The second contribution is to present an experimentation of running theMapReduce system in a cloud environment to validate the proposed framework and to present theevaluation of the experiment based on the criteria such as speed of processing, data-storage usage,response time and cost efficiency.The rest of the paper is organized as follows. Section II provides background information anddefinitionof MapReduce and Hadoop. Section III describesworkflow ofMapReduce, the generalintroduction of Map and Reduce functions and it also describes Hadoop, an implementation of theMapReduce framework. Section IV we present MapReduce in cloud computing. Section Vpresents the relatedMapReducesystems. Section VI capturesa set of requirements to develop theframework. Section VII showstheproposed framework andthe implementation of the framework onAmazon Web Services for running a MapReduce system in a cloud environment it alsopresents anexperimentation of running a MapReduce system in a cloud environmentto validatethe proposed framework and resulting outcomes with evaluation criteria.

2.BACKGROUND OF MAP REDUCE AND HADOOP

2.1.MapReduce Definition& History

MapReduce has been facilitated by Google as a programming framework to analyse massiveamounts of data. It uses for distributed data processing on large datasets acrossacluster of machines. Since the input data is too large, the computationneedsto be distributed acrossthousands of machines within a cluster in order to finish each partof computation in a reasonableamount of time. This distributed concept implies to parallelize computations easily and using re-execution as the main technique for faulttolerance. J.Dean and S.Ghemawat from Google Inc.published a paper in 2004 describing MapReduce. Google never released their implementation of MapReduce. Finally,the Apache Company made available a concrete implementation of MapReduce named Hadoop.

MapReduce allows developers to perform complex computationsin a simple way while hiding the details of data distributation, parallelization, and fault tolerance.

The unique feature of MapReduce is that it can both interpret and analyse both structuredandunstructured data across many nodes through using of a distributed share nothing architecture.Share nothing architecture is a distributed computing architecture consisting of multiple nodes.Each node is independent, has its own disks, memory, and I/Odevices in the network. In this typeof architecture, each node is self-sufficient and shares nothing over the network: therefore,thereare no points of contention across the system.A MapReduce programming model drives from three fundamental phases:1.

Map phase

: partition into M Map function (Mapper);each Mapper runs in parallel. Theoutputs of Map phase are intermediate key and value pairs.2.

Shuffle and Sort phase

: the output of each Mapper is partitioned by hashing the output key.In this phase,the number of partitions is equal to the number of reducers;all key and valuepairs in shuffle phase share the same key that belongs to the same partition. Afterpartitioning the Map output, each partition is stored by a key to merge all values for thatkey.3.

Reduce phase

: partition into R Reduce function (Reducer);each Reducer also runs in paralleland processes different intermediate keys.MapReduce libraries have been written in several programming languages,include, LISP, Java,C++, Python, Ruby and C. A presentation oftheMapReduceworkflowincludes;adataset is dividedinto several unitsof dataand then each unit of data is processed in a Map phase.Finally,they arecombined in Reduce phase to produce the finaloutput.Map function takes input pairs and produces a set of intermediate key and value pairs and passesthem to the Reduce function in order to combine all the values associated with the same key.Reduce function accepts an intermediate key as a set ofvalues for that key; it merges togetherthese values to prepare a proper smaller set of values to produce the output file .

MAP REDUCE IN CLOUD COMPUTING

Cloud Computing refers to both the applications delivered as services over the Internet,thehardware andthe systemsoftware in the datacenters that provide those services .Cloudplatform, or platform as a service, refers to providea computer platform or software stack as aservice.Developersby using the cloud computing paradigm are enabled to perform paralleldata processing in a distributed environment with affordable cost and reasonable time. Thus, theadvantage of data processing using cloud computing is the capability to easily do parallel anddistributed computing on large clusters.It is a fact that many cloud computing based computational processes will handle large datasets ina fully distributed environment in both a wired and wireless computer networking andcommunication environment.By using cloud computing, we enableto store,collect, share andtransfer large amounts of data at very high speeds in a seamless and transparent manner that
would out of sheer necessity classify all data to be "totally virtual". Hence, all data in cloud
computing captures the concept of data virtualization through a new programming paradigm ormodel which treats all data as a single entity through a process called MapReduce.MapReduce isa popular computing framework that is widely used for big data processing in cloud platforms.Cloud computing as adistributed computing paradigm, providesan environment to performlarge-scale data processing. It enablesmassive data analyticson available computer nodes byusing MapReduce platform.MapReduce and its open-source implementation Hadoop, allowdevelopers to process terabytes of data that take hours to finish while hiding the complexity of parallel execution across hundreds of servers in a cloud environment. The main reason of usingMapReduce with cloud computing is a key point of MapReduce that hideshowparallelprogramming work away from the developer.A Major web company, Amazon web services platform offers a service called Amazon ElasticMapReduce to store and process massive datasets by running MapReducesystemon Amazoncloud.It utilizes a hostedHadoop framework running on the

web-scale infrastructure of AmazonElastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3). EC2is a web service platform that provides resizable compute capacity in a cloud .Amazon S3providesa simple web services interface that can be used to store and retrieve any amount of data,at any time, from anywhere on the web.

5.MAP REDUCE REQUIREMENTS

5.1.Fundamental and Specific Requirements

MapReduce platform is a distributed runtime engine for managing, scheduling and runningMapReduce systems in a cluster of servers across an open distributed file system. To develop aMapReduce system based on the proposed framework, fundamental and specific requirementshave been captured.Table 1lists the summary of fundamental requirements for MapReducesystem. Table 2lists the summary of specific requirements. Both requirements essentially must bemet to make a MapReduce system for large-scale processing more efficient.

Table1.Fundamental Requirements.

NoFundamentalRequirementsDescription

1 Scalability To scale petabytes of data on thousands of machines.

2 Parallelism All tasks must run in parallel.

3 Distributed Data MapReduce distributesadata file to all nodesacross a cluster to execute the application.

4 Cost Efficiency Afford to buy cheaper hardware and pay lessfor operation, especially if the size of dataset istoo big.Table2.Specific Requirements.

Specific Requirements Description

1 Availability Many machine nodes and servers should beavailable in a computing cluster in failure mode.

2 Reliability Multiple copies of data should be automaticallystored in case of failure.

3 F l e x i b i l i t y The system should enable toanalyse and processvarious kinds of structured and unstructureddata.
4 S e c u r i t y Before running thesystem, user authenticationisrequired.
5 U s a b i l i t y The system should be developed as a service forrunning arbitrary code.
6 L o c a l i t y T h e s y s t e m s h o u l d d i v i d e t a s k s b a s e d o n location of inputfile;each part is 64 MB samesize of Google File System.
7DataConsistencyThe system should support coordination of datachanges and it helps to provideconsistency of data to ensure correctness of the execution andresult.
8 T r u s t When all nodes faithfully execute their tasks, theresult is accurate and can trust the result.

Amazon EC2 (Elastic Compute Cloud)

Amazon Elastic Compute Cloud (Amazon EC2) is a web-based service that allows businesses to run application programs in the Amazon Web Services (AWS) public cloud. Amazon EC2 allows a developer to spin up virtual machines (VMs), which provide compute capacity for IT projects and cloud workloads that run with global AWS data centers.

An AWS user can increase or decrease instance capacity as needed within minutes using the Amazon EC2 web interface or an application programming interface (API). A developer can code an application to scale instances automatically with AWS Auto Scaling. A developer can also define an autoscaling policy and group to manage multiple instances at once.

EC2 history

EC2 was the idea of engineer Chris Pinkham who conceived it as a way to scale Amazon's internal infrastructure. Pinkham and engineer Benjamin Black presented a paper on their ideas to Amazon CEO Jeff Bezos, who liked what he read and requested details on virtual cloud servers.

EC2 was then developed by a team in Cape Town, South Africa. Pinkham provided the initial architecture guidance for EC2, gathered a development team and led the project along with Willem van Biljon.

In 2006, Amazon announced a limited public beta test of EC2, and in 2007 added two new instance types -- Large and Extra-Large. Amazon announced the addition of static IP addresses, availability zones, and user selectable kernels in spring 2008, followed by the release of the Elastic Block Store (EBS) in August.

Amazon EC2 went into full production on October 23, 2008. Amazon also released a service level agreement (SLA) for EC2 that day, along with Microsoft Windows and SQL Server in beta form on EC2. Amazon added the AWS Management Console, load balancing, autoscaling, and cloud monitoring services in 2009.

As of 2019, EC2 and Amazon Simple Storage Service (S3) are the most popular of Amazon's AWS products.

How EC2 works

To begin using EC2, developers sign up for an account at Amazon's AWS website. They can then use the AWS Management Console, the AWS Command Line Tools (CLI), or AWS Software Developer Kits (SDKs) to manage EC2.

A developer then chooses EC2 from the AWS Services dashboard and 'launch instance' in the EC2 console. At this point, they select either an Amazon Machine Image (AMI) template or create an AMI containing an operating system, application programs, and configuration settings. The AMI is then uploaded to the Amazon S3 and registered with Amazon EC2, creating an AMI identifier. Once this has been done, the subscriber can requisition virtual machines on an as-needed basis.

Data only remains on an EC2 instance while it is running, but a developer can use an Amazon Elastic Block Store volume for an extra level of durability and Amazon S3 for EC2 data backup.

VM Import/Export allows a developer to import on-premises virtual machine images to Amazon EC2, where they are turned into instances.

EC2 also offers Amazon CloudWatch which monitors Amazon cloud applications and resources, allowing users to set alarms, view graphs, and get statistics for AWS data; and AWS Marketplace, an online store where users can buy and sell software that runs on AWS.

Amazon EC2 instance types

Instances allow developers to expand computing capabilities by 'renting' virtual machines rather than purchasing hardware. An EC2 instance is used to run applications on the Amazon Web Services infrastructure.

Amazon EC2 provides different instance types, sizes and pricing structures designed for different computing and budgetary needs. In addition to general purpose instances, Amazon EC2 offers an instance type for compute, memory, accelerated computing, and storage-optimized workloads. AWS limits how many instances a user can run in a region at a time, depending on the type of instance. Each instance type comes with different

size options corresponding to the CPU, memory and storage needs of each enterprise.

Cost

On-Demand instances allow a developer to create resources as needed and to pay for them by the hour. Reserved instances (RIs) provide a price discount in exchange for one and three-year contract commitments -- a developer can also opt for a convertible RI, which allows for the flexibility to change the instance type, operating system or tenancy. There's also an option to purchase a second-hand RI from the Amazon EC2 reserved instances marketplace. A developer can also submit a bid for spare Amazon EC2 capacity, called Spot instances, for a workload that has a flexible start and end time. If a business needs dedicated physical server space, a developer can opt for EC2 dedicated hosts, which charge hourly and let the business use existing server-bound software licenses, including Windows Server and SQL Server.

A breakdown of Amazon EC2 instances and their associated prices.

Benefits

Getting started with EC2 is easy, and because EC2 is controlled by APIs developers can commission any number of server instances at the same time to quickly increase or decrease capacity. EC2 allows for complete control of

instances which makes operation as simple as if the machine were in-house.

The flexibility of multiple instance types, operating systems, and software packages and the fact that EC2 is integrated with most AWS Services -- S3, Relational Database Service (RDS), Virtual Private Cloud (VPC) -- makes it a secure solution for computing, query processing, and cloud storage.

Challenges

Resource utilization -- developers must manage the number of instances they have to avoid costly large, long-running instances.

Security -- developers must make sure that public facing instances are running securely.

Deploying at scale -- running a multitude of instances can result in cluttered environments that are difficult to manage.

Management of AMI lifecycle -- developers often begin by using default Amazon Machine Images. As computing needs change, custom configurations will likely be required.

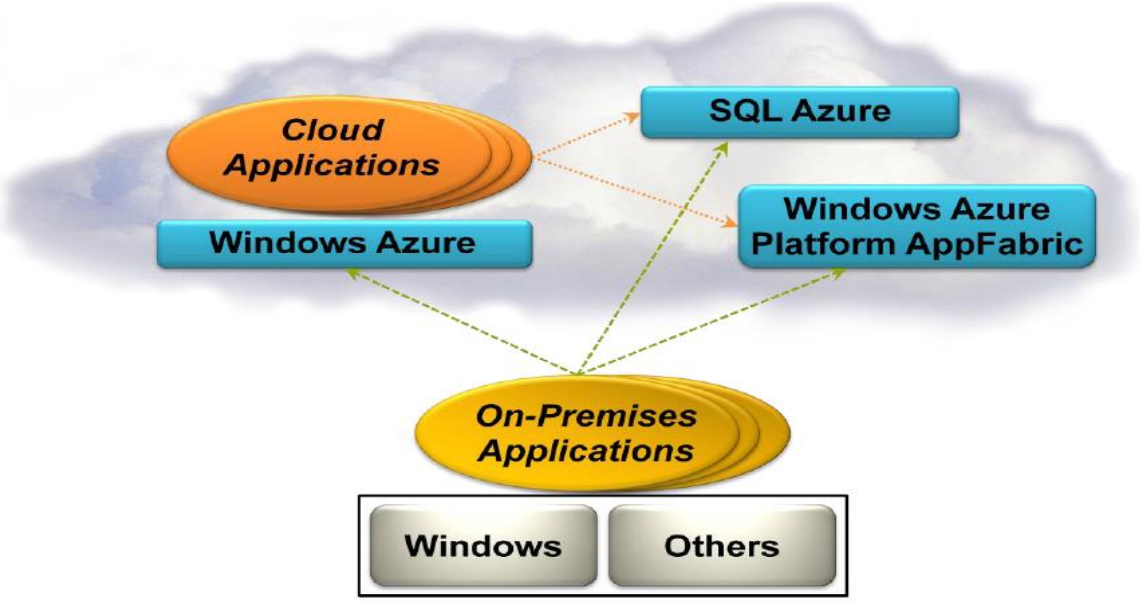
Ongoing maintenance -- Amazon EC2 instances are virtual machines that run in Amazon's cloud. However, they ultimately run on physical hardware which can fail. AWS alerts developers when an instance must be moved due to hardware maintenance. This requires ongoing monitoring.

EC2 vs. S3

Both Amazon EC2 and Amazon S3 are important services that allow developers to maximize use of the AWS cloud. The main difference between Amazon EC2 and S3 is that EC2 is a computing service that allows companies to run servers in the cloud. While S3 is an object storage service

used to store and retrieve data from AWS through the Internet. S3 is like a giant hard drive in the cloud, while EC2 offers CPU and RAM in addition to storage. Many developers use both services for their cloud computing needs.

Case Study – Microsoft Azure



Execution Environment

The Windows Azure execution environment consists of a platform for applications and services hosted within one or more roles. The types of roles you can implement in Windows Azure are:

- **Azure Compute (Web and Worker Roles)**. A Windows Azure application consists of one or more hosted roles running within the Azure data centers. Typically there will be at least one Web role that is exposed for access by users of the application. The application may contain additional roles, including Worker roles that are typically used to perform background processing and support tasks for Web roles. For more detailed information see "Overview of Creating a Hosted Service for Windows Azure" athttp://technet.microsoft.com/en-au/library/gg432976.aspx and "Building an Application that Runs in a Hosted Service" at http://technet.microsoft.com/en-au/library/hh180152.aspx.

- **Virtual Machine (VM role)**. This role allows you to host your own custom instance of the Windows Server 2008 R2 Enterprise or Windows Server 2008 R2 Standard operating system within a Windows Azure data center. For more detailed information see "Creating Applications by Using a VM Role in Windows Azure" athttp://technet.microsoft.com/en-au/library/gg465398.aspx.

Data Management

Windows Azure, SQL Azure, and the associated services provide opportunities for storing and managing data in a range of ways. The following data management services and features are available:

- **Azure Storage:** This provides four core services for persistent and durable data storage in the cloud. The services support a REST interface that can be accessed from within Azure-hosted or on-premises (remote) applications. For information about the REST API, see "Windows Azure Storage Services REST API Reference" athttp://msdn.microsoft.com/en-us/library/dd179355.aspx. The four storage services are:
  - **The Azure Table Service** provides a table-structured storage mechanism based on the familiar rows and columns format, and supports queries for managing the data. It is primarily aimed at scenarios where large volumes of data must be stored, while being easy to access and update. For more detailed information see "Table Service Concepts" at http://msdn.microsoft.com/en-us/library/dd179463.aspx and "Table Service API" at http://msdn.microsoft.com/en-us/library/dd179423.aspx.
  - **The Binary Large Object (BLOB) Service** provides a series of containers aimed at storing text or binary data. It provides both Block BLOB containers for streaming data, and Page BLOB containers for random read/write operations. For more detailed information see "Understanding Block Blobs and Page Blobs" athttp://msdn.microsoft.com/en-us/library/ee691964.aspx and "Blob Service API" at http://msdn.microsoft.com/en-us/library/dd135733.aspx.
  - **The Queue Service** provides a mechanism for reliable, persistent messaging between role instances, such as between a Web role and a Worker role. For more detailed information see "Queue Service Concepts" at http://msdn.microsoft.com/en-us/library/dd179353.aspx and "Queue Service API" athttp://msdn.microsoft.com/en-us/library/dd179363.aspx.
  - Windows Azure Drives provide a mechanism for applications to mount a single volume NTFS VHD as a Page BLOB, and upload and download

VHDs via the BLOB. For more detailed information see "Windows Azure Drive" (PDF) at http://go.microsoft.com/?linkid=9710117.

- **SQL Azure Database:** This is a highly available and scalable cloud database service built on SQL Server technologies, and supports the familiar T-SQL based relational database model. It can be used with applications hosted in Windows Azure, and with other applications running on-premises or hosted elsewhere. For more detailed information see "SQL Azure Database" at http://msdn.microsoft.com/en-us/library/ee336279.aspx.
- **Data Synchronization:** SQL Azure Data Sync is a cloud-based data synchronization service built on Microsoft Sync Framework technologies. It provides bi-directional data synchronization and data management capabilities allowing data to be easily shared between multiple SQL Azure databases and between on-premises and SQL Azure databases. For more detailed information see "Microsoft Sync Framework Developer Center" at http://msdn.microsoft.com/en-us/sync.
- **Caching:** This service provides a distributed, in-memory, low latency and high throughput application cache service that requires no installation or management, and dynamically increases and decreases the cache size automatically as required. It can be used to cache application data, ASP.NET session state information, and for ASP.NET page output caching. For more detailed information see "Caching Service (Windows Azure AppFabric)" at http://msdn.microsoft.com/en-us/library/gg278356.aspx.

Networking Services

Windows Azure provides several networking services that you can take advantage of to maximize performance, implement authentication, and improve manageability of your hosted applications. These services include the following:

- **Content Delivery Network (CDN).** The CDN allows you to cache publicly available static data for applications at strategic locations that are closer (in network delivery terms) to end users. The CDN uses a number of data centers at many locations around the world, which store the data in BLOB storage that has anonymous access. These do not need to be locations where the application is actually running. For more detailed information see "Delivering High-Bandwidth Content with the Windows Azure CDN" athttp://msdn.microsoft.com/en-us/library/ee795176.aspx.
- **Virtual Network Connect.** This service allows you to configure roles of an application running in Windows Azure and computers on your on-premises network so that they appear to be on the same network. It uses a software agent

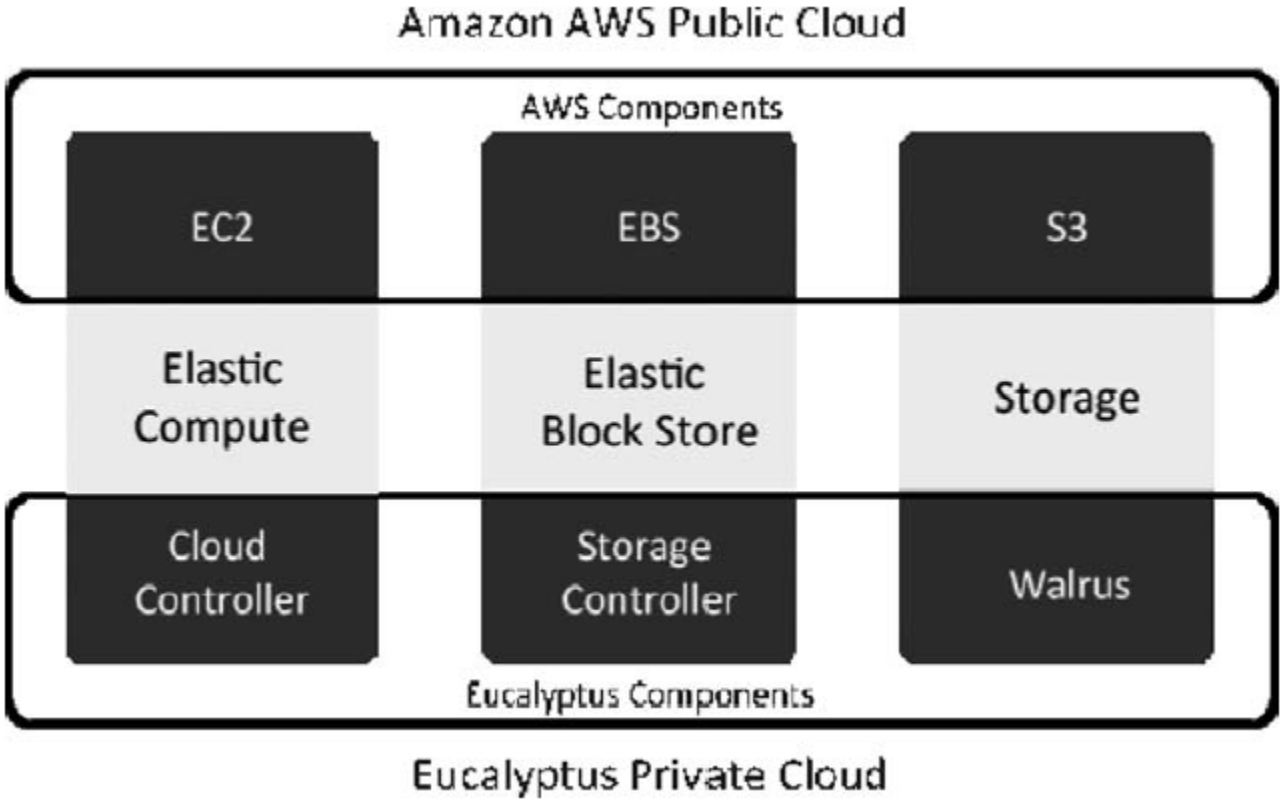
running on the on-premises computer to establish an IPsec-protected connection to the Windows Azure roles in the cloud, and provides the capability to administer, manage, monitor, and debug the roles directly. For more detailed information see "Connecting Local Computers to Windows Azure Roles" at http://msdn.microsoft.com/en-us/library/gg433122.aspx.

- **Virtual Network Traffic Manager.** This is a service that allows you to set up request redirection and load balancing based on three different methods. Typically you will use Traffic Manager to maximize performance by redirecting requests from users to the instance in the closest data center using the Performance method. Alternative load balancing methods available are Failover and Round Robin. For more detailed information see "Windows Azure Traffic Manager" at http://msdn.microsoft.com/en-us/WAZPlatformTrainingCourse_WindowsAzureTrafficManager.
- **Access Control.** This is a standards-based service for identity and access control that makes use of a range of identity providers (IdPs) that can authenticate users. ACS acts as a Security Token Service (STS), or token issuer, and makes it easier to take advantage of federation authentication techniques where user identity is validated in a realm or domain other than that in which the application resides. An example is controlling user access based on an identity verified by an identity provider such as Windows Live ID or Google. For more detailed information see "Access Control Service 2.0" at http://msdn.microsoft.com/en-us/library/gg429786.aspx and "Claims Based Identity & Access Control Guide" at http://claimsid.codeplex.com/.
- **Service Bus.** This provides a secure messaging and data flow capability for distributed and hybrid applications, such as communication between Windows Azure hosted applications and on-premises applications and services, without requiring complex firewall and security infrastructures. It can use a range of communication and messaging protocols and patterns to provide delivery assurance, reliable messaging; can scale to accommodate varying loads; and can be integrated with on-premises BizTalk Server artifacts. For more detailed information see "AppFabric Service Bus" at http://msdn.microsoft.com/en-us/library/ee732537.aspx

EUCALYPTUS—OPEN SOURCE SOFTWARE SUPPORTING - CLOUD COMPUTING

A popular way of integrating public and private clouds is using Eucalyptus. Eucalyptus is an open source software platform that implements IaaS-style cloud computing using the Linux-based infrastructure found in many modern data centers. While it can be deployed solely for private clouds, because it is interface-compatible with Amazon's AWS, it is possible to move workloads between AWS and the data center without code modification.

Eucalyptus for hybrid public and private clouds.



Many other cloud vendors support Eucalyptus, so today, it is the most portable option available. Eucalyptus also works with most of the currently available Linux distributions, including Ubuntu, Red Hat Enterprise Linux (RHEL), CentOS, SUSE Linux Enterprise Server (SLES), openSUSE, Debian, and Fedora. Importantly, Eucalyptus can use a variety of virtualization technologies, including VMware, Xen, and KVM, to implement the cloud abstractions it supports.

Eucalyptus's Walrus is an S3-compatible implementation of cloud storage. It is well-described in *The Eucalyptus Open-source Cloud-computing System*.

The Ubuntu Enterprise Cloud (UEC) is powered by Eucalyptus and brings an Amazon EC2-like infrastructure inside the firewall.12 It appears that the recently announced Nimbula, which supports private versions of EC2, is similar.

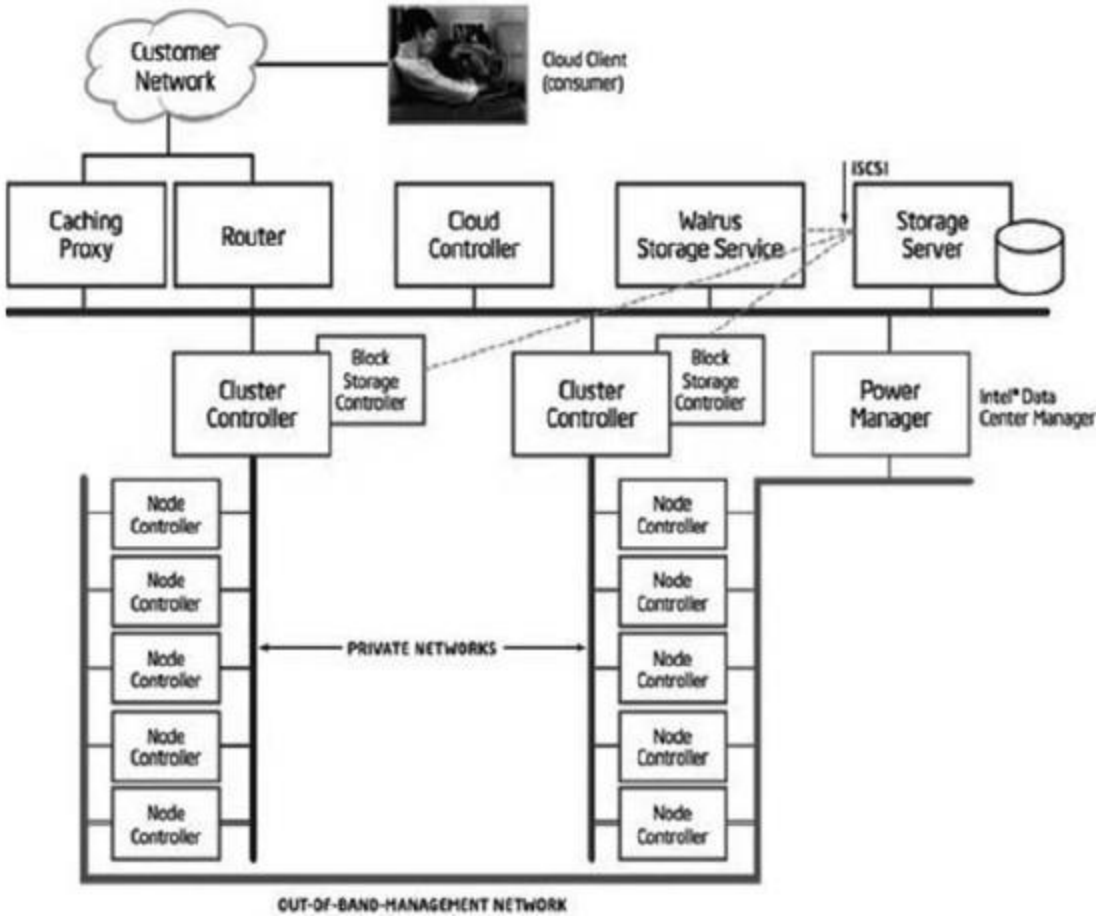
Ubuntu UEC is open source, with commercial support available from Canonical Ltd., a company founded (and funded) by South African entrepreneur Mark Shuttleworth (formerly the official maintainer of Debian, a version of Linux, and founder of Thawte Consulting) for the promotion of free software projects. Canonical is registered in the Isle of Man (part of the Channel Islands), a favorable tax jurisdiction, and employs staff around the world, as well as in its main offices in London. Ubuntu JeOS is an efficient variant of Ubuntu configured specifically for virtual appliances.

Eucalyptus Features and Benefits

The most attractive features of Eucalyptus are:

- Is compatible with Amazon AWS (EC2, S3, and EBS)
- Includes Walrus, an Amazon S3 interface-compatible storage manager
- Has added support for elastic IP assignment
- Has a Web-based interface for cloud configuration
- Provides image registration and image attribute manipulation
- Provides configurable scheduling policies and service level agreements (SLAs)
- Supports multiple hypervisor technologies within the same cloud

The benefits of Eucalyptus include:



- The ability to build a private cloud that can "cloud-burst" into Amazon AWS
- Easy deployment on all types of legacy hardware and software
- Leveraging of the development strength of a worldwide user community
- Compatibility with multiple distributions of Linux, including support for the commercial Linux distributions, Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES)

Eucalyptus Enterprise Edition 2.0 was built on the core Eucalyptus open source platform, with additional functionality designed to optimize the building and deploying of massively scalable, high performance private clouds in the enterprise. The latest release adds support for Windows Server 2003 and 2008 and Windows 7 virtual machines. (Previously, only Linux images were supported). Other changes include new accounting and user group management capabilities, allowing administrators to easily define groups of users and allocate different levels of access based on a group's needs.

The benefits of Ubuntu Enterprise Cloud (UEC) include:

- It incorporates Ubuntu 9.04 Server Edition (April 2009); an enhanced version of Eucalyptus that uses the KVM hypervisor was integrated into the distribution. This allows any user to deploy a cloud that matches the same API employed by AWS.

- Official Ubuntu images have been released that work both on AWS (a Xen-based hypervisor system) and a UEC cloud (a KVMbased hypervisor system)