Cloud Computing Course Outline

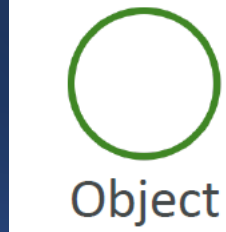
Farid Afzali, Ph.D., P.Eng.

Amazon S3 Bucket

Amazon S3 bucket

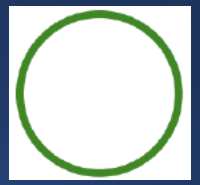
- **Amazon S3 as a main building block**: Amazon S3 is described as a fundamental component of AWS's suite of services. It is commonly used for a wide range of applications due to its scalability, durability, and availability.
- **Infinitely scaling storage**: S3 is advertised as offering "infinitely scaling" storage, meaning that it can grow to accommodate any amount of data, which is only limited by the AWS account and service limits.
- **Use by many websites**: Numerous websites rely on S3 to host and serve their content. This is because S3 can serve as a repository for images, videos, files, and other static assets that a website might need.
- **Use by many AWS services**: S3 integrates with a multitude of other AWS services, acting as a storage endpoint or repository. For example, it can store logs from AWS CloudTrail, data for Amazon Machine Learning, or input/output for AWS Lambda.
- **Step-by-step approach to S3**: There will be a structured guide or tutorial following this introduction, which will likely go into detail on how to use S3, including how to set it up, upload files, retrieve data, and possibly manage access and security.

Amazon S3 bucket- Objects



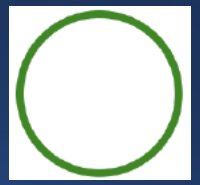
- **Objects have a Key**: In Amazon S3, each file is considered an "object" and has a unique identifier known as a "key".
- **The key is the FULL path**: Unlike traditional file systems, the key in S3 represents the entire path of the object. For example, **s3://my-bucket/my_file.txt** shows the key for a file named **my_file.txt** in the **my-bucket** bucket.
- **Key composition**: A key is composed of a prefix (which can be thought of as the "folder" path, even though S3 doesn't use real folders) and the object name itself. For instance, **s3://my-bucket/my_folder/another_folder/my_file.txt** has a prefix of **my_folder/another_folder/** and the object name is **my_file.txt**.
- **No concept of “directories” within buckets**: S3 doesn't have actual directories or folders, but it simulates this structure using prefixes in keys. This allows the S3 user interface to display a hierarchy for ease of navigation, even though the underlying structure is flat.
- **Keys with slashes**: The keys can include slashes (**/**) to simulate folder structures, but these are part of the object's key name rather than indicating actual directories.

Amazon S3 bucket- Objects



- **User-Based**: Refers to Identity and Access Management (IAM) policies which determine which API calls should be allowed for a specific user in IAM. These are permissions attached to IAM users or roles.
- **Resource-Based**: This includes:
  - **Bucket Policies**: These are bucket-wide rules set from the S3 console, which can allow cross-account access.
  - **Object Access Control List (ACL)**: Provides a finer grain of access control and can be disabled.
  - **Bucket Access Control List (ACL)**: This is less common but can also be disabled.
- **Note**: It's stated that an IAM principal (a user, role, or AWS service) can access an S3 object if the IAM permissions allow it or the resource policy allows it, provided there is no explicit deny in place.
- **Encryption**: It advises to encrypt objects in Amazon S3 using encryption keys, which secures the data at rest.

Amazon S3 bucket- Objects



- **Object values**: These refer to the actual content or data within the object. The maximum size for an object in S3 is 5 terabytes (TB), or 5000 gigabytes (GB).
- **Uploading large objects**: For objects larger than 5 GB, Amazon S3 requires the use of "multi-part upload," which is a method to upload large files by splitting them into smaller parts and uploading each part separately.
- **Metadata**: This is additional information about the object, which can be system-defined or user-defined. Metadata is stored as a set of key/value pairs.
- **Tags**: Similar to metadata, tags are Unicode key/value pairs that can be associated with an S3 object, up to a limit of 10 tags per object. They are useful for managing access control, setting up lifecycle policies, and tracking costs.
- **Version ID**: If versioning is enabled on the S3 bucket, each object will have a unique version ID. This allows for the retrieval of different versions of an object over time, providing a way to recover from accidental deletions or overwrites.

Amazon S3 bucket policies

- **JSON based policies**: These are access policies written in the JavaScript Object Notation (JSON) format that define permissions for S3 buckets and objects.
- **Resources**: The specific S3 buckets and objects to which the policy applies.
- **Effect**: Specifies whether the policy will allow or deny access.
- **Actions**: Defines the set of S3 API operations (like **s3:GetObject**) that are allowed or denied by the policy.
- **Principal**: Identifies the account, user, role, or service that is allowed or denied access by the policy

Amazon S3 bucket policies

HANDS ON PRACTICE

Amazon S3 bucket policies- Example

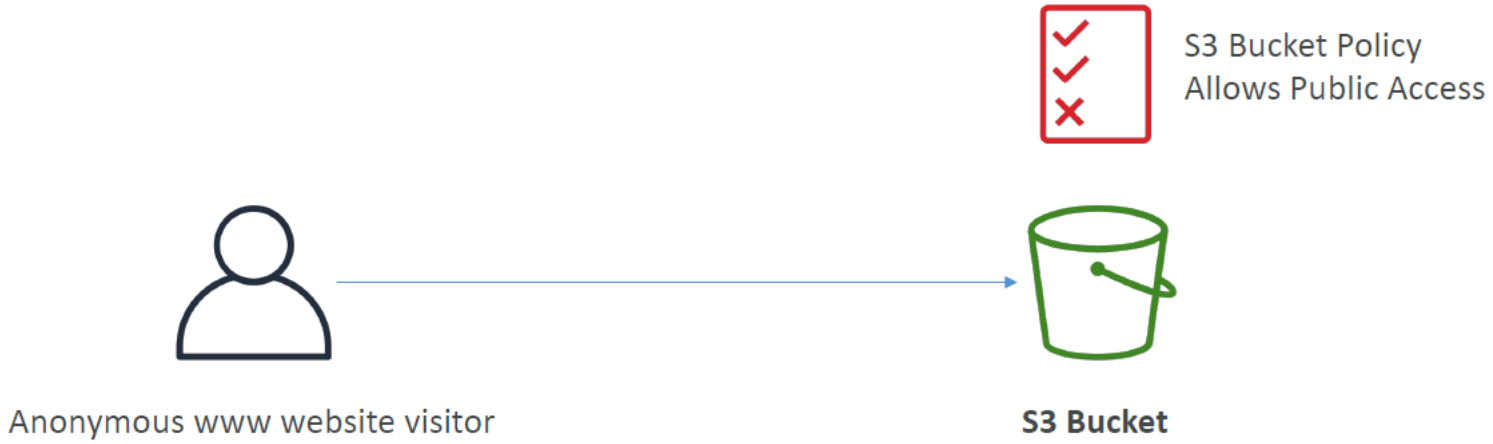
- An example of a JSON policy snippet that grants public read access to all objects in an S3 bucket.
- It uses the **"Effect": "Allow"** to permit the action, **"Principal": "*"** to apply to all users, and **"Action": ["s3:GetObject"]** to specify the allowed operation.
- The **"Resource": "arn:aws:s3:::examplebucket/*"** line indicates that the policy applies to all objects within the 'examplebucket'.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicRead",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::examplebucket/*"
            ]
        }
    ]
}
```

Amazon S3 bucket policies- Example

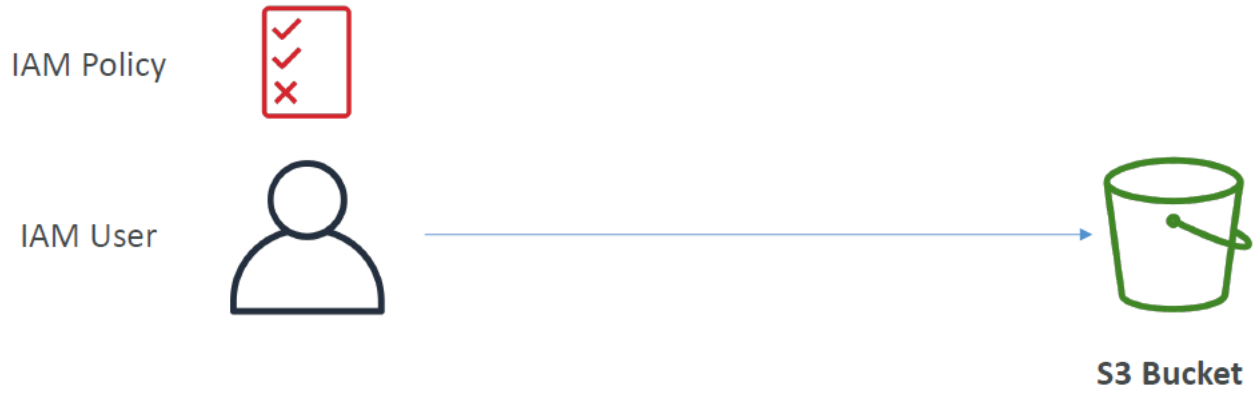
```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PublicReadAccess",
         "Effect": "Allow",
         "Principal": "*",
         "Action": [
            "s3:GetObject"
         ],
         "Resource": "arn:aws:s3:::BUCKET-NAME/*"
      },
      {
         "Sid": "PublicWriteAccess",
         "Effect": "Allow",
         "Principal": "*",
         "Action": [
            "s3:PutObject",
            "s3:DeleteObject"
         ],
         "Resource": "arn:aws:s3:::BUCKET-NAME/*"
      }
   ]
}
```

Amazon S3 bucket policies- Public Access-User Bucket policy



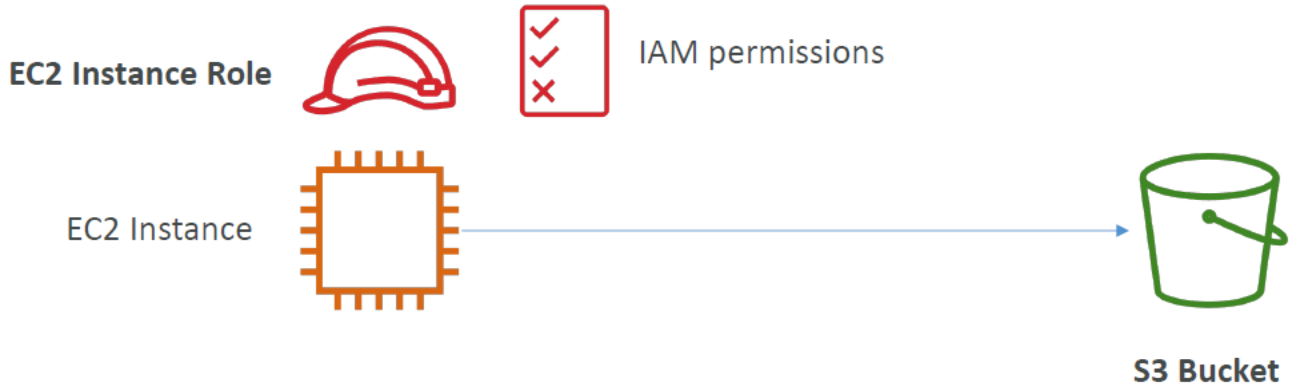
- This is a schematic representation of public access to an Amazon S3 bucket using a bucket policy.
- It shows an icon representing an anonymous website visitor on the left, a line connecting to an Amazon S3 bucket icon on the right, and a checkmark indicating that the S3 bucket policy allows for public access.
- This implies that the bucket has been configured to allow anyone on the internet to access or download its contents without requiring AWS credentials or any form of user authentication.

Amazon S3 bucket policies- IAM Permission



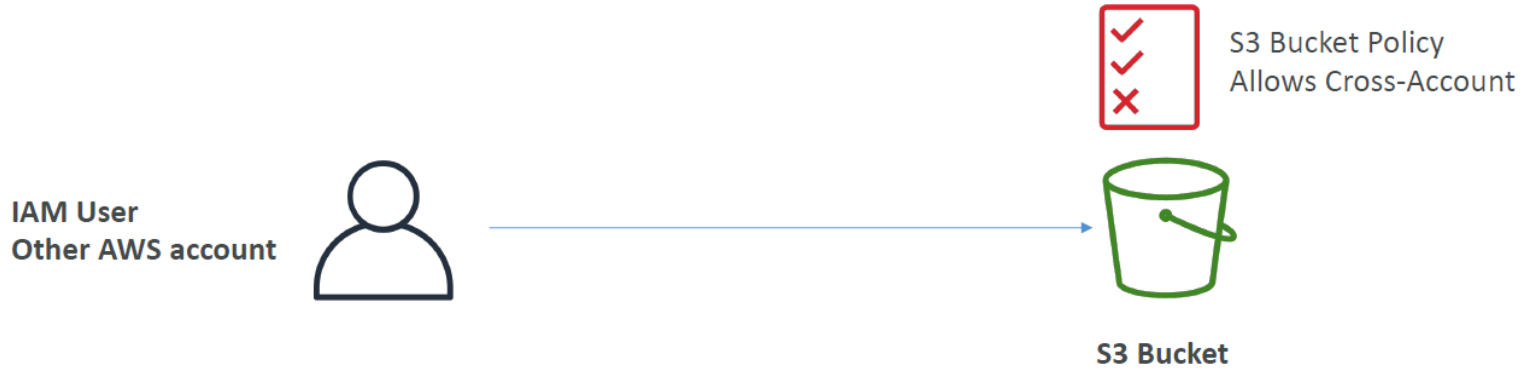
- This is a simple diagram explaining user access to an Amazon S3 bucket via IAM (Identity and Access Management) permissions.
- It includes a representation of an IAM user and an S3 bucket, with a line connecting the two, suggesting that the IAM user has some level of access to the S3 bucket.
- Next to the IAM user icon, there is an IAM policy icon with a checkmark and a cross, indicating that the user's access is governed by specific permissions set within the IAM policy.
- This policy dictates what actions the IAM user is allowed or denied to perform on the S3 bucket.

Amazon S3 bucket policies- EC2 instance Access- Use IAM rules



- This illustrates the concept of an Amazon EC2 instance utilizing IAM roles to interact with an S3 bucket.
- The EC2 instance is depicted with an associated EC2 instance role, which is equipped with specific IAM permissions.
- These permissions define what actions the EC2 instance is authorized to perform on the S3 bucket, such as reading or writing data.
- The use of IAM roles for EC2 instances is a best practice in AWS for securely managing credentials and permissions. It allows the EC2 instance to make API calls to AWS services without the need to manage explicit credentials.

Amazon S3 bucket policies- cross-account access



- This describes how cross-account access is managed in AWS using an S3 bucket policy.
- It shows an IAM user from another AWS account trying to access an S3 bucket. The S3 bucket policy is configured to allow such cross-account interactions.
- This means that the bucket owner has set up permissions that explicitly allow users from a different AWS account to access or perform specific actions within the bucket, which can include listing objects, reading and writing files, etc.
- This setup is commonly used for collaboration between different AWS accounts, allowing for secure sharing of resources without sharing AWS account credentials.

S3 bucket settings to block public access

Block *all* public access
On

- **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
  On
- **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
  On
- **Block public access to buckets and objects granted through *new* public bucket or access point policies**
  On
- **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
  On

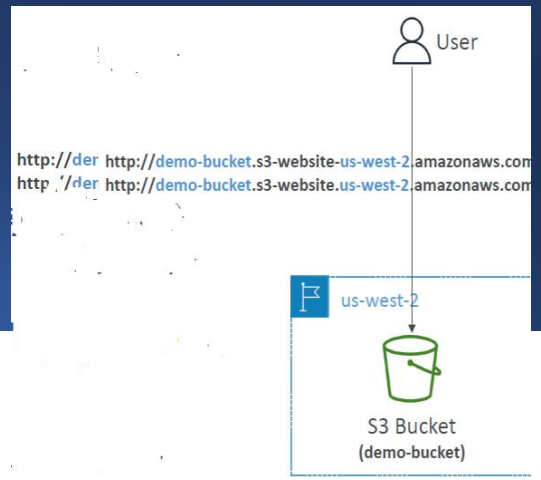
S3 bucket settings to block public access

- These settings are an important aspect of cloud storage security, ensuring that buckets and objects within Amazon S3 are not inadvertently exposed to the public. The settings include:
- **Block all public access:** A master switch that overrides individual permissions, ensuring no public access is possible.
- **Block public access to buckets and objects granted through new access control lists (ACLs):** Prevents the granting of public access via new ACLs.
- **Block public access to buckets and objects granted through any access control lists (ACLs):** Ensures that no public access is allowed, regardless of existing or new ACLs.
- **Block public access to buckets and objects granted through new public bucket or access point policies:** Prevents new policies from inadvertently granting public access.
- **Block public and cross-account access to buckets and objects through any public bucket or access point policies:** Extends the block to cross-account access if it's made public through bucket policies.
- These settings are a proactive measure to prevent data leaks and ensure that sensitive company data is not accessible to anyone without proper authorization. It's recommended to keep these settings enabled if there's no requirement for the bucket to be public, and they can be applied at the AWS account level for broad protection.

Hosting a static website on Amazon S3

- **Amazon S3 – Static Website Hosting**: Static websites typically contain web pages with fixed content coded in HTML, CSS, and JavaScript.
- **S3 can host static websites and have them accessible on the Internet**: Amazon S3 is not only a storage service but can also be used to host static websites, making them accessible via a standard web browser over the internet.
- **The website URL will be (depending on the region)**: The URL structure of the hosted static website on S3 will vary depending on the AWS region where the S3 bucket is located.
  - **http://bucket-name.s3-website-aws-region.amazonaws.com**: This is an example URL format provided by AWS. Here **bucket-name** should be replaced with the actual name of your S3 bucket, and **aws-region** should be replaced with the identifier for the region, like **us-west-2** for the US West (Oregon) region.
- **If you get a 403 Forbidden error, make sure the bucket policy allows public reads**: This is troubleshooting advice. A 403 Forbidden error typically indicates that the S3 bucket's permissions are not set to allow public access. To resolve this, the bucket policy needs to be configured to allow anyone to read the files, if public access is intended.

Hosting a static website on Amazon S3



- **Amazon S3 – Static Website Hosting**: Static websites typically contain web pages with fixed content coded in HTML, CSS, and JavaScript.
- **S3 can host static websites and have them accessible on the Internet**: Amazon S3 is not only a storage service but can also be used to host static websites, making them accessible via a standard web browser over the internet.
- **The website URL will be (depending on the region)**: The URL structure of the hosted static website on S3 will vary depending on the AWS region where the S3 bucket is located.
  - **http://bucket-name.s3-website-aws-region.amazonaws.com**: This is an example URL format provided by AWS. Here **bucket-name** should be replaced with the actual name of your S3 bucket, and **aws-region** should be replaced with the identifier for the region, like **us-west-2** for the US West (Oregon) region.
- **If you get a 403 Forbidden error, make sure the bucket policy allows public reads**: This is troubleshooting advice. A 403 Forbidden error typically indicates that the S3 bucket's permissions are not set to allow public access. To resolve this, the bucket policy needs to be configured to allow anyone to read the files, if public access is intended.

Hosting a static website on Amazon S3

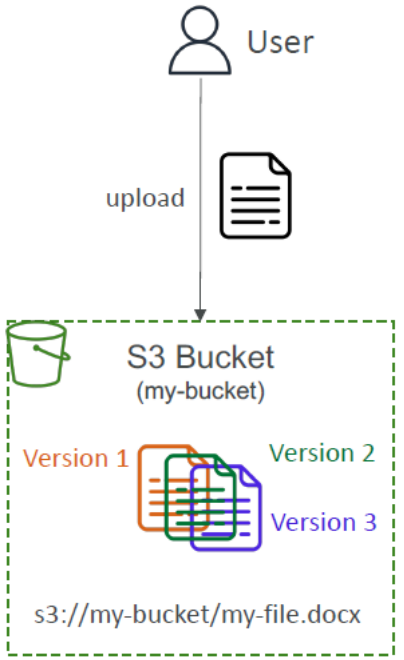
HANDS ON PRACTICE

Amazon S3 - Versioning

- **Amazon S3 - Versioning**: This is the heading of the section. It indicates that the content will discuss the versioning capability within Amazon S3, which is a web service offering scalable object storage.
- **You can version your files in Amazon S3**: This sentence explains that Amazon S3 allows you to keep multiple versions of an object in the same bucket. Versioning is a means of keeping multiple variants of an object in the same bucket.
- **It is enabled at the bucket level**: Versioning is a setting that is applied to the entire S3 bucket, not to individual files. Once enabled, it affects all objects added to that bucket.
- **Same key overwrite will change the "version"**: When you upload a new version of a file with the same key (the file name in the bucket), the version ID changes. S3 maintains a list of all versions of the object.
- **It is best practice to version your buckets**: The image suggests that it is a good practice to enable versioning for S3 buckets for the following reasons:
  - **Protect against unintended deletes (ability to restore a version)**: If you accidentally delete an object, you can restore it to a previous version.
  - **Easy roll back to previous version**: If an object is updated and you need to revert to an earlier version, versioning makes this possible.
- **Notes**:
  - **Any file that is not versioned prior to enabling versioning will have version "null"**: Objects that were in the bucket before versioning was enabled will have a version ID of "null".
  - **Suspending versioning does not delete the previous versions**: If you suspend versioning, the existing versions of the objects are preserved. New objects will not have version IDs unless you re-enable versioning.

Amazon S3 - Versioning

- A user is uploading a document to the S3 bucket (named "my-bucket").
- The document ("my-file.docx") has multiple versions within the bucket. Each time the document is updated and re-uploaded, a new version is created and stored.
- The S3 bucket has a graphical representation of three versions of an object, suggesting that this bucket has versioning enabled and is storing different versions of the same object.

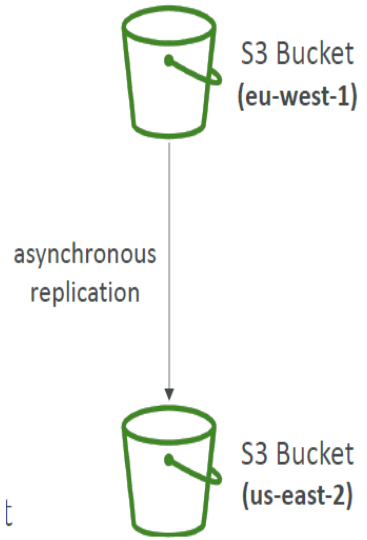


Amazon S3 - Versioning

HANDS ON PRACTICE

Amazon S3 Replication Cross-Region Replication (CRR) and Same-Region Replication (SRR)

- **Amazon S3 – Replication (CRR & SRR)**: features available in Amazon S3 that allow you to automatically copy objects across S3 buckets in different AWS Regions or within the same Region.
- **Must enable Versioning in source and destination buckets**: For replication to work, both the source bucket (where the original files are stored) and the destination bucket (where the files are replicated to) must have versioning enabled.
- **Cross-Region Replication (CRR)**: This feature is used to copy objects across buckets that are located in different AWS Regions. It's typically used for geographic redundancy and compliance.
- **Same-Region Replication (SRR)**: This allows for the automatic copying of objects within the same AWS Region, which can be useful for log aggregation or live replication between production and testing environments.
- **Buckets can be in different AWS accounts**: The replication can occur not only within the same account but also across different AWS accounts, providing flexibility for access and data management.
- **Copying is asynchronous**: The replication process is not instantaneous; it happens asynchronously, meaning there may be a short delay between the original object's upload and its replication.
- **Must give proper IAM permissions to S3**: The AWS Identity and Access Management (IAM) permissions must be correctly configured to allow for the replication to occur.
- **Use cases**:
  - **CRR**: Use cases for Cross-Region Replication include compliance requirements, improving data locality for lower latency access, and replication across different accounts for organizational purposes.
  - **SRR**: Use cases for Same-Region Replication include the aggregation of logs into a central account and real-time replication between production and test environments.



Amazon S3 Replication

HANDS ON PRACTICE

Amazon S3 Storage Classes

- **Amazon S3 Standard - General Purpose**: This class is designed for frequently accessed data, providing high durability, availability, and performance.
- **Amazon S3 Standard-Infrequent Access (IA)**: Ideal for data that is accessed less frequently, but requires rapid access when needed. It offers a lower storage price compared to S3 Standard.
- **Amazon S3 One Zone-Infrequent Access**: Similar to IA, but stores data in a single Availability Zone only. It's cheaper than Standard-IA and suitable for data that can be recreated if the data in the AZ is lost.
- **Amazon S3 Glacier Instant Retrieval**: Offers the lowest cost storage for long-lived, infrequently accessed data that requires retrieval in milliseconds.
- **Amazon S3 Glacier Flexible Retrieval**: Used for archiving data where retrieval can be planned a few minutes to a few hours in advance.
- **Amazon S3 Glacier Deep Archive**: Provides the lowest cost storage for data archiving where retrieval times of up to 12 hours are acceptable.
- **Amazon S3 Intelligent Tiering**: Automatically moves data between different access tiers based on usage patterns, without performance impact or operational overhead.

Durability and availability characteristics of Amazon S3 storage

- **Durability**:
  - Amazon S3 offers high durability, with a guarantee of 99.999999999% (often referred to as "11 9's"), meaning that data is extremely resilient against loss.
  - The example given illustrates that if you were to store 10,000,000 objects on Amazon S3, you could expect a loss of a single object once every 10,000 years on average.
  - This high level of durability is consistent across all S3 storage classes.
- **Availability**:
  - Availability refers to how readily available the service is for operations.
  - The availability guarantee varies by storage class, with S3 Standard offering 99.99% availability.
  - An example provided states that S3 Standard's 99.99% availability translates to the service potentially being unavailable for about 53 minutes in a year.

Amazon S3 Standard storage class

- **99.99% Availability**: Amazon S3 Standard offers a high level of availability, ensuring that data is accessible when needed nearly all of the time.
- **Use for Frequently Accessed Data**: It is optimized for data that is accessed often, making it suitable for active workloads.
- **Low Latency and High Throughput**: The service is designed to provide quick response times and high data transfer rates, which is critical for performance-sensitive applications.
- **Sustain 2 Concurrent Facility Failures**: The data is resilient and can withstand the failure of two facilities simultaneously without data loss, emphasizing the reliability of the storage.
- **Use Cases**: Ideal for a variety of applications, including Big Data analytics, mobile and gaming applications, and content distribution, where frequent and fast access to the data is necessary.

Amazon S3's infrequent access storage classes

- **Amazon S3 Standard-Infrequent Access (Standard-IA)**:
  - **99.9% Availability**: This class has a slightly lower availability compared to S3 Standard.
  - **Use Cases**: It is ideal for data that is accessed less frequently but requires fast access when needed, like disaster recovery and backups.
- **Amazon S3 One Zone-Infrequent Access (One Zone-IA)**:
  - **High Durability**: The durability is very high, with a probability of 99.999999999% of durability over a given year.
  - **99.5% Availability**: This storage class has a slightly lower availability compared to Standard-IA and is designed to store data in a single Availability Zone.
  - **Use Cases**: Suitable for storing secondary backup copies of on-premise data, or data that can be recreated. It’s important to note that data stored in this class may be lost if the Availability Zone is destroyed, hence it is not suitable for critical data that cannot be recreated.

Amazon S3 Glacier storage classes

- **Amazon S3 Glacier Instant Retrieval**:
  - Offers millisecond retrieval times, making it suitable for data that is accessed infrequently (about once a quarter) but requires immediate access when it is needed.
  - There is a minimum storage duration charge of 90 days.
- **Amazon S3 Glacier Flexible Retrieval (formerly Amazon S3 Glacier)**:
  - Provides three retrieval options:
    - Expedited (1 to 5 minutes),
    - Standard (3 to 5 hours),
    - Bulk (5 to 12 hours), which is the most cost-effective option and is free.
  - This class also has a minimum storage duration charge of 90 days.
- **Amazon S3 Glacier Deep Archive**:
  - Is intended for long-term storage of data that may be accessed very infrequently.
  - Offers two retrieval options:
    - Standard (12 hours),
    - Bulk (48 hours), which is the lowest-cost retrieval option.
  - This class has a minimum storage duration charge of 180 days.
- The pricing for these Glacier storage classes includes the cost of storage per gigabyte and the cost associated with retrieving objects. These classes are optimized for cost savings, making them ideal for archiving and backing up data that does not require frequent access.

Amazon S3 Intelligent-Tiering storage class

- **Small monthly monitoring and auto-tiering fee**: This refers to a nominal fee charged by Amazon S3 for monitoring the access patterns of the objects and automatically moving them between tiers.
- **Moves objects automatically between Access Tiers based on usage**: S3 Intelligent-Tiering automatically shifts objects to the most cost-effective access tier without performance impact or operational overhead.
- **There are no retrieval charges in S3 Intelligent-Tiering**: Unlike other S3 storage classes that may charge for retrieval, Intelligent-Tiering does not have additional fees for accessing objects.
- **Frequent Access tier (automatic)**: This is the default tier for objects that are accessed regularly. If an object has not been accessed for 30 consecutive days, it is moved to the Infrequent Access tier.
- **Infrequent Access tier (automatic)**: Objects are moved here from the Frequent Access tier if they have not been accessed for 30 days. This tier has a lower storage cost, but higher access costs compared to the Frequent Access tier.
- **Archive Instant Access tier (automatic)**: Objects that haven't been accessed for 90 days are moved here, offering lower storage costs for long-lived, infrequently accessed data.
- **Archive Access tier (optional)**: Users can optionally activate this tier for objects to further reduce storage costs for data that is accessed less frequently. The configurable access period ranges from 90 to 700+ days.
- **Deep Archive Access tier (optional)**: This tier is for data that may be accessed very infrequently and can tolerate a retrieval time of hours. It offers the lowest storage cost and is suitable for long-term archiving. The access period for this tier is also configurable from 180 to 700+ days.

Amazon S3 Class Comparison

	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Glacier Deep Archive
Durability	99.999999999% == (11 9's)						
Availability	99.99%	99.9%	99.9%	99.5%	99.9%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99%	99.9%	99.9%
Availability Zones	>= 3	>= 3	>= 3	1	>= 3	>= 3	>= 3
Min. Storage Duration Charge	None	None	30 Days	30 Days	90 Days	90 Days	180 Days
Min. Billable Object Size	None	None	128 KB	128 KB	128 KB	40 KB	40 KB
Retrieval Fee	None	None	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved

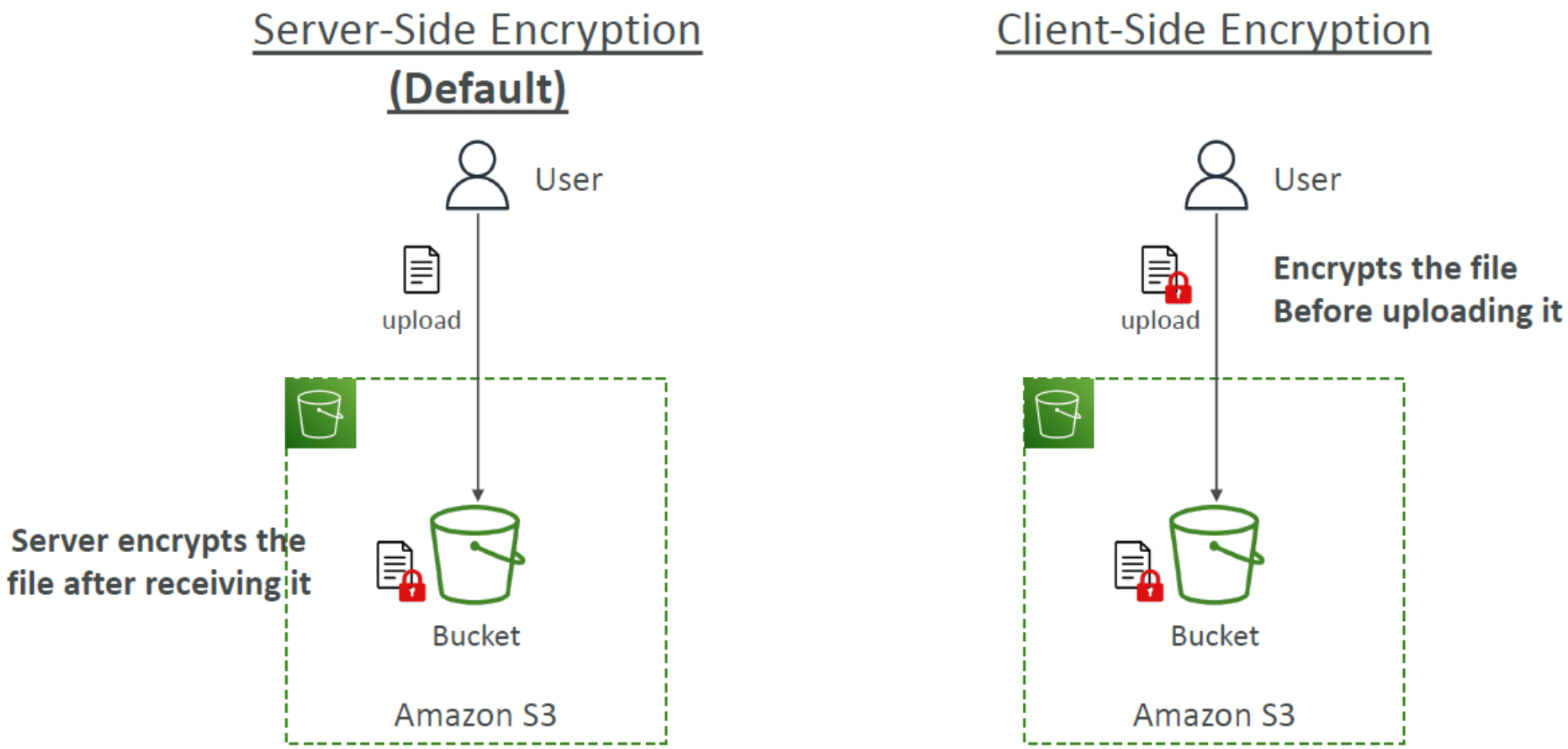
Amazon S3 Class Price Comparison us-east-1

	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Glacier Deep Archive
Storage Cost (per GB per month)	$0.023	$0.0025 - $0.023	$0.0125	$0.01	$0.004	$0.0036	$0.00099
Retrieval Cost (per 1000 request)	**GET:** $0.0004 **POST:** $0.005	**GET:** $0.0004 **POST:** $0.005	**GET:** $0.001 **POST:** $0.01	**GET:** $0.001 **POST:** $0.01	**GET:** $0.01 **POST:** $0.02	**GET:** $0.0004 **POST:** $0.03 **Expedited:** $10 **Standard:** $0.05 **Bulk:** free	**GET:** $0.0004 **POST:** $0.05 **Standard:** $0.10 **Bulk:** $0.025
Retrieval Time	Instantaneous					**Expedited** (1 – 5 mins) **Standard** (3 – 5 hours) Bulk (5 – 12 hours)	**Standard** (12 hours) **Bulk** (48 hours)
Monitoring Cost (pet 1000 objects)		$0.0025					

Amazon S3 Class

HANDS ON PRACTICE

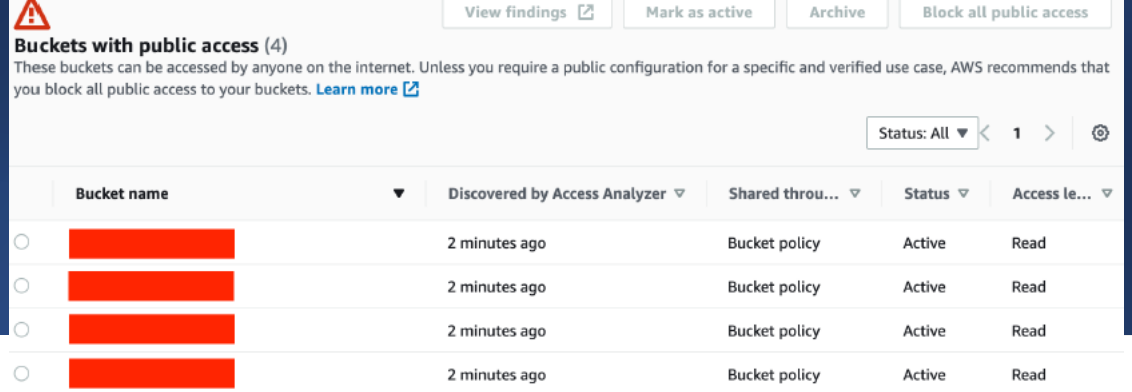
Amazon S3 Encryption



Amazon S3 Encryption

- Server-Side Encryption (SSE) and Client-Side Encryption (CSE).
- **Server-Side Encryption (Default)**:
  - The user uploads their files to an S3 bucket.
  - Once the file is uploaded, Amazon S3 encrypts the file. This process is handled by the server after receiving the file, hence the name server-side encryption.
  - The lock icon on the S3 bucket symbolizes that the data is secured within the S3 environment.
- **Client-Side Encryption**:
  - The user encrypts the file on their side (client-side) before uploading it to the S3 bucket.
  - The encryption is done using the user's own encryption mechanisms or tools before the data leaves the user's environment.
  - The padlock on the document before it is uploaded signifies that the file is encrypted before it is sent to the S3 bucket.

IAM Access Analyzer for S3



- **IAM Access Analyzer for S3**: This tool is designed to ensure that only intended users have access to your Amazon S3 buckets. It helps to identify and mitigate security issues related to unintended access.
- **Functionality**:
  - It can point out if a bucket is publicly accessible or if a bucket is shared with another AWS account, which could be a potential security risk if not intended.
  - The analyzer evaluates S3 bucket policies, access control lists (ACLs), and access point policies to determine access permissions.
- **Evaluation Example**: The image shows an example warning from the IAM Access Analyzer indicating that there are four S3 buckets with public access. The tool lists these buckets and marks them with a red bar, suggesting that they can be accessed by anyone on the internet. This serves as an alert for the user to review and potentially restrict the access settings.
- **Actions**: The interface provides options to view findings, mark them as active, archive them, or block all public access immediately.

Shared Responsibility Model for Amazon S3 (Simple Storage Service)

- **AWS Responsibilities**:
  - **Infrastructure**: AWS is responsible for the global security, durability, and availability of the S3 service, including ensuring that data can survive the concurrent loss of two facilities without data loss.
  - **Configuration and vulnerability analysis**: AWS handles the overall configuration of the S3 environment and regularly analyzes vulnerabilities to maintain security standards.
  - **Compliance validation**: AWS ensures that the S3 service complies with various industry and regulatory standards.
- **User Responsibilities**:
  - **S3 Versioning**: Users are responsible for enabling and managing versioning of their S3 buckets to keep track of and recover previous versions of objects.
  - **S3 Bucket Policies**: Users need to create and manage policies that define access permissions to their S3 buckets.
  - **S3 Replication Setup**: Users must set up and manage the replication of their S3 objects across different regions or within the same region.
  - **Logging and Monitoring**: Users must implement and maintain logging and monitoring to track access and usage of their S3 resources.
  - **S3 Storage Classes**: Users are tasked with selecting the appropriate storage class for their data based on access patterns and cost considerations.
  - **Data encryption at rest and in transit**: Users must manage the encryption of their data both at rest (while stored in S3) and in transit (when moving to or from S3).