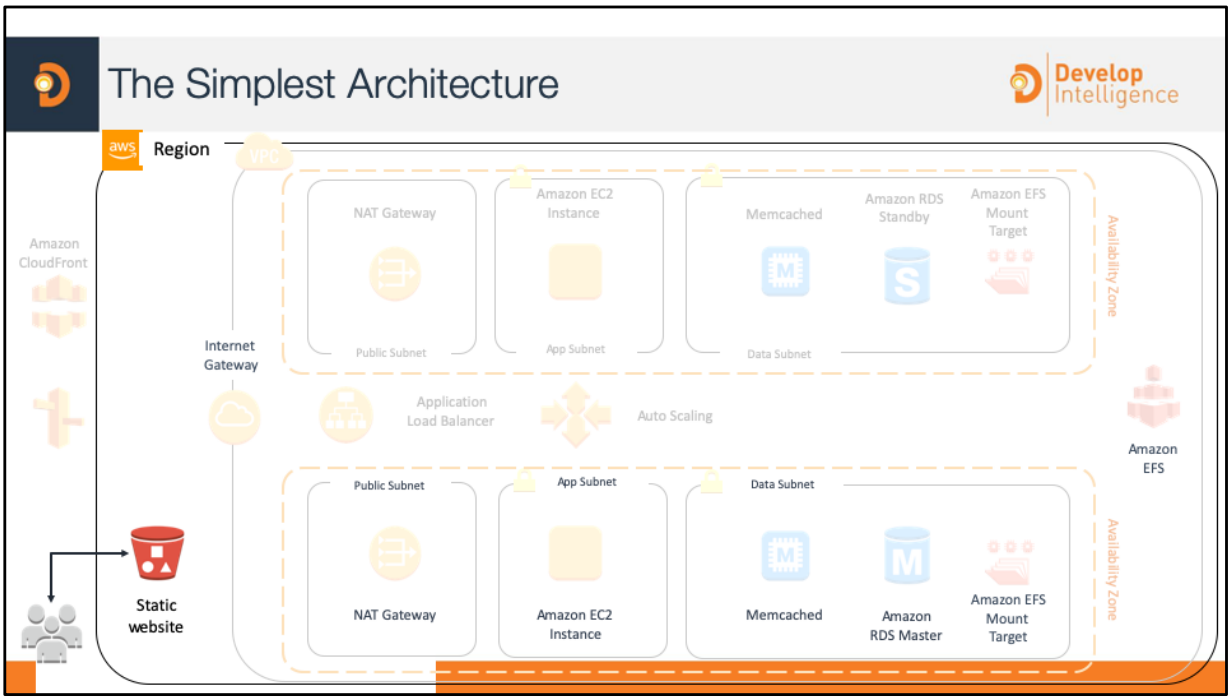
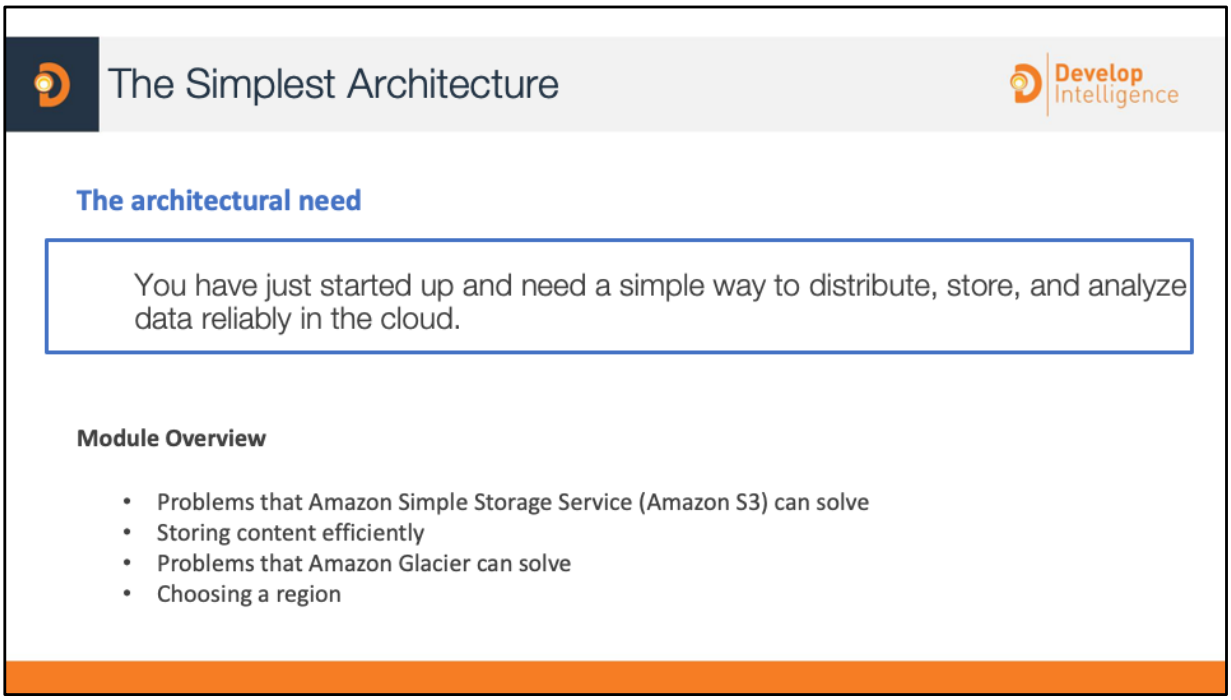
The Simplest Architectures

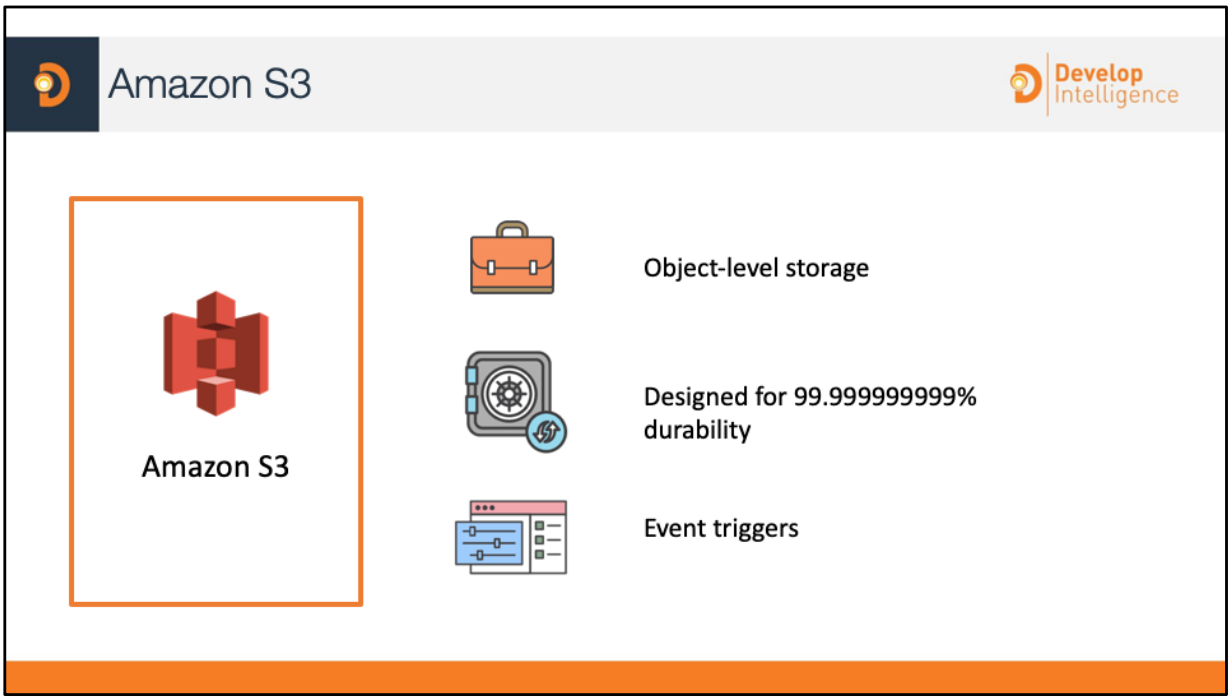
Develop Intelligence

- Module 2



By the end of class, you will be able to understand all of the components of this architectural diagram. You will also be able to construct your own architectural solutions that are just as large and robust.


The Simplest Architecture
Develop Intelligence
The architectural need
You have just started up and need a simple way to distribute, store, and analyze data reliably in the cloud.
Module Overview
• Problems that Amazon Simple Storage Service (Amazon S3) can solve
• Storing content efficiently
• Problems that Amazon Glacier can solve
• Choosing a region



Amazon S3 is *object-level storage*, which means that if you want to change a part of a file, you have to make the change and then re-upload the entire modified file.

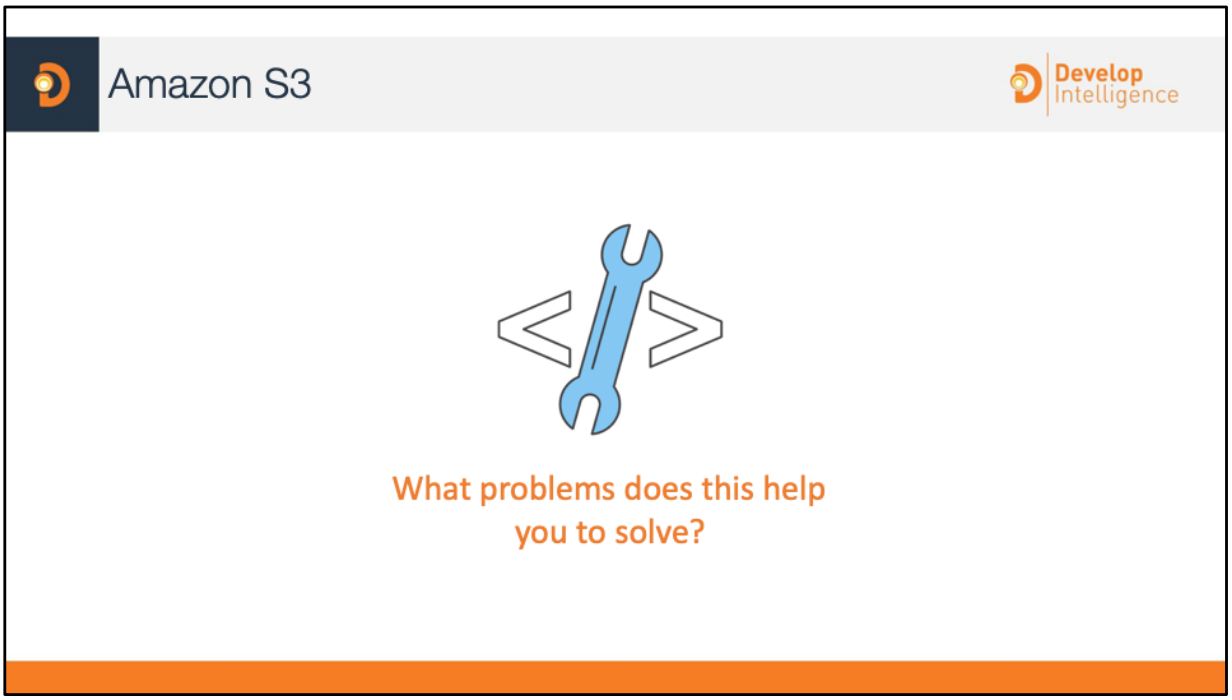
Amazon S3 allows you to store as much data as you want. Individual objects cannot be larger than 5 TB; however, you can store as much total data as you need.

By default, data in Amazon S3 is stored redundantly across multiple facilities and multiple devices in each facility.

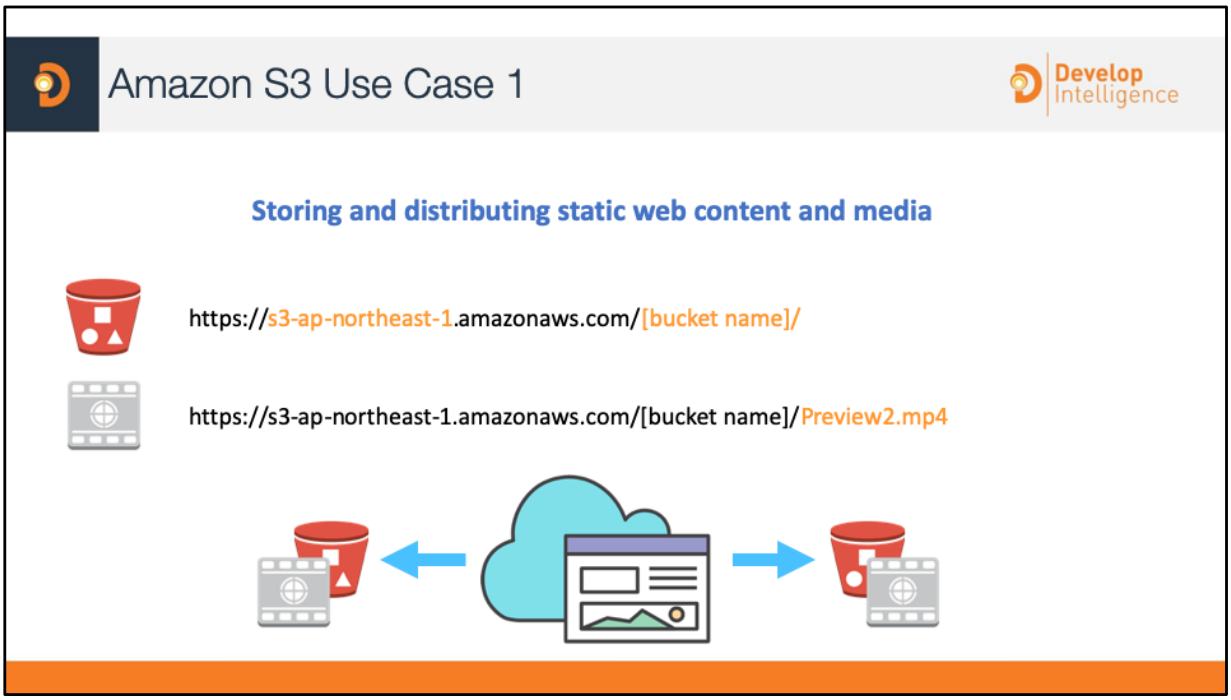
Amazon S3 can be accessed via the web-based AWS Management Console, programmatically via the API and SDKs, or with third-party solutions (which use the API/SDKs).

Amazon S3 includes *event notifications* that allow you to set up automatic notifications when certain events occur, such as an object being uploaded to or deleted from a specific bucket. Those notifications can be sent to you, or they can be used to trigger other processes, such as AWS Lambda scripts.

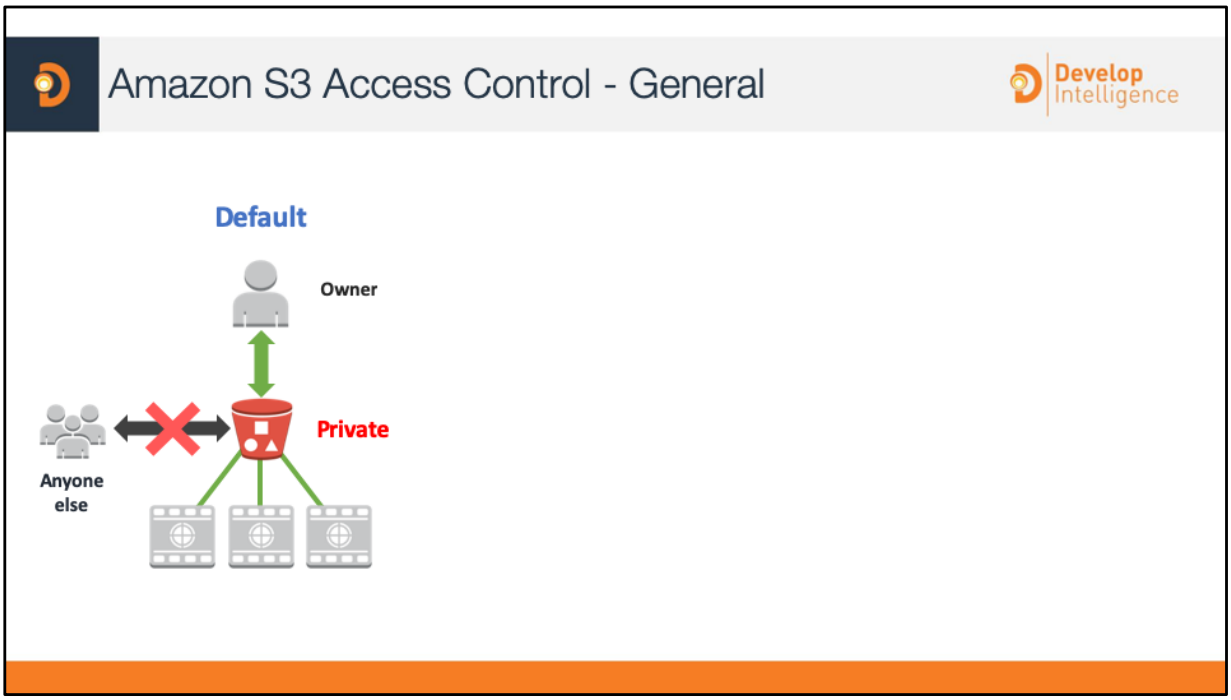
With *storage class analysis*, you can analyze storage access patterns and transition the right data to the right storage class. This new S3 Analytics feature automatically identifies the optimal lifecycle policy to transition less frequently accessed storage to Amazon S3 Standard-Infrequent Access (S3 Standard-IA). You can configure a storage class analysis policy to monitor an entire bucket, a prefix, or object tag. Once an infrequent access pattern is observed, you can easily create a new lifecycle age policy based on the results. Storage class analysis also provides daily visualizations of your storage usage in the AWS Management Console. You can export these to an S3 bucket to analyze using the business intelligence tools of your choice, such as Amazon QuickSight.



So how can you use these features of Amazon S3 to address your needs?



First, you can use Amazon S3 to store and distribute static web content or media. These files can be delivered directly from Amazon S3 because each object is associated with a unique HTTP URL. Amazon S3 can also be used as an origin for a content delivery network (such as Amazon CloudFront). Amazon S3 works well for fast-growing websites that require strong elasticity. This might include workloads with large amounts of user generated content, such as video or photo sharing.



By default, all Amazon S3 resources—buckets, objects, and related sub-resources (for example, lifecycle configuration and website configuration)—are private: only the resource owner, an AWS account that created it, can access the resource. The resource owner can grant access permissions to others by writing an access policy.

Module 7 covers AWS Identity and Access Management (IAM) vs. access control lists (ACL) and bucket policies. For more information about access control in Amazon S3, see https://docs.aws.amazon.com/AmazonS3/latest/dev/access-control-overview.html

Amazon S3 buckets are protected by default. The only entities with access to a newly created, unmodified bucket are the account administrator and root user.
Modifications to bucket policies can enable additional access, and AWS provides a number of different tools to enable developers to configure buckets for a wide variety of workloads. Amazon S3 includes a “block public access” feature, which acts as an additional layer of protection to prevent accidental exposure of customer data.
In the public access settings for a bucket, customers can specify the following four options. All options are enabled by default.

- Block new public ACLs and uploading public objects.

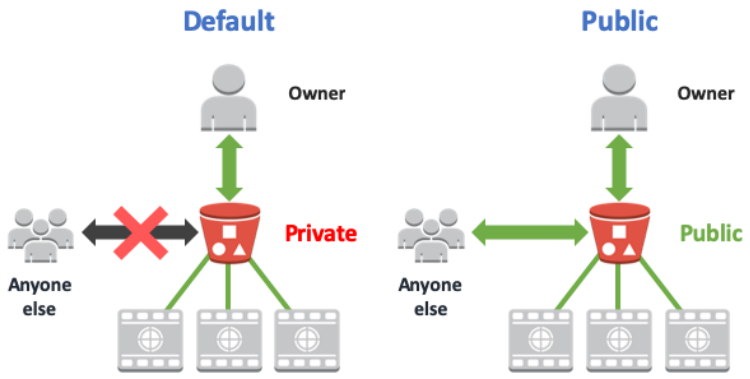
- Remove public access granted though public ACLs.
- Block new public bucket policies.
- Block public and cross-account access to buckets that have public policies.

Public Access Settings
These settings must be manually disabled for public access settings, such as with a public, static website.
Possible demo to show creating a public bucket and attempting to access a file, then disabling the block public access settings to show the file now being downloadable.

https://aws.amazon.com/blogs/aws/amazon-s3-block-public-access-another-layer-of-protection-for-your-accounts-and-buckets/

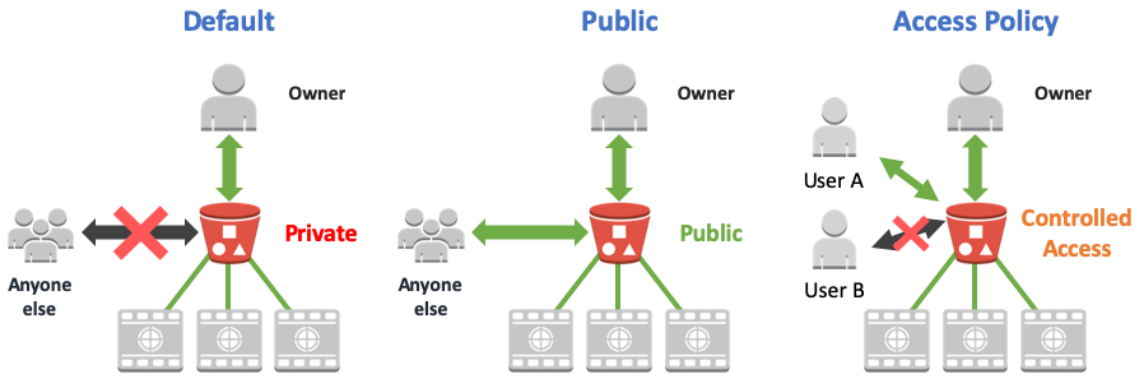
Amazon S3 Access Control - General

Develop Intelligence

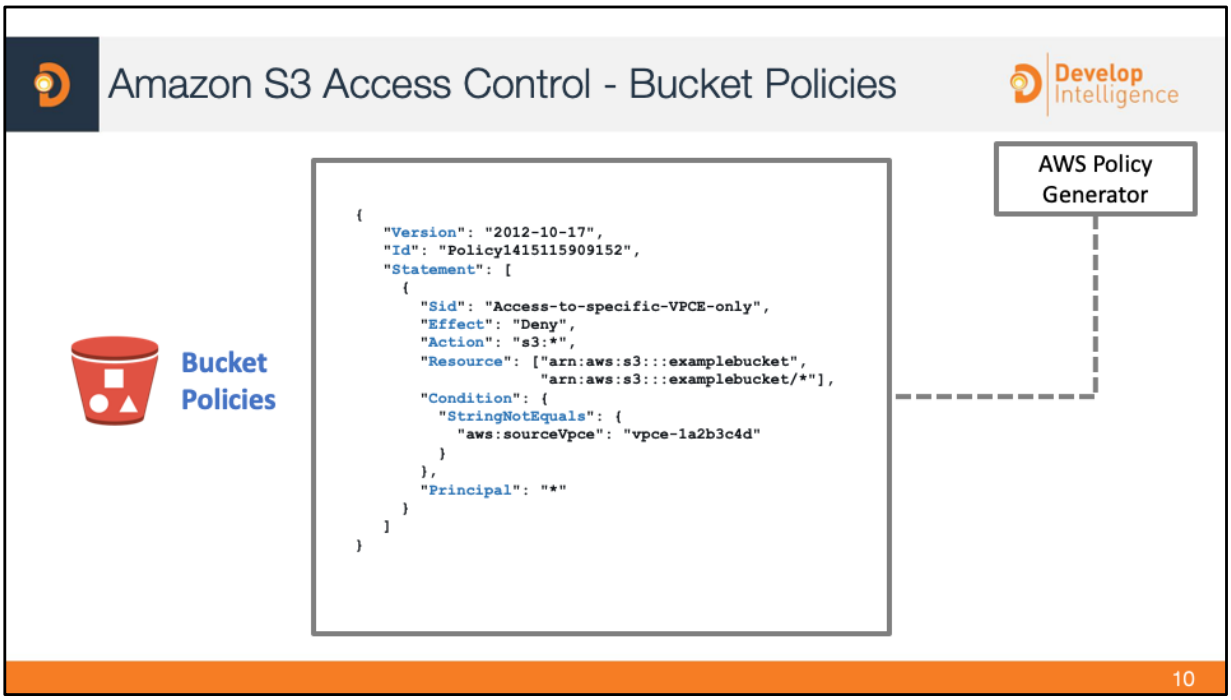


Amazon S3 Access Control - General

Develop Intelligence

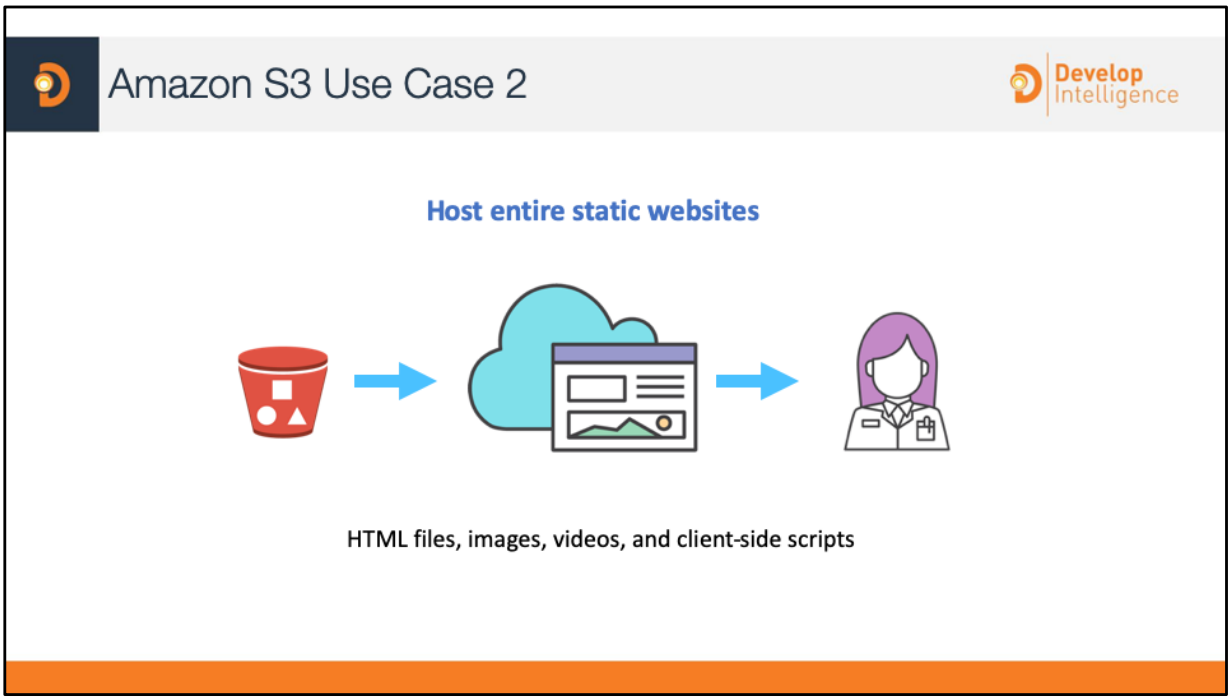


All buckets have an ACL associated with them. This ACL is a list of permissions for various grantees. You can allow read/write permissions to other AWS accounts through the Amazon S3 XML schema.

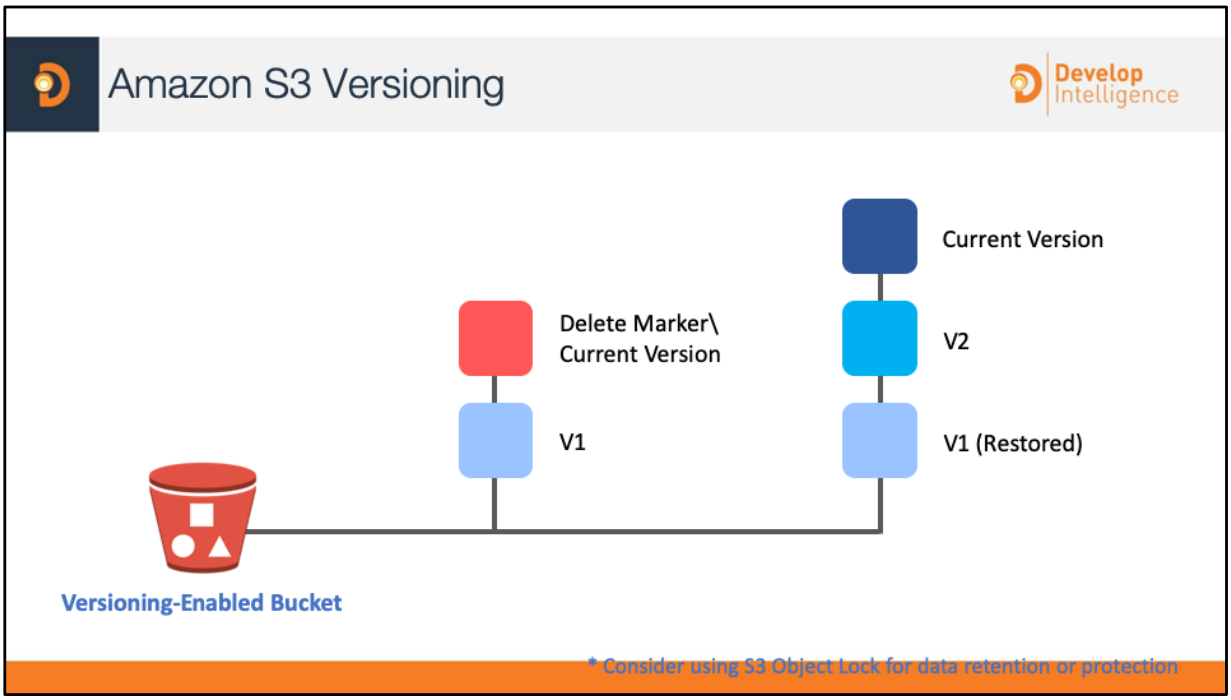


In your S3 buckets, you can add polices to allow other AWS accounts or users to access the objects stored within. Bucket policies can supplement and, in some cases, replace standard ACL access policies.

Bucket policies are limited to 20 KB in size.



You can use Amazon S3 to host entire static websites. Amazon S3 provides a low-cost, highly available, and highly scalable solution, including storage for static HTML files, images, videos, and client-side scripts in formats such as JavaScript.



Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite. For example:

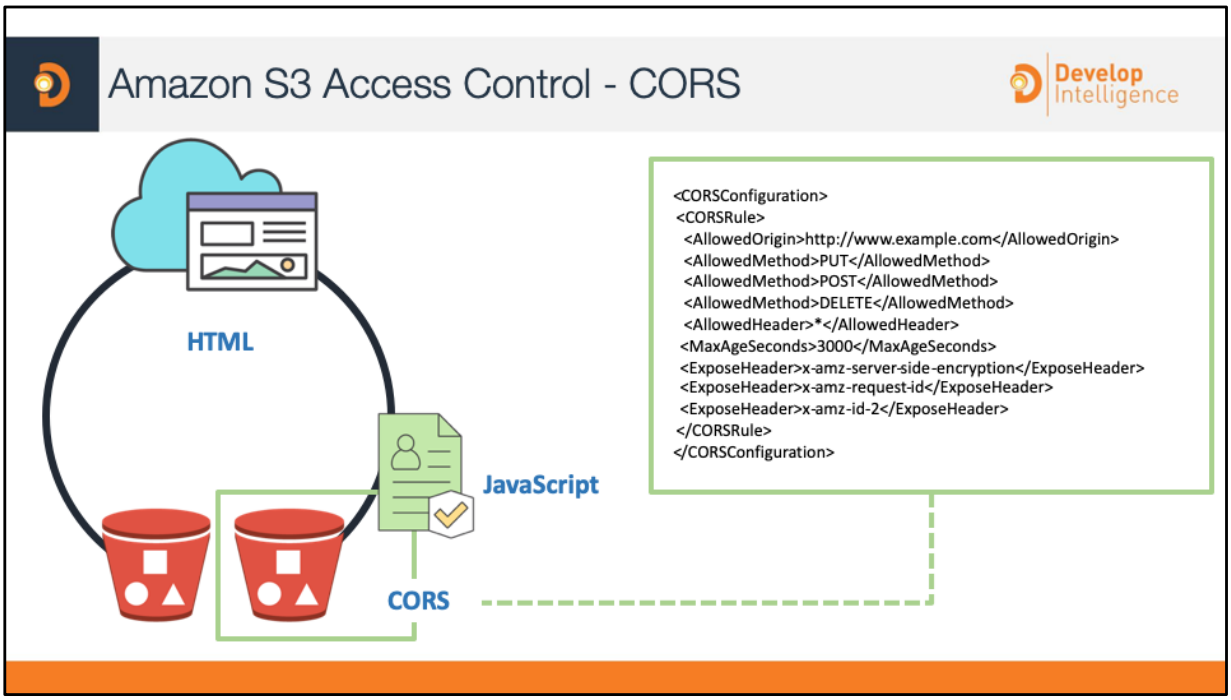
- If you delete an object, instead of removing it permanently, Amazon S3 inserts a *delete marker*, which becomes the current object version. You can always restore the previous version.
- If you overwrite an object, it results in a new object version in the bucket. You can always restore the previous version.

For more information about versioning, see https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html

You can use S3 Object Lock for data retention or protection. By using the Write-Once-Read-Many (WORM) model, you can prevent accidental overwrites or deletions within S3 storage.

Use Retention Periods for locking an object for a fixed period of time, or a Legal Hold for a lock until explicitly removed.

This feature works only on versioned buckets with the retention periods and legal holds applying to individual object versions and Amazon S3 stores the lock information in the metadata for that object version. This does not prevent a new version from being created. Object Lock helps comply with **SEC 17a-4, CTCC, and FINRA.**
https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lock.html

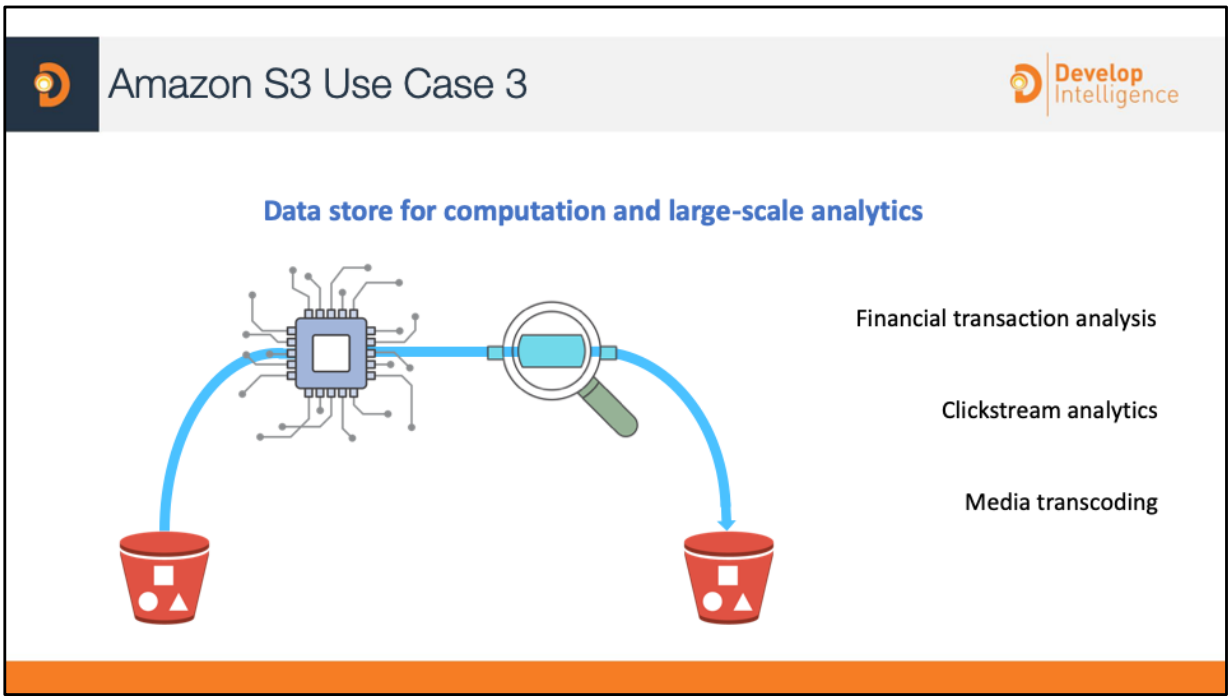


Cross-origin resource sharing (CORS) defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. With CORS support, you can build rich client-side web applications with Amazon S3 and selectively allow cross-origin access to your Amazon S3 resources.

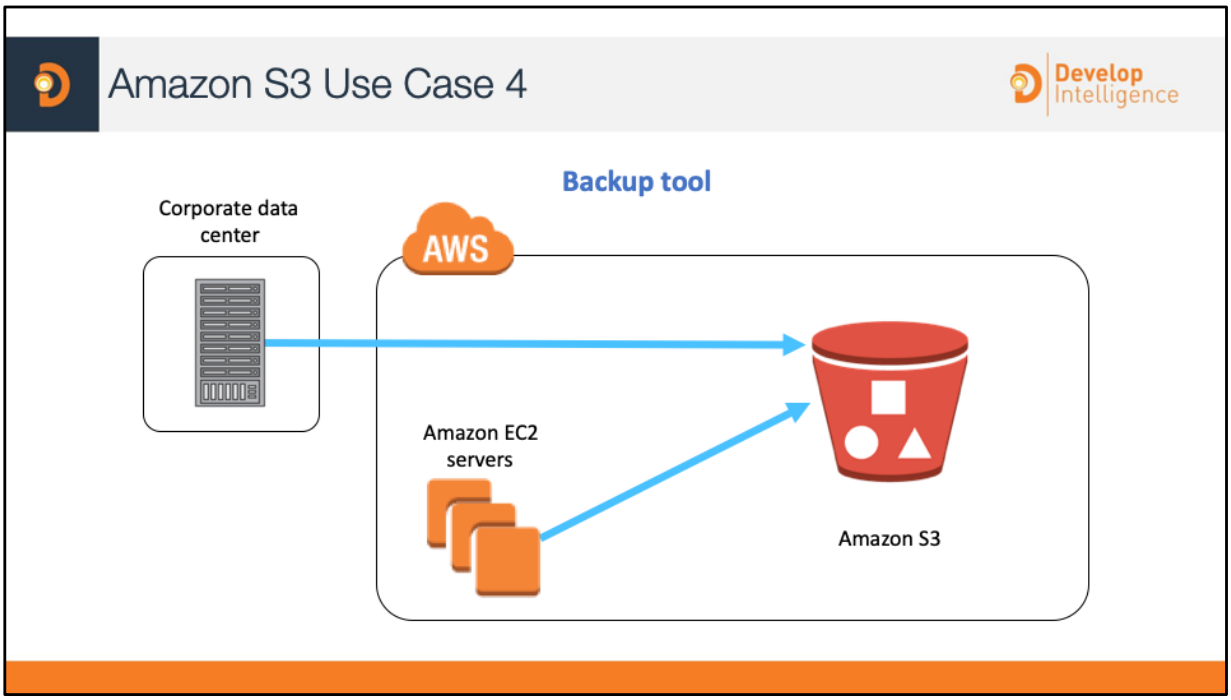
To configure your bucket to allow cross-origin requests, you create a CORS configuration, which is an XML document with rules that identify:

- The origins that you will allow to access your bucket.
- The operations (HTTP methods) that will support for each origin.
- Other operation-specific information.

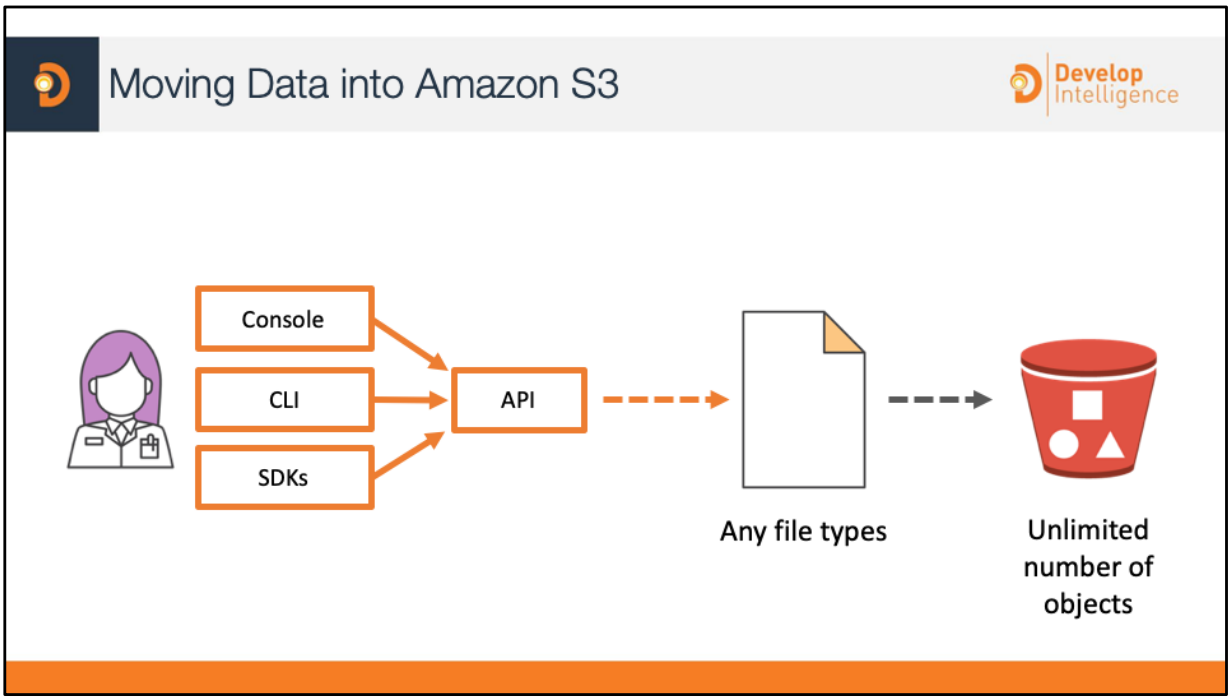
For more information about CORS, see
https://docs.aws.amazon.com/AmazonS3/latest/dev/cors.html



You can also use Amazon S3 as a data store for computation or large-scale analytics, such as financial transaction analysis, clickstream analytics, and media transcoding. Amazon S3 can support these workloads because of its horizontal scaling ability, which easily allows multiple concurrent transactions.



Because of its highly durable and scalable nature, Amazon S3 also works well as a backup and archival tool. Additionally, you can move long term data in to Amazon Glacier through the use of lifecycle policies. For more durability, you can use cross-region replication to automatically copy objects into other Amazon S3 buckets in different regions.



When you upload a file to Amazon S3, it is stored as an S3 object. Objects consist of the file data and metadata that describes the object. A bucket can hold an unlimited number of objects.

You can move data into Amazon S3 in a few different ways:

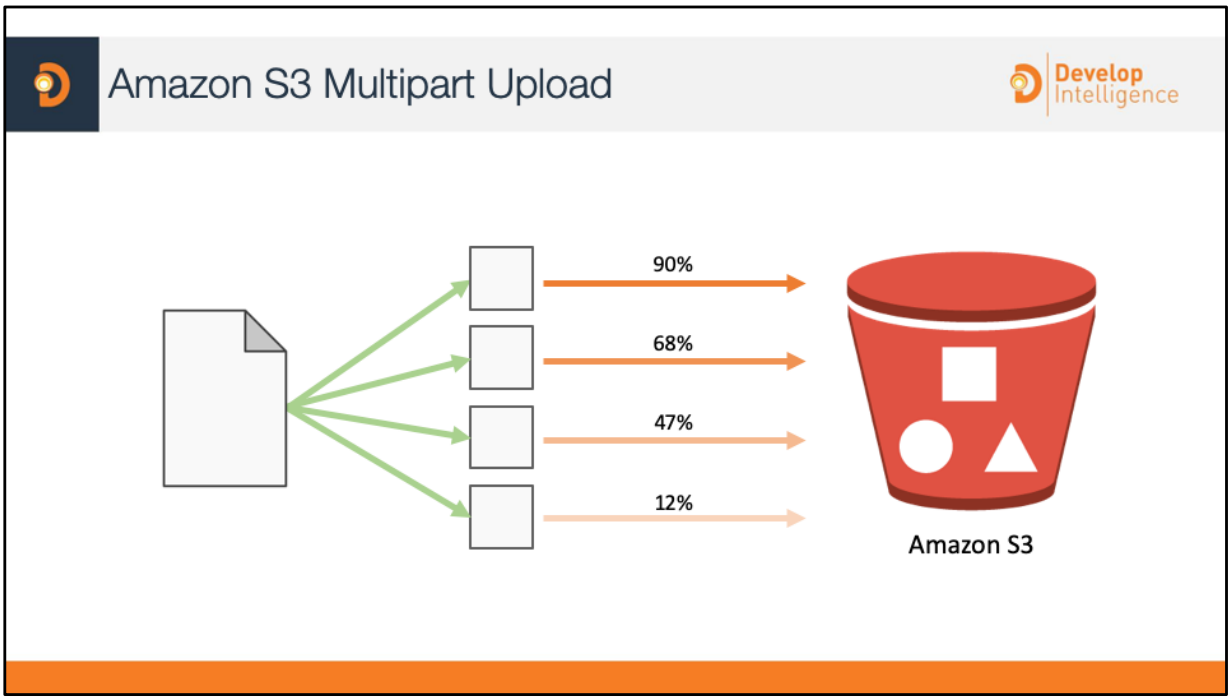
- **Transfer it using the console, AWS Command Line Interface (CLI), or API.** If you have small amounts of data, or data that is already within the AWS network, you can transfer it into Amazon S3 easily by using the console, CLI or API.
- **Upload it into an S3 bucket.** You can upload any file type—images, backups, data, movies, etc.—into an S3 bucket. The maximum size of a file that you can upload by using the Amazon S3 console is 78 GB. Using the CLI or the API will allow you to move more.
- **AWS DataSync** is a data transfer service that makes it easy for you to automate moving data between on-premises storage and Amazon S3 or Amazon Elastic File System (Amazon EFS).
- **AWS Transfer for SFTP** is a fully-managed, highly-available Secure File Transfer Protocol, or SFTP, service that enables applications to transfer files over SFTP directly into Amazon S3.

You can use DataSync to transfer your data up to 10 times faster than open-source tools while DataSync also automatically handles many tasks that can slow down migrations, including running your own instances, handling encryption, managing scripts, network optimization, and data integrity validation.

DataSync uses an on-premises software to connect to your existing storage or files systems using NFS protocol and you pay only for the data you copy.
https://aws.amazon.com/datasync/

AWS Transfer for SFTP is a fully-managed, highly-available Secure File Transfer Protocol, or SFTP, service that enables applications to transfer files over SFTP directly into Amazon S3. You create a server, set up user accounts, and associate the server with one or more Amazon S3 buckets. Your customers and your partners continue to connect and make transfers as usual, with no changes to their existing workflows. Some of the other benefits include having control over user identity, permissions, and keys; migrating to AWS Transfer for SFTP by using your existing DNS name and SSH public keys; and writing AWS Lambda functions to build an “intelligent” FTP site for processing and querying files.

https://aws.amazon.com/blogs/aws/new-aws-transfer-for-sftp-fully-managed-sftp-service-for-amazon-s3/



Multipart upload enables you to consistently upload large objects in manageable parts. This process involves three steps:

- Initiating the upload
- Uploading the object parts
- Complete the multipart upload

Once the multipart upload request is completed, Amazon S3 will recreate the full object from the individual pieces.

Here's how this benefits you:

Improved throughput: You can upload parts in parallel to improve throughput.

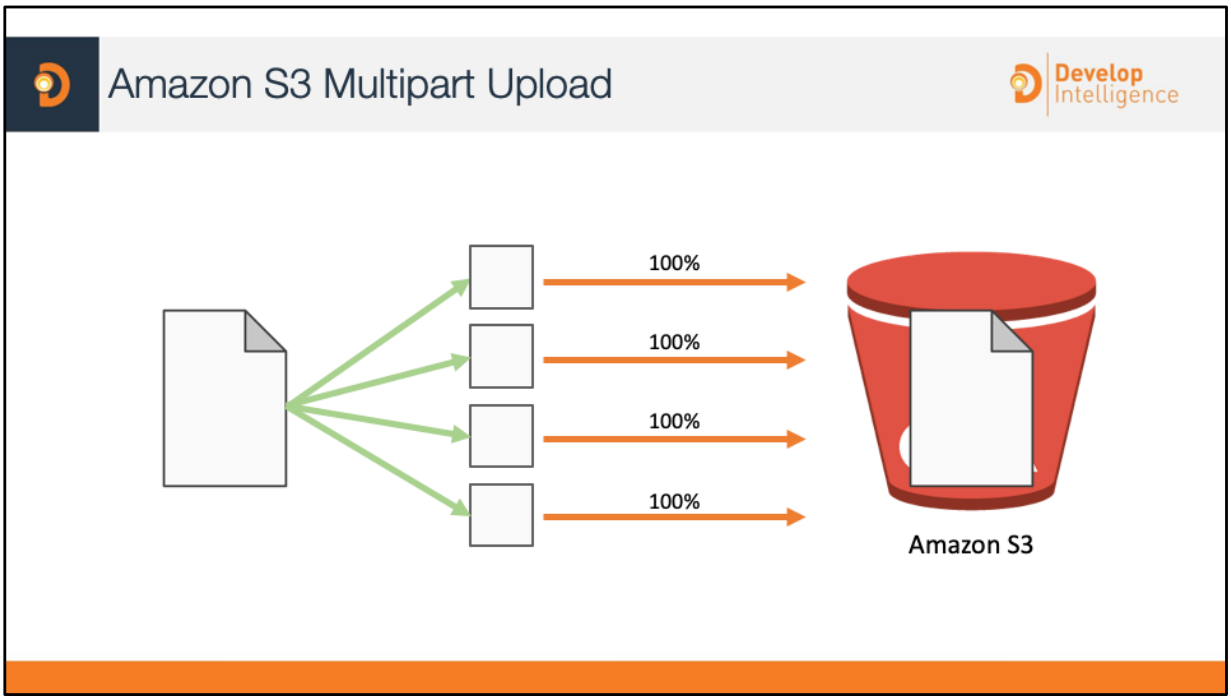
Quick recovery from any network issues: Smaller part sizes minimize the impact of restarting a failed upload due to a network error.

Pause and resume object uploads: You can upload object parts over time. Once you initiate a multipart upload, there is no expiry; you must explicitly complete or abort the multipart upload.

Begin an upload before you know the final object size: You can upload an object as you are creating it.

Upload large objects: Using the multipart upload API, you can upload large objects, up to 5 TB.

For more information about multipart uploads, see https://docs.aws.amazon.com/AmazonS3/latest/dev/mpuoverview.html



Multipart upload enables you to consistently upload large objects in manageable part. This process involves three steps: you initiate the upload, you upload the object parts, and after you have uploaded all the parts, you complete the multipart upload. Once the multipart upload request is completed, Amazon S3 will recreate the full object from the individual pieces.

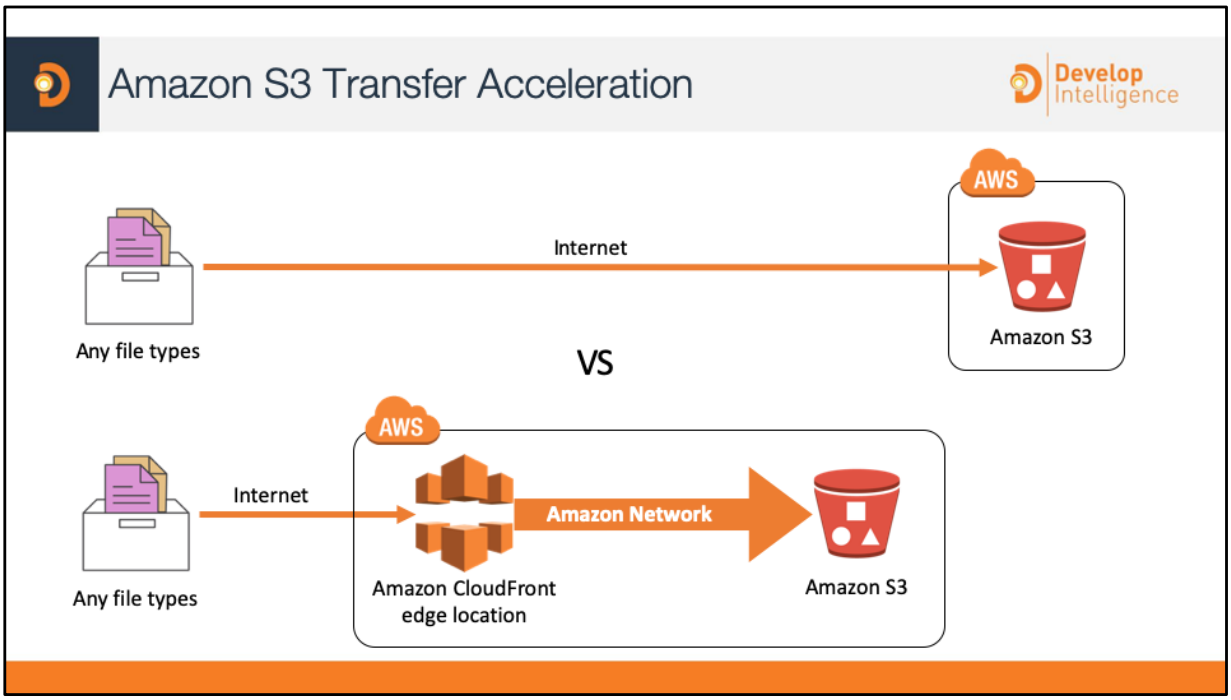
Improved throughput - You can upload parts in parallel to improve throughput.
Quick recovery from any network issues - Smaller part size minimizes the impact of restarting a failed upload due to a network error.

Pause and resume object uploads - You can upload object parts over time. Once you initiate a multipart upload there is no expiry; you must explicitly complete or abort the multipart upload.

Begin an upload before you know the final object size - You can upload an object as you are creating it.

Using the multipart upload API, you can upload large objects, up to 5 TB.

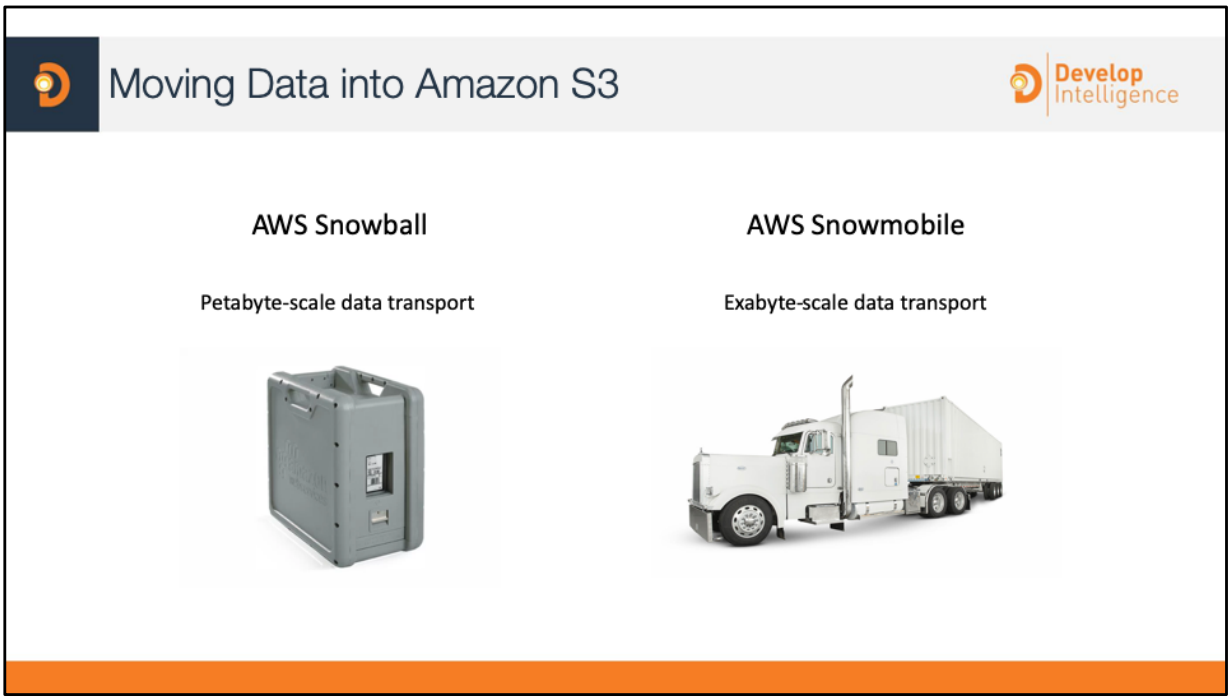
https://docs.aws.amazon.com/AmazonS3/latest/dev/mpuoverview.html



Amazon S3 Transfer Acceleration allows for fast and easy data transfer into an S3 bucket by taking advantage of Amazon CloudFront's globally distributed edge locations. This data is then routed to Amazon S3 over an optimized network path.

Use Transfer Acceleration when you…

- Have customers all over the world who upload to a centralized bucket.
- Transfer gigabytes or terabytes of data across continents on a regular basis.
- Underutilize the available bandwidth when uploading to Amazon S3 over the internet.



AWS Snowball is a petabyte-scale data transport option that doesn't require you to write any code or purchase any hardware to transfer your data. All you need to do is create a job in the AWS management console and a Snowball appliance will be shipped to you. Simply attach the appliance into your local network and transfer the files directly onto it. Once completed, the E ink shipping label will automatically update and can be tracked via Amazon Simple Notification Service (Amazon SNS) or in the console. The Snowball will then be shipped back into a secure Amazon facility and transferred into the network.

AWS Snowball Edge Optimized is ideal for edge processing usage cases that require additional computing power in remote, disconnected, or harsh environments. This service provides 52 vCPUs, 208 GB of memory, 7.68TB of NVMe SSD, and 42 TB of S3-compatible storage. Typical usage scenarios include advanced machine learning and full-motion video analysis in disconnected environments.

For more information about Snowball, see https://aws.amazon.com/snowball/ https://aws.amazon.com/snowball-edge/

AWS Snowmobile is an even larger data transfer option that operates in exabyte-scale. It should only be used to move extremely large amounts of data into AWS. A Snowmobile is 45-foot-long ruggedized shipping container that is pulled by a semi-trailer truck. You can transfer 100 PB per Snowmobile.

Snowmobile uses multiple layers of security designed to protect your data, including dedicated security personnel, GPS tracking, alarm monitoring, 24/7 video surveillance, and an optional escort security vehicle while in transit. All data is encrypted with 256-bit encryption keys managed through the AWS Key Management Service (AWS KMS) and designed to ensure both security and full chain-of-custody of your data.

For more information about Snowmobile, see https://aws.amazon.com/snowmobile/

When Should You Use Amazon S3?

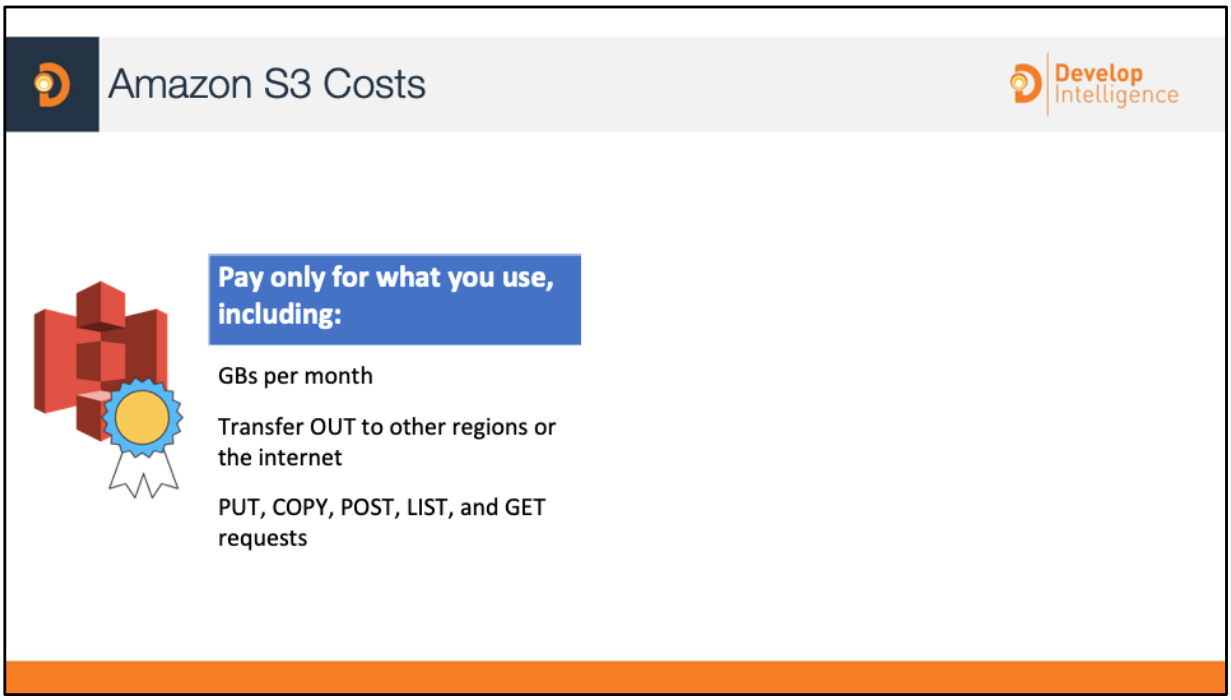
Good use cases
When you need to write once, read many times
Spiky data access
Large number of users and diverse amounts of content
Growing data sets

When Should You Use Amazon S3?

Develop Intelligence

Good use cases
When you need to write once, read many times
Spiky data access
Large number of users and diverse amounts of content
Growing data sets

Not ideal use cases
Block storage requirements
Frequently changing data
Long-term archival storage



Specific costs may vary depending on region and the specific requests made. As a general rule, you only pay for transfers that cross the boundary of your region, which means you do not pay for transfers to Amazon CloudFront edge locations within that same region.

Amazon S3 Costs

Develop Intelligence

Pay only for what you use, including:

GBs per month

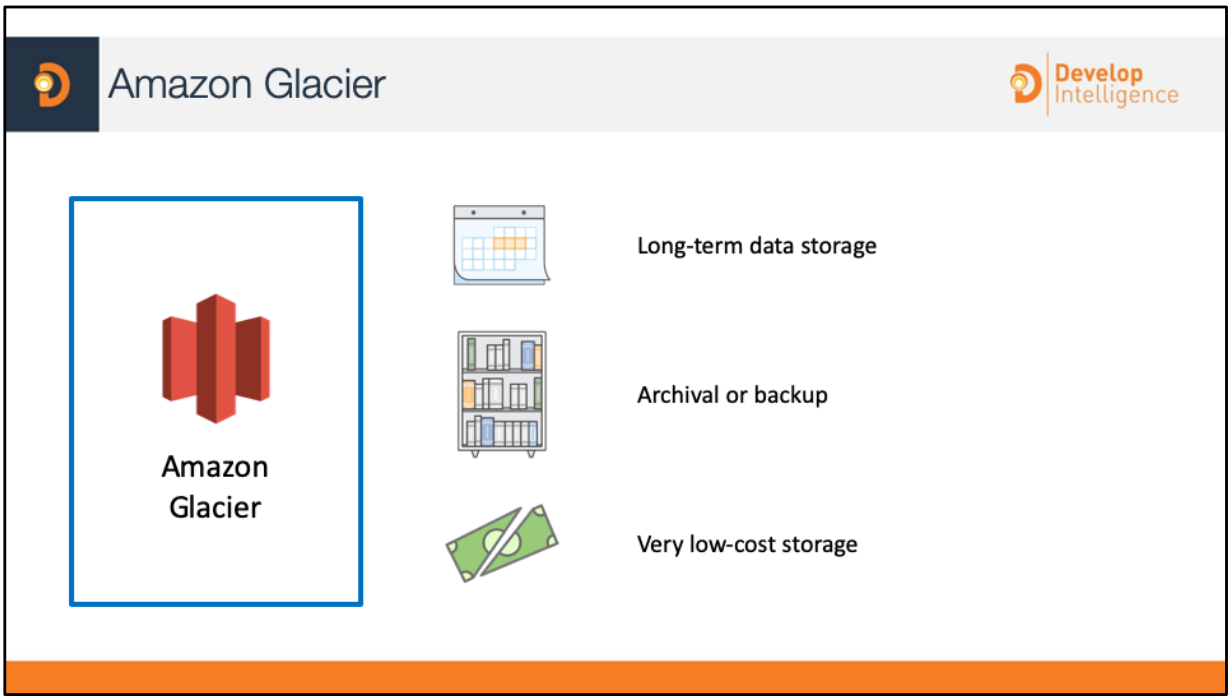
Transfer OUT to other regions or the internet

PUT, COPY, POST, LIST, and GET requests

You do NOT have to pay for:

Transfer IN to Amazon S3

Transfer OUT to Amazon EC2 in the same region, or to CloudFront



Amazon Glacier is a great storage choice when low storage cost is paramount, your data is rarely retrieved, and retrieval latency of several hours is acceptable. If your application requires fast or frequent access to your data, consider using Amazon S3.

Amazon Glacier's data archiving means that although you can store your data at an extremely low cost (even in comparison to Amazon S3), you cannot retrieve your data immediately when you want it. Data stored in Amazon Glacier takes several hours to retrieve, which is why it's ideal for archiving.

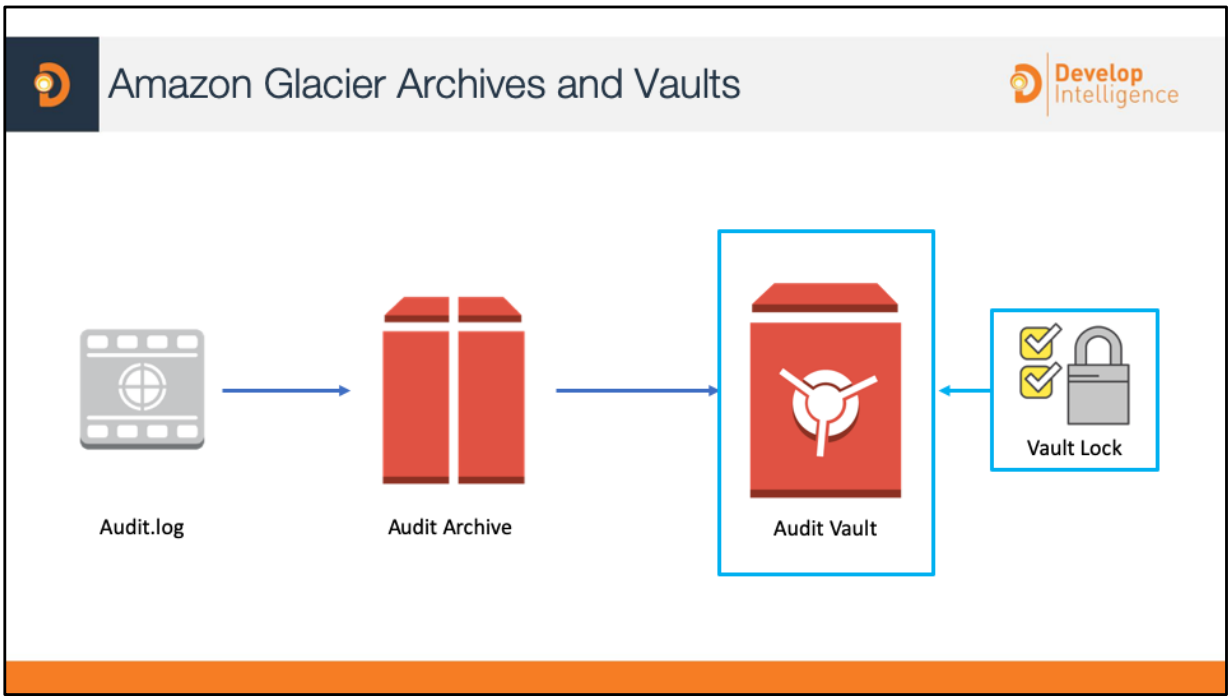
You have three options for retrieving data with varying access times and cost:

- **Expedited** retrievals are typically made available within 1 – 5 minutes.
- **Standard** retrievals typically complete within 3 – 5 hours.
- **Bulk** retrievals typically complete within 5 – 12 hours.

A few more details:

- Amazon Glacier is a **data archiving service** designed for **security**, **durability**, and **extremely low cost**.

- Designed for durability of 99.999999999% of objects.
- Supports SSL/TLS encryption of data in transit and at rest.
- The Vault Lock feature enforces compliance via a lockable policy.
- Extremely low-cost design is ideal for long-term archiving.

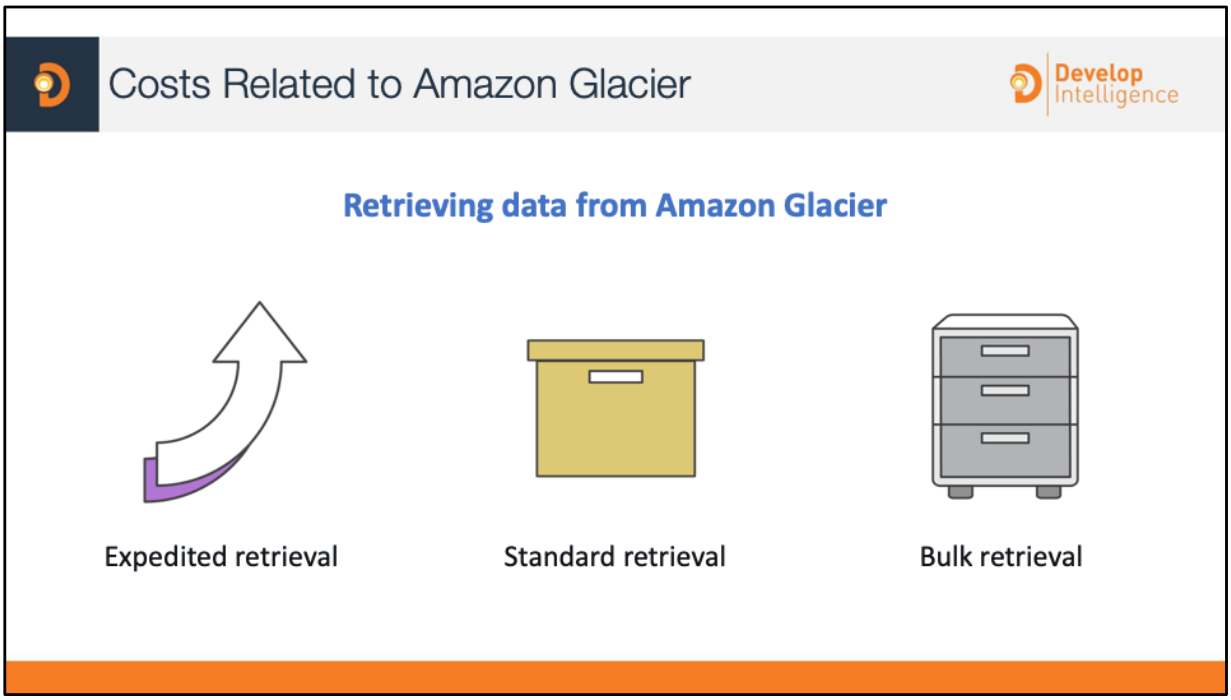


An *archive* is any object, such as a photo, video, or document, that you store in a vault. It is a base unit of storage in Amazon Glacier. Each archive has a unique ID and an optional description. When you upload an archive, Amazon Glacier returns a response that includes an archive ID. This archive ID is unique in the region in which the archive is stored.

Amazon Glacier provides a management console. You can use the console to create and delete vaults. However, all other interactions with Amazon Glacier require that you use the CLI or write code. For example, to upload data, such as photos, videos, and other documents, you must either use the AWS CLI or write code to make requests, using either the REST API directly or by using the AWS SDKs.

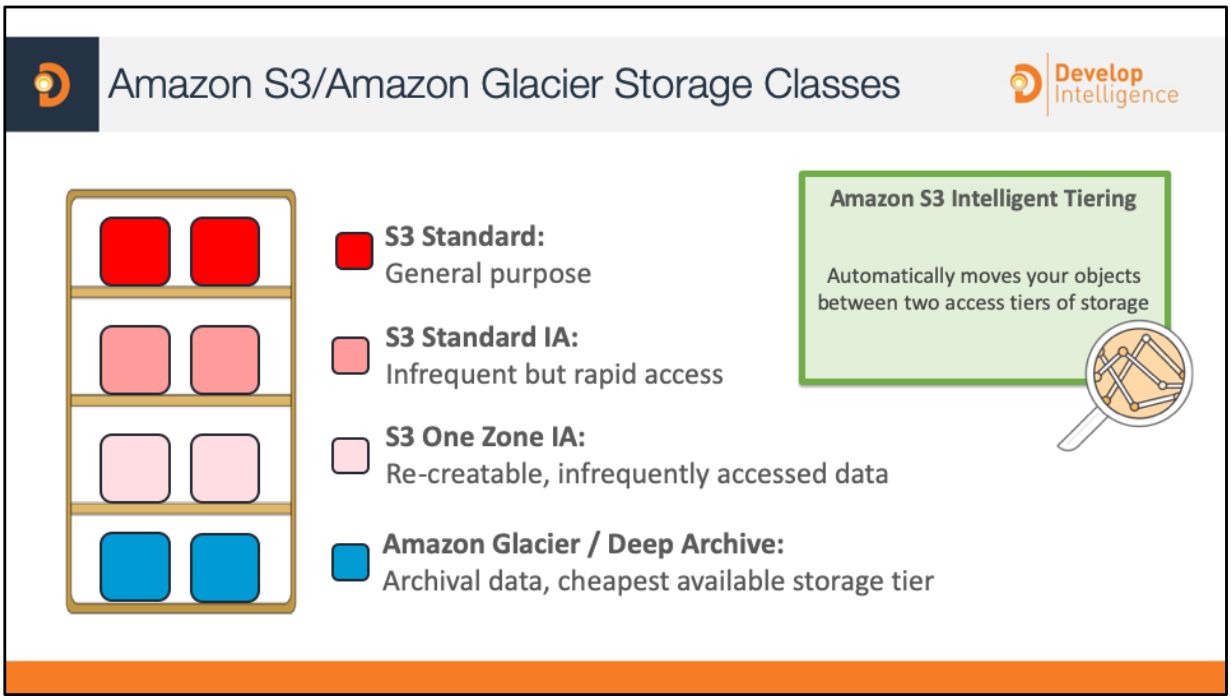
A *vault* is a container for storing archives. When you create a vault, you specify a vault name and the AWS Region in which you want to create the vault.

The Vault Lock feature enforces compliance via a lockable policy.



Your retrieval options enable you to access all the archives you need, when you need them, for a simple, low price:

- You can use **Expedited retrievals** to access data in 1 – 5 minutes for a flat rate of $0.03 per GB retrieved. Expedited retrievals allow you to quickly access your data when occasional urgent requests for a subset of archives are required.
- If you have large amounts of data to retrieve, even petabytes, you can use **Bulk retrievals** to access your data in approximately 5 – 12 hours for a flat rate of just $0.0025 per GB retrieved. Bulk retrievals allow you to cost-effectively access significant portions of your data for things like big data analytics and media transcoding.



For the sake of comparison, here is a description of Amazon S3 storage classes.

General purpose: Amazon S3 Standard
    Higher availability requirements: Use cross-region replication
Infrequently accessed data: Amazon S3 Standard - Infrequent Access
    Lower cost per GB stored.
    High cost per PUT, COPY, POST or GET request
    30-day storage minimum
Infrequent but rapid access: Amazon S3 One Zone-Infrequent Access
    Single Availability Zone
    Cost 20% less than S3 Standard – infrequent Access

Storage Class Analytics

For storing data that needs to be immediately accessible, just like standard data, but which isn't expected to be requested very often, we provide Amazon S3 Standard – Infrequent Access.

Amazon S3 Standard – IA offers all of the benefits of Amazon S3, including its durability, availability, and security; it simply runs on a different cost model to provide solutions for storing infrequently accessed data, such as a user's older digital images or older log files.

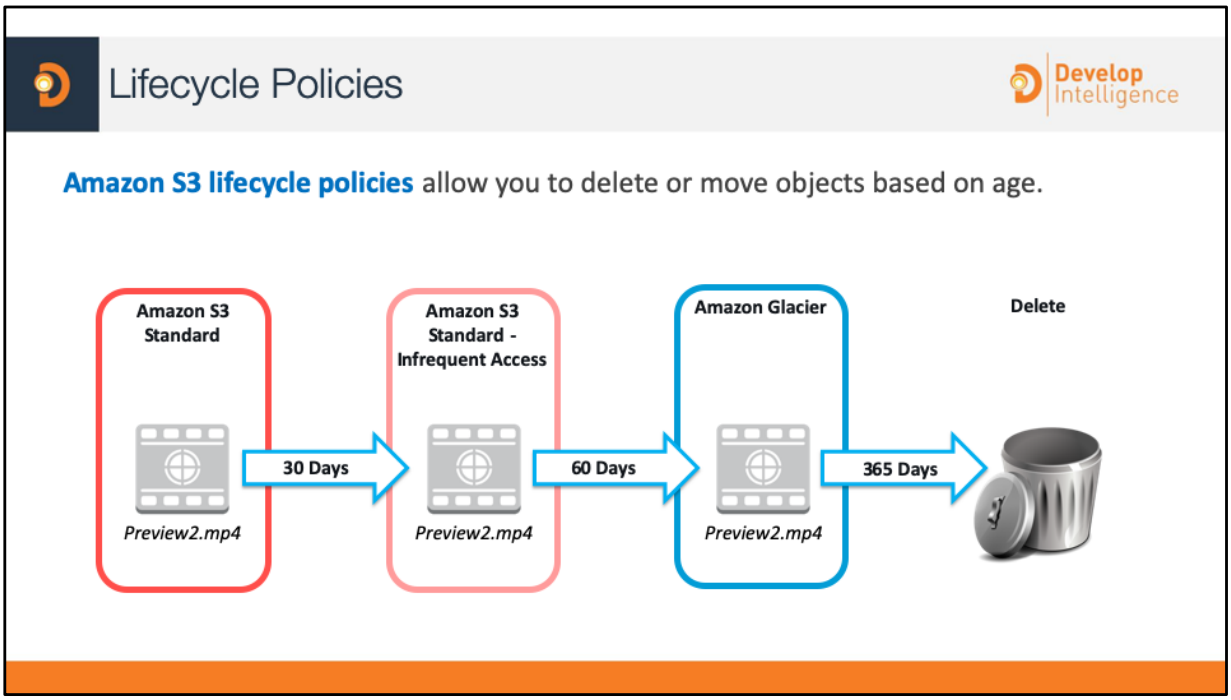
Amazon S3 One Zone-Infrequent Access is an Amazon S3 storage class for data that is accessed less frequently but requires rapid access when needed. Unlike other Amazon object storage classes, which store data in a minimum of three Availability Zones, S3 One Zone-IA stores data in a single Availability Zone. Because of this, storing data in S3 One Zone-IA costs 20% less than storing it in S3 Standard-IA.

By using Amazon S3 analytics *storage class analysis,* you can analyze storage access patterns to help you decide when to transition the right data to the right storage class. This new Amazon S3 analytics feature observes data access patterns to help you determine when to transition less frequently accessed STANDARD storage to the STANDARD_IA (IA, for infrequent access) storage class.

For more information about Amazon S3 storage classes, see
https://aws.amazon.com/s3/storage-classes/

Amazon S3 Intelligent Tiering is a storage class for Amazon Simple Storage Service (Amazon S3) that optimizes storage costs by automatically moving objects between two access tiers of storage when access patterns change. Amazon S3 Intelligent Tiering is ideal when you access storage that is retained for more than a month and has unknown or changing access patterns. For example, you might have newly launched applications and data lakes, where accss patterns can vary across different subsets of storage.
Amazon S3 Intelligent Tiering offers the same milliseconds latency and a 99% availability SLA regardless of which S3 tier objects are stored in. Because Amazon S3 Intelligent Tiering automates storage cost optimization, you don't have to analyze or audit storage access patterns in order to save on storage that is infrequently accessed.

S3 Glacier Deep Archive will be the cheapest available storage tier for users while still maintaining its durability and long term data retention. This storage type is ideal for customers who need to make archival, durable copies of data that rarely or never need to be accessed. It will also allow customers to eliminate the need for on-premises tape libraries. Can be retrieved within 12 hours.
https://aws.amazon.com/about-aws/whats-new/2018/11/s3-glacier-deep-archive/

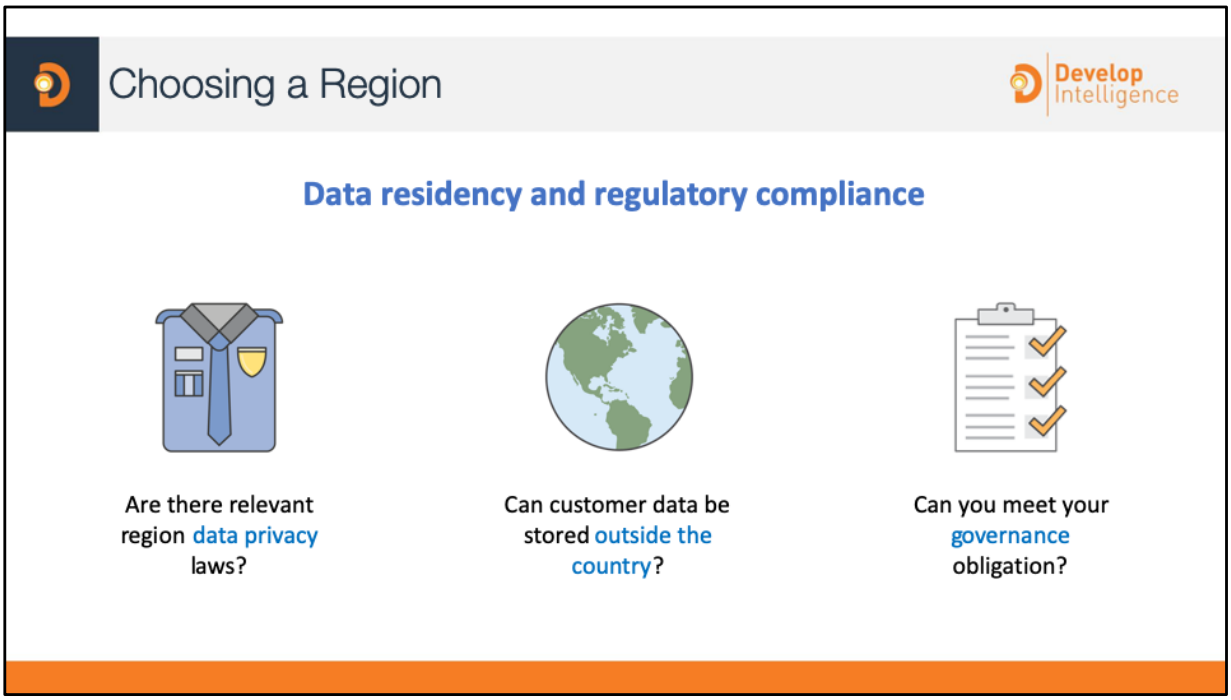


You should automate the lifecycle of your data stored in Amazon S3. Using lifecycle policies, you can have data cycled at regular intervals between different Amazon S3 storage types.

This reduces your overall cost, because you are paying less for data as it becomes less important with time.

In addition to being able to set lifecycle rules per *object*, you can also set lifecycle rules per *bucket*.

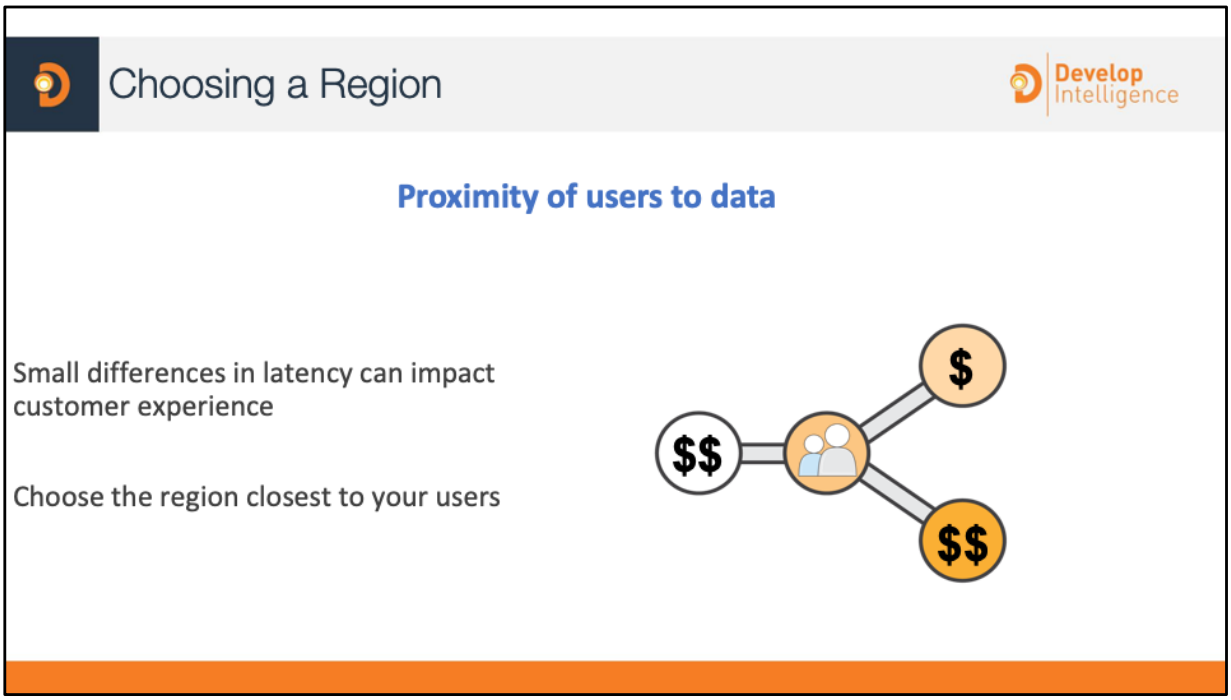
For more information, see
https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html



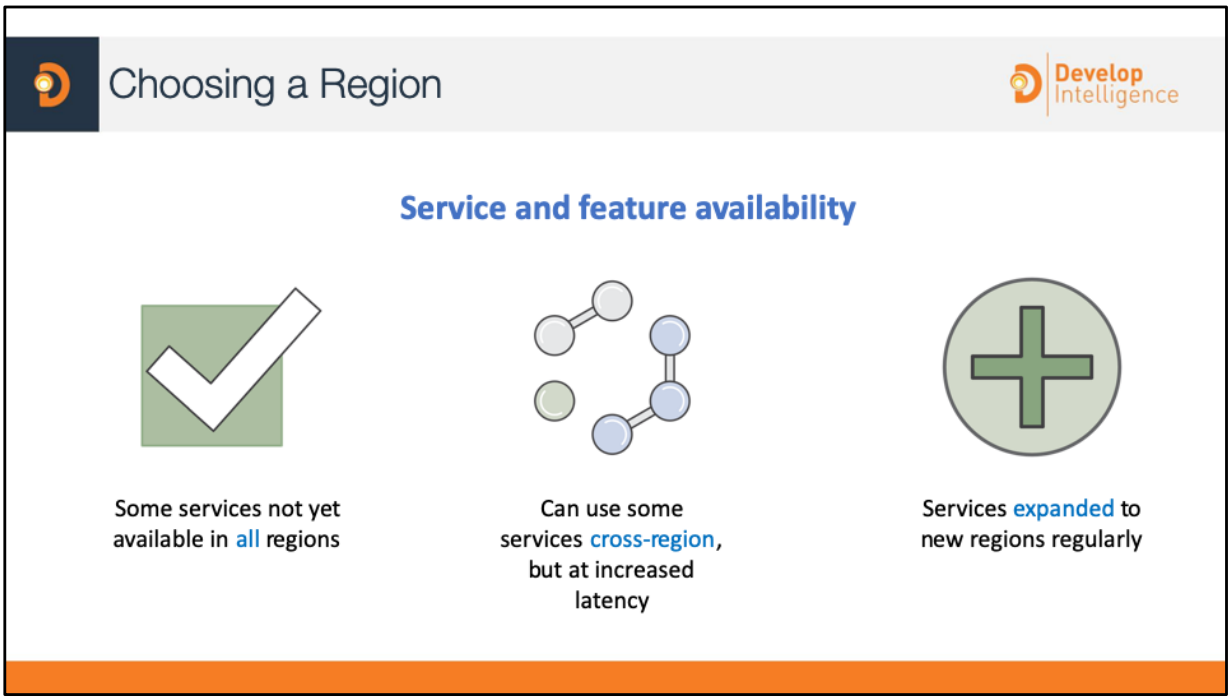
Your data will be subject to the laws of the country and locality in which it's stored. In addition, some laws dictate that if you're operating your business in their jurisdiction, you cannot store that data anywhere else. Similarly, compliance standards (such as the United States' Health Insurance Portability and Accountability Act, or HIPAA) have strict guidelines on how and where data can be stored. Also, AWS opened its first carbon-neutral region in 2011 and now offers five separate carbon-neutral regions.

Take all of these things into account when evaluating where to place your environment.

To find more information about carbon-neutral options, see https://aws.amazon.com/about-aws/sustainability.



Proximity is a big factor in choosing your region, especially when latency is critical. In most cases, the latency difference between using the closest region and the farthest region is relatively small, but even small differences in latency can impact customer experience. Customers expect responsive environments, and as time goes by and technology becomes more and more powerful, those expectations rise as well.



While AWS strives to make our services and features available everywhere, the complications that arise from having a global reach make accomplishing that goal extremely challenging. But rather than wait until a service is available everywhere before launching it, we release our service when it's ready, and expand its availability as soon as possible.

Service costs can differ depending on which region they're used in. For example, an Amazon EC2 instance in US-East 1 may not cost the same as if it were running in EU-West 1. Typically, the difference in cost may not be enough to supersede the other three considerations—however, in cases where the latency/compliance/service availability differences between regions are minimal, you may be able to save by using the lower-cost region for your environment.

In circumstances where your customers are in different areas of the globe, consider optimizing their experience by replicating your environment in multiple regions that are closer to them. Since you would then be distributing your load across multiple environments, your costs for components in each environment may go down even as you add more infrastructure. For example, adding a second application environment might allow you to cut your processing and storage capacity requirements in half in each environment. Since AWS is designed to allow you that kind of flexibility, and since you only pay for what you use, you could easily scale your existing environment down as a way to mitigate the cost of adding another environment.

The downside to that approach is that you now have two environments to manage, and not all of your components will scale down enough to mitigate all of the new component costs. Additionally, you may have to maintain one single storage "source of truth" in one region (such as a Master RDS instance), with which your secondary region would have to communicate with, increasing latency and cost for those operations.

Lab 1:
Hosting a Static Website

Develop Intelligence