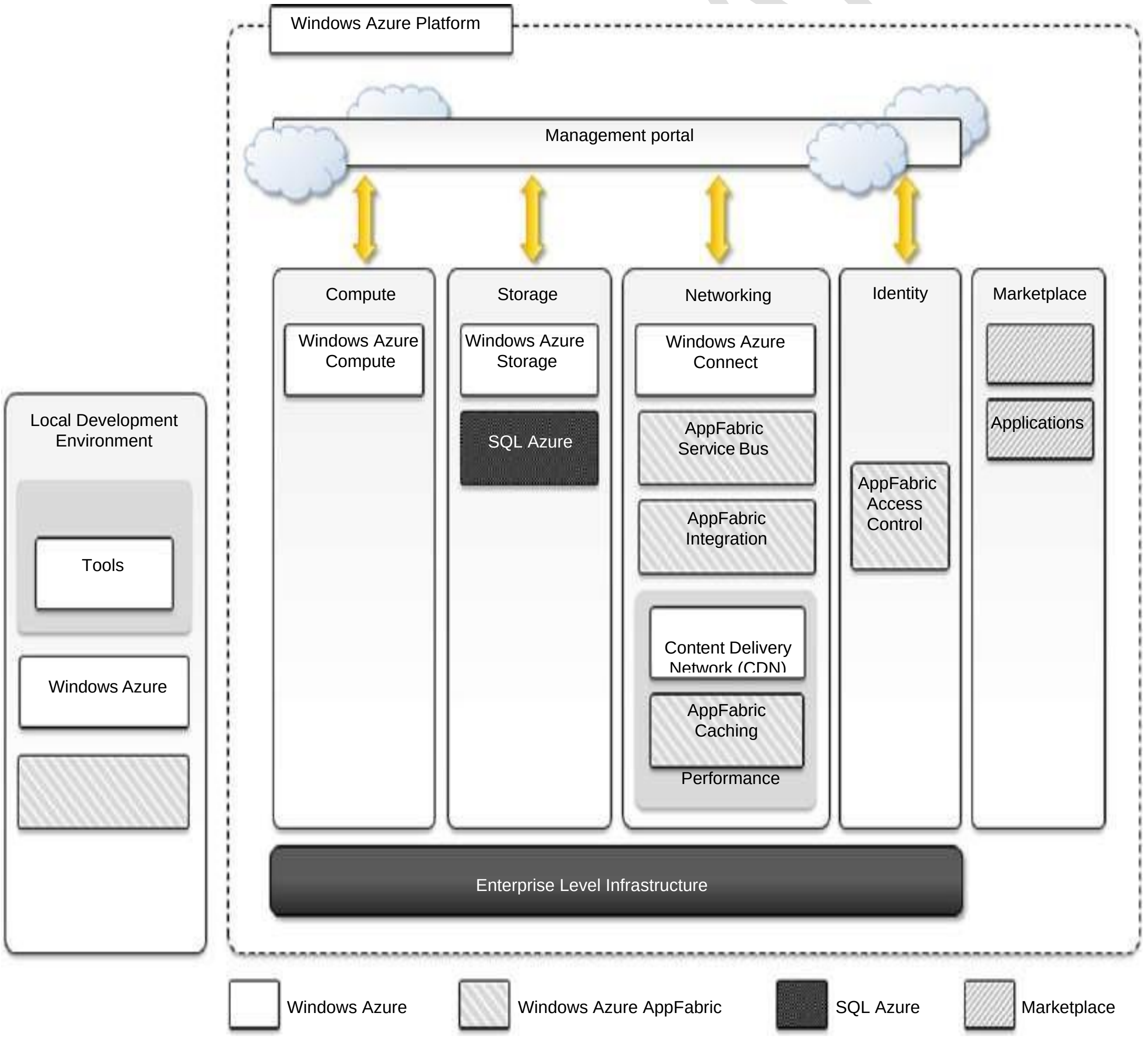
CLOUD COMPUTING- MODULE 5

Microsoft Azure

Microsoft Windows Azure is a cloud operating system built on top of Microsoft data centers' infra- structure and provides developers with a collection of services for building applications with cloud technology. Services range from compute, storage, and networking to application connectivity, access control, and business intelligence.. Any application that is built on the Microsoft technology can be scaled using the Azure platform.

Microsoft Windows Azure Platform Architecture

Figure provides an overview of services provided by Azure. These services can be managed and controlled through the Windows Azure Management Portal, which acts as an administrative console for all the services offered by the Azure platform. In this section, we present the core features of the major services available with Azure. The Windows Azure platform is made up of a foundation layer and a set of developer services that can be used to build scalable applications. These services cover compute, storage, networking, and identity management, which are tied together by middleware called AppFabric. This scalable com- puting environment is hosted within Microsoft datacenters and accessible through the Windows Azure Management Portal. Alternatively, developers can recreate a Windows Azure environment (with limited capabilities) on their own machines for development and testing purposes. In this sec- tion, we provide an overview of the Azure middleware and its services.



Compute services

Compute services are the core components of Microsoft Windows Azure, and they are delivered by means of the abstraction of roles. A role is a runtime environment that is customized for a specific compute task. Roles are managed by the Azure operating system and instantiated on demand in order to address surges in application demand. Currently, there are three different roles: Web role, Worker role, and Virtual Machine (VM) role.

Web role

The Web role is designed to implement scalable Web applications. Web roles represent the units of deployment of Web applications within the Azure infrastructure. They are hosted on the IIS 7 Web Server, which is a component of the infrastructure that supports Azure. When Azure detects peak loads in the request made to a given application, it instantiates multiple Web roles for that application and distributes the load among them by means of a load balancer. Since version 3.5, the .NET technology natively supports Web roles; developers can directly develop their applications in Visual Studio, test them locally, and upload to Azure.

Worker role

Worker roles are designed to host general compute services on Azure. They can be used to quickly provide compute power or to host services that do not communicate with the external world through HTTP. A common practice for Worker roles is to use them to provide background processing for Web applications developed with Web roles. Developing a worker role is like a developing a service. Compared to a Web role whose computation is triggered by the interaction with an HTTP client (i.e., a browser), a Worker role runs continuously from the creation of its instance until it is shut down. For example, Worker roles can be used to host Tomcat and serve JSP-based applications.

Virtual machine role

The Virtual Machine role allows developers to fully control the computing stack of their compute service by defining a custom image of the Windows Server 2008 R2 operating system and all the service stack required by their applications. The Virtual Machine role is based on the Windows Hyper-V virtualization technology which is natively integrated in the Windows server technology at the base of Azure. Developers can image a Windows server installation complete with all the required applications and components, save it into a Virtual Hard Disk (VHD).

Storage services

Compute resources are equipped with local storage in the form of a directory on the local file sys- tem that can be used to temporarily store information that is useful for the current execution cycle of a role. If the role is restarted and activated on a different physical machine, this information is lost. Windows Azure provides different types of storage solutions that complement compute services with a more durable and redundant option compared to local storage. Compared to local storage, these services can be accessed by multiple clients at the same time and from everywhere, thus becoming a general solution for storage.

Blobs

Azure allows storing large amount of data in the form of binary large objects (BLOBs) by means of the blobs service. This service is optimal to store large text or binary files. Two types of blobs are available:

- Block blobs. Block blobs are composed of blocks and are optimized for sequential access; therefore they are appropriate for media streaming. Currently, blocks are of 4 MB, and a single block blob can reach 200 GB in dimension.
- Page blobs. Page blobs are made of pages that are identified by an offset from the beginning of the blob. A page blob can be split into multiple pages or constituted of a single page. This type of blob is optimized for random access and can be used to host data different from streaming. Currently, the maximum dimension of a page blob can be 1 TB.

Azure drive

Page blobs can be used to store an entire file system in the form of a single Virtual Hard Drive (VHD) file. This can then be mounted as a part of the NTFS file system by Azure compute resources, thus providing persistent and durable storage. A page blob mounted as part of an NTFS tree is called an Azure Drive.

Tables

Tables constitute a semistructured storage solution, allowing users to store information in the form of entities with a collection of properties. Entities are stored as rows in the table and are identified by a key, which also constitutes the unique index built for the table. Users can insert, update, delete, and select a subset of the rows stored in the table. Unlike SQL tables, there are no schema enforcing constraints on the properties of entities and there is no facility for representing relation- ships among entities. For this reason, tables are more similar to spreadsheets rather than SQL tables. The service is designed to handle large amounts of data and queries returning huge result sets.

Queues

Queue storage allows applications to communicate by exchanging messages through durable queues, thus avoiding lost or unprocessed messages. Applications enter messages into a queue, and other applications can read them in a first-in, first-out (FIFO) style.

To ensure that messages get processed, when an application reads a message it is marked as invisible; hence it will not be available to other clients. Once the application has completed processing the message, it needs to explicitly delete the message from the queue. This two-phase process ensures that messages get processed before they are removed from the queue, and the client failures do not prevent messages from being processed. At the same time, this is also a reason that the queue does not enforce a strict FIFO model.

Microsoft Hyper-V

Hyper-V is an infrastructure virtualization solution developed by Microsoft for server virtualization. As the name recalls, it uses a hypervisor-based approach to hardware virtualization, which leverages several techniques to support a variety of guest operating systems. Hyper-V is currently shipped as a component of Windows Server 2008 R2 that installs the hypervisor as a role within the server.

Hyper-V supports multiple and concurrent execution of guest operating systems by means of partitions. A partition is a completely isolated environment in which an operating system is installed and run.

Despite its straightforward installation as a component of the host operating system, Hyper-V takes control of the hardware, and the host operating system becomes a virtual machine instance with special privileges, called the parent partition. The parent partition (also called the root partition) is the only one that has direct access to the hardware. It runs the virtualization stack, hosts all the drivers required to configure guest operating systems, and creates child partitions through the hypervisor. Child partitions are used to host guest operating systems and do not have access to the underlying hardware, but their interaction with it is controlled by either the parent partition or the hypervisor itself.

Azure Virtual Machine

Microsoft's cloud service Azure uses a special operating system called as Microsoft Azure and runs a layer called as 'Fabric Controller' over it which manages computing and storage resources of clusters at Microsoft's data center.

Users can create virtual machines over this layer using Azure Virtual Machines to use scalable computing infrastructure on-demand. Fabric Controller manages the scaling.

The operating system of this is also called as "cloud layer". It is built over Windows server system and combines a customized version of Hyper-V being known as the Microsoft Azure Hypervisor.

Microsoft offers the options of choosing between Windows and Linux operating systems at the time of launching the virtual machines in Azure. Azure provides per-minute billing system for their virtual machines. Azure Virtual Machine with Windows OS runs all Microsoft enterprise applications with outstanding ease and also supports a full range of Linux distributions like Red Hat, Ubuntu and SUSE.

Virtual machines in Azure are offered in two categories as basic tier and standard tier. The basic tier provides economical option for development or testing purpose where features like load balancing or scaling are not required. Standard tier provides all of the advanced features. Virtual machines of different sizes are offered under several series like A, D, DS, G etc.

Consumers can launch machines from the list of available choices to fulfill various requirements. Pre-configured machines are designed to meet the needs of compute-intensive application, network-intensive application, better local disk performance, high memory demand and so on.

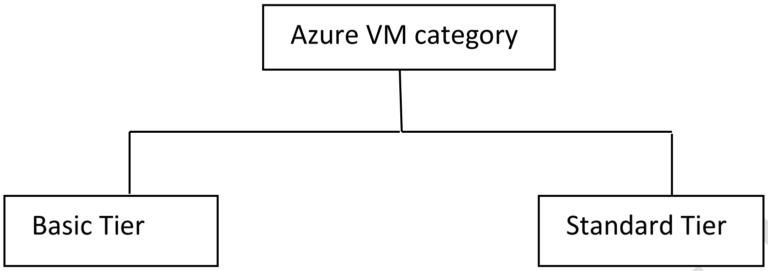


Fig: Azure virtual machine tiers

# Google Cloud

Google is presently one among the leaders in the cloud computing market and offers a vast collection of web-based applications as cloud services. Presently Google divides its cloud services under the following categories as Compute, Storage, Networking, Database, Services and Management.

Google's IaaS Offerings

Google's Infrastructure-as-a-Service (IaaS) offering include two popular services as Google Compute Engine and Google Cloud Storage.

Google Compute Engine (GCE)

Google ventured into Infrastructure-as-a-Service (IaaS) after gaining success in SaaS and PaaS markets and the GCE is the outcome of that effort. Developers can run large-scale distributed applications on virtual machines hosted on Google's infrastructure by using GCE. GCE uses KVM (Kernel-based Virtual Machine) as the hypervisor and supports guest machines to run Linux as operating system. Instances can use different popular distributions of Linux using a standard image provided by Google Compute Engine or a modified version of one of these images as customized by customer.

Google provides both pre-defined and custom-configured machines. the available machine categories offered by GCE are as following:

- Standard Machine,
- High Memory Machine,
- High CPU Machine,
- Shared Core Machine, and
- Custom Machine Types.

Shared-core machine type becomes cost-effective where small amount of resources are required. The instance types under each machine category specify the number of virtual processors and size of memory among other features.

When the pre-defined machine types do not match with requirements, then consumers have the option of building custom-defined machine. There one can specify the memory and CPU needs.

Machines under GCE offering are charged per minutes basis after first 10 minutes. For usage of less than 10 minutes, consumers need to pay the charge for 10 minutes. Apart from machine instances, Google separately charges its customers for network usages, load balancing, persistence disk usages etc.

Google Cloud Storage

Google Cloud Storage is an Infrastructure-as-a-Service (IaaS) offering from Google and comparable to Amazon S3. The storage can be accessed through simple APIs or can be used with Google Compute Engine, App Engine and so on.

Data in Google Cloud Storage are stored as objects. Each object has two parts like data and metadata. These objects are stored into containers called as *buckets*. Buckets are used to organize data and to specify access control on that data. Buckets are stored as part of a project. One bucket cannot be shared among multiple projects. Unlike directories or folders, buckets in a project cannot be organized in nested manner. There is no limit on the number of buckets a project can contain. A consumer can be part of one project or multiple projects.

Google's PaaS Offering

The PaaS facility offered by Google is called as Google App Engine (GAE). Google also offers a number of database services as part of their cloud offering.

Google App Engine

App Engine (GAE) is Google's offering as Platform-as–a-Service (PaaS) facility. On GAE, applications can be developed in highlevel programming languages by using Google App Engine Framework. The frameworks ease the development efforts required to build an application. GAE allows developers to create and deploy web applications without worrying about necessary managing tasks to run their applications. GAE also provides facility for cloud-based web hosting services on Google's infrastructure.

Google App Engine provides several high-level programming language run-times like Java runtime, Python runtime and PHP runtime. GAE provides a persistent storage facility with it and applications deployed over App Engine are easy to scale when traffic or data storage needs the growth. GAE also has auto load balancing feature.

Like other cloud platform services, there are some limitations from developer's point of view though. Applications developed on App Engine must comply with Google's infrastructure. User's programs communicate with GAE APIs and uses objects and properties from the App Engine framework. This narrows the range of application types that can be developed and run on GAE. Again it may become difficult to port application to another platform since it may be dependent on some unique GAE feature.

Apart from this limitation, the Google App Engine provides a low-cost and flexible option for developers to build applications that can run on the Google's cloud infrastructure. There are no set-up costs and consumers only pay for what they use. Moreover, to encourage to write applications using GAE, Google allows free application development and deployment up to a certain level of resource consumption for the developers.

Database Services

1. **Cloud SQL**

Google Cloud SQL offers a fully-managed database solution MySQL which is the most popular open-source database. This Database-as-a-Service (DBaaS) allows users to use a relational database that is highly scalable. Users can concentrate more on database design and application development since other database management activities are managed by Google.

Google Cloud SQL integrates well with Google App Engine and very easy to use. It provides most of the features of MySQL and is ideal for small or medium size applications. It also presents a set of tools enabling users to easily move data in and out of the cloud.

2. **Cloud Datastore**

Google Cloud Datastore is a fully-managed, schema-less NoSQL data store for storing non relational data. Cloud Data store is very easy-to-use and supports SQL-like queries being called as GQL. The Datastore is a NoSQL key/value store, where users can store data as key/value pairs. Cloud Datastore also supports ACID transactions for concurrency control. Although Cloud SQL and Datastore are totally different, many applications require to use both of them together.

3. **BigQuery**

BigQuery is Google's offering that provides facility of analyzing *Big Data* in the cloud. *Big Data* refers to a massive volume of both structured and unstructured data which is so large that it becomes difficult to process it using traditional database management tools and software techniques. BigQuery enables for interactive analysis of massive datasets having billions of rows in real time using SQL-like queries.

Google's SaaS Offerings

Among many services they offer, cloud-based word processing (Google Docs), email service (Gmail), or calendar/scheduling service (Google Calendar) are to mention a few. These SaaS facilities from Google are offered under 'G Suite' and are popularly known as 'Google Apps'.

1. *Gmail*

   This is a popular SaaS offering from Google. Gmail is offered under the communicate section of 'G Suite' (formerly known as 'Google Apps for Work'). Other offerings under this communicate suite include Calendar and Google+. Gmail is a free service and apart from personal use Gmail offers services for enterprise users too.

2. *Docs*

   Google Docs is a SaaS offering for managing documents. It comes under the collaborative application suit offered by Google which resembles traditional office suites and provides functionalities to create text documents, spreadsheets and presentations over the cloud using any web-browser. The features of Google's collaborative suite are almost similar to those present in traditional Microsoft Office suite. Like other offering under this collaborative suite (namely Sheets, Forms, Slides), Google Docs is also free for anyone to use and allows multiple users to edit and update same document which is very useful feature in this age of global collaboration.

3. *Google Drive*

   This is Google's SaaS offerings for general users to store file. It is basically built for the users of 'G Suite' to enable them for storing data in cloud seamlessly. Users can access their stored data from anywhere and share it with other Google Drive users. It acts as the default storage for Gmail or Google Docs. It also provides easy and simple way of transferring data from user's own computer to the cloud.

# Amazon Web Services (AWS)

Amazon Web Services (AWS) is a platform that allows the development of flexible applications by providing solutions for elastic infrastructure scalability, messaging, and data storage. The platform is accessible through SOAP or RESTful Web service interfaces and provides a Web-based console where users can handle administration and monitoring of the resources required, as well as their expenses computed on a pay-as-you-go basis.

Compute services

Compute services constitute the fundamental element of cloud computing systems. The fundamental service in this space is Amazon EC2, which delivers an IaaS solution that has served as a reference model for several offerings from other vendors in the same market segment. Amazon EC2 allows deploying servers in the form of virtual machines created as instances of a specific image. Images come with a preinstalled operating system and a software stack, and instances can be con- figured for memory, number of processors, and storage. Users are provided with credentials to remotely access the instance and further configure or install software if needed.

Amazon machine images

Amazon Machine Images (AMIs) are templates from which it is possible to create a virtual machine. They are stored in Amazon S3 and identified by a unique identifier in the form of ami-xxxxxx anda manifest XML file. An AMI contains a physical file system layout with a predefined operating system installed. These are specified by the Amazon Ramdisk Image (ARI, id: ari-yyyyyy) and the Amazon Kernel Image (AKI, id: aki-zzzzzz), which are part of the configuration of the template. AMIs are either created from scratch or "bundled" from existing EC2 instances. A common prac- tice is to prepare new AMIs to create an instance from a preexisting AMI, log into it once it is booted and running, and install all the software needed. Using the tools provided by Amazon, we can convert the instance into a new image. Once an AMI is created, it is stored in an S3 bucket and the user can decide whether to make it available to other users or keep it for personal use. Finally, it is also possible to associate a product code with a given AMI, thus allowing the owner of the AMI to get revenue every time this AMI is used to create EC2 instances.

EC2 instances

EC2 instances represent virtual machines. They are created using AMI as templates, which are specialized by selecting the number of cores, their computing power, and the installed memory. The processing power is expressed in terms of virtual cores and EC2 Compute Units (ECUs). The ECU is a measure of the computing power of a virtual core; it is used to express a predictable quantity of real CPU power that is allocated to an instance. By using compute units instead of real frequency values, Amazon can change over time the mapping of such units to the underlying real amount of computing power allocated, thus keeping the performance of EC2 instances consistent with standards set by the times.

Some currently available configurations for EC2 instances are:

- Standard instances. This class offers a set of configurations that are suitable for most applications. EC2 provides three different categories of increasing computing power, storage, and memory.
- Micro instances. This class is suitable for those applications that consume a limited amount of computing power and memory and occasionally need bursts in CPU cycles to process surges in the workload. Micro instances can be used for small Web applications with limited traffic.
- High-memory instances. This class targets applications that need to process huge workloads and require large amounts of memory. Three-tier Web applications characterized by high traffic are the target profile. Three categories of increasing memory and CPU are available, with memory proportionally larger than computing power.
- High-CPU instances. This class targets compute-intensive applications. Two configurations are available where computing power proportionally increases more than memory.
- Cluster Compute instances. This class is used to provide virtual cluster services. Instances in this category are characterized by high CPU compute power and large memory and an extremely high I/O and network performance, which makes it suitable for HPC applications.
- Cluster GPU instances. This class provides instances featuring graphic processing units (GPUs) and high compute power, large memory, and extremely high I/O and network performance.

EC2 instances are priced hourly according to the category they belong to. At the beginning of every hour of usage, the user will be charged the cost of the entire hour. The hourly expense charged for one instance is constant. Instance owners are responsible for providing their own backup strategies, since there is no guarantee that the instance will run for the entire hour.

Advanced compute services

Amazon Web Services provide more sophisticated services that allow the easy packaging and deploying of applications and a computing platform that supports the execution of MapReduce-based applications.

1. *AWS CloudFormation*

AWS CloudFormation constitutes an extension of the simple deployment model that characterizes EC2 instances. CloudFormation introduces the concepts of templates, which are JSON formatted text files that describe the resources needed to run an application or a service in EC2 together with the relations between them. CloudFormation allows easily and explicitly linking EC2 instances together and introducing dependencies among them. Templates provide a simple and declarative way to build complex systems and integrate EC2 instances with other AWS services such as S3, SimpleDB, SQS, SNS, Route 53, Elastic Beanstalk, and others.

2. *AWS elastic beanstalk*

AWS Elastic Beanstalk constitutes a simple and easy way to package applications and deploy them on the AWS Cloud. This service simplifies the process of provisioning instances and deploying application code and provides appropriate access to them. Currently, this service is available only for Web applications developed with the Java/Tomcat technology stack. Developers can conve- niently package their Web application into a WAR file and use Beanstalk to automate its deploy- ment on the AWS Cloud.

3. *Amazon elastic MapReduce*

Amazon Elastic MapReduce provides AWS users with a cloud computing platform for MapReduce applications. It utilizes Hadoop as the MapReduce engine, deployed on a virtual infrastructure com- posed of EC2 instances, and uses Amazon S3 for storage needs.

Apart from supporting all the application stack connected to Hadoop (Pig, Hive, etc.), Elastic MapReduce introduces elasticity and allows users to dynamically size the Hadoop cluster according to their needs, as well as select the appropriate configuration of EC2 instances to compose the clus- ter (Small, High-Memory, High-CPU, Cluster Compute, and Cluster GPU).

Storage services

AWS provides a collection of services for data storage and information management. The core ser- vice in this area is represented by Amazon Simple Storage Service (S3). This is a distributed object store that allows users to store information in different formats. The core components of S3 are two: buckets and objects. Buckets represent virtual containers in which to store objects; objects rep- resent the content that is actually stored. Objects can also be enriched with metadata that can be used to tag the stored content with additional information.

Key features of Simple Storage Service are:

- *The storage is organized in a two-level hierarchy*. S3 organizes its storage space into buckets that cannot be further partitioned. This means that it is not possible to create directories or other kinds of physical groupings for objects stored in a bucket. Despite this fact, there are few limitations in naming objects, and this allows users to simulate directories and create logical groupings.
- *Stored objects cannot be manipulated like standard files*. S3 has been designed to essentially provide storage for objects that will not change over time. Therefore, it does not allow renaming, modifying, or relocating an object. Once an object has been added to a bucket, its content and position is immutable, and the only way to change it is to remove the object from the store and add it again.
- *Content is not immediately available to users*. The main design goal of S3 is to provide an eventually consistent data store. As a result, because it is a large distributed storage facility, changes are not immediately reflected. For instance, S3 uses replication to provide redundancy and efficiently serve objects across the globe; this practice introduces latencies when adding objects to the store—especially large ones—which are not available instantly across the entire globe.
- *Requests will occasionally fail.* Due to the large distributed infrastructure being managed, requests for object may occasionally fail. Under certain conditions, S3 can decide to drop a request by returning an internal server error. Therefore, it is expected to have a small failure rate during day-to-day operations, which is generally not identified as a persistent failure.

Resource naming

Buckets, objects, and attached metadata are made accessible through a REST interface. Therefore, they are represented by uniform resource identifiers (URIs) under the s3.amazonaws.com domain. All the operations are then performed by expressing the entity they are directed to in the form of a request for a URI.

Amazon offers three different ways of addressing a bucket:

- Canonical form: http://s3.amazonaws.com/bukect_name/. The bucket name is expressed as a path component of the domain name s3.amazonaws.com. This is the naming convention that has less restriction in terms of allowed characters, since all the characters that are allowed for a path component can be used.
- Subdomain form: http://bucketname.s3.amazon.com/. Alternatively, it is also possible to reference a bucket as a subdomain of s3.amazonaws.com. To express a bucket name in this form, the name has to do all of the following:
  - Be between 3 and 63 characters long
  - Contain only letters, numbers, periods, and dashes
  - Start with a letter or a number
  - Contain at least one letter
  - Have no fragments between periods that start with a dash or end with a dash or that are empty strings

This form is equivalent to the previous one when it can be used, but it is the one to be preferred since it works more effectively for all the geographical locations serving resources stored in S3.

•Virtual hosting form: http://bucket-name.com/. Amazon also allows referencing of its resources with custom URLs. This is accomplished by entering a CNAME record into the DNS that points to the subdomain form of the bucket URI.

Buckets

A bucket is a container of objects. It can be thought of as a virtual drive hosted on the S3 distributed storage, which provides users with a flat store to which they can add objects. Buckets are top- level elements of the S3 storage architecture and do not support nesting. That is, it is not possible to create “subbuckets” or other kinds of physical divisions.

A bucket is located in a specific geographic location and eventually replicated for fault tolerance and better content distribution. Users can select the location at which to create buckets, which by default are created in Amazon’s U.S. data centers. Once a bucket is created, all the objects that belong to the bucket will be stored in the same availability zone of the bucket. Users create a bucket by sending a PUT request to http://s3.amazonaws.com/ with the name of the bucket and, if they want to specify the availability zone, additional information about the preferred location. The content of a bucket can be listed by sending a GET request specifying the name of the bucket. Once created, the bucket cannot be renamed or relocated. If it is necessary to do so, the bucket needs to be deleted and recreated. The deletion of a bucket is performed by a DELETE request, which can be successful if and only if the bucket is empty.

Objects and metadata

Objects constitute the content elements stored in S3. Users either store files or push to the S3 text stream representing the object’s content. An object is identified by a name that needs to be unique within the bucket in which the content is stored. The name cannot be longer than 1,024 bytes when encoded in UTF-8, and it allows almost any character. Since buckets do not support nesting, even characters normally used as path separators are allowed. This actually compensates for the lack of a structured file system, since directories can be emulated by properly naming objects.

Objects can be tagged with metadata, which are passed as properties of the PUT request. Such properties are retrieved either with a GET request or with a HEAD request, which only returns the object’s metadata without the content. Metadata are both system and user defined: the first ones are used by S3 to control the interaction with the object, whereas the second ones are meaningful to the user, who can store up to 2 KB per metadata property represented by a key-value pair of strings.

Amazon elastic block store

The Amazon Elastic Block Store (EBS) allows AWS users to provide EC2 instances with persistent storage in the form of volumes that can be mounted at instance startup. They accommodate up to 1 TB of space and are accessed through a block device interface, thus allowing users to format them according to the needs of the instance they are connected to (raw storage, file system, or other). The content of an EBS volume survives the instance life cycle and is persisted into S3. EBS volumes can be cloned, used as boot partitions, and constitute durable storage since they rely on S3 and it is possible to take incremental snapshots of their content.

EBS volumes normally reside within the same availability zone of the EC2 instances that will use them to maximize the I/O performance. It is also possible to connect volumes located in differ- ent availability zones. Once mounted as volumes, their content is lazily loaded in the background and according to the request made by the operating system. This reduces the number of I/O requests that go to the network. Volume images cannot be shared among instances, but multiple (separate) active volumes can be created from them. In addition, it is possible to attach multiple volumes to a single instance or create a volume from a given snapshot and modify its size, if the formatted file system allows such an operation.

The expense related to a volume comprises the cost generated by the amount of storage occupied in S3 and by the number of I/O requests performed against the volume. Currently, Amazon charges $0.10/GB/month of allocated storage and $0.10 per 1 million requests made to the volume.

Communication services
Amazon provides facilities to structure and facilitate the communication among existing applications and services residing within the AWS infrastructure. These facilities can be organized into two major categories: virtual networking and messaging.

Virtual networking
Virtual networking comprises a collection of services that allow AWS users to control the connectivity to and between compute and storage services. Amazon Virtual Private Cloud (VPC) and Amazon Direct Connect provide connectivity solutions in terms of infrastructure; Route 53 facili- tates connectivity in terms of naming.
Amazon VPC provides a great degree of flexibility in creating virtual private networks within the Amazon infrastructure and beyond. The service providers prepare either templates covering most of the usual scenarios or a fully customizable network service for advanced configurations. Prepared templates include public subnets, isolated networks, private networks accessing Internet through network address translation (NAT), and hybrid networks including AWS resources and pri- vate resources. Also, it is possible to control connectivity between different services (EC2 instances and S3 buckets) by using the Identity Access Management (IAM) service. During 2011, the cost of Amazon VPC was $0.50 per connection hour.
Amazon Direct Connect allows AWS users to create dedicated networks between the user private network and Amazon Direct Connect locations, called ports. This connection can be further partitioned in multiple logical connections and give access to the public resources hosted on the Amazon infrastructure. The advantage of using Direct Connect versus other solutions is the consistent performance of the connection between the users' premises and the Direct Connect locations. This service is compatible with other services such as EC2, S3, and Amazon VPC and can be used in scenarios requiring high bandwidth between the Amazon network and the outside world.

Messaging
Messaging services constitute the next step in connecting applications by leveraging AWS capabili- ties. The three different types of messaging services offered are Amazon Simple Queue Service (SQS), Amazon Simple Notification Service (SNS), and Amazon Simple Email Service (SES).

Amazon Simple Queue Service (SQS):
Amazon SQS constitutes disconnected model for exchanging messages between applications by means of message queues, hosted within the AWS infrastructure. Using the AWS console or directly the underlying Web service AWS, users can create an unlimited number of message queues and configure them to control their access. Applications can send messages to any queue they have access to. These messages are securely and redundantly stored within the AWS infrastructure for a limited period of time, and they can be accessed by other (authorized) applications. While a mes- sage is being read, it is kept locked to avoid spurious processing from other applications. Such a lock will expire after a given period.

Amazon Simple Notification Service (SNS) :
Amazon SNS provides a publish-subscribe method for connecting heterogeneous applications. With respect to Amazon SQS, where it is necessary to continuously poll a given queue for a new message to process, Amazon SNS allows applications to be notified when new content of interest is available. This feature is accessible through a Web service whereby AWS users can create a topic, which other applications can subscribe to. At any time, applications can publish content on a given topic and subscribers can be automatically notified. The service provides subscribers with different notification models (HTTP/HTTPS, email/email JSON, and SQS).

Amazon Simple Email Service (SES).
Amazon SES provides AWS users with a scalable email service that leverages the AWS infra- structure. Once users are signed up for the service, they have to provide an email that SES will use to send emails on their behalf. To activate the service, SES will send an email to verify the given address and provide the users with the necessary information for the activation. Upon verification, the user is given an SES sandbox to test the service, and he can request access to the production version. Using SES, it is possible to send either SMTP-compliant emails or raw emails by specifying email headers and Multipurpose Internet Mail Extension (MIME) types. Emails are queued for

delivery, and the users are notified of any failed delivery. SES also provides a wide range of statistics that help users to improve their email campaigns for effective communication with customers.