Object Storage Options

[Edition 01]

[Last Update 210508]

K21Academy
Learn Cloud from Experts

amazon web services

Contents

1 DOCUMENTATION LINK

1. Block vs Object vs File Storage
   https://www.redhat.com/en/topics/data-storage/fileblock-object-storage
2. Amazon S3
   https://aws.amazon.com/s3/
3. Getting started with Amazon S3
   https://aws.amazon.com/s3/getting-started/
4. Hosting a Static Website using Amazon S3
   https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html
5. AWS CloudFront
   https://aws.amazon.com/cloudfront/
6. AWS CloudFront Features
   https://aws.amazon.com/cloudfront/features/?nc=sn&loc=2
7. AWS CloudFront Pricing
   https://aws.amazon.com/cloudfront/pricing/?nc=sn&loc=3

2 AMAZON S3

- Amazon S3 is object storage built to store and retrieve any amount of data from anywhere on the Internet.
- It's a simple storage service that offers an extremely durable, highly available, and infinitely scalable data storage infrastructure at very low costs.
- Amazon S3 is a distributed architecture and objects are redundantly stored on multiple devices across multiple facilities (AZs) in an Amazon S3 region.
- Amazon S3 is a simple key-based object store.
- Keys can be any string, and they can be constructed to mimic hierarchical attributes.
- Alternatively, you can use S3 Object Tagging to organize your data across all of your S3 buckets and/or prefixes.
- Amazon S3 provides a simple, standards-based REST web services interface that is designed to work with any Internet-development toolkit.
- Files can be from 0 bytes to 5TB.
- The largest object that can be uploaded in a single PUT is 5 gigabytes.
- For objects larger than 100 megabytes use the Multipart Upload capability.
- Updates to an object are atomic – when reading an updated object, you will either get the new object or the old one, you will never get partial or corrupt data.
- There is unlimited storage available.
- It is recommended to access S3 through SDKs and APIs (the console uses APIs).
- Event notifications for specific actions, can send alerts or trigger actions.

Notifications can be sent to:

- SNS Topics.
- SQS Queue.
- Lambda functions.
- Need to configure SNS/SQS/Lambda before S3.
- No extra charges from S3 but you pay for SNS, SQS and Lambda.
- Requester pays function causes the requester to pay (removes anonymous access).
- Can provide time-limited access to objects.
- Provides read after write consistency for PUTS of new objects.
- Provides eventual consistency for overwrite PUTS and DELETES (takes time to propagate).

- You can only store files (objects) on S3.
- HTTP 200 code indicates a successful write to S3.

S3 data is made up of:

- Key (name)
- Value (data)
- Version ID
- Metadata
- Access Control Lists
- Amazon S3 automatically scales to high request rates.

For example, your application can achieve at least 3,500 PUT/POST/DELETE and 5,500 GET requests per second per prefix in a bucket.

There are no limits to the number of prefixes in a bucket. It is simple to increase your read or write performance exponentially.

For read intensive requests you can also use CloudFront edge locations to offload from S3.

2.1 Additional Capabilities

Additional capabilities offered by Amazon S3 include:

Additional S3 Capability	How it Works
Transfer Acceleration	Speed up data uploads using CloudFront in reverse
Requester Pays	The requester rather than the bucket owner pays for requests and data transfer
Tags	Assign tags to objects to use in costing, billing, security etc.
Events	Trigger notifications to SNS, SQS, or Lambda when certain events happen in your bucket
Static Web Hosting	Simple and massively scalable static website hosting
BitTorrent	Use the BitTorrent protocol to retrieve any publicly available object by automatically generating a .torrent file

2.2 Use Cases

Typical use cases include:

- Backup and Storage – Provide data backup and storage services for others.
- Application Hosting – Provide services that deploy, install, and manage web applications.
- Media Hosting – Build a redundant, scalable, and highly available infrastructure that hosts video, photo, or music uploads and downloads.
- Software Delivery – Host your software applications that customers can download.
- Static Website – you can configure a static website to run from an S3 bucket.
- S3 is a persistent, highly durable data store.
- Persistent data stores are non-volatile storage systems that retain data when powered off.
- This is in contrast to transient data stores and ephemeral data stores which lose the data when powered off.

The following table provides a description of persistent, transient and ephemeral data stores and which AWS service to use:

Storage Type	Description	Examples
Persistent Data Store	Data is durable and sticks around after reboots, restarts, or power cycles	S3, Glacier, EBS, EFS
Transient Data Store	Data is just temporarily stored and passed along to another process or persistent store	SQS, SNS
Ephemeral Data Store	Data is lost when the system is stopped	EC2 Instance Store, Memcached

2.3 Buckets

Files are stored in buckets:

- A bucket can be viewed as a container for objects.
- A bucket is a flat container of objects.
- It does not provide a hierarchy of objects.
- You can use an object key name to mimic folders.
- 100 buckets per account by default.
- You can store unlimited objects in your buckets.
- You can create folders in your buckets (only available through the Console).
- You cannot create nested buckets.
- Bucket ownership is not transferrable.
- Bucket names cannot be changed after they have been created.
- If a bucket is deleted its name becomes available again.
- Bucket names are part of the URL used to access the bucket.
- An S3 bucket is region specific.
- S3 is a universal namespace so names must be unique globally.
- URL is in this format: https://s3-eu-west-1.amazonaws.com/<bucketname>.
- Can backup a bucket to another bucket in another account.
- Can enable logging to a bucket.

Bucket naming:

- Bucket names must be at least 3 and no more than 63 character in length.
- Bucket names must start and end with a lowercase character or a number.
- Bucket names must be a series of one or more labels which are separated by a period.
- Bucket names can contain lowercase letters, numbers and hyphens.
- Bucket names cannot be formatted as an IP address.

For better performance, lower latency, and lower cost, create the bucket closer to your clients.

2.4 Objects

Each object is stored and retrieved by a unique key (ID or name).

An object in S3 is uniquely identified and addressed through:

- Service end-point.
- Bucket name.
- Object key (name).
- Optionally, an object version.
- Objects stored in a bucket will never leave the region in which they are stored unless you move them to another region or enable cross-region replication.
- You can define permissions on objects when uploading and at any time afterwards using the AWS Management Console.

2.5 Sub-Resources

- Sub-resources are subordinate to objects, they do not exist independently but are always associated with another entity such as an object or bucket.
- Sub-resources (configuration containers) associated with buckets include:
- Lifecycle – define an object's lifecycle.
- Website – configuration for hosting static websites.
- Versioning – retain multiple versions of objects as they are changed.
- Access Control Lists (ACLs) – control permissions access to the bucket.
- Bucket Policies – control access to the bucket.
- Cross Origin Resource Sharing (CORS).
- Logging
- Sub-resources associated with objects include:
- ACLs – define permissions to access the object.
- Restore – restoring an archive.

2.6 Storage Classes

There are six S3 storage classes.

1. S3 Standard (durable, immediately available, frequently accessed).
2. S3 Intelligent-Tiering (automatically moves data to the most cost-effective tier).
3. S3 Standard-IA (durable, immediately available, infrequently accessed).

4. S3 One Zone-IA (lower cost for infrequently accessed data with less resilience).
5. S3 Glacier (archived data, retrieval times in minutes or hours).
6. S3 Glacier Deep Archive (lowest cost storage class for long term retention).

The table below provides the details of each Amazon S3 storage class:

	S3 Standard	S3 Standard-IA	S3 One Zone-IA	Amazon Glacier
Designed for durability	99.999999999%	99.999999999%	99.999999999%	99.999999999%
Designed for availability	99.99%	99.9%	99.5%	N/A
Availability SLA	99.9%	99%	99%	N/A
Availability Zones	≥3	≥3	1	≥3
Minimum capacity charge per object	N/A	128KB	128KB	N/A
Minimum storage duration charge	N/A	30 days	30 days	90 days
Retrieval fee	N/A	Per GB retrieved	Per GB retrieved	Per GB retrieved
First byte latency	milliseconds	milliseconds	milliseconds	Select minutes or hours
Storage type	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes

Objects stored in the S3 One Zone-IA storage class are stored redundantly within a single Availability Zone in the AWS Region you select.

2.7 Access & Access Policies

There are four mechanisms for controlling access to Amazon S3 resources:

- IAM policies.
- Bucket policies.
- Access Control Lists (ACLs).
- Query string authentication (URL to an Amazon S3 object which is only valid for a limited time).
- Access auditing can be configured by configuring an Amazon S3 bucket to create access log records for all requests made against it.
- For capturing IAM/user identity information in logs configure AWS CloudTrail Data Events.

- By default, a bucket, its objects, and related sub-resources are all private.
- By default, only a resource owner can access a bucket.
- The resource owner refers to the AWS account that creates the resource.
- With IAM the account owner rather than the IAM user is the owner.
- Within an IAM policy you can grant either programmatic access or AWS Management Console access to Amazon S3 resources.
- Amazon Resource Names (ARN) are used for specifying resources in a policy.

The format for any resource on AWS is:

- arn:partition:service:region:namespace:relative-id.

For S3 resources:

- aws is a common partition name.
- s3 is the service.
- You don't specify Region and namespace.
- For Amazon S3, it can be a bucket-name or a bucket-name/object-key. You can use wild card

The format for S3 resources is:

- arn:aws:s3:::bucket_name.
- arn:aws:s3:::bucket_name/key_name.
- A bucket owner can grant cross-account permissions to another AWS account (or users in an account) to upload objects.
- The AWS account that uploads the objects owns them.

The bucket owner does not have permissions on objects that other accounts own, however:

- The bucket owner pays the charges.
- The bucket owner can deny access to any objects regardless of ownership.
- The bucket owner can archive any objects or restore archived objects regardless of ownership.

Access to buckets and objects can be granted to:

- Individual users.
- AWS accounts.
- Everyone (public/anonymous).
- All authenticated users (AWS users).
- Access policies define access to resources and can be associated with resources (buckets and objects) and users.
- You can use the AWS Policy Generator to create a bucket policy for your Amazon S3 bucket.
- The categories of policy are resource-based policies and user policies.

Resource-based policies:

- Attached to buckets and objects.
- ACL-based policies define permissions.
- ACLs can be used to grant read/write permissions to other accounts.
- Bucket policies can be used to grant other AWS accounts or IAM users permission to the bucket and objects.

User policies:

- Can use IAM to manage access to S3 resources.
- Using IAM you can create users, groups and roles and attach access policies to them granting them access to resources.
- You cannot grant anonymous permissions in an IAM user policy as the policy is attached to a user.
- User policies can grant permissions to a bucket and the objects in it.

ACLs:

- S3 ACLs enable you to manage access to buckets and objects.
- Each bucket and object has an ACL attached to it as a subresource.
- Bucket and object permissions are independent of each other.
- The ACL defines which AWS accounts (grantees) or pre-defined S3 groups are granted access and the type of access.
- A grantee can be an AWS account or one of the predefined Amazon S3 groups.

- When you create a bucket or an object, S3 creates a default ACL that grants the resource owner full control over the resource.

Cross account access:

- You grant permission to another AWS account using the email address or the canonical user ID.
- However, if you provide an email address in your grant request, Amazon S3 finds the canonical user ID for that account and adds it to the ACL.
- Grantee accounts can then then delegate the access provided by other accounts to their individual users.

2.8 Pre-Defined Groups

Authenticated Users group:

- This group represents all AWS accounts.
- Access permission to this group allows any AWS account access to the resource.
- All requests must be signed (authenticated).
- Any authenticated user can access the resource.

All Users group:

- Access permission to this group allows anyone in the world access to the resource.
- The requests can be signed (authenticated) or unsigned (anonymous).
- Unsigned requests omit the authentication header in the request.
- AWS recommends that you never grant the All Users group WRITE, WRITE_ACP, or FULL_CONTROL permissions.

Log Delivery group:

- Providing WRITE permission to this group on a bucket enables S3 to write server access logs.
- Not applicable to objects.

The following table lists the set of permissions that Amazon S3 supports in an ACL:

- The set of ACL permissions is the same for an object ACL and a bucket ACL.
- Depending on the context (bucket ACL or object ACL), these ACL permissions grant permissions for specific buckets or object operations.

- The table lists the permissions and describes what they mean in the context of objects and buckets.

Permission	When granted on a bucket	When granted on an object
READ	Allows grantee to list the objects in the bucket	Allows grantee to read the object data and its metadata
WRITE	Allows grantee to create, overwrite, and delete any object in the bucket	Not applicable
READ_ACP	Allows grantee to read the bucket ACL	Allows grantee to read the object ACL
WRITE_ACP	Allows grantee to write the ACL for the applicable bucket	Allows grantee to write the ACL for the applicable object
FULL_CONTROL	Allows grantee the READ, WRITE, READ_ACP, and WRITE_ACP permissions on the bucket	Allows grantee the READ, READ_ACP, and WRITE_ACP permissions on the object

Note the following:

- Permissions are assigned at the account level for authenticated users.
- You cannot assign permissions to individual IAM users.
- When Read is granted on a bucket it only provides the ability to list the objects in the bucket.
- When Read is granted on an object the data can be read.
- ACP means access control permissions and READ_ACP/WRITE_ACP control who can read/write the ACLs themselves.
- WRITE is only applicable to the bucket level (except for ACP).
- Bucket policies are limited to 20 KB in size.
- Object ACLs are limited to 100 granted permissions per ACL.
- The only recommended use case for the bucket ACL is to grant write permissions to the S3 Log Delivery group.

There are limits to managing permissions using ACLs:

- You cannot grant permissions to individual users.
- You cannot grant conditional permissions.
- You cannot explicitly deny access.
- When granting other AWS accounts the permissions to upload objects, permissions to these objects can only be managed by the object owner using object ACLs.

You can use bucket policies for:

- Granting users permissions to a bucket owned by your account.
- Managing object permissions (where the object owner is the same account as the bucket owner).
- Managing cross-account permissions for all Amazon S3 permissions.

You can use user policies for:

- Granting permissions for all Amazon S3 operations.
- Managing permissions for users in your account.
- Granting object permissions to users within the account.
- For an IAM user to access resources in another account the following must be provided:
- Permission from the parent account through a user policy.
- Permission from the resource owner to the IAM user through a bucket policy, or the parent account through a bucket policy, bucket ACL or object ACL.
- If an AWS account owns a resource it can grant permissions to another account, that account can then delegate those permissions or a subset of them to uses in the account (permissions delegation).
- An account that receives permissions from another account cannot delegate permissions cross-account to a third AWS account.

2.9 Charges

- No charge for data transferred between EC2 and S3 in the same region.
- Data transfer into S3 is free of charge.
- Data transferred to other regions is charged.
- Data Retrieval (applies to S3 Standard-IA and S3 One Zone-IA, S3 Glacier and S3 Glacier Deep Archive).

Charges are:

- Per GB/month storage fee.
- Data transfer out of S3.
- Upload requests (PUT and GET).
- Retrieval requests (S3-IA or Glacier).

Requester pays:

- The bucket owner will only pay for object storage fees.
- The requester will pay for requests (uploads/downloads) and data transfers.
- Can only be enabled at the bucket level.

2.10 Multipart Upload

- Can be used to speed up uploads to S3.
- Multipart upload uploads objects in parts independently, in parallel and in any order.
- Performed using the S3 Multipart upload API.
- It is recommended for objects of 100MB or larger.
- Can be used for objects from 5MB up to 5TB.
- Must be used for objects larger than 5GB.
- If transmission of any part fails it can be retransmitted.
- Improves throughput.
- Can pause and resume object uploads.
- Can begin upload before you know the final object size.

2.11 Copy

- You can create a copy of objects up to 5GB in size in a single atomic operation.
- For files larger than 5GB you must use the multipart upload API.
- Can be performed using the AWS SDKs or REST API.

The copy operation can be used to:

- Generate additional copies of objects.
- Renaming objects.
- Changing the copy's storage class or encryption at rest status.
- Move objects across AWS locations/regions.
- Change object metadata.
- Once uploaded to S3 some object metadata cannot be changed, copying the object can allow you to modify this information.

2.12 Transfer Acceleration

- Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and your Amazon S3 bucket.
- S3 Transfer Acceleration leverages Amazon CloudFront's globally distributed AWS Edge Locations.
- Used to accelerate object uploads to S3 over long distances (latency).
- Transfer acceleration is as secure as a direct upload to S3.
- You are charged only if there was a benefit in transfer times.
- Need to enable transfer acceleration on the S3 bucket.
- Cannot be disabled, can only be suspended.
- May take up to 30 minutes to implement.
- URL is: <bucketname>.s3-accelerate.amazonaws.com.
- Bucket names must be DNS compliance and cannot have periods between labels.
- Now HIPAA compliant.
- You can use multipart uploads with transfer acceleration.

Key Difference	REST API Endpoint	Website Endpoint
Access Control	Supports both public and private content	Supports only publicly readable content
Error message handling	Returns an XML-formatted error response	Returns an HTML document
Redirection support	Not applicable	Supports both object-level and bucket-level redirects
Requests support	Supports all bucket and object operations	Supports only GET and HEAD requests on objects
Responses to GET and HEAD requests at the root of the bucket	Returns a list of the object keys in the bucket	Returns the Index document that is specified in the website configuration
SSL support	Supports SSL connections	Does not support SSL connections

2.13 Pre-Signed URL's

- Pre-signed URLs can be used to provide temporary access to a specific object to those who do not have AWS credentials.
- By default, all objects are private and can only be accessed by the owner.
- To share an object, you can either make it public or generate a pre-signed URL.
- Expiration date and time must be configured.
- These can be generated using SDKs for Java and .Net and AWS explorer for Visual Studio.
- Can be used for downloading and uploading S3 objects.

2.14 Versioning

- Versioning stores all versions of an object (including all writes and even if an object is deleted).
- Versioning protects against accidental object/data deletion or overwrites.

- Enables “roll-back” and “un-delete” capabilities.
- Versioning can also be used for data retention and archive.
- Old versions count as billable size until they are permanently deleted.
- Enabling versioning does not replicate existing objects.
- Can be used for backup.
- Once enabled versioning cannot be disabled only suspended.
- Can be integrated with lifecycle rules.
- Multi-factor authentication (MFA) delete can be enabled.
- MFA delete can also be applied to changing versioning settings.

MFA delete applies to:

- Changing the bucket’s versioning state.
- Permanently deleting an object.

Cross Region Replication requires versioning to be enabled on the source and destination buckets.

Reverting to previous versions isn’t replicated.

By default, a HTTP GET retrieves the most recent version.

Only the S3 bucket owner can permanently delete objects once versioning is enabled.

When you try to delete an object with versioning enabled a DELETE marker is placed on the object.

You can delete the DELETE marker and the object will be available again.

Deletion with versioning replicates the delete marker. But deleting the delete marker is not replicated.

Bucket versioning states:

- Enabled
- Versioned
- Un-versioned

Objects that existed before enabling versioning will have a version ID of NULL.

Suspension:

- If you suspend versioning the existing objects remain as they are however new versions will not be created.
- While versioning is suspended new objects will have a version ID of NULL and uploaded objects of the same name will overwrite the existing object.

2.15 Lifecycle Management

Used to optimize storage costs, adhere to data retention policies and to keep S3 volumes well-maintained.

A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. There are two types of actions:

1. Transition actions - Define when objects transition to another storage class. For example, you might choose to transition objects to the STANDARD_IA storage class 30 days after you created them, or archive objects to the GLACIER storage class one year after creating them.

There are costs associated with the lifecycle transition requests. For pricing information, see Amazon S3 Pricing.

2. Expiration actions - Define when objects expire. Amazon S3 deletes expired objects on your behalf.

Lifecycle configuration is an XML file applied at the bucket level as a subresource.

Can be used in conjunction with versioning or independently.

Can be applied to current and previous versions.

Can be applied to specific objects within a bucket: objects with a specific tag or objects with a specific prefix.

Supported Transitions and Related Constraints

Amazon S3 supports the following lifecycle transitions between storage classes using a lifecycle configuration:

- You can transition from the STANDARD storage class to any other storage class.
- You can transition from any storage class to the GLACIER or DEEP_ARCHIVE storage classes.
- You can transition from the STANDARD_IA storage class to the INTELLIGENT_TIERING or ONEZONE_IA storage classes.
- You can transition from the INTELLIGENT_TIERING storage class to the ONEZONE_IA storage class.

- You can transition from the GLACIER storage class to the DEEP_ARCHIVE storage class.

The following lifecycle transitions are not supported:

- You can't transition from any storage class to the STANDARD storage class.
- You can't transition from any storage class to the REDUCED_REDUNDANCY storage class.
- You can't transition from the INTELLIGENT_TIERING storage class to the STANDARD_IA storage class.
- You can't transition from the ONEZONE_IA storage class to the STANDARD_IA or INTELLIGENT_TIERING storage classes.
- You can transition from the GLACIER storage class to the DEEP_ARCHIVE storage class only.
- You can't transition from the DEEP_ARCHIVE storage class to any other storage class.

The lifecycle storage class transitions have the following constraints:

- From the STANDARD or STANDARD_IA storage class to INTELLIGENT_TIERING. The following constraints apply:
- For larger objects, there is a cost benefit for transitioning to INTELLIGENT_TIERING. Amazon S3 does not transition objects that are smaller than 128 KB to the INTELLIGENT_TIERING storage class because it's not cost effective.
- From the STANDARD storage classes to STANDARD_IA or ONEZONE_IA. The following constraints apply:
- For larger objects, there is a cost benefit for transitioning to STANDARD_IA or ONEZONE_IA. Amazon S3 does not transition objects that are smaller than 128 KB to the STANDARD_IA or ONEZONE_IA storage classes because it's not cost effective.
- Objects must be stored at least 30 days in the current storage class before you can transition them to STANDARD_IA or ONEZONE_IA. For example, you cannot create a lifecycle rule to transition objects to the STANDARD_IA storage class one day after you create them.
- Amazon S3 doesn't transition objects within the first 30 days because newer objects are often accessed more frequently or deleted sooner than is suitable for STANDARD_IA or ONEZONE_IA storage.

- If you are transitioning noncurrent objects (in versioned buckets), you can transition only objects that are at least 30 days noncurrent to STANDARD_IA or ONEZONE_IA storage.
- From the STANDARD_IA storage class to ONEZONE_IA. The following constraints apply:
- Objects must be stored at least 30 days in the STANDARD_IA storage class before you can transition them to the ONEZONE_IA class.

2.16 Encryption

You can securely upload/download your data to Amazon S3 via SSL endpoints using the HTTPS protocol (In Transit – SSL/TLS).

Encryption options:

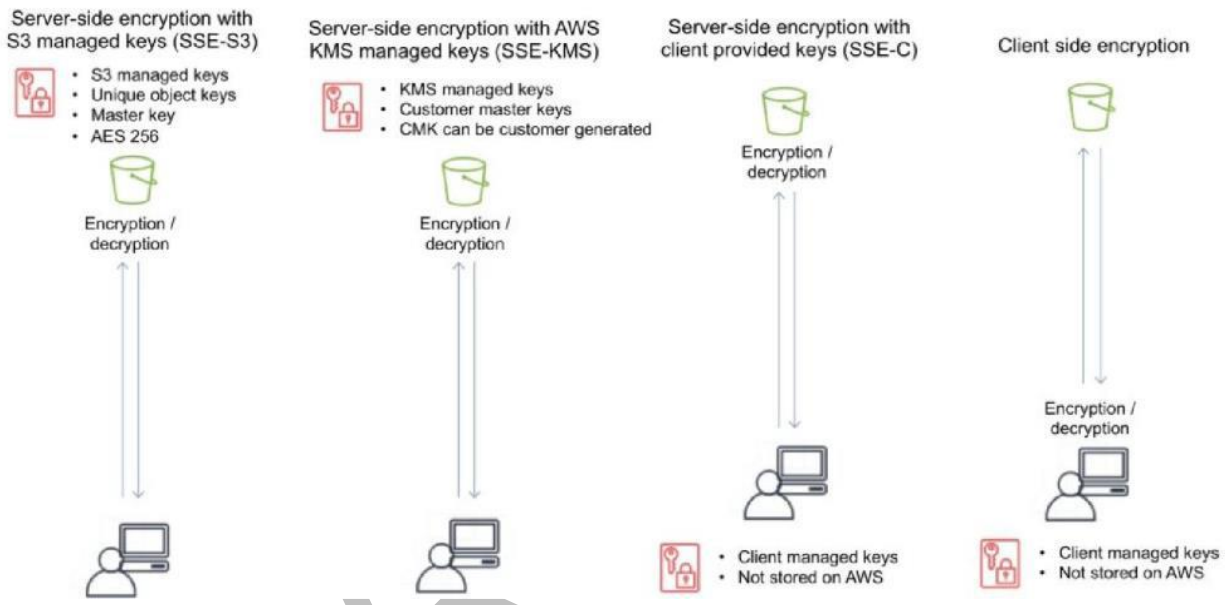
Encryption Option	How it Works
SSE-S3	Use S3's existing encryption key for AES-256
SSE-C	Upload your own AES-256 encryption key which S3 uses when it writes objects
SSE-KMS	Use a key generated and managed by AWS KMS
Client-Side	Encrypt objects using your own local encryption process before uploading to S3

Server side encryption options:

- SSE-S3 – Server Side Encryption with S3 managed keys
- Each object is encrypted with a unique key.
- Encryption key is encrypted with a master key.
- AWS regularly rotate the master key.
- Uses AES 256.
- SSE-KMS – Server Side Encryption with AWS KMS keys
- KMS uses Customer Master Keys (CMKs) to encrypt.
- Can use the automatically created CMK key.
- OR you can select your own key (gives you control for management of keys).
- An envelope key protects your keys.
- Chargeable.

- SSE-C – Server Side Encryption with client provided keys
- Client manages the keys, S3 manages encryption.
- AWS does not store the encryption keys.
- If keys are lost data cannot be decrypted.

The following diagram depicts the options for enabling encryption and shows you where the encryption is applied and where the keys are managed:



2.17 Event Notifications

Amazon S3 event notifications can be sent in response to actions in Amazon S3 like PUTs, POSTs, COPYs, or DELETEs.

Amazon S3 event notifications enable you to run workflows, send alerts, or perform other actions in response to changes in your objects stored in S3.

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

You can configure notifications to be filtered by the prefix and suffix of the key name of objects.

Amazon S3 can publish notifications for the following events:

- New object created events
- Object removal events
- Restore object events
- Reduced Redundancy Storage (RRS) object lost events
- Replication events

Amazon S3 can send event notification messages to the following destinations:

- Publish event messages to an Amazon Simple Notification Service (Amazon SNS) topic.
- Publish event messages to an Amazon Simple Queue Service (Amazon SQS) queue.
- Publish event messages to AWS Lambda by invoking a Lambda function and providing the event message as an argument.
- Need to grant Amazon S3 permissions to post messages to an Amazon SNS topic or an Amazon SQS queue.
- Need to also grant Amazon S3 permission to invoke an AWS Lambda function on your behalf. For information about granting these permissions.

2.18 Object Tags

- S3 object tags are key-value pairs applied to S3 objects which can be created, updated or deleted at any time during the lifetime of the object.
- Allow you to create Identity and Access Management (IAM) policies, setup S3 Lifecycle policies, and customize storage metrics.
- Up to ten tags can be added to each S3 object and you can use either the AWS Management Console, the REST API, the AWS CLI, or the AWS SDKs to add object tags.
- S3 CLOUDWATCH METRICS
- You can use the AWS Management Console to enable the generation of 1-minute CloudWatch request metrics for your S3 bucket or configure filters for the metrics using a prefix or object tag.
- Alternatively, you can call the S3 PUT Bucket Metrics API to enable and configure publication of S3 storage metrics.
- CloudWatch Request Metrics will be available in CloudWatch within 15 minutes after they are enabled.
- CloudWatch Storage Metrics are enabled by default for all buckets and reported once per day.

The S3 metrics that can be monitored include:

- S3 requests
- Bucket storage
- Bucket size
- All requests
- HTTP 4XX/5XX errors

2.19 Cross Region Replication

- CRR is an Amazon S3 feature that automatically replicates data across AWS Regions.
- With CRR, every object uploaded to an S3 bucket is automatically replicated to a destination bucket in a different AWS Region that you choose.
- Provides automatic, asynchronous copying of objects between buckets in different regions.
- CRR is configured at the S3 bucket level.

- You enable a CRR configuration on your source bucket by specifying a destination bucket in a different Region for replication.
- You can use either the AWS Management Console, the REST API, the AWS CLI, or the AWS SDKs to enable CRR.
- Versioning must be enabled for both the source and destination buckets.
- With CRR you can only replication between regions, not within a region (see SRR below for single region replication).
- Replication is 1:1 (one source bucket, to one destination bucket).
- You can configure separate S3 Lifecycle rules on the source and destination buckets.
- You can replicate KMS-encrypted objects by providing a destination KMS key in your replication configuration.
- You can set up CRR across AWS accounts to store your replicated data in a different account in the target region.
- Provides low latency access for data by copying objects to buckets that are closer to users.

To activate CRR you need to configure the replication on the source bucket:

- Define the bucket in the other region to replicate to.
- Specify to replicate all objects or a subset of objects with specific key name prefixes.
- The replicas will be exact replicas and share the same key names and metadata.
- You can specify a different storage class (by default the source storage class will be used).
- AWS S3 will encrypt data in-transit with SSL.
- AWS S3 must have permission to replicate objects.
- Bucket owners must have permission to read the object and object ACL.
- Can be used across accounts but the source bucket owner must have permission to replicate objects into the destination bucket.

Triggers for replication are:

- Uploading objects to the source bucket.
- DELETE of objects in the source bucket.
- Changes to the object, its metadata, or ACL.

What is replicated:

- New objects created after enabling replication.
- Changes to objects.
- Objects created using SSE-S3 using the AWS managed key.
- Object ACL updates.

What isn't replicated:

- Objects that existed before enabling replication (can use the copy API).
- Objects created with SSE-C and SSE-KMS.
- Objects to which the bucket owner does not have permissions.
- Updates to bucket-level subresources.
- Actions from lifecycle rules are not replicated.
- Objects in the source bucket that are replicated from another region are not replicated.

Deletion behavior:

- If a DELETE request is made without specifying an object version ID a delete marker will be added and replicated.
- If a DELETE request is made specifying an object version ID the object is deleted but the delete marker is not replicated.

Charges:

- Requests for upload
- Inter-region transfer
- S3 storage in both regions

2.20 Same Region Replication (SRR)

- As the name implies you can use SRR to replication objects to a destination bucket within the same region as the source bucket.
- This feature was released in September 2018.
- Replication is automatic and asynchronous.
- New objects uploaded to an Amazon S3 bucket are configured for replication at the bucket, prefix, or object tag levels.
- Replicated objects can be owned by the same AWS account as the original copy or by different accounts, to protect from accidental deletion.
- Replication can be to any Amazon S3 storage class, including S3 Glacier and S3 Glacier Deep Archive to create backups and long-term archives.
- When an S3 object is replicated using SRR, the metadata, Access Control Lists (ACL), and object tags associated with the object are also part of the replication.
- Once SRR is configured on a source bucket, any changes to the object, metadata, ACLs, or object tags trigger a new replication to the destination bucket.

2.21 S3 Analytics

- Can run analytics on data stored on Amazon S3.
- This includes data lakes, IoT streaming data, machine learning, and artificial intelligence.

The following strategies can be used:

S3 Analytics Strategies	Service Used
Data Lake Concept	Athena, RedShift Spectrum, QuickSight
IoT Streaming Data Repository	Kinesis Firehose
Machine Learning and AI Storage	Rekognition, Lex, MXNet
Storage Class Analysis	S3 Management Analytics

2.22 S3 Performance Guidelines

AWS provide some performance guidelines for Amazon S3. These are summarized here:

Measure Performance - When optimizing performance, look at network throughput, CPU, and DRAM requirements. Depending on the mix of demands for these different resources, it might be worth evaluating different Amazon EC2 instance types.

Scale Storage Connections Horizontally - You can achieve the best performance by issuing multiple concurrent requests to Amazon S3. Spread these requests over separate connections to maximize the accessible bandwidth from Amazon S3.

Use Byte-Range Fetches - Using the Range HTTP header in a GET Object request, you can fetch a byte-range from an object, transferring only the specified portion. You can use concurrent connections to Amazon S3 to fetch different byte ranges from within the same object. This helps you achieve higher aggregate throughput versus a single whole-object request. Fetching smaller ranges of a large object also allows your application to improve retry times when requests are interrupted.

Retry Requests for Latency- Sensitive Applications - Aggressive timeouts and retries help drive consistent latency. Given the large scale of Amazon S3, if the first request is slow, a retried request is likely to take a different path and quickly succeed. The AWS SDKs have configurable timeout and retry values that you can tune to the tolerances of your specific application.

Combine Amazon S3 (Storage) and Amazon EC2 (Compute) in the Same AWS Region - Although S3 bucket names are globally unique, each bucket is stored in a Region that you select when you create the bucket. To optimize performance, we recommend that you access the bucket from Amazon EC2 instances in the same AWS Region when possible. This helps reduce network latency and data transfer costs.

Use Amazon S3 Transfer Acceleration to Minimize Latency Caused by Distance - Amazon S3 Transfer Acceleration manages fast, easy, and secure transfers of files over long geographic distances between the client and an S3 bucket. Transfer Acceleration takes advantage of the globally distributed edge locations in Amazon CloudFront. As the data arrives at an edge location, it is routed to Amazon S3 over an optimized network path. Transfer Acceleration is ideal for transferring gigabytes to terabytes of data regularly across continents. It's also useful for clients that upload to a centralized bucket from all over the world.

2.23 Sample Questions

Q1 : Which of the following are reasons to select the S3 standard storage class? (Choose two.)

a. Need for high durability
b. Need for highest available throughput
c. Infrequent access of objects
d. Objects can easily be re-created if lost.

Answer: A, B

Explanation: S3 shares the durability of all S3 storage classes at 11 9s. It also provides the highest availability throughput of all S3 storage classes. Infrequent access is a use case for S3-IA, while the ability to re-create objects would suggest S3 One Zone-IA.

Q2 : Your company needs a storage solution that can support millions of customers accessing billing data. The data should be instantly accessible for users, but individual bills are not accessed that often. What is the most cost-efficient storage for this use case?

a. Glacier with expedited retrieval
b. S3 with Transfer Acceleration
c. Standard S3
d. S3-IA

Answer: D

Explanation: This is a pretty “by the book” question, and in this case, is the exact use case for which S3-IA (Infrequent Access) was built. Instant access with less frequent requests is ideal for S3-IA.

Q3: Is the S3-IA storage class less expensive than S3?

a. Yes
b. No
c. Their costs are identical.
d. It depends on how the storage class is used.

Answer: A

Explanation: S3-IA is less expensive than S3, regardless of use case. It is certainly possible that S3-IA is not appropriate for a certain use case, but it is less expensive on a “per byte retrieved” case.

For more Questions Please check Certification Sample Quiz under each module

Link: https://k21academy.com/awssaquizm04

3 GLACIER

Glacier is an archiving storage solution for infrequently accessed data.

There are two storage tiers:

1. **S3 Glacier:**
   - Same low latency and high throughput performance of S3 Standard.
   - Designed for durability of 99.999999999% of objects in a single Availability Zone†.
   - Designed for 99.5% availability over a given year.
   - Backed with the Amazon S3 Service Level Agreement for availability.
   - Supports SSL for data in transit and encryption of data at rest.
   - S3 Lifecycle management for automatic migration of objects to other S3 Storage Classes.

2. **S3 Glacier Deep Archive.**
   - Designed for durability of 99.999999999% of objects across multiple Availability Zones.
   - Data is resilient in the event of one entire Availability Zone destruction.
   - Supports SSL for data in transit and encryption of data at rest.
   - Low-cost design is ideal for long-term archive.
   - Configurable retrieval times, from minutes to hours.
   - S3 PUT API for direct uploads to S3 Glacier, and S3 Lifecycle management for automatic migration of objects.
   - The key difference between the tiers is that Deep Archive is lower cost, but retrieval times are much longer (12 hours).
   - The S3 Glacier tier has configurable retrieval times from minutes to hours (you pay accordingly).
   - Archived objects are not available for real time access and you need to submit a retrieval request.
   - Glacier must complete a job before you can get its output.
   - Requested archival data is copied to S3 One Zone-IA.
   - Following retrieval, you have 24 hours to download your data.
   - You cannot specify Glacier as the storage class at the time you create an object.
   - Glacier is designed to sustain the loss of two facilities.

- Glacier automatically encrypts data at rest using AES 256 symmetric keys and supports secure transfer of data over SSL.
- Glacier may not be available in all AWS regions.
- Glacier objects are visible through S3 only (not Glacier directly).
- Glacier does not archive object metadata; you need to maintain a client-side database to maintain this information.
- Archives can be 1 bytes up to 40TB.
- Glacier file archives of 1 byte – 4 GB can be performed in a single operation.
- Glacier file archives from 100MB up to 40TB can be uploaded to Glacier using the multipart upload API.
- Uploading archives is synchronous.
- Downloading archives is asynchronous.
- The contents of an archive that has been uploaded cannot be modified.
- You can upload data to Glacier using the CLI, SDKs or APIs – you cannot use the AWS Console.
- Glacier adds 32-40KB (indexing and archive metadata) to each object when transitioning from other classes using lifecycle policies.
- AWS recommends that if you have lots of small objects they are combined in an archive (e.g. zip file) before uploading.
- A description can be added to archives, no other metadata can be added.
- Glacier archive IDs are added upon upload and are unique for each upload.

Archive retrieval:

- Expedited is 1-5 minutes retrieval (most expensive).
- Standard is 3.5 hours retrieval (cheaper, 10GB data retrieval free per month).
- Bulk retrieval is 5-12 hours (cheapest, use for large quantities of data).
- You can retrieve parts of an archive.
- When data is retrieved it is copied to S3 and the archive remains in Glacier and the storage class therefore does not change.
- AWS SNS can send notifications when retrieval jobs are complete.
- Retrieved data is available for 24 hours by default (can be changed).
- To retrieve specific objects within an archive you can specify the byte range (Range) in the HTTP GET request (need to maintain a DB of byte ranges).
- Glacier Charges:

- There is no charge for data transfer between EC2 and Glacier in the same region.
- There is a charge if you delete data within 90 days.

When you restore you pay for:

- The Glacier archive.
- The requests.
- The restored data on S3.

3.1 Sample Questions

Q1 : You have a large archive of documents that must be backed up. The documents will be accessed very infrequently, if at all. However, when the documents are accessed, they must be delivered within 10 minutes of a retrieval request. What is the most cost-effective option for storing these documents?

A. S3

B. S3-IA

C. Glacier

D. Glacier with expedited retrieval

Answer: D

Explanation: All the description here suggests using Glacier. The documents are a large archive, and many will never be accessed. However, the requirement for quick retrieval points to a need for expedited retrieval. Glacier with expedited retrieval is still going to cost less than S3-IA for access that isn't that frequent.

Q2 : You are in charge of storage for large datasets at a predictive analytics firm. You are tasked with minimizing storage costs. You need to store data 30–59 days old in a storage class that makes the data immediately available and data older than 60 days in a class that makes the data available within 10 hours. You want to use the least expensive classes available. Which two storage classes would you choose? (Choose two.)

A. S3 standard

B. S3 Infrequent Access

C. S3 RRS

D. Glacier

Answer:B, D

Explanation: Glacier is the easy choice, as it can handle the oldest data and still meet the 10-hour retrieval time. S3 RRS is deprecated and shouldn't be considered. This leaves S3 and S3-IA. S3-IA is always less expensive than S3, so it's the better option here.

Q3 : Which of the following statements regarding S3 storage classes is true?

A.The availability of S3 and S3-IA is the same.

B. The durability of S3 and S3-IA is the same.

C. The latency of S3 and Glacier is the same.

D. The latency of S3 is greater than that of Glacier.

Answer: B

This is a common question on AWS exams, and relates to your understanding of the various S3 classes. S3 and S3-IA have the same durability, but the availability of S3 is one 9 greater than S3-IA. S3 has 99.99 availability, while S3-IA has 99.9 availability. Glacier has much greater first-byte latency than S3, so both C and D are false.

For more Questions Please check Certification Sample Quiz under each module

Link: https://k21academy.com/awssaquizm04

4 AWS STORAGE GATEWAY

- The AWS Storage Gateway service enables hybrid storage between on-premises environments and the AWS Cloud.
- It provides low-latency performance by caching frequently accessed data on premises, while storing data securely and durably in Amazon cloud storage services.
- Implemented using a virtual machine that you run on-premises (VMware or Hyper-V virtual appliance).
- Provides local storage resources backed by AWS S3 and Glacier.
- Often used in disaster recovery preparedness to sync data to AWS.
- Useful in cloud migrations.
- AWS Storage Gateway supports three storage interfaces: file, volume, and tape.

The table below shows the different gateways available and the interfaces and use cases:

New Name	Old Name	Interface	Use Case
File Gateway	None	NFS, SMB	Allow on-prem or EC2 instances to store objects in S3 via NFS or SMB mount points
Volume Gateway Stored Mode	Gateway-Stored Volumes	iSCSI	Asynchronous replication of on-prem data to S3
Volume Gateway Cached Mode	Gateway-Cached Volumes	iSCSI	Primary data stored in S3 with frequently accessed data cached locally on-prem
Tape Gateway	Gateway-Virtual Tape Library	ISCSI	Virtual media changer and tape library for use with existing backup software

- Each gateway you have can provide one type of interface.
- All data transferred between any type of gateway appliance and AWS storage is encrypted using SSL.
- By default, all data stored by AWS Storage Gateway in S3 is encrypted server-side with Amazon S3-Managed Encryption Keys (SSE-S3).
- When using the file gateway, you can optionally configure each file share to have your objects encrypted with AWS KMS-Managed Keys using SSE-KMS.

4.1 File Gateway

- File gateway provides a virtual on-premises file server, which enables you to store and retrieve files as objects in Amazon S3.
- Can be used for on-premises applications, and for Amazon EC2-resident applications that need file storage in S3 for object-based workloads.
- Used for flat files only, stored directly on S3.
- File gateway offers SMB or NFS-based access to data in Amazon S3 with local caching.
- File gateway supports Amazon S3 Standard, S3 Standard – Infrequent Access (S3 Standard – IA) and S3 One Zone – IA.
- File gateway supports clients connecting to the gateway using NFS v3 and v4.1.
- Microsoft Windows clients that support NFS v3 can connect to file gateway.
- The maximum size of an individual file is 5 TB.

4.2 Volume Gateway

- The volume gateway represents the family of gateways that support block-based volumes, previously referred to as gateway-cached and gateway-stored modes.
- Block storage – iSCSI based.
- Cached Volume mode – the entire dataset is stored on S3 and a cache of the most frequently accessed data is cached on-site.
- Stored Volume mode – the entire dataset is stored on-site and is asynchronously backed up to S3 (EBS point-in-time snapshots). Snapshots are incremental and compressed.
- Each volume gateway can support up to 32 volumes.
- In cached mode, each volume can be up to 32 TB for a maximum of 1 PB of data per gateway (32 volumes, each 32 TB in size).
- In stored mode, each volume can be up to 16 TB for a maximum of 512 TB of data per gateway (32 volumes, each 16 TB in size).

4.3 Gateway Virtual Tape Libraray

- Used for backup with popular backup software.
- Each gateway is preconfigured with a media changer and tape drives. Supported by NetBackup, Backup Exec, Veeam etc.
- When creating virtual tapes, you select one of the following sizes: 100 GB, 200 GB, 400 GB, 800 GB, 1.5 TB, and 2.5 TB.
- A tape gateway can have up to 1,500 virtual tapes with a maximum aggregate capacity of 1 PB.

4.4 Sample Questions

Q1 : A small business specializing in video processing wants to prototype cloud storage in order to lower its costs. However, management is wary of storing its client files in the cloud rather than on premises. They are focused on cost savings and experimenting with the cloud at this time. What is the best solution for their prototype?

A. Install a VPN, set up an S3 bucket for their files created within the last month, and set up an additional S3-IA bucket for older files. Create a lifecycle policy in S3 to move files older than 30 days into the S3-IA bucket nightly.

B. Install an AWS storage gateway using stored volumes.

C. Set up a Direct Connect and back all local hard drives up to S3 over the Direct Connect nightly.

D. Install an AWS storage gateway using cached volumes.

Answer:B

Explanation : Anytime the primary consideration is storage with a local data presence-where data must be stored or seen to be stored locally-a storage gateway gives you the best option. This reduces the choices to B and D. B will store the files in S3 and provide local cached copies, while D will store the files locally and push them to S3 as a backup. Since management is concerned about storage in the cloud of primary files, B is the best choice; local files are the primary source of data, while still allowing the company to experiment with cloud storage without “risking” its data being stored primarily in the cloud.

Q2 : You have been tasked with ensuring that data stored in your organization's RDS instance exists in a minimum of two geographically distributed locations. Which of the following solutions are valid approaches? (Choose two.)

A. Enable RDS in a Multi-AZ configuration.

B. Enable RDS in a read replica configuration.

C. Install a storage gateway with stored volumes.

D. Enable RDS in a cross-region read replica configuration.

Answer: A, D

Explanation : A Multi-AZ setup is the easiest solution, and the most common. Turning on read replicas (option B) is not a guarantee, as read replicas are not automatically installed in different AZs or regions. However, with option D, a cross-region replica configuration will ensure multiple regions are used. A storage gateway (option C) is backed by S3, not RDS.

For more Questions Please check Certification Sample Quiz under each module

Link: https://k21academy.com/awssaquizm04

5 AWS SNOWBALL

- Petabyte scale data transport solution for transferring data into or out of AWS.
- Uses a secure storage device for physical transportation.
- AWS Snowball Client is software that is installed on a local computer and is used to identify, compress, encrypt, and transfer data.
- Uses 256-bit encryption (managed with the AWS KMS) and tamper-resistant enclosures with TPM.
- Snowball must be ordered from and returned to the same region.
- To speed up data transfer it is recommended to run simultaneous instances of the AWS Snowball Client in multiple terminals and transfer small files as batches.
- Snowball can import to S3 or export from S3.

5.1 The Snowball Family

- Several services are offered in the Snowball family.

The table below describes these at a high-level:

Service	What it Is
AWS Import/Export	Ship an external hard drive to AWS. Someone at AWS plugs it in and copies your data to S3
AWS Snowball	Ruggedized NAS in a box that AWS ships to you. You can copy up to 80TB of data and ship it back to AWS. They copy the data over to S3
AWS Snowball Edge	Same as Snowball, but with onboard Lambda and clustering
AWS Snowmobile	A literal shipping container full of storage (up to 100PB) and a truck to transport it

- Snowball (80TB) (50TB model available only in the USA).
- Snowball Edge (100TB) comes with onboard storage and compute capabilities.
- Snowmobile – exabyte scale with up to 100PB per Snowmobile.
- AWS Import/export is when you send your own disks into AWS – this is being deprecated in favour of Snowball.

5.2 Sample Questions

Q1 : You need to transfer 50 TB of data into S3 and want to avoid lengthy network exchanges and network saturation. What option would provide you with inexpensive data transfer at a large scale?

A. Storage Gateway
B. S3 Transfer Acceleration
C. Glacier
D. Snowball

Answer: D
Explanation: Snowball is almost always the most cost-effective approach to data transfer when you approach 50 TB, and there are good reasons to consider it even at 10 TB or more.

Q2 : You need to transfer several petabytes of data into AWS at the lowest possible costs. What AWS services could help?

A. Large Data Transfer Service
B. S3 Transfer Acceleration
C. Snowball
D. CloudFront

Answer:C
Explanation: Large data should always make you think, “Snowball.” Snowball gives you a reliable, scalable, petabyte-scale data transfer solution.

For more Questions Please check Certification Sample Quiz under each module

Link: https://k21academy.com/awssaquizm04

6 AMAZON CLOUDFRONT

- CloudFront is a web service that gives businesses and web application developers an easy and cost-effective way to distribute content with low latency and high data transfer speeds.
- CloudFront is a good choice for distribution of frequently accessed static content that benefits from edge delivery—like popular website images, videos, media files or software downloads.
- Used for dynamic, static, streaming, and interactive content.

CloudFront is a global service:

- Ingress to upload objects.
- Egress to distribute content.

Amazon CloudFront provides a simple API that lets you:

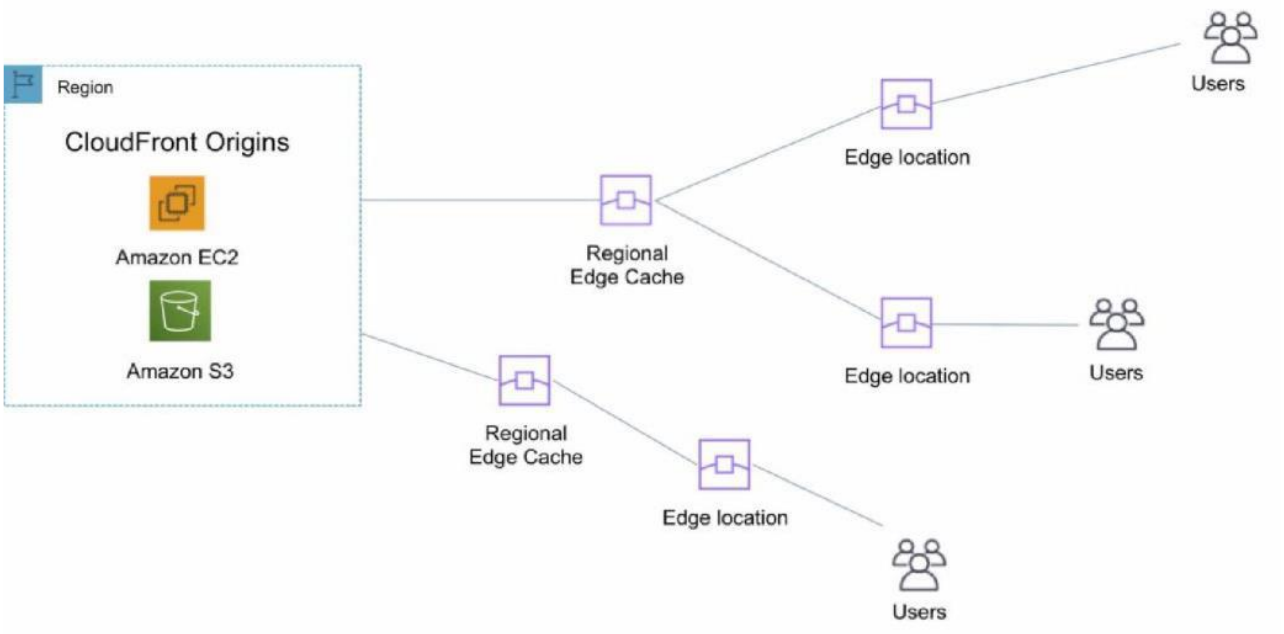
- Distribute content with low latency and high data transfer rates by serving requests using a network of edge locations around the world.
- Get started without negotiating contracts and minimum commitments.
- You can use a zone apex name on CloudFront.
- CloudFront supports wildcard CNAME.
- Supports wildcard SSL certificates, Dedicated IP, Custom SSL and SNI Custom SSL (cheaper). Supports Perfect Forward Secrecy which creates a new private key for each SSL session.

6.1 Edge Locations & Regional Edge Caches

- An edge location is the location where content is cached (separate to AWS regions/AZs).
- Requests are automatically routed to the nearest edge location.
- Edge locations are not tied to Availability Zones or regions.
- Regional Edge Caches are located between origin web servers and global edge locations and have a larger cache.
- Regional Edge Caches have larger cache-width than any individual edge location, so your objects remain in cache longer at these locations.
- Regional Edge caches aim to get content closer to users.

- Proxy methods PUT/POST/PATCH/OPTIONS/DELETE go directly to the origin from the edge locations and do not proxy through Regional Edge caches.
- Dynamic content goes straight to the origin and does not flow through Regional Edge caches.
- Edge locations are not just read only, you can write to them too.

The diagram below shows where Regional Edge Caches and Edge Locations are placed in relation to end users:



6.2 Origins

- An origin is the origin of the files that the CDN will distribute.
- Origins can be either an S3 bucket, an EC2 instance, an Elastic Load Balancer, or Route 53 – can also be external (non-AWS).
- When using Amazon S3 as an origin you place all of your objects within the bucket.
- You can use an existing bucket and the bucket is not modified in any way.
- By default, all newly created buckets are private.

You can setup access control to your buckets using:

- Bucket policies.
- Access Control Lists.
- You can make objects publicly available or use CloudFront signed URLs.

- A custom origin server is a HTTP server which can be an EC2 instance or an on-premise/non-AWS based web server.
- When using an on-premise or non-AWS based web server you must specify the DNS name, ports and protocols that you want CloudFront to use when fetching objects from your origin.
- Most CloudFront features are supported for custom origins except RTMP distributions (must be an S3 bucket).

When using EC2 for custom origins Amazon recommend:

- Use an AMI that automatically installs the software for a web server.
- Use ELB to handle traffic across multiple EC2 instances.
- Specify the URL of your load balancer as the domain name of the origin server.

S3 static website:

- Enter the S3 static website hosting endpoint for your bucket in the configuration.
- Example: http://<bucketname>.s3-website-<region>.amazonaws.com.
- Objects are cached for 24 hours by default.
- The expiration time is controlled through the TTL.
- The minimum expiration time is 0.
- Static websites on Amazon S3 are considered custom origins.
- AWS origins are Amazon S3 buckets (not a static website).
- CloudFront keeps persistent connections open with origin servers.
- Files can also be uploaded to CloudFront.

High availability with Origin Failover:

- Can set up CloudFront with origin failover for scenarios that require high availability.
- Uses an origin group in which you designate a primary origin for CloudFront plus a second origin that CloudFront automatically switches to when the primary origin returns specific HTTP status code failure responses.
- Also works with Lambda@Edge functions.

6.3 Distribution

To distribute content with CloudFront you need to create a distribution.

The distribution includes the configuration of the CDN including:

- Content origins.
- Access (public or restricted).
- Security (HTTP or HTTPS).
- Cookie or query-string forwarding.
- Geo-restrictions.
- Access logs (record viewer activity).

There are two types of distribution.

1. Web Distribution:
   - Static and dynamic content including .html, .css, .php, and graphics files.
   - Distributes files over HTTP and HTTPS.
   - Add, update, or delete objects, and submit data from web forms.
   - Use live streaming to stream an event in real time.

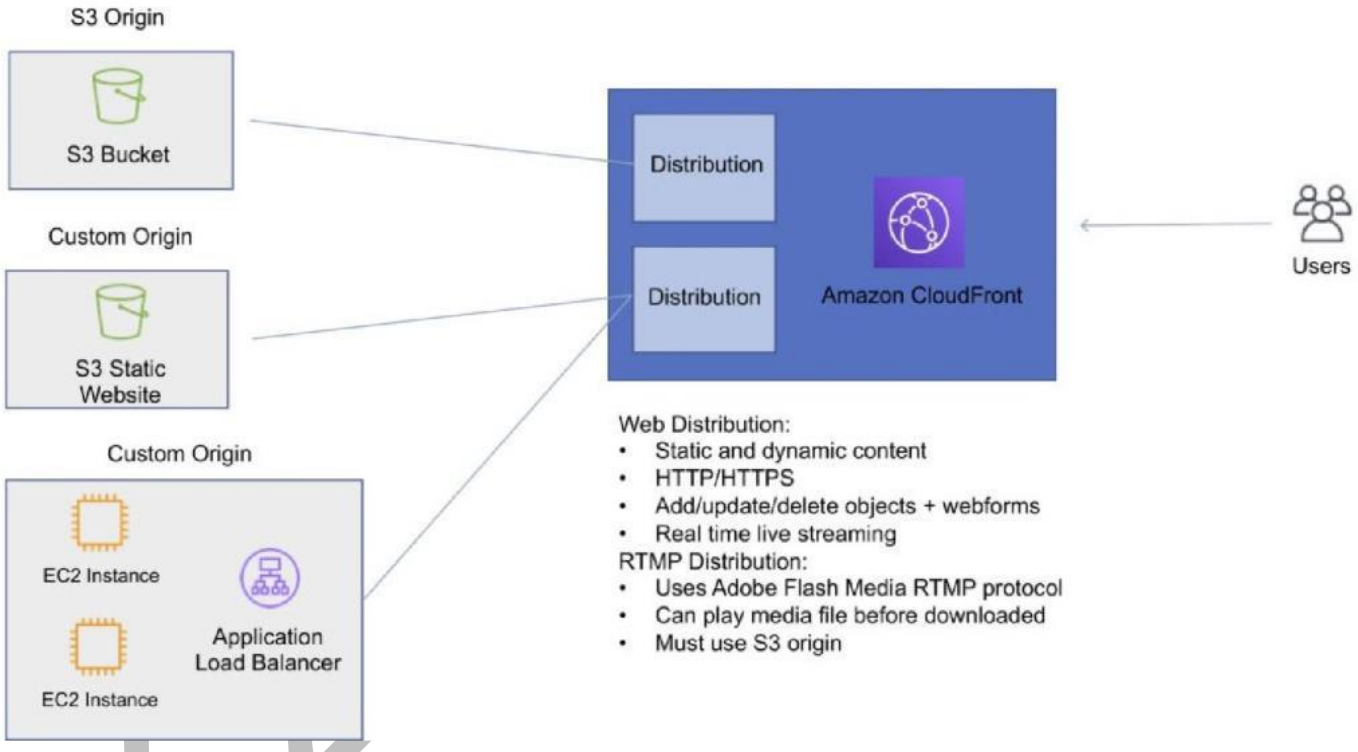
2. RTMP:
   - Distribute streaming media files using Adobe Flash Media Server's RTMP protocol.
   - Allows an end user to begin playing a media file before the file has finished downloading from a CloudFront edge location.
   - Files must be stored in an S3 bucket.
   - To use CloudFront live streaming, create a web distribution.

For serving both the media player and media files you need two types of distributions:

- A web distribution for the media player.
- An RTMP distribution for the media files.
- S3 buckets can be configured to create access logs and cookie logs which log all requests made to the S3 bucket.
- Amazon Athena can be used to analyze access logs.
- CloudFront is integrated with CloudTrail.

- CloudTrail saves logs to the S3 bucket you specify.
- CloudTrail captures information about all requests whether they were made using the CloudFront console, the CloudFront API, the AWS SDKs, the CloudFront CLI, or another service.
- CloudTrail can be used to determine which requests were made, the source IP address, who made the request etc.
- To view CloudFront requests in CloudTrail logs you must update an existing trail to include global services.
- To delete a distribution, it must first be disabled (can take up to 15 minutes).

The diagram below depicts Amazon CloudFront Distributions and Origins:



6.4 Cache Behavior

Allows you to configure a variety of CloudFront functionality for a given URL path pattern.

For each cache behavior you can configure the following functionality:

- The path pattern (e.g. /images/*.jpg, /images*.php).
- The origin to forward requests to (if there are multiple origins).
- Whether to forward query strings.

- Whether to require signed URLs.
- Allowed HTTP methods.
- Minimum amount of time to retain the files in the CloudFront cache (regardless of the values of any cache-control headers).
- The default cache behavior only allows a path pattern of /*.
- Additional cache behaviors need to be defined to change the path pattern following creation of the distribution.

You can restrict access to content using the following methods:

- Restrict access to content using signed cookies or signed URLs.
- Restrict access to objects in your S3 bucket.
- A special type of user called an Origin Access Identity (OAI) can be used to restrict access to content in an Amazon S3 bucket.
- By using an OAI you can restrict users so they cannot access the content directly using the S3 URL, they must connect via CloudFront.

You can define the viewer protocol policy:

- HTTP and HTTPS
- Redirect HTTP to HTTPS
- HTTPS only

You can define the Allowed HTTP Methods:

- GET, HEAD
- GET, HEAD, OPTIONS
- GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE
- For web distributions you can configure CloudFront to require that viewers use HTTPS.

Field-Level Encryption:

- Field-level encryption adds an additional layer of security on top of HTTPS that lets you protect specific data so that it is only visible to specific applications.
- Field-level encryption allows you to securely upload user-submitted sensitive information to your web servers.

- The sensitive information is encrypted at the edge closer to the user and remains encrypted throughout application processing.

Origin policy:

- HTTPS only.
- Match viewer – CloudFront matches the protocol with your custom origin.
- Use match viewer only if you specify Redirect HTTP to HTTPS or HTTPS only for the viewer protocol policy.
- CloudFront caches the object once even if viewers makes requests using HTTP and HTTPS.

Object invalidation:

- You can remove an object from the cache by invalidating the object.
- You cannot cancel an invalidation after submission.
- You cannot invalidate media files in the Microsoft Smooth Streaming format when you have enabled Smooth Streaming for the corresponding cache behavior.
- Objects are cached for the TTL (always recorded in seconds, default is 24 hours, default max is 1 year).
- Only caches for GET requests (not PUT, POST, PATCH, DELETE).
- Dynamic content is cached.
- Consider how often your files change when setting the TTL.
- Invalidation can be used to immediately revoke cached objects – chargeable.
- Deletions propagate.

6.5 Restrictions

- Blacklists and whitelists can be used for geography – you can only use one at a time.
- There are two options available for geo-restriction (geo-blocking):
- Use the CloudFront geo-restriction feature (use for restricting access to all files in a distribution and at the country level).
- Use a 3rd party geo-location service (use for restricting access to a subset of the files in a distribution and for finer granularity at the country level).

AWS WAF

- AWS WAF is a web application firewall that lets you monitor HTTP and HTTPS requests that are forwarded to CloudFront and lets you control access to your content.
- With AWS WAF you can shield access to content based on conditions in a web access control list (web ACL) such as:
- Origin IP address.
- Values in query strings.
- CloudFront responds to requests with the requested content or an HTTP 403 status code (forbidden).
- CloudFront can also be configured to deliver a custom error page.
- Need to associate the relevant distribution with the web ACL.

6.6 Security

- PCI DSS compliant but recommended not to cache credit card information at edge locations.
- HIPAA compliant as a HIPAA eligible service.

Distributed Denial of Service (DDoS) protection:

- CloudFront distributes traffic across multiple edge locations and filters requests to ensure that only valid HTTP(S) requests will be forwarded to backend hosts. CloudFront also supports geoblocking, which you can use to prevent requests from particular geographic locations from being served.

6.7 Domain Names

- CloudFront typically creates a domain name such as a232323.cloudfront.net.
- Alternate domain names can be added using an alias record (Route 53).
- For other service providers use a CNAME (cannot use the zone apex with CNAME).

Moving domain names between distributions:

- You can move subdomains yourself.
- For the root domain you need to use AWS support.
-

6.8 Charges

- There is an option for reserved capacity over 12 months or longer (starts at 10TB of data transfer in a single region).

You pay for:

- Data Transfer Out to Internet.
- Data Transfer Out to Origin.
- Number of HTTP/HTTPS Requests.
- Invalidation Requests.
- Dedicated IP Custom SSL.
- Field level encryption requests.

You do not pay for:

- Data transfer between AWS regions and CloudFront.
- Regional edge cache.
- AWS ACM SSL/TLS certificates.
- Shared CloudFront certificates.

6.9 Sample Questions

Q1 : You have a website running on a fleet of EC2 instances behind an ELB. You also have an Auto Scaling group running across multiple availability zones. The instances are serving files from an EFS file system, but you are incurring lag and significant cost from serving these files from disk over and over. What would you recommend as a solution for reducing costs while still handling high traffic without degradation?

A. Move the files into S3 standard.

B. Use Elastic Transcoder to reduce the file sizes.

C. Cache the files using CloudFront.

D. Use reserved EC2 instances instead of on-demand instances.

Answer: C

Explanation: CloudFront will allow you to cache files that are frequently accessed. In this case, that should actually reduce costs. While CloudFront does incur a new additional cost, it would likely be offset by reduced egress from the EFS as well as the compute of additional EC2 instances to handle requests.

Q2 : Which of the following would not incur a charge?

A. Transferring data from S3 to CloudFront

B. Distributing data via CloudFront to an Internet client in a different region

C. Transferring data from an EC2 instance to an instance in another region

D. Importing data to S3 via Transfer Acceleration

Answer: A

Explanation: It is always free to move data into CloudFront. There may be a cost associated with egress from CloudFront, but the transfer to CloudFront is cost-free.

For more Questions Please check Certification Sample Quiz under each module

Link: https://k21academy.com/awssaquizm04