**Introduction**

1. **What is Cloud Computing?**

Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider like Amazon Web Services (AWS).

## ***What is a Cloud Deployment Model?***

Cloud Deployment Model functions as a virtual computing environment with a deployment architecture that varies depending on the amount of data you want to store and who has access to the infrastructure.

[Different types of cloud](https://www.geeksforgeeks.org/types-of-cloud/) computing deployment models are described below.

* [Public Cloud](https://www.geeksforgeeks.org/difference-between-public-cloud-and-private-cloud/)
* Private Cloud
* [Hybrid Cloud](https://www.geeksforgeeks.org/public-cloud-vs-private-cloud-vs-hybrid-cloud/)
* Community Cloud
* [Multi-Cloud](https://www.geeksforgeeks.org/overview-of-multi-cloud/)

1. ***Cloud Service Models?***

There are the following three types of cloud service models -

1. [Infrastructure as a Service (IaaS)](https://www.javatpoint.com/cloud-service-models#IaaS)
2. [Platform as a Service (PaaS)](https://www.javatpoint.com/cloud-service-models#PaaS)
3. [Software as a Service (SaaS)](https://www.javatpoint.com/cloud-service-models#SaaS)

## **Infrastructure as a Service (IaaS)**

IaaS is also known as **Hardware as a Service (HaaS)**. It is a computing infrastructure managed over the internet. The main advantage of using IaaS is that it helps users to avoid the cost and complexity of purchasing and managing the physical servers.

## **Platform as a Service (PaaS)**

PaaS cloud computing platform is created for the programmer to develop, test, run, and manage the applications.

## **Software as a Service (SaaS)**

SaaS is also known as "**on-demand software**". It is a software in which the applications are hosted by a cloud service provider. Users can access these applications with the help of internet connection and web browser.

# *Architecture of Cloud Computing?*

# Lightbox

# AWS Global Infrastructure count?

* 32 Launched Regions each with multiple Availability Zones (AZs)
* 102 Availability Zones
* 550+ Points of Presence and 13 Regional Edge Caches

## 32 Launched Regions

* each with multiple Availability Zones (AZs
* What are the three parts of AWS global infrastructure?

The components are: Availability Zones (AZs) Regions. Edge Locations.

1. ***why do we use region in cloud computing***

The closer your customers are to the region where your cloud resources are located, the faster and better their experience will be. For example, if your customers are located in Germany, it makes sense to choose a European region for your cloud region, even if your office is in Delaware.

1. ***What are service in cloud computing?***

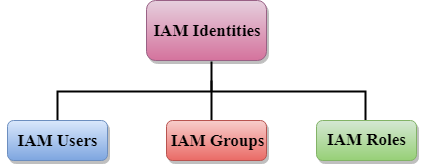
Cloud services are application and infrastructure resources that exist on the Internet. Third-party providers contract with subscribers for these services, allowing customers to leverage powerful computing resources without having to purchase or maintain hardware and software.

1. **what are resources in cloud computing?**

Cloud computing is the delivery of computing resources — including storage, processing power, databases, networking, analytics, artificial intelligence, and software applications — over the internet (the cloud).

**IAM Service**

1. ***how many resources do we have in IAM ?***
2. ***Deployment model in IAM ?***
3. ***Identities in IAM ?***



1. ***What is an IAM User ?***

An AWS Identity and Access Management (IAM) user is an entity that you create in AWS. The IAM user represents the human user or workload who uses the IAM user to interact with AWS. A user in AWS consists of a name and credentials.

1. ***What is the IAM group ?***

An IAM group is an identity that specifies a collection of IAM users. You can't use a group to sign-in. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users.

1. ***What is the IAM policy ?***

IAM policies define permissions for an action regardless of the method that you use to perform the operation.

1. ***What is the IAM Role ?***

An IAM role is an IAM identity that you can create in your account that has specific permissions.

An IAM role is similar to an IAM user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS.

1. Where do we attach identity based policy ?
2. Identity-based policies are attached to an IAM user, group, or role.
3. These policies let you specify what that identity can do (its permissions).
4. ***Where do we attach resource based policy ?***

Resource-based policies are attached to a resource.

For example, you can attach resource-based policies to Amazon S3 buckets, Amazon SQS queues, VPC endpoints, and AWS Key Management Service encryption keys. For a list of services that support resource-based policies, see AWS services that work with IAM.

1. ***Can we able to create policy via json code ?***

Yes, You can type or paste policies in JSON by choosing the JSON option.

 This method is useful for copying an example policy to use in your account. Or, you can type your own JSON policy document in the JSON editor.

1. ***If one user can created by it by default. Which permission has been assigned to that user?***

no permissions and is not authorized to perform any AWS actions or to access any AWS resources.

1. ***What is dominator policy ?***
2. ***What is ARN? What are the fields in ARN ?***

An Amazon Resource Name is a file naming convention used to identify a particular resource in the Amazon Web Services (AWS) public cloud. ARNs, which are specific to AWS, help an administrator track and use AWS items and policies across AWS products and API calls.

1. ***how many types of ARN partition ?***

In AWS, there are three types of ARN (Amazon Resource Name) partitions:

1. **AWS ARN partition:** This is the default partition used for most AWS resources, such as EC2 instances, S3 buckets, and Lambda functions. The ARN for resources in this partition starts with **"arn:aws".**
2. **AWS China ARN partition:** This partition is specific to the AWS China region. It is used for resources created in the AWS China (Beijing) and AWS China (Ningxia) regions. The ARN for resources in this partition starts with **"arn:aws-cn".**
3. **AWS GovCloud (US) ARN partition:** This partition is specific to the AWS GovCloud (US) region. It is used for resources created in the AWS GovCloud (US) region, which is designed to host sensitive data and regulated workloads for US government agencies and their partners. The ARN for resources in this partition starts with **"arn:aws-us-gov".**
4. ***What are Tags ?***

In AWS IAM (Identity and Access Management) service, tags are key-value pairs that can be assigned to IAM users, groups, roles, and policies to help manage and organize these resources. Tags can be used for various purposes such as categorizing resources, tracking costs, or implementing security policies.

***S3***

1. **Difference between Block storage & Object Storage?**

|  |  |
| --- | --- |
| **Block Storage** | **Object Storage** |
| 1. Block storage manages data in fixed-sized blocks or chunks. | Object storage manages data as discrete objects. |
| 1. These blocks are accessed like individual hard drives. | Each object consists of data, metadata, and a unique identifier. |
| 1. It is typically used for storing data at the operating system level. | Objects are accessed through a unique URL or API. |
| 1. Block storage is suitable for applications that require low latency and high I/O performance, such as databases and virtual machines. | It is suitable for storing unstructured data such as images, videos, documents, and backups. |
| 1. You can mount block storage devices to instances as mounted block devices. | Object storage provides high durability, scalability, and availability. |
|  | Objects can be accessed over the internet from anywhere. |
|  | You can organize objects in directories or buckets. |

1. **Difference between static website & dynamic website?**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Static Website** | **Dynamic Website** | | 1. Content: Static websites have fixed content that does not change based on user input or actions. | Dynamic websites have content that can change based on user input or actions. | | 1. Hosting: Static websites are hosted on a file hosting service like AWS S3 bucket, where the files are stored and served to users as-is. | Dynamic websites require a server-side programming language and a hosting platform to generate and serve content to users dynamically. | | 1. Interactive Elements: Static websites are typically more limited in interactive elements such as forms, user logins, or personalized content. | Dynamic websites can have interactive elements like contact forms, user registrations, comments, etc., as they can execute server-side scripts to handle user inputs. | | 1. Scalability: Static websites can handle high traffic efficiently as the files are served directly from the server without processing any dynamic content. | Dynamic websites may require more resources to handle high traffic as they need to generate content dynamically for each user request. | | 1. Programming: Static websites do not require server-side programming skills, as they only consist of HTML, CSS, and JavaScript files. | Dynamic websites require server-side programming skills like PHP, Ruby, Python, etc., to handle user requests and generate content dynamically. | | 1. Maintenance: Static websites are easier to maintain as there are no server-side components or databases to manage. | Dynamic websites may require regular maintenance to update server-side scripts, databases, and handle security patches. | | 1. Speed: Static websites are generally faster as they serve pre-generated files directly from the server without any processing. | Dynamic websites may have slightly slower response times as they need to generate content on the fly. | | 1. Cost: Static websites are often cheaper to host and maintain compared to dynamic websites as they require fewer resources and server-side components | Dynamic websites may have higher hosting costs as they require server-side processing power, databases, and potentially additional services for handling dynamic content. | | 1. Flexibility: Static websites are limited in their ability to customize and tailor content to individual users | Dynamic websites offer more flexibility to personalize content and provide a dynamic user experience based on user inputs. | | 1. Examples: Personal blogs, portfolio websites, company landing pages are often static websites. | E-commerce websites, social media platforms, online booking systems are typically dynamic websites. | |

1. **What is the Naming Rules ?**
2. Bucket names must be globally unique across all AWS accounts.
3. Bucket names must be at least 3 and no more than 63 characters long.
4. Bucket names can contain lowercase letters, numbers, periods, and hyphens.
5. Bucket names must start with a lowercase letter or number.
6. Bucket names cannot contain underscores, uppercase letters, or special characters.
7. Bucket names cannot be formatted as an IP address (e.g., "192.168.5.4").
8. Bucket names cannot end with a hyphen.

It's important to choose a unique and descriptive name for your S3 Bucket as it will be part of the unique URL used to access the bucket's contents.

1. **What is the major resource of S3 Bucket ?**
2. The major resource of an S3 (Simple Storage Service) bucket in AWS (Amazon Web Services) is the ability to store and retrieve large amounts of data.
3. S3 buckets provide highly durable object storage, allowing users to store and retrieve any amount of data at any time, from anywhere on the web.
4. It can be used to store various types of data such as images, videos, documents, databases, and backups.
5. **Why do we need to host static websites instead of dynamic websites?**

There are several reasons why hosting static websites instead of dynamic websites in AWS S3 may be preferred:

1. **Cost-effective:** Static websites typically have fewer resources and infrastructure requirements compared to dynamic, resulting in lower hosting costs. AWS S3 provides a cost-effective solution for hosting static websites as it charges only for the storage and data transfer.
2. **Scalability:** AWS S3 offers excellent scalability for static websites. It can handle large amounts of traffic without compromising performance. Additionally, AWS CloudFront can be used to further enhance scalability by caching static website content across multiple edge locations worldwide.
3. **Simplicity**: Static websites are simple to deploy and manage compared to dynamic websites. There is no need for complex server-side processing or databases. AWS S3 provides a straightforward and easy-to-use solution for hosting static websites.
4. **Performance:** Static websites can offer faster page load times as there is no need to generate or fetch content dynamically. The content is already available on the server and can be directly delivered to the user's browser, resulting in a smoother and faster user experience.
5. **Security:** Hosting static websites can be more secure as there is no need for server-side scripting or database interactions that could potentially introduce security vulnerabilities. AWS S3 includes various security features, such as encryption, access control, and logging, ensuring the protection of your static website's files.
6. **What is versioning & Why do we need versioning ?**
7. Versioning is a feature in Amazon Simple Storage Service (S3) that allows you to keep multiple versions of an object in the same bucket.
8. Each time you overwrite or delete an object, a new version is created and can be retained. This ensures that every change made to an object in S3 is stored and can be accessed at any time.

**There are several reasons why versioning is important in AWS S3:**

1. **Data protection**: Versioning helps protect against accidental deletions or overwrites of important data.

If an object is mistakenly deleted or modified, you can easily restore a previous version of the object.

1. **Disaster recovery:** By enabling versioning, you can create backup copies of your data.

This allows you to recover from accidental data corruption, software bugs, or any other issues that may affect your data.

1. **Compliance and governance:** Versioning is often required for compliance with data retention policies.

It ensures that all changes made to the objects are stored, allowing for auditing and tracking of data modifications.

1. **Collaborative development:** Versioning can be beneficial in scenarios where multiple users or teams are working on the same objects.

It allows for easy tracking and management of changes made by different users.

1. **Testing and development:** Versioning provides a convenient way to test new versions of objects or roll back to previous versions if there are issues with the latest version.

Overall, versioning in AWS S3 offers **data protection, disaster recovery, compliance, collaboration, and testing benefits,** making it an essential feature for managing and safeguarding your data.

1. **What are the objects and types of objects that we are uploading into the S3 Bucket?**
2. **Text files:** This includes documents, plain text files, configuration files, etc.
3. **Image files:** This includes JPEG, PNG, GIF, or any other image format.
4. **Audio and video files: This** includes MP3, MP4, WAV, etc.
5. **Compressed files:** This includes ZIP, GZ, TAR, BZ2, etc.
6. **Data files:** This includes CSV, JSON, XML, etc.
7. **Executable files:** This includes binary files, scripts, application installers, etc.
8. **Web files:** This includes HTML, CSS, JavaScript files.
9. **Database backups:** This includes SQL dumps, database backup files, etc.
10. **Log files:** This includes server logs, application logs, etc.
11. **Virtual machine disk images**: This includes VMDK, OVA, QCOW2, etc.

These are just a few examples, and **in reality, you can upload any type of file or object** into an S3 bucket as long as it meets the size and format limitations imposed by AWS.

1. **Why is MFA Delete important in S3 Bucket object level ?**

MFA (Multi-Factor Authentication) Delete is important in AWS S3 bucket object level to provide an additional layer of security and prevent accidental or unauthorized deletions of objects stored in the bucket.

Here are some reasons why MFA Delete is important:

1. **Prevents accidental deletions:** MFA Delete requires the use of a second authentication factor (usually a virtual or hardware-based device) along with the usual username and password combination.

This makes it difficult for a user to accidentally delete objects, as they need to provide the additional authentication factor.

1. **Protects against unauthorized deletions**: MFA Delete ensures that only authorized individuals with access to the required MFA device can delete objects.

Even if an attacker gains access to the account's username and password, they would still need the MFA device to perform deletions.

1. **Safeguards against** **data loss:** Unauthorized deletions of objects can result in permanent data loss. By enabling MFA Delete, the chances of accidental or malicious deletions are greatly reduced, providing an extra layer of protection for important data stored in S3 buckets.
2. **Regulatory compliance:** Many industry regulations and security standards require the use of strong access controls and multi-factor authentication.

By enabling MFA Delete, organizations can meet these compliance requirements and ensure the security of their data.

Overall, MFA Delete enhances the security and integrity of objects stored in AWS S3 buckets by adding an extra layer of authentication for deletions, preventing accidental or unauthorized actions, and reducing the risk of data loss.

1. **What is S3 Multipart upload ?**

AWS S3 Multipart upload is a feature provided by Amazon Simple Storage Service (S3) that allows users to upload large objects (greater than 5 GB) in smaller parts.

Instead of uploading the entire object in one go, S3 Multipart upload breaks it into several parts and uploads them in parallel.

Example:

1. **Resumable uploads:** If the upload is interrupted, only the failed parts need to be re-uploaded, reducing the need to start the entire upload process again.
2. **Improved upload performance:** Multipart upload allows concurrent uploads of smaller parts, which can result in faster uploads compared to uploading the entire object at once.
3. **Handling of large objects:** Multipart upload enables the upload of objects that are larger than the size allowed by a single HTTP request (5 GB).

AWS S3 Multipart upload helps in managing the risk of network failures and service interruptions, ensuring the successful completion of large file uploads.

This feature is particularly useful for applications that deal with large files such as video streaming, backup and restore, content distribution, and big data processing.

1. **What are the storage classes in Amazon S3 ?   —------------------- IMP**

In AWS S3, there are several storage classes available. These include:

**1. Standard:** This is the default storage class in S3 and offers high durability and availability. It is suitable for frequently accessed data.

**2. Intelligent-Tiering:** This storage class automatically moves objects between two access tiers (frequent and infrequent) based on their usage patterns. It helps to optimize costs while maintaining high performance.

**3. Standard-IA (Infrequent Access):** This storage class is designed for data that is accessed less frequently but still requires immediate access when needed. It offers lower storage costs compared to Standard class.

**4. One Zone-IA:** This is similar to Standard-IA but stores data in a single availability zone, which reduces costs but may have slightly higher risk of data loss in case of zone failure.

**5. Glacier:** Glacier is a secure and durable storage class used for long-term archiving of data. It offers low storage costs but longer retrieval times (minutes to hours) compared to other classes.

**6. Glacier Deep Archive:** This is the lowest-cost storage class in S3 and is designed for long-term archival data that is rarely accessed. It has the longest retrieval times (hours) but the lowest storage costs.

Each storage class has different pricing and performance characteristics, allowing users to choose the most suitable option based on their specific requirements.

1. **What is ACL ?**
2. ACL in AWS S3 stands for Access Control List.
3. It is a security feature that allows you to manage access to your S3 buckets and objects.
4. ACLs allow you to specify permissions on a per-bucket or per-object basis, controlling which AWS accounts or groups can access your resources and what level of access they have.
5. ACLs can be configured to grant permissions such as read, write, read/write, or full control to different users or groups.
6. **Why do we need ACL ?**

ACL (Access Control List) in S3 AWS (Amazon Web Services) is required for managing access to the S3 buckets and objects within them. Here are a few reasons why ACL is necessary:

1. **Control Access:** ACL allows you to grant or deny access to S3 resources, such as buckets and objects, based on specific permissions.

You can define who can read, write, or delete objects in your S3 buckets.

It helps you enforce security and prevent unauthorized access.

1. **Fine-grained Permissions:** ACL provides granular control over access permissions by allowing you to specify access for individual AWS accounts or specific pre-defined groups.

It allows you to grant different permissions to different users or groups, ensuring that only authorized individuals can access and modify your data.

1. **Data Sharing and Collaboration:** ACL helps in easily sharing data with others by allowing you to specify access permissions for specific users or groups.

It enables controlled collaboration and data sharing without compromising security.

1. **Compliance Requirements**: Many industries have strict compliance regulations regarding data privacy and security.

ACL enables you to implement required access controls and permissions to meet compliance requirements.

It helps you demonstrate that you have proper access controls in place for your data.

1. **Object-Level Access Control:** ACL allows you to set access permissions at the object level.

This means you can control access to individual files within a bucket, providing more flexibility and control over who can access specific data within your S3 resources.

Overall, ACL in S3 AWS is crucial for maintaining data security, managing access permissions, and ensuring compliance with regulatory requirements. It helps you control and monitor who can access your S3 buckets and objects, preventing unauthorized access and data breaches.

1. **What is a Life cycle policy ? Why do we need to use the life cycle rule ?**

A life cycle policy, in the context of AWS S3 (Simple Storage Service), is a set of rules that define actions to be taken on objects stored in S3 buckets over time.

The life cycle policy allows you to automate the management of your objects, including actions like transitioning to different storage classes or deleting them after a certain period.

1. **Cost optimization:** S3 offers different storage classes with varying costs.

By using life cycle rules, you can automatically transition objects to less expensive storage classes based on their age or usage patterns.

This helps in reducing storage costs.

1. **Data retention and compliance:** Life cycle rules enable you to enforce data retention by automatically deleting or archiving objects based on specified time periods.

This ensures compliance with data retention policies or regulatory requirements.

1. **Efficient data organization:** By setting life cycle rules, you can automatically organize your data by specifying folders or prefixes for objects to be moved or deleted. This helps in maintaining a well-structured and organized data repository.
2. **Performance optimization:** Life cycle policies allow you to optimize performance by automatically moving infrequently accessed objects to lower-cost storage or deleting unnecessary objects.

This helps improve access latency for frequently accessed data.

Overall, life cycle policies in AWS S3 help in automating the management of your objects, **reducing costs, ensuring data compliance, optimizing performance, and maintaining an organized data repository**.

1. **How can we make our bucket public ?**

To make a bucket public, follow these steps:

1. Open the Amazon S3 Management Console.
2. Select the bucket you want to make public.
3. Click on the "Permissions" tab.
4. Under the "Public access" section, click on the "Edit" button.
5. Choose the option "Public" for the "Block public access" settings. This allows for public access to the bucket and its contents.
6. Scroll down to the "Access control list (ACL)" section.
7. Click on the "Everyone" group and grant the appropriate permissions (e.g., read, write, or full control) based on your requirements.
8. Click on the "Save changes" button to apply the changes.
9. **How can we give public access to our bucket ?**
10. **Aws pricing factor of the S3 Service?**
11. **Storage usage:** The amount of data stored in Amazon S3, measured in gigabytes (GB) or terabytes (TB).

There are different tiers available based on the frequency of access, such as Standard, Intelligent-Tiering, Glacier, and Glacier Deep Archive.

1. **Data transfer:** The amount of data transferred in and out of Amazon S3.

This includes data uploads, downloads, and data transfer between different regions or availability zones.

1. **Requests:** The number of requests made to Amazon S3, including PUT, GET, COPY, and DELETE requests. This factor includes both standard and Glacier requests.
2. **Additional features**: Some additional features, such as data transfer acceleration, versioning, and cross-region replication, may incur additional costs.

It's worth noting that Amazon S3 has a tiered pricing structure, meaning that the pricing per GB or TB decreases as you store more data.

For specific and up-to-date pricing details, it's recommended **to refer to the official Amazon S3 pricing page.**

1. **How can we make our object public?**
2. Go to the AWS Management Console and open the S3 service.
3. Select the bucket containing the object you want to make public.
4. Navigate to the object by clicking on the bucket name and then the desired folder (if applicable).
5. Select the checkbox next to the object you want to make public.
6. Click on the "Actions" dropdown menu and select "Make Public".
7. A dialog box will appear asking for confirmation. Click "Make Public" to confirm the action.
8. The selected object is now public, and you can access it using the object's URL or by using the S3 API.
9. **How can we configure the static website logs in s3 ?**
10. Open the AWS Management Console and navigate to the S3 service.
11. Choose the bucket that you want to enable logging for and click on its name.
12. In the bucket properties, click on the "Properties" tab.
13. Scroll down to the "Static website hosting" section and click on the "Edit" button.
14. In the "Static website hosting" dialog box, scroll down and find the "Logging" section.
15. Enable the "Enable access logs" option by selecting the checkbox.
16. Specify the target bucket where you want to store the access logs. This can be the same bucket or a different one. Note that the target bucket must in the same AWS region as the source bucket.
17. Optionally, specify a prefix for the log files to easily identify them.
18. Click on the "Save changes" button to apply the configuration.

Once the logging configuration is saved, S3 will start writing access logs to the specified bucket. The logs contain information about every request made to the static website, such as the request time, request source, HTTP status code, and more.

1. **What is CORS ?**
2. CORS (Cross-Origin Resource Sharing) is a mechanism that allows a web page or a client, executing in a browser, to request resources from a different domain.
3. In the context of AWS S3 (Simple Storage Service), CORS is an S3 bucket-level configuration that controls which origins (domains) are allowed to access the resources (files) stored in the bucket.

By default, S3 buckets do not allow cross-origin requests.

1. To enable CORS in AWS S3, you need to define a CORS configuration for your bucket. The configuration is an XML document that specifies the rules for allowing cross-origin requests.

These rules typically include the list of allowed origins, HTTP methods, headers, and other parameters.

1. Once the CORS configuration is applied to an S3 bucket, the bucket will respond to the requests received from the allowed origins according to the defined rules.

This enables secure and controlled access to the bucket's resources from different domains.

1. **What is S3 Inventory?**
2. S3 Inventory in AWS S3 is a feature that provides a detailed inventory of objects stored in an S3 bucket.
3. It provides a report in CSV or ORC (Apache Hive-compatible open-source columnar storage format) format, containing information such as object metadata, size, and encryption status.
4. S3 Inventory can help with compliance, security, and inventory management tasks by providing insights into the objects stored in an S3 bucket.

It can be scheduled to generate reports at specific intervals and can be integrated with other AWS services for further analysis and automation.

1. **What does it mean by Requester pays ?**
2. "Requester pays" in AWS S3 (Simple Storage Service) means that the person or entity making the request for data stored in S3 is responsible for paying the data transfer costs associated with retrieving the data.
3. Normally, the account that owns the S3 bucket where data is stored pays for all data transfer costs.

However, with the "requester pays" option enabled, the requestor takes on the cost of transferring the data out of the S3 bucket.

1. **What is the secondary word to Transfer acceleration ?, why we need to use this transfer acceleration ?**
2. The secondary word for Transfer acceleration in AWS S3 is "speed".
3. Uses:

We need to use Transfer acceleration in AWS S3 to improve the speed of uploading and downloading objects to and from S3 buckets.

It achieves this by utilizing the Amazon CloudFront content delivery network (CDN) globally distributed edge locations.

When Transfer acceleration is enabled, it routes the S3 data transfer through an optimized network path, thereby accelerating the transfer speeds.

This can be particularly useful when there is a need for faster data transfers over long distances or for users located far from the S3 bucket's region.