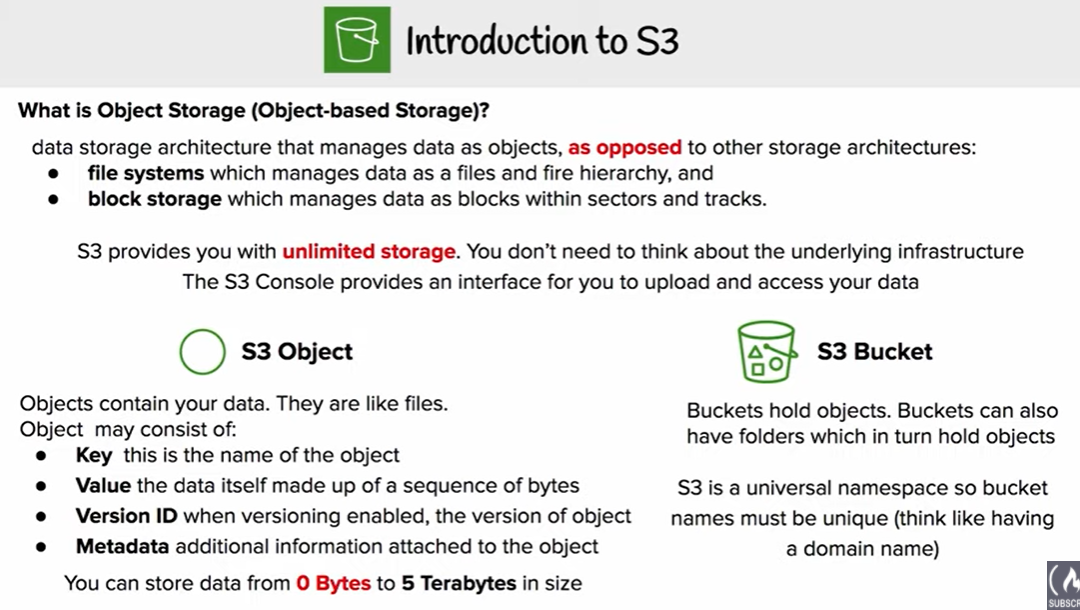
AWS Simple Storage Service – S3

# AWS Simple Storage Service – S3



* Amazon S3 is a simple key, value object store designed for the Internet
* S3 provides unlimited storage space and works on the pay as you use model. Service rates gets cheaper as the usage volume increases
* S3 offers an extremely durable, highly available, and infinitely scalable data storage infrastructure at very low costs.
* **S3 is an Object level storage (not a Block level storage) and cannot be used to host OS or dynamic websites**
* **S3 resources for e.g. buckets and objects are private by default**

## S3 Buckets & Objects

### ****S3 Buckets****

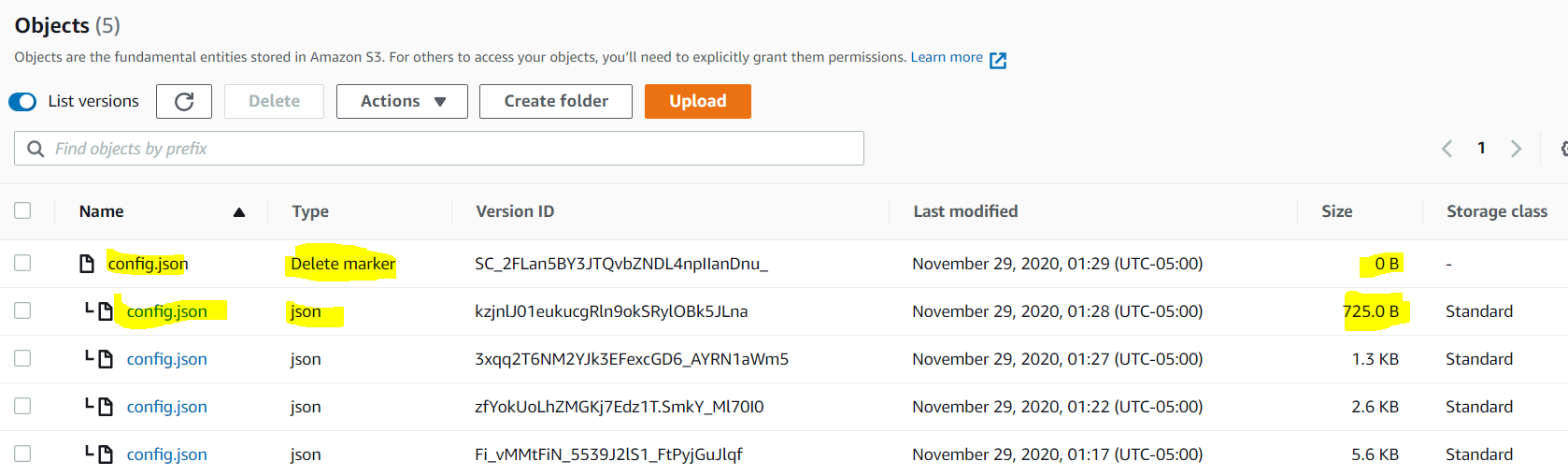
* A bucket is a container for objects stored in S3 and help organize the S3 namespace.
* A bucket is owned by the AWS account that creates it and helps identify the account responsible for storage and data transfer charges.
* **S3 bucket names are globally unique, regardless of the AWS region in which it was created and the namespace is shared by all AWS accounts**
* **Even though S3 is a global service, buckets are created within a region specified during the creation of the bucket**
* Every object is contained in a bucket
* **There is no limit to the number of objects that can be stored in a bucket and no difference in performance whether a single bucket or multiple buckets are used to store all the objects**
* S3 data model is a flat structure i.e. there are no hierarchies or folders within the buckets. However, logical hierarchy can be inferred using the keyname prefix e.g. Folder1/Object1
* Restrictions
  + 100 buckets (soft limit) and maximum of 1000 buckets can be created in each of AWS account
  + Bucket names has to be all lower-case.
  + Bucket names should be globally unique and DNS compliant
  + Bucket ownership is not transferable
  + Buckets cannot be nested and cannot have bucket within another bucket
  + Bucket name and region cannot be changed, once created
* Empty or a non-empty bucket can be deleted
* S3 allows retrieval of 1000 objects and provides pagination support

### ****Objects****

* Objects are the fundamental entities stored in S3 bucket
* **Object is uniquely identified within a bucket by a keyname and version ID**
* Objects consist of object data, metadata and others
  + **Key** is object name
  + **Value** is data portion is opaque to S3
  + **Metadata** is the data about the data and is a set of name-value pairs that describe the object for e.g. content-type, size, last modified. Custom metadata can also be specified at the time the object is stored.
  + **Version ID**is the version id for the object and in combination with the key helps to unique identify an object within a bucket
  + **Subresources** helps provide additional information for an object
  + **Access Control Information** helps control access to the objects
* S3 objects allow two kinds of metadata
  + System metadata
    - Metadata such as the Last-Modified date is controlled by the system. Only S3 can modify the value.
    - System metadata that user can control, for e.g., the storage class, encryption configured for the object.
  + User-defined metadata
    - User-defined metadata can be assigned during uploading the object or after the object has been uploaded.
    - User-defined metadata is stored with the object and is returned when object is downloaded
    - S3 does not process user-defined metadata.
    - User-defined metadata must begin with the prefix “x-amz-meta“, otherwise S3 will not set the key value pair as you define it
* Object metadata cannot be modified after the object is uploaded and it can be only modified by performing copy operation and setting the metadata
* Objects belonging to a bucket reside in a specific AWS region never leave that region, unless explicitly copied using Cross Region replication
* Object can be retrieved as a whole or a partially
* With [Versioning](https://jayendrapatil.com/aws-s3-object-versioning/) enabled, current as well as previous versions of an object can be retrieved

## Bucket & Object Operations

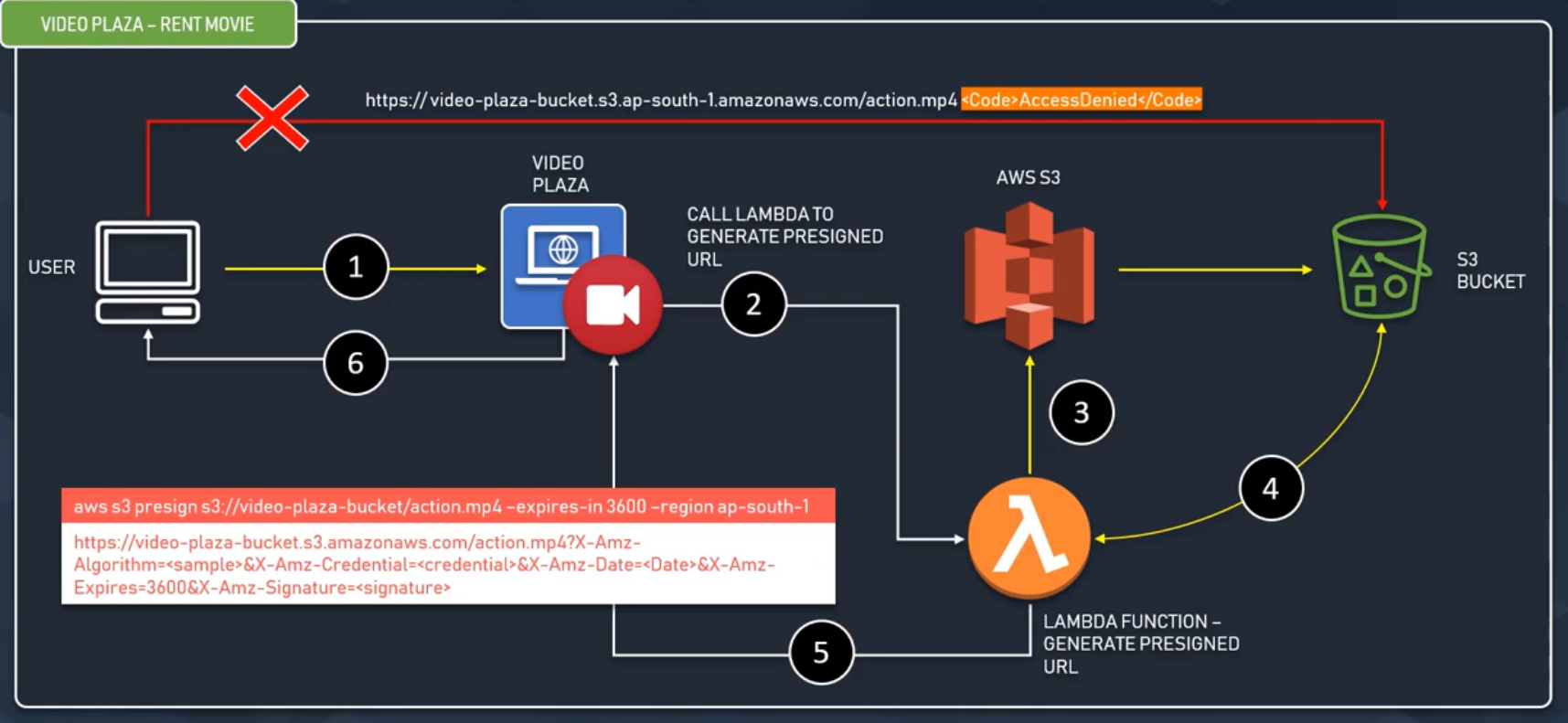
* Listing
  + S3 allows listing of all the keys within a bucket
  + A single listing request would return a max of 1000 object keys with pagination support using an indicator in the response to indicate if the response was truncated
  + Keys within a bucket can be listed using Prefix and Delimiter.
  + Prefix limits results to only those keys (kind of filtering) that begin with the specified prefix, and delimiter causes list to roll up all keys that share a common prefix into a single summary list result.
* Retrieval
  + Object can be retrieved as a whole
  + Object can be retrieved in parts or partially (specific range of bytes) by using the Range HTTP header.
  + Range HTTP header is helpful
    - if only partial object is needed for e.g. multiple files were uploaded as a single archive
    - for fault tolerant downloads where the network connectivity is poor
  + Objects can also be downloaded by sharing Pre-Signed urls
  + Metadata of the object is returned in the response headers
* Object Uploads
  + **Single Operation – Objects of size 5GB can be uploaded in a single PUT operation**
  + **Multipart upload – can be used for objects of size > 5GB and supports max size of 5TB can is recommended for objects above size 100MB**
  + Pre-Signed URLs can also be used shared for uploading objects
  + Uploading object if successful, can be verified if the request received a success response. Additionally, returned ETag can be compared to the calculated MD5 value of the upload object
* Copying Objects
  + **Copying of object up to 5GB can be performed using a single operation and multipart upload can be used for uploads up to 5TB**
  + When an object is copied
    - user-controlled system metadata e.g. storage class and user-defined metadata are also copied.
    - system controlled metadata e.g. the creation date etc is reset
  + Copying Objects can be needed to
    - Create multiple object copies
    - Copy object across locations or regions
    - Renaming of the objects
    - Change object metadata for e.g. storage class, encryption etc
    - **Updating any metadata for an object requires all the metadata fields to be specified again**
* Deleting Objects
  + **S3 allows deletion of a single object or multiple objects (max 1000) in a single call**
  + For Non Versioned buckets,
    - the object key needs to be provided and object is permanently deleted
  + For Versioned buckets,
    - if an object key is provided, S3 inserts a **delete marker** and the current object becomes non-current object
    - if an object key with a version ID is provided, that object is permanently deleted
    - if the version ID is of the delete marker, the delete marker is removed and the previous non-current version becomes the current version object (i.e. it makes file visible again).
    - If you want to delete a file completely (along with all versions), select all version & delete them.

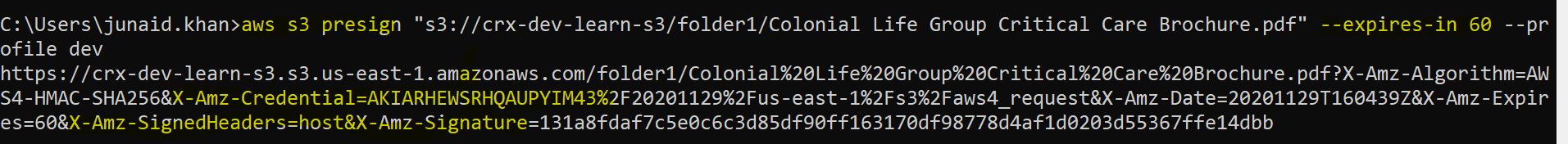


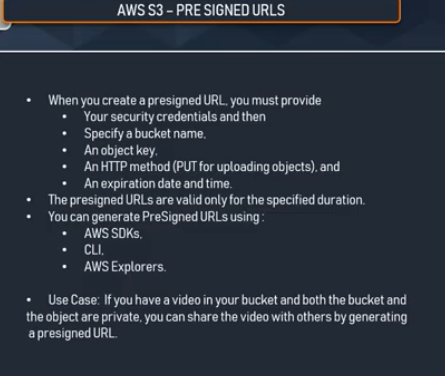
* + Deletion can be MFA enabled for adding extra security
* Restoring Objects from Glacier
  + Objects must be restored before accessing an archived object
  + **Restoration of an Object can take about 3 to 5 hours for standard retrievals. Glacier now offers expedited retrievals within 1-5 minutes.**
    - * **Bulk Retrievals: 5 to12 Hrs.**
      * **Standard Retrievals: 3 to 5 Hrs.**
      * **Expedited Retrievals: 1 to 5 Minutes (Objects less than 250 MB)**
  + **Restoration request also needs to specify the number of days for which the object copy needs to be maintained.**
  + During this period, storage cost applies for both archive and the copy

## Pre-Signed URLs

* All buckets and objects are by default private
* **Pre-signed URLs allows user to be able to download or upload a specific object without requiring AWS security credentials or permissions**
* Pre-signed URL allows anyone access to the object identified in the URL, provided the creator of the URL has permissions to access that object
* Creation of the pre-signed urls requires the creator to provide his security credentials, specify a bucket name, an object key, an HTTP method (GET for download object & PUT of uploading objects), and expiration date and time
* Pre-signed urls are valid only till the expiration date & time







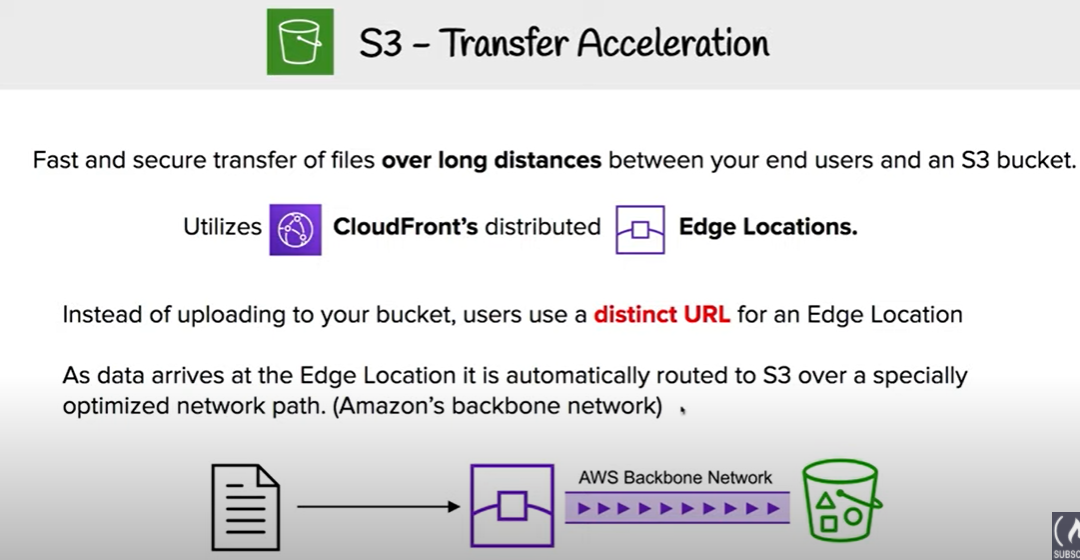
## Multipart Upload

* Multipart upload allows the user to upload a single large object as a set of parts. Each part is a contiguous portion of the object’s data.
* Multipart uploads supports 1 to 10000 parts and each part can be from 5MB to 5GB with last part size allowed to be less than 5MB
* **Multipart uploads allow max upload size of 5TB**
* **Object parts can be uploaded independently and in any order. If transmission of any part fails, it can be retransmitted without affecting other parts.**
* After all parts of the object are uploaded and complete initiated, S3 assembles these parts and creates the object.
* Using multipart upload provides the following advantages:
  + **Improved throughput** – parallel upload of parts to improve throughput
  + **Quick recovery from any network issues – Smaller part size minimizes the impact of restarting a failed upload due to a network error.**
  + **Pause and resume object uploads** – Object parts can be uploaded over time. Once a multipart upload is initiated there is no expiry; you must explicitly complete or abort the multipart upload.
  + **Begin an upload before the final object size is known** – an object can be uploaded as is it being created

**Three Step process** : <https://aws.amazon.com/premiumsupport/knowledge-center/s3-multipart-upload-cli/>

* + **Multipart Upload Initiation**
    - Initiation of a Multipart upload request to S3 returns a unique ID (I.e., UploadID) for each multipart upload.
    - This ID needs to be provided for each part uploads, completion or abort request and listing of parts call.
    - All the Object metadata required needs to be provided during the Initiation call
  + **Parts Upload**
    - Parts upload of objects can be performed using the unique upload ID
    - **A part number (between 1 – 10000) needs to be specified with each request which identifies each part and its position in the object**
    - If a part with the same part number is uploaded, the previous part would be overwritten
    - After the part upload is successful, S3 returns an ETag header in the response which must be recorded along with the part number to be provided during the multipart completion request
  + **Multipart Upload Completion or Abort**
    - On Multipart Upload Completion request, S3 creates an object by concatenating the parts in ascending order based on the part number and associates the metadata with the object
    - Multipart Upload Completion request should include the unique upload ID with all the parts and the ETag information
    - S3 response includes an ETag that uniquely identifies the combined object data
    - On Multipart upload Abort request, the upload is aborted and all parts are removed. Any new part upload would fail. However, any in progress part upload is completed and hence and abort request must be sent after all the parts upload have been completed
    - S3 should receive a multipart upload completion or abort request else it will not delete the parts and storage would be charged

## Transfer Acceleration



* S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between the client and an S3 bucket.
* Transfer Acceleration takes advantage of CloudFront’s globally distributed edge locations. As the data arrives at an edge location, data is routed to S3 over an optimized network path.

## Virtual Hosted Style vs Path-Style Request

S3 allows the buckets and objects to be referred in Path-style or Virtual hosted-style URLs

### Path-style

* Bucket name is not part of the domain (unless region specific endpoint used)
* Endpoint used must match the region in which the bucket resides for e.g, if you have a bucket called *mybucket*that resides in the EU (Ireland) region with object named puppy.jpg, the correct path-style syntax URI is http://s3-eu-west-1.amazonaws.com/mybucket/puppy.jpg*.*
* A “PermanentRedirect” error is received with an HTTP response code 301, and a message indicating what the correct URI is for the resource if a bucket is accessed outside the US East (N. Virginia) region with path-style syntax that uses either of the following:
  + http://s3.amazonaws.com
  + An endpoint for a region different from the one where the bucket resides for e.g., if you use http://s3-eu-west-1.amazonaws.com for a bucket that was created in the US West (N. California) region
* Path-style requests would not be supported after September 30, 2020

### Virtual hosted-style

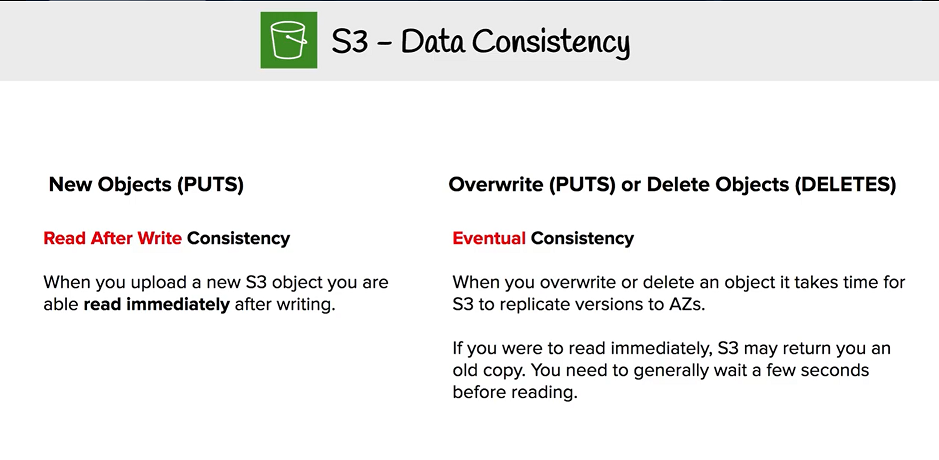
* S3 supports virtual hosted-style and path-style access in all regions.
* In a virtual-hosted-style URL, the bucket name is part of the domain name in the URL for e.g. **http://bucketname.s3.amazonaws.com/objectname**
* S3 virtual hosting can be used to address a bucket in a REST API call by using the HTTP Host header
* Benefits
  + attractiveness of customized URLs,
  + provides an ability to publish to the “root directory” of the bucket’s virtual server. This ability can be important because many existing applications search for files in this standard location.
* S3 updates DNS to reroute the request to the correct location when a bucket is created in any region, which might take time.
* **S3 routes any virtual hosted-style requests to the US East (N.Virginia) region, by default,** if the US East (N. Virginia) endpoint s3.amazonaws.com is used, instead of the region-specific endpoint (for e.g., s3-eu-west-1.amazonaws.com) and S3 redirects it with HTTP 307 redirect to the correct region.
* When using virtual hosted-style buckets with SSL, the SSL wild card certificate only matches buckets that do not contain periods.To work around this, use HTTP or write your own certificate verification logic.
* If you make a request to the **http://bucket.s3.amazonaws.com** endpoint, the DNS has sufficient information to route the request directly to the region where your bucket resides.

## S3 Pricing

* S3 costs vary by Region
* Charges in S3 are incurred for
  + **Storage – cost is per GB/month**
  + **Requests – per request cost varies depending on the request type GET, PUT**
  + **Data Transfer**
    - **data transfer in is free**
    - **data transfer out is charged per GB/month (except in the same region or to Amazon CloudFront)**

# AWS S3 Data Consistency Model

* S3 achieves high availability by replicating data across multiple servers within Amazon’s data centers.
* S3 provides **Read-after-Write consistency for PUTS of new objects**
  + For a PUT request, S3 synchronously stores data across multiple facilities before returning SUCCESS
  + A process writes a new object to S3 and will be immediately able to read the Object i.e. **PUT 200 -> GET 200**
  + A process writes a new object to S3 and immediately lists keys within its bucket. Until the change is fully propagated, the object might not appear in the list.
  + However, if a HEAD or GET request to a key name is made before the object is created, then create the object shortly after that, a subsequent GET might not return the object due to eventual consistency. i.e. **GET 404 -> PUT 200 -> GET 404**
* S3 provides**Eventual Consistency for overwrite PUTS and DELETES** in all regions.
  + For updates and deletes to Objects, the changes are eventually reflected and not available immediately i.e. **PUT 200 -> PUT 200 -> GET 200 (might be older version) OR DELETE 200 -> GET 200**
  + if a process replaces an existing object and immediately attempts to read it, S3 might return the prior data till the change is fully propagated
  + if a process deletes an existing object and immediately attempts to read it, S3 might return the deleted data until the deletion is fully propagated
  + if a process deletes an existing object and immediately lists keys within its bucket. Until the deletion is fully propagated, S3 might list the deleted object.
* Updates to a single key are atomic. for e.g., if you PUT to an existing key, a subsequent read might return the old data or the updated data, but it will never write corrupted or partial data.
* **S3 does not currently support object locking. for e.g. If two PUT requests are simultaneously made to the same key, the request with the latest time stamp wins. If this is an issue, you will need to build an object-locking mechanism into your application.**
* S3 Object Lock is different as it allows to store objects using a **write-once-read-many (WORM)** model, which prevents an object from being deleted or overwritten for a fixed amount of time or indefinitely.
* Updates are key-based; there is no way to make atomic updates across keys. for e.g, update of one key cannot be dependent on the update of another key unless you design this functionality into the application.



## Additional Topics

* [Amazon S3 Consistency Model](https://jayendrapatil.com/aws-s3-data-consistency-model/)
* [Amazon S3 Subresources](https://jayendrapatil.com/aws-s3-subresources/)
* [Amazon S3 Storage Tiers](https://jayendrapatil.com/aws-s3-storage-classes/)
* [Amazon S3 Object Versioning](https://jayendrapatil.com/aws-s3-object-versioning/)
* [Amazon S3 Lifecycle Management](https://jayendrapatil.com/aws-s3-object-lifecycle-management/)
* [Amazon S3 Permission](https://jayendrapatil.com/aws-s3-permisions/)
* [Amazon S3 Data Protection](https://jayendrapatil.com/aws-s3-data-protection/)
* [Amazon S3 Best Practices](https://jayendrapatil.com/aws-s3-best-practices/)

## AWS Certification Exam Practice Questions

1. What does Amazon S3 stand for?
   1. Simple Storage Solution.
   2. Storage Storage Storage (triple redundancy Storage).
   3. Storage Server Solution.
   4. **Simple Storage Service**
2. What are characteristics of Amazon S3? Choose 2 answers
   1. **Objects are directly accessible via a URL**
   2. S3 should be used to host a relational database
   3. S3 allows you to store objects or virtually unlimited size
   4. **S3 allows you to store virtually unlimited amounts of data**
   5. S3 offers Provisioned IOPS
3. You are building an automated transcription service in which Amazon EC2 worker instances process an uploaded audio file and generate a text file. You must store both of these files in the same durable storage until the text file is retrieved. You do not know what the storage capacity requirements are. Which storage option is both cost-efficient and scalable?
   1. Multiple Amazon EBS volume with snapshots
   2. A single Amazon Glacier vault
   3. **A single Amazon S3 bucket**
   4. Multiple instance stores
4. A user wants to upload a complete folder to AWS S3 using the S3 Management console. How can the user perform this activity?
   1. Just drag and drop the folder using the flash tool provided by S3
   2. Use the Enable Enhanced Folder option from the S3 console while uploading objects
   3. The user cannot upload the whole folder in one go with the S3 management console
   4. **Use the Enable Enhanced Uploader option from the S3 console while uploading objects**(NOTE – Its no longer supported by AWS)
5. A media company produces new video files on-premises every day with a total size of around 100GB after compression. All files have a size of 1-2 GB and need to be uploaded to Amazon S3 every night in a fixed time window between 3am and 5am. Current upload takes almost 3 hours, although less than half of the available bandwidth is used. What step(s) would ensure that the file uploads are able to complete in the allotted time window?
   1. Increase your network bandwidth to provide faster throughput to S3
   2. **Upload the files in parallel to S3 using mulipart upload**
   3. Pack all files into a single archive, upload it to S3, then extract the files in AWS
   4. Use AWS Import/Export to transfer the video files
6. A company is deploying a two-tier, highly available web application to AWS. Which service provides durable storage for static content while utilizing lower Overall CPU resources for the web tier?
   1. Amazon EBS volume
   2. **Amazon S3**
   3. Amazon EC2 instance store
   4. Amazon RDS instance
7. You have an application running on an Amazon Elastic Compute Cloud instance, that uploads 5 GB video objects to Amazon Simple Storage Service (S3). Video uploads are taking longer than expected, resulting in poor application performance. Which method will help improve performance of your application?
   1. Enable enhanced networking
   2. **Use Amazon S3 multipart upload**
   3. Leveraging Amazon CloudFront, use the HTTP POST method to reduce latency.
   4. Use Amazon Elastic Block Store Provisioned IOPs and use an Amazon EBS-optimized instance
8. When you put objects in Amazon S3, what is the indication that an object was successfully stored?
   1. Each S3 account has a special bucket named\_s3\_logs. Success codes are written to this bucket with a timestamp and checksum.
   2. A success code is inserted into the S3 object metadata.
   3. **A HTTP 200 result code and MD5 checksum, taken together, indicate that the operation was successful.**
   4. Amazon S3 is engineered for 99.999999999% durability. Therefore there is no need to confirm that data was inserted.
9. You have private video content in S3 that you want to serve to subscribed users on the Internet. User IDs, credentials, and subscriptions are stored in an Amazon RDS database. Which configuration will allow you to securely serve private content to your users?
   1. **Generate pre-signed URLs for each user as they request access to protected S3 content**
   2. Create an IAM user for each subscribed user and assign the GetObject permission to each IAM user
   3. Create an S3 bucket policy that limits access to your private content to only your subscribed users’ credentials
   4. Create a CloudFront Origin Identity user for your subscribed users and assign the GetObject permission to this user
10. You run an ad-supported photo sharing website using S3 to serve photos to visitors of your site. At some point you find out that other sites have been linking to the photos on your site, causing loss to your business. What is an effective method to mitigate this?
    1. **Remove public read access and use signed URLs with expiry dates.**
    2. Use CloudFront distributions for static content.
    3. Block the IPs of the offending websites in Security Groups.
    4. Store photos on an EBS volume of the web server.
11. You are designing a web application that stores static assets in an Amazon Simple Storage Service (S3) bucket. You expect this bucket to immediately receive over 150 PUT requests per second. What should you do to ensure optimal performance?
    1. Use multi-part upload.
    2. **Add a random prefix to the key names.**
    3. Amazon S3 will automatically manage performance at this scale.
    4. Use a predictable naming scheme, such as sequential numbers or date time sequences, in the key names
12. What is the maximum number of S3 buckets available per AWS Account?
    1. 100 Per region
    2. There is no Limit
    3. **100 Per Account**(Refer [documentation](http://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html))
    4. 500 Per Account
    5. 100 Per IAM User
13. Your customer needs to create an application to allow contractors to upload videos to Amazon Simple Storage Service (S3) so they can be transcoded into a different format. She creates AWS Identity and Access Management (IAM) users for her application developers, and in just one week, they have the application hosted on a fleet of Amazon Elastic Compute Cloud (EC2) instances. The attached IAM role is assigned to the instances. As expected, a contractor who authenticates to the application is given a pre-signed URL that points to the location for video upload. However, contractors are reporting that they cannot upload their videos. Which of the following are valid reasons for this behavior? Choose 2 answers { “Version”: “2012-10-17”, “Statement”: [ { “Effect”: “Allow”, “Action”: “s3:\*”, “Resource”: “\*” } ] }
    1. The IAM role does not explicitly grant permission to upload the object. (The role has all permissions for all activities on S3)
    2. The contractorsˈ accounts have not been granted “write” access to the S3 bucket. (using pre-signed urls the contractors account don’t need to have access but only the creator of the pre-signed urls)
    3. **The application is not using valid security credentials to generate the pre-signed URL.**
    4. The developers do not have access to upload objects to the S3 bucket. (developers are not uploading the objects but its using pre-signed urls)
    5. The S3 bucket still has the associated default permissions. (does not matter as long as the user has permission to upload)
    6. **The pre-signed URL has expired.**
14. Which of the following are valid statements about Amazon S3? Choose 2 answers
    1. S3 provides read-after-write consistency for any type of PUT or DELETE.
    2. Consistency is not guaranteed for any type of PUT or DELETE.
    3. **A successful response to a PUT request only occurs when a complete object is saved**
    4. Partially saved objects are immediately readable with a GET after an overwrite PUT.
    5. **S3 provides eventual consistency for overwrite PUTS and DELETES**
15. A customer is leveraging Amazon Simple Storage Service in eu-west-1 to store static content for web-based property. The customer is storing objects using the Standard Storage class. Where are the customers’ objects replicated?
    1. Single facility in eu-west-1 and a single facility in eu-central-1
    2. Single facility in eu-west-1 and a single facility in us-east-1
    3. **Multiple facilities in eu-west-1**
    4. A single facility in eu-west-1
16. A user has an S3 object in the US Standard region with the content “color=red”. The user updates the object with the content as “color=”white”. If the user tries to read the value 1 minute after it was uploaded, what will S3 return?
    1. It will return “color=white”
    2. It will return “color=red”
    3. It will return an error saying that the object was not found
    4. **It may return either “color=red” or “color=white” i.e. any of the value**