**Cloud computing**

Delivery of computing services from a remote location over a network. Cloud model has five essential characteristics:

1. On-demand self service
2. Broad network access
3. Resource pooling
4. Rapid elasticity
5. Measured service

**Cloud computing models**

* IaaS (Infrastructure as a Service – processing, network connectivity, storage etc.)
* PaaS (Platform as a Service function calls to perform tasks, e.g. backup files etc.)
* SaaS (Software as a Service – application level access to the user)

**Cloud deployment models**

* On premises (Private, only authorized users can use it)
* Hybrid (Best of both worlds)
* Public cloud (Anyone can use it)

**AWS**

* In March 2006, Amazon Web Services (AWS) began offering IT infrastructure services to businesses in the form of web services – now commonly called as cloud computing.
* Cloud computing provides a simple way to access servers, storage, databases and a broad set of application services over the internet.
* S3 service (Simple Storage Service) was the 1st service offered by Amazon to general public. An S3 object can be as large as 5 terabytes.
* Using AWS to build your application is like purchasing electricity from a power company instead of running your own generator. It’s a pay-as-you-go model that scales with usage
* To start use AWS, you need nothing more than an email address and a credit card
* Xen Hypervisor is the software foundation of Amazon’s virtualization environment.
* Amazon makes AWS highly resilient by implementing resource redundancy

**AWS services**

* Administrative services : manage your AWS resources
* Application services : add functionality to your application
* Foundation services : deploy IT infrastructure in the cloud

**The AWS cloud infrastructure is built on the following**

* Regions (a physical location of the world where they have multiple AZs)
* Availability Zones (consists of one or more discrete data centers each with redundant power, networking and connectivity)

**AWS Management Console:** A simple and intuitive user interface to access and manage Amazon Web Services

**Terminologies**

|  |  |
| --- | --- |
| Regions and Availability zones | The AWS Cloud infrastructure is built around Regions and Availability Zones (AZs). A Region is a physical location in the world where we have multiple AZs. AZs consist of one or more discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities.  Each AWS Region is designed to be completely isolated from the other Amazon Regions. This achieves the greatest possible fault tolerance and stability. Each AZ is isolated, but the AZs in a Region are connected through low-latency links. |
| Amazon EC2 instance | Amazon Elastic Compute Cloud is a service that provides secure, resizable compute capacity in the cloud.  An EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure. |
| Amazon CloudFront | A web service that speeds up the distribution of your static and dynamic web content |
|  |  |
| Edge locations | Worldwide network of data centers through which CloudFront delivers the content. When user requests a content that is being served by CloudFront. User is routed to the edge location that provides the lowest latency (time delay) so that content is delivered with the best possible performance. Edge locations are collections of servers in geographically dispersed data centers where CloudFront caches copies of your objects. |
| Origin server | An Amazon S3 bucket or your own HTTP server from which CloudFront gets your files to be distributed to CloudFront edge locations all over the world. An origin server stores the original, definitive version of your objects. |
| CloudFront Distribution | Tells CloudFront which origin servers to get your files from when users request the files through your web site or application. CloudFront sends your distribution’s configuration (but not your content) to all of its edge locations. |
|  |  |
|  |  |
|  |  |

**DNS (Domain Naming System)**

Communication using the Internet Protocol (IP) requires each endpoint to be identified by a numeric IP address. IP addresses are needed to actually deliver a message to a receiver, just like a numeric phone number is required to place a call.

The domain name system is a distributed way to share DomainName-to-IP associations instead of requiring each computer to synchronize a hosts file. A name server publishes the IP address for a domain and provides a single location to update when an IP changes.

**How DNS works**

Domain names are issued by registrars and name servers provide the details about which IP is associated with the domain. Domain registrations are valid for 1 to 10 years and registrars typically provide name servers to their customers as well.   
  
Here’s a step-by-step description of how DNS works:

1. A user enters a URL like www.example.com in their browser.
2. The computer performs a DNS request to resolve example.com to an actual IP address.
3. The DNS request starts with a set of authoritative name servers and determines if any have the name server for example.com. Name servers have separate “zones of authority” and the request may be routed to a different zone.
4. The computer follows the hierarchy until it eventually reaches the name server for example.com. For a web request, the computer requests the record for “www” and receives the associated IP address (10.20.30.40).
5. Using this IP address the computer can communicate with the server and load the content of http://www.example.com.

Instead of performing a full DNS lookup on every request, a computer may cache the results for a time period defined by the name server. A typical DNS entry looks like this:

**www.example.com A 3600 10.20.30.40** (A maps to IPv4 and AAAA to IPv6)

This entry indicates that the result should be cached for 3600 seconds (1 hour). Administrators will often change the timeout value to a low number (minutes) before they change an IP address.

TLD: Top Level Domain (com, edu, org etc.)

SLD: Second Level Domain (Ubuntu.com, example.com etc.)

FQDN: Fully Qualified Domain Name (mail.google.com)

**Amazon Route 53**

Amazon Route 53 is a highly available and scalable **Domain Name System** (DNS) web service. It is designed for developers and corporates to route the end users to Internet applications by translating human readable names like **www.mydomain.com**, into the numeric IP addresses like **192.0.2.1** that computers use to connect to each other.

Amazon Route 53 performs three main functions:

• **Domain registration**

Amazon Route 53 lets you register domain names such as exelaproducts.com.

• **Domain Name System (DNS) service**

Amazon Route 53 translates friendly domains names like www.exelaproducts.com into IP addresses like 192.0.2.1. Amazon Route 53 responds to DNS queries using a global network of authoritative DNS servers, which reduces latency.

• **Health checking**

Amazon Route 53 sends automated requests over the Internet to your application to verify that it's reachable, available, and functional.

You can use any combination of these functions. For example, you can use Amazon Route 53 as both your registrar and your DNS service, or you can use Amazon Route 53 as the DNS service for a domain that you registered with another domain registrar.

**Amazon S3**

Amazon S3 is cloud storage for the internet. To upload your data (photos, videos, documents etc.), you first create a bucket in one of the AWS Regions. You can then upload any number of objects to the bucket.

In terms of implementation, buckets and objects are resources, and Amazon S3 provides APIs for you to manage them. For example, you can create a bucket and upload objects using the Amazon S3 API. You can also use the Amazon S3 console to perform these operations. The console uses the Amazon S3 APIs to send requests to Amazon S3.

Amazon S3 (Simple Storage Service) is a scalable, high-speed, low-cost web-based service designed for online backup and archiving of data and application programs. It allows to upload, store, and download any type of files up to 5 GB in size.

This service allows the subscribers to access the same systems that Amazon uses to run its own web sites. The subscriber has control over the accessibility of data, i.e. privately/publicly accessible.

**Buckets**

A bucket is a container for objects stored in Amazon S3. Every object is contained in a bucket. For example, if the object named **photos/puppy.jpg** is stored in the **johnsmith** bucket, then it is addressable using the URL [**http://johnsmith.s3.amazonaws.com/photos/puppy.jpg**](http://johnsmith.s3.amazonaws.com/photos/puppy.jpg)**.** You can store an infinite amount of data in a bucket.

An Amazon S3 bucket name is globally unique, regardless of the AWS Region in which you create the bucket. You specify the name at the time you create the bucket.

Amazon S3 creates buckets in a region you specify. To optimize latency, minimize costs, or address regulatory requirements, choose any AWS Region that is geographically close to you.

Buckets serve several purposes: they organize the Amazon S3 namespace at the highest level, they identify the account responsible for storage and data transfer charges, they play a role in access control, and they serve as the unit of aggregation for usage reporting.

**Objects and Keys**

Objects are the fundamental entities stored in Amazon S3 buckets. Objects consist of object data and metadata. The data portion is opaque to Amazon S3. The metadata is a set of name-value pairs that describe the object. An object is uniquely identified within a bucket by a key (name) and a version ID

A key is the unique identifier for an object within a bucket. Every object in a bucket has exactly one key. Because the combination of a bucket, key, and version ID uniquely identify each object, Amazon S3 can be thought of as a basic data map between "bucket + key + version" and the object itself.

Every object in Amazon S3 can be uniquely addressed through the combination of the web API Version 2006-03-01 3 Amazon Simple Storage Service Developer Guide Regions service endpoint, bucket name, key, and optionally, a version. For example, in the URL http:// doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl, "doc" is the name of the bucket and "2006-03-01/ AmazonS3.wsdl" is the key.

**AWS CloudFront**

Amazon CloudFront is an economical, but a dynamic Content Delivery Network or CDN. It administers and association of Edge locations to store copies of frequently accessed files in proximity to its users. Edge locations refer to the boundaries between locations of major cloud servers.