**[S3 (Simple Storage Service):](https://aws.amazon.com/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.
* Object based storage only for files, can not install OS or applications
* Data is spread across multiple devices and multiple facilities
* Can loose 2 facilities and still have access to your files
* Files can be between 1 byte and 5TB, and has no storage limit
* Files are stored flatly in buckets, Folders don't really exist, but are part of the file name
* S3 bucket names have a universal name-space, meaning each bucket name must be globally unique
* S3 Stores data in alphabetical order (lexigraphical order)
* S3 URL structures are region/amazon.aws.com/bucketname (https://s3-eu-west-1.amazonaws.com/myawesomebucket)
* Read after write consistency for PUTS of new objects (As soon as you write an object, it is immediately available)
* Eventual consistency for overwrite PUTS and DELETES. (Updating or deleting an object could take time to propagate)
* S3 is basically a key value store and consists of the following:
  + Key - Name of the object
  + Value - Data made up of bytes
  + Version ID (important for versioning)
  + Meta-data - Data about what you are storing
  + ACLs - Permissions for stored objects
* Amazon guarantees 99.99% availability for the S3 platform
* Amazon guarantees 99.999999999% durability for S3 information (11 x 9's)
* Tiered storage, and life-cycle management available
* Versioning is available but must be enabled. It is off by default
* Offers encryption, and allows you to secure the data using ACLs
* S3 charges for storage, requests, and data transfer
* Bucket names must be all lowercase, however in US-Standard if creating with the CLI tool, it will allow capital letters
* The transfers tab shows uploads, downloads, permission changes, storage class changes, etc..
* When you upload a file to S3, by default it is set private
* You can transfer files up to 5GB using PUT requests
* You can setup access control to control your buckets access by using bucket policies or ACLs
* Change the storage class under the Properties tab when an object is selected
* S3 buckets can be configured to create access logs which logs all requests to the S3 bucket
* S3 Events include SNS, or SQS events or Lambda functions. Lambda is location specific, not available in South Korea
* All storage tiers have SSL support, millisecond first byte latency, and support life-cycle management policies.
* **Storage Tiers:**
  + **Standard S3:**
    - Stored redundantly across multiple devices in multiple facilities
    - Designed to sustain the loss of 2 facilities concurrently
    - 11-9's durability, 99.99% availability
  + **S3-IA (Infrequently Accessed):**
    - For data that is accessed less frequently, but requires rapid access when needed
    - Lower fee than S3, but you are charged a retrieval fee
    - Also designed to sustain the loss of 2 facilities concurrently
    - 11-9's durability, 99.99% availability
  + **Reduced Redundancy Storage (RSS):**
    - Use for data such as thumbnails or data that could be regenerated
    - Costs less than Standard S3
    - Designed to provide 99.99% durability and 99.99% availability of objects over a year
    - Designed to sustain the loss of a single facility
  + **Glacier:**
    - Very cheap, Stores data for as little as $0.01 per gigabyte, per month
    - Optimized for data that is infrequently accessed. Used for archival only
    - It takes 3-5 hours to restore access to files from Glacier

**Using Versioning:**

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.

In one bucket, for example, you can have two objects with the same key, but different version IDs, such as photo.gif (version 111111) and photo.gif (version 121212).



On buckets which have had versioning enabled or versioning can only be disabled, but not removed. If you want to get rid of versioning, you’ll need to copy files to a new bucket which has never had versioning enabled, and update any references pointing to the old bucket to point to the new bucket instead.

If you enable versioning on an existing bucket, versioning will not be applied to existing objects; versioning will only apply to any new or updated objects.

* **Life-cycle Management**:
  + When clicking on Life-cycle, and adding a rule, a rule can be applied to either the entire bucket or a single 'folder' in a bucket
  + Rules can be set to move objects to either separate storage tiers or delete them all together
  + Can be applied to current version and previous versions
  + If multiple actions are selected for example transition from STD to IA storage 30 days after upload, and then Archive 60 days after upload is also selected, once an object is uploaded, 30 days later the object will be moved to IA storage. 30 days after that the object will be moved to glacier.
  + Calculates based on UPLOAD date not Action data
  + Transition from STD to IA storage class requires MINIMUM of 30 days. You cannot select or set any data range less than 30 days
  + Archive to Glacier can be set at a minimum of 1 day If STD->IA is NOT set
  + If STD->IA IS set, then you will have to wait a minimum of 60 days to archive the object because the minimum for STD->IA is 30 days, and the transition to glacier then takes an additional 30 days
  + When you enable versioning, there will be 2 sections in life-cycle management tab. 1 for the current version of an object, and another for previous versions
  + Minimum file size for IA storage is 128K for an object
  + Can set policy to permanently delete an object after a given time frame
  + If versioning is enabled, then the object must be set to expire, before it can be permanently deleted
  + Can not move objects to Reduced Redundancy using life-cycle policies
* **S3 Transfer Acceleration**:
  + Utilizes the CloudFront Edge Network to accelerate your uploads to S3
  + Instead of uploading directly to your S3 bucket, you can use a distinct URL to upload directly to an edge location which will then transfer the file to S3
  + This topic is covered in [AWS Solutions Architect Study Guide](http://clusterfrak.com/notes/certs/aws_saa_notes/)
  + There is a test utility available that will test uploading direct to S3 vs through Transfer Acceleration, which will show the upload speed from different global locations
  + Turning on and using Transfer Acceleration will incur an additional fee
* **2 types of encryption available**:
  + In transit:
    - Uses SSL/TLS to encrypt the transfer of the object
  + At Rest (AES 256):
    - Server Side: S3 Managed Keys (SSE-S3)
    - Server Side: AWS Key Management Service, Managed Keys (SSE-KMS)
    - Server Side: Encryption with Customer provided Keys (SSE-C)
    - Client Side Encryption
* Pricing (What your charged for when using S3):
  + Storage used
  + Number of Requests
  + Data Transfer

Encryption

Data in transit is encrypted via TLS

Data at rest can be encrypted by:

* [Server-side encryption](https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html)
  + SSE-S3 - Amazon S3-Managed Keys where S3 manages the keys, encrypting each object with a unique key using AES-256, and even ecrypts the key itself with a master key which regularly rotates.
  + SSE-KMS - AWS KMS-Managed Keys - Similar to SSE-S3, but with an option to provide an audit trail of when your key is used, and by whom, and also the option to create and manage keys yourself.
  + SSE-C - Customer-Provided Keys - Where you manage the encryption keys, and AWS manages encryption and decryption as it reads from and writes to disk.
* [Client Side Encryption](https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html)

Storage

S3 objects are stored in buckets, which can be thought of as folders

Buckets are private by default

S3 buckets are suitable for objects or flat files. S3 is NOT suitable for storing an OS or Database (use block storage for this)

There is a maximum of 100 buckets per account by default.

S3 objects are stored, and sorted by name in lexographical order, which means that there can be performance bottlenecks is you have a large number of objects in your S3 bucket which have similar names.

For S3 buckets with a large number of files, it’s recommended that you add a salt to the beginning of each file, to help avoid performance bottlenecks, and ensure that files are evenly distributed throughout the datacenter.

S3 objects are stored in multiple facilities; S3 is designed to sustain the loss of 2 facilities concurrently.

There is unlimited storage available with support for objects with sizes from 0 bytes to 5TB.

Versioning

On buckets which have had versioning enabled - versioning can only be disabled, but not removed. If you want to get rid of versioning, you’ll need to copy files to a new bucket which has never had versioning enabled, and update any references pointing to the old bucket to point to the new bucket instead.

If you enable versioning on an existing bucket, versioning will not be applied to existing objects; versioning will only apply to any new or updated objects.

Cross-region replication requires that versioning is enabled.

Side note: Dropbox uses S3 versioning.

Lifecycle management

There is tiered storage available, and you can use lifecycle management to transition though the tiers. For example, there might be a requirement that invoices for the last 24 months are immediately available, and that older invoices don’t need to be immediately available, but must be stored for compliance reasons for 7 years. For this scenario, you may decide to keep the invoices younger than 24 months in S3 for immediate access, and use lifecycle management to move the invoices to Glacier (where storage is extremely cheap, with the tradeoff that it takes 3-5 hours to restore an object) for long term storage.

Lifecycle management also supports permanently deleting files after a configurable amount of time i.e. after the file has been migrated to Glacier.

Cloudfront

Cloudfront is AWS’s CDN.

Cloudfront supports Perfect Forward Secrecy.

Edge Locations

Edge locations are separate from and different to AWS AZ’s and Regions

Edge locations support:

* Reads
* Writes - enabling writes means that customers can upload files to their local edge location, which can speed up data transfer for them

Origin

The origin is the name of the source location

The origin can be:

* A S3 bucket
* An EC2 Instance
* An ELB
* Route53
* Outside of AWS

Distribution

A “distribution” is the collection of edge locations in the CDN

There are two different types of distribution:

* Web distribution - used for websites
* RTMP distribution - used for streaming content (i.e. video), and only supported if the origin is S3 - other origins such as EC2, etc do not support RTMP

It’s possible to clear cached items. If the cache isn’t cleared, all items live for their configured TTL (Time To Live)

Read access can be restricted via pre-signed urls and cookies. i.e. to ensure only certain customers can access certain objects.

Geo restrictions can be created for whitelists/blacklists

[**CloudFront:**](https://aws.amazon.com/cloudfront/)

* **CloudFront** is a **CDN (Content Delivery Network)**. It retrieves data from Amazon S3 bucket and distributes it to multiple datacenter locations. It delivers the data through a network of data centers called **edge locations**. The nearest edge location is routed when the user requests for data, resulting in lowest latency, low network traffic, fast access to data, etc.

## How AWS CloudFront Delivers the Content?

AWS CloudFront delivers the content in the following steps.

**Step 1** − The user accesses a website and requests an object to download like an image file.

**Step 2** − DNS routes your request to the nearest CloudFront edge location to serve the user request.

**Step 3** − At edge location, CloudFront checks its cache for the requested files. If found, then returns it to the user otherwise does the following −

* First CloudFront compares the request with the specifications and forwards it to the applicable origin server for the corresponding file type.
* The origin servers send the files back to the CloudFront edge location.
* As soon as the first byte arrives from the origin, CloudFront starts forwarding it to the user and adds the files to the cache in the edge location for the next time when someone again requests for the same file.

**Step 4** − The object is now in an edge cache for 24 hours or for the provided duration in file headers. CloudFront does the following −

* CloudFront forwards the next request for the object to the user’s origin to check the edge location version is updated or not.
* If the edge location version is updated, then CloudFront delivers it to the user.
* If the edge location version is not updated, then origin sends the latest version to CloudFront. CloudFront delivers the object to the user and stores the latest version in the cache at that edge location.

## Features of CloudFront

**Fast** − The broad network of edge locations and CloudFront caches copies of content close to the end users that results in lowering latency, high data transfer rates and low network traffic. All these make CloudFront fast.

**Simple** − It is easy to use.

**Can be used with other AWS Services** − Amazon CloudFront is designed in such a way that it can be easily integrated with other AWS services, like Amazon S3, Amazon EC2.

**Cost-effective** − Using Amazon CloudFront, we pay only for the content that you deliver through the network, without any hidden charges and no up-front fees.

**Elastic** − Using Amazon CloudFront, we need not worry about maintenance. The service automatically responds if any action is needed, in case the demand increases or decreases.

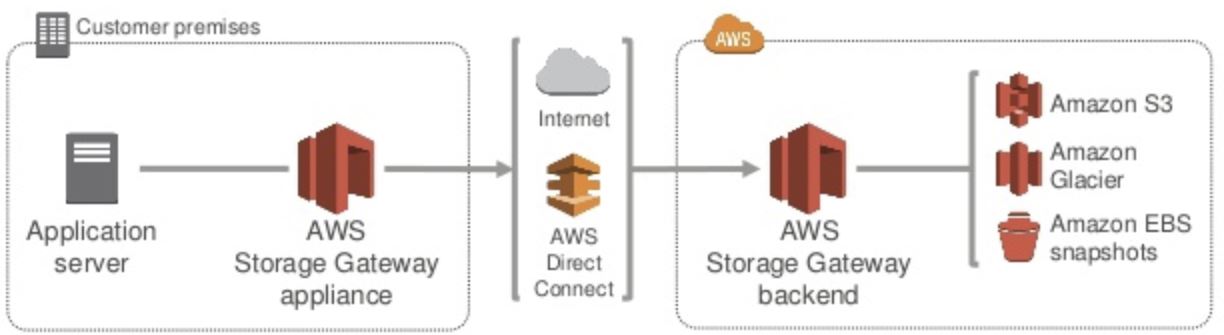
**Reliable** − Amazon CloudFront is built on Amazon’s highly reliable infrastructure, i.e. its edge locations will automatically re-route the end users to the next nearest location, if required in some situations.

**Global** − Amazon CloudFront uses a global network of edge locations located in most of the regions.

* Edge Location is the location where content will be cached, separate from an AWS Region/AZ
* Origin is the origin of all files, can be S3, EC2 instance, a ELB, or Route53
* Distribution is the name given to the CDN which consists a collection of edge locations
* Web Distributions are used for websites
* RTMP - (Real-Time Messaging Protocol) used for streaming media typically around adobe flash files
* Edge locations can be R/W and will accept a PUT request on an edge location, which then will replicate the file back to the origin
* Objects are cached for the life of the TTL (24 hours by default)
* You can clear objects from edge locations, but you will be charged
* When enabling cloudfront from an S3 origin, you have the option to restrict bucket access; this will disable the direct link to the file in the S3 bucket, and ensure that the content is only served from cloudfront
* The path pattern uses regular expressions
* You can restrict access to your distributions using signed URLS
* You can assign Web Application Firewall rules to your distributions
* Distribution URLs are going to be non-pretty names such as random\_characters.cloudfront.com; you can create a CNAME that points to the cloudfront name to make the URL user friendly
* You can restrict content based on geographical locations in the behaviors tab
* You can create custom error pages via the error pages tab
* Purging content is handled in the Invalidations tab

Storage Gateway

A Storage Gateway is a software appliance which sits in your data center, and securely connects your on-premise



There are four types of Storage Gateway:

* File Gateway - using NFS to store files in S3
* Volume Gateway - a virtual iSCSI disk. Block based, not object based like S3.
  + Cached volumes - the entire data set is stored on the cloud, with recently-read data on site, for quick retrieval of frequently accessed data.
  + Stored volumes - similar to Volume Gateway, but the entire data-set is stored on premise with data being incrementally backed up to S3
* Tape Gateway - virtual tapes, backed up to Glacier. Used by popular backup applications such as NetBackup

All data transferred between Storage Gateway and S3 is encrypted using SSL. By default, all data stored in S3 is encrypted server-side with SSE-S3, so your data is automatically encrypted at rest.