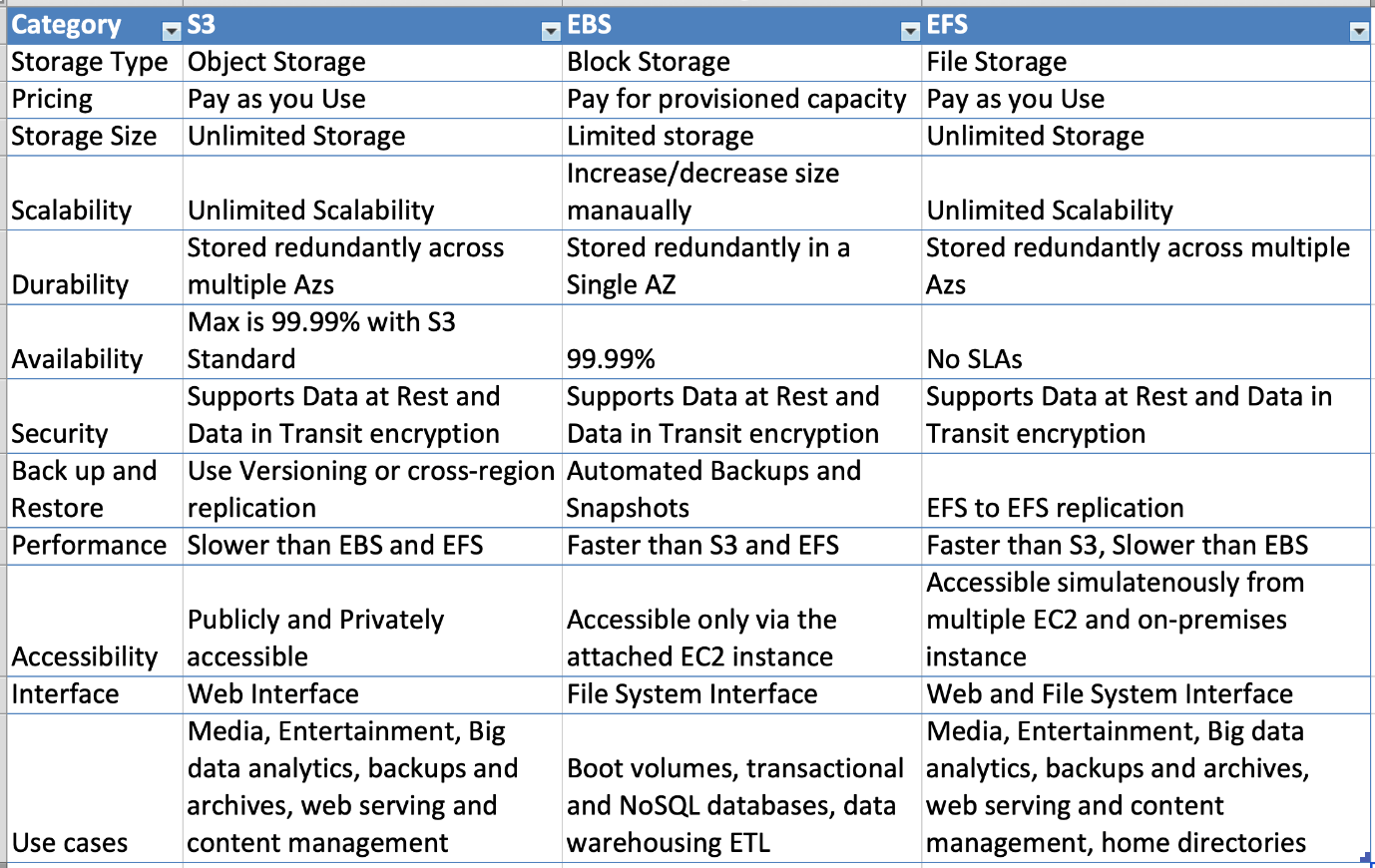
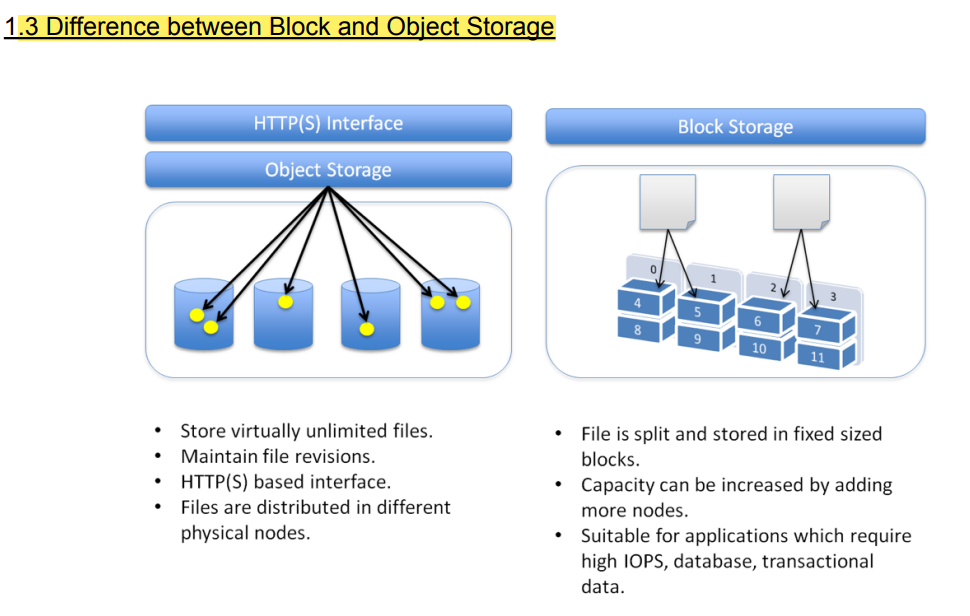
Storage

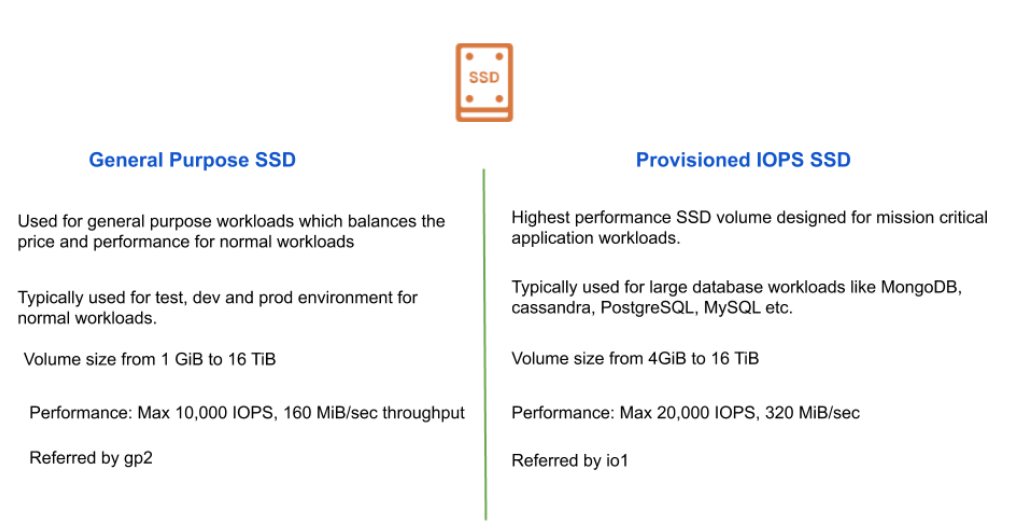
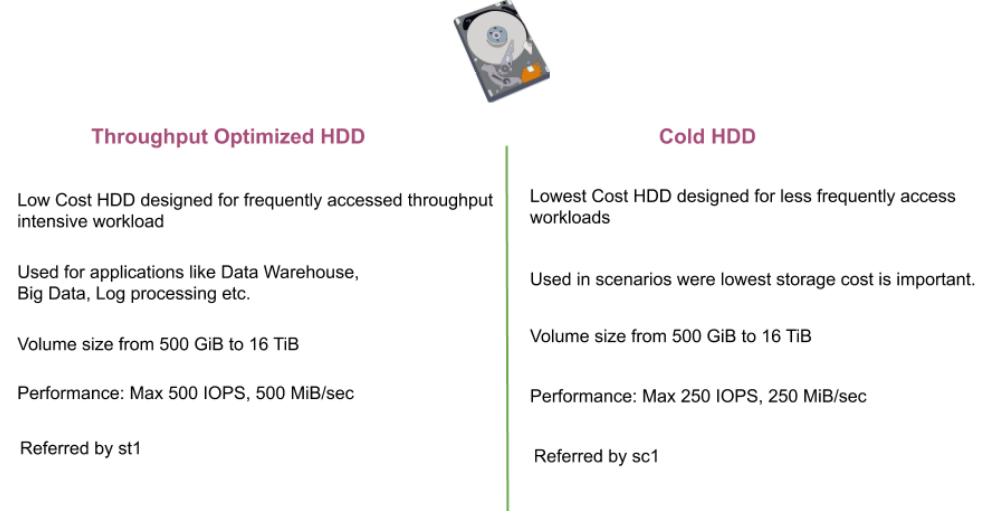


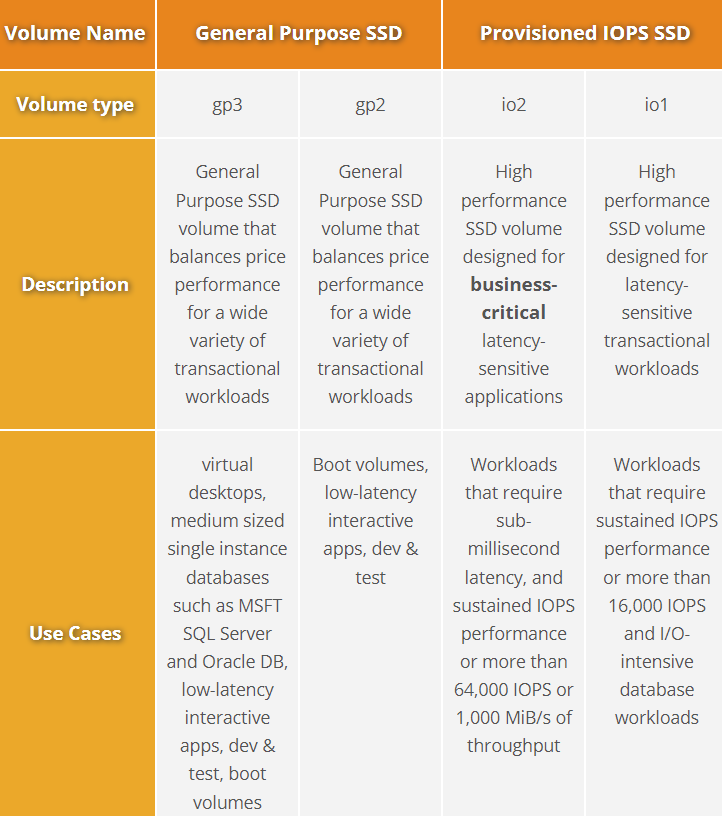


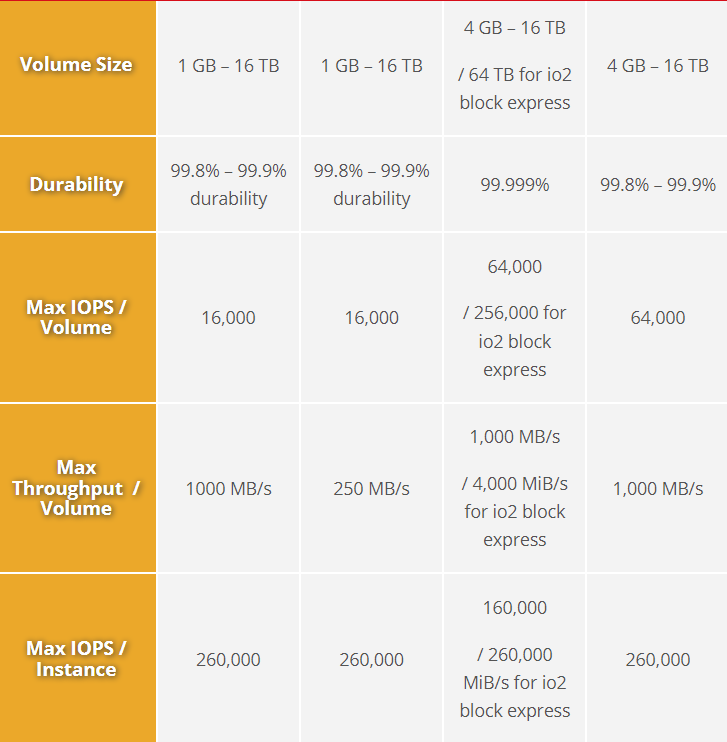
EBS

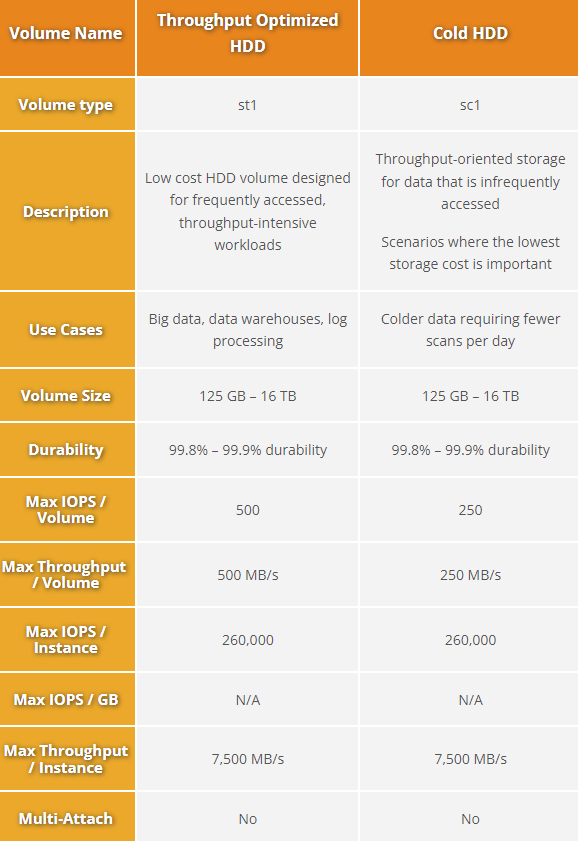
* AWS Elastic Block Store is based on Network Attached Storage, can be easily detached as well hence providing the portability based feature
* Each EBS volume is designed for 99.9999% availability & is automatically replicated within its availability zone and at no additional charge.- True
* EBS Volume replicated to physical hardware with in the same available zone, So if AZ fails then EBS volume will fail. That's why AWS recommend to always keep EBS volume snapshot in s3 bucket for high durability.
* In order for a volume to be available in another region, you need to first create a snapshot from the volume. Then creating a volume from the snapshot , you can then specify the new availability zone accordingly.

EBS Types:









Instance Store

* AWS Instance Store provides temporary block storage volumes for use with EC2.
* This storage is located on the disks that are physically attached to the host computer.
* The size of instance store varies depending on your instance type
* Data in instance store is lost in the following situation:
  + The underlying disk drive fails.
  + The instance stops.
  + The instance terminates
* If planning to use an instance store, make sure you backup your data to central storage places like S3.

S3

* AWS S3 is object storage designed to store and retrieve any amount of data from anywhere
* It is designed for 99.999999999% (11 9s) durability and 99.99% availability.
* Versioning must be enabled for the s3 lifecycle policy to occur- True
* Max Object Size is **5TB (5000GB)**
* If uploading more than 5GB, must use “multi-part upload”
* There are 4 methods of encrypting objects in S3
  + SSE-S3: encrypts S3 objects using keys handled & managed by AWS**. AES-256 encryption type. Must set header: “x-amz-server-side-encryption": "AES256"**
  + SSE-KMS: leverage AWS Key Management Service to manage encryption keys. **Must set header: “x-amz-server-side-encryption": ”aws:kms"**
  + SSE-C: when you want to manage your own encryption keys. **HTTPS must be used. Encryption key must provided in HTTP headers**, for every HTTP request made
  + Client Side Encryption

S3 Storage Classes

S3 offers various kinds of storage classes for different use cases:-

* Standard : General purpose ,support lifecycle policy
* Intelligent-Tiering: Ideal for Unknown or changing access. No retrieval fee, no minimum storage duration
* Standard-IA: Infrequent access, high durability &availability, supports lifecycle policy. Ideal for long-term storage, backups, and as a data store for disaster recovery files.
* One Zone-IA: for infrequently accesed data. Costs 20% less than S3 Standard-IA. For storing secondary backup copies of on-premises data or easily re-creatable data
* Glacier:
* Glacier Deep Archive

S3 Replication:

* S3 Reduced Redundancy(SRR): for frequently accessed data.

AWS S3 Reduced Redundancy storage enables customers to reduce their costs by storing

non-critical, reproducible data at lower levels of redundancy than Amazon S3’s standard storage

* Designed for durability of 99.99% only of objects
* Designed for 99.99% availability over a given year
* you have to designs automation around replacing lost objects.
* S3 Cross-Region Replication (SRR)
  + Many compliances has a requirement that the data must be replicated across greater distances.
  + Cross-Region Replication allows data from S3 buckets to be replicated across regions

S3 Security:

* User based
  + IAM policies - which API calls should be allowed for a specific user from IAM console
* Resource Base
  + Bucket Policies - bucket wide rules from the S3 console - allows cross account
  + Object Access Control List (ACL) – finer grain
  + Bucket Access Control List (ACL) – less common

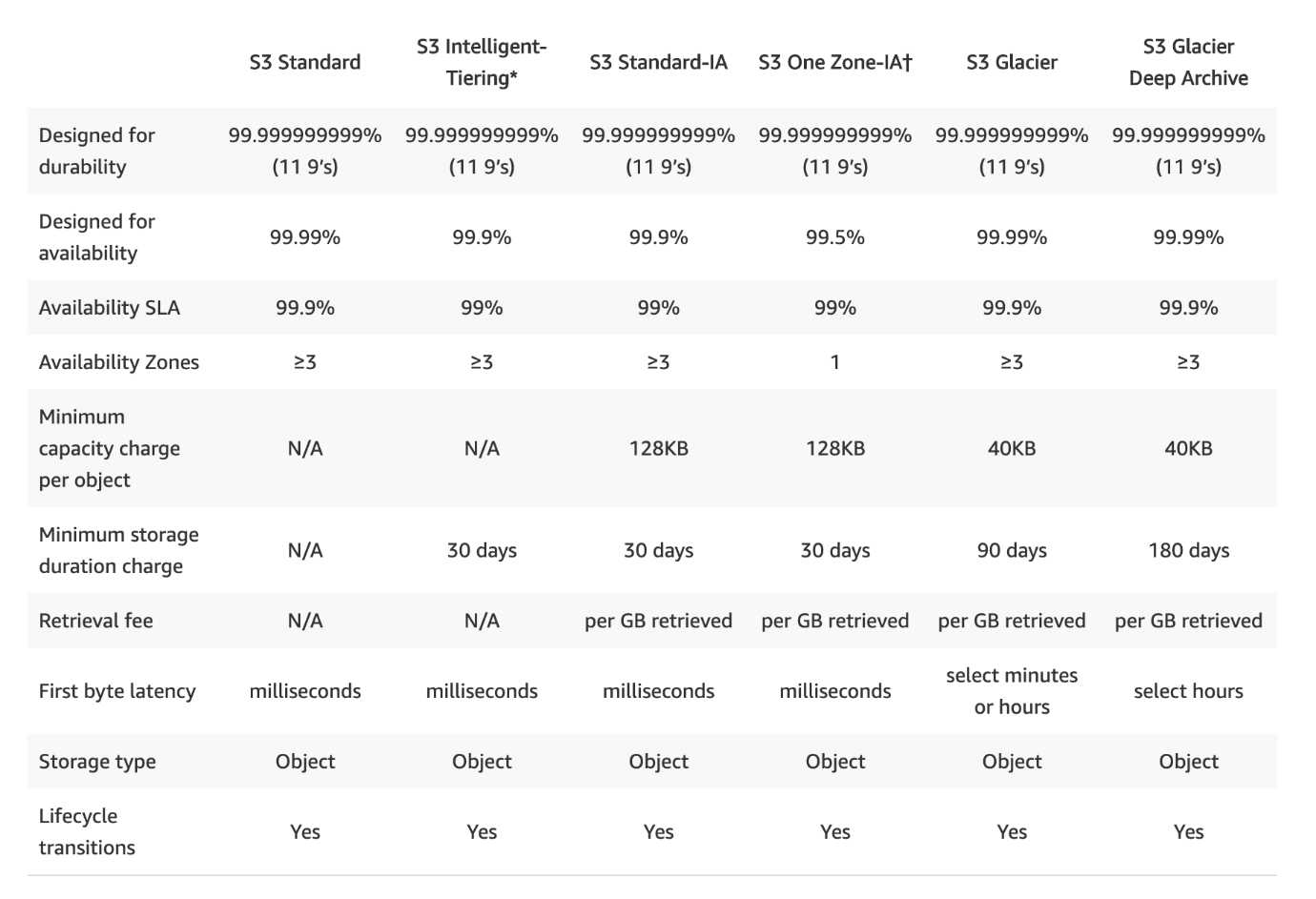
S3 on Outpost:

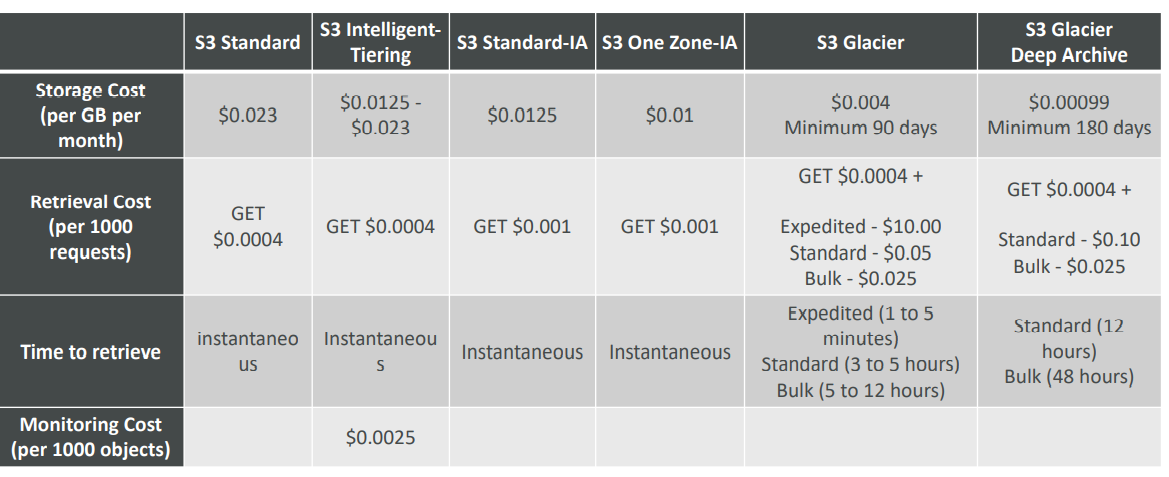
S3 on Outposts makes it easy to store and retrieve data on your Outpost, as well as secure the data, control access, tag, and report on it. S3 on Outposts provides a single Amazon S3 storage class, named S3 Outposts, which uses the S3 APIs, and is designed to durably and redundantly store data across multiple devices and servers on your Outposts. S3 Outposts storage class is ideal for workloads with local data residency requirements, and to satisfy demanding performance needs by keeping data close to on-premises applications.

Aws outposts:

AWS Outposts is a fully managed service that offers the same AWS infrastructure, AWS services, APIs, and tools to virtually any datacenter, co-location space, or on-premises facility for a truly consistent hybrid experience. AWS Outposts is ideal for workloads that require low latency access to on-premises systems, local data processing, data residency, and migration of applications with local system interdependencies.

AWS compute, storage, database, and other services run locally on Outposts, and you can access the full range of AWS services available in the Region to build, manage, and scale your on-premises applications using familiar AWS services and tools.





**S3 MFA-Delete**

* To use MFA-Delete, enable Versioning on the S3 bucket
* You will need MFA to
  + permanently delete an object version
  + suspend versioning on the bucket
* Only the bucket owner (root account) can enable/disable MFA-Delete
* MFA-Delete currently can only be enabled using the CLI

Static Website Hosting

AWS S3 allows us to host static websites.

The static website includes individual webpages which might include static content.

Note:AWS S3 does not support dynamic websites that rely on server-side scripts like PHP, JSP,

ASP.NET, and many more

S3 Presigned URL

All objects in S3 are ‘Private’ by default.

However, Object owner can optionally share objects with others by creating a pre-signed URL to

grant time-limited permission to download the object.

S3 Transfer Acceleration

* Increase transfer speed by transferring file to an AWS edge location which will forward the data to the S3 bucket in the target region
* Compatible with multi-part upload

S3 Byte-Range Fetches

* Parallelize GETs by requesting specific byte ranges
* Better resilience in case of failure
* Can be used to speed up downloads
* Can be used to retrieve only partial data (for example the head of a file)

S3 Select & Glacier Select

• Retrieve less data using SQL by performing server side filtering

• Can filter by rows & columns (simple SQL statements)

• Less network transfer, less CPU cost client-side

S3 Event Notifications:

* S3 event notifications typically deliver events in seconds but can sometimes take a minute or longer
* If **two writes are made to a single non-versioned object** at the same time, **it is possible that only a single event notification** will be sent
* If you want to ensure that an event notification is sent for every successful write, you can enable versioning on your bucket.

S3 – Requester Pays:

* In general, bucket owners pay for all Amazon S3 storage and data transfer costs associated with their bucket
* With Requester Pays buckets, the requester instead of the bucket owner pays the cost of the request and the data download from the bucket
* Helpful when you want to share large datasets with other accounts
* The requester must be authenticated in AWS (cannot be anonymous)

Glacier Vault Lock:

* Adopt a WORM (Write Once Read Many) model
* Lock the policy for future edits (can no longer be changed)
* Helpful for compliance and data retention

S3 Object Lock (versioning must be enabled):

* Adopt a WORM (Write Once Read Many) model
* Block an object version deletion for a specified amount of time
* Object retention:
  + Retention Period: specifies a fixed period
  + Legal Hold: same protection, no expiry date
* Modes:
  + Governance mode: users can't overwrite or delete an object version or alter its lock settings unless they have special permissions
  + Compliance mode: a protected object version can't be overwritten or deleted by any user, including the root user in your AWS account.

When an object is locked in compliance mode, its retention mode can't be changed, and its retention period can't be shortened.

Amazon Athena:

* Serverless query service to perform analytics against S3 objects.
* Uses standard SQL language to query the files
* Supports CSV, JSON, ORC, Avro, and Parquet (built on Presto)
* Pricing: $5.00 per TB of data scanned • Use compressed or columnar data for cost-savings (less scan)
* Use cases: Business intelligence / analytics / reporting, analyze & query VPC Flow Logs, ELB Logs, CloudTrail trails, etc...
* Exam Tip: analyze data in S3 using serverless SQL, use Athena

AWS Storage Gateway

AWS Storage Gateway is a hybrid storage service that allows the on-premise application to

easily use the cloud storage

The storage gateway appliance runs on Ec2 server/ any virtualization server

Types of gateway :

1. File gateway: use file share to synchronously(at the same time) replicate file to s3. It uses NFS (Network File System) and SMB (Server Message Block) protocols for sharing files. we create an NFS mount point on the on-prem server and any files added to that mount point automatically get replicated on s3.

2. volume gateway

Stored volume gateway: async backing i.e. every few hours or weekly, as u have set it. Data is 1st stored in gateway present on the premises and then is backed up onto s3/EBS snapshots. Good to access a large amount of data fast.

Cached volume gateway:  data is not stored on-premises side gateway but directly uploaded to s3, only the cache of recently read or recently written data is present on-prem. Point in time backups to EBS snapshots.

3. Tape gateway

Virtual tape library gateway: same as a cached gateway on-prem side but Data is stored in S3 in the form of virtual tapes.

Snowball:

* With a Snowball, you can transfer hundreds of terabytes or petabytes of data between your on-premises data centers and Amazon Simple Storage Service (Amazon S3).
* In the US regions, Snowballs come in two sizes: 50 TB and 80 TB. All other regions have 80 TB Snowballs only.
* Encryption is enforced, protecting your data at rest and in physical transit.
* You don't have to buy or maintain your own hardware devices.
* First 10 days of onsite usage are free and each extra onsite day is $15
* Asia Pacific (Singapore) is $20 per extra onsite day

If you want to transfer less than 10 terabytes of data between your on-premises data centers

and Amazon S3, Snowball might not be your most economical choice. Use Data Sync or Storage Gateway

Amazon EFS

* Amazon EFS is a serverless, scalable, high-performance file system in the cloud for Linux-based workloads for use with AWS Cloud services and on-premises resources.
* Shared files are generally stored in NAS (network-attached storage)
* It is built to scale on-demand to petabytes without disrupting applications, growing and shrinking

automatically as you add and remove files

* It is designed to provide massively parallel shared access to thousands of Amazon EC2 instances
* To access the file system from an instance inside the VPC, we need to create a mount target in the VPC
* Amazon EFS has four storage classes: Standard, Standard Infrequent Access, One Zone, and One Zone Infrequent Access
* You can create lifecycle management rules to move your data from standard storage classes to infrequent access storage classes.
* Every EFS file system object of Standard storage is redundantly stored across multiple AZs.
* EFS offers the ability to encrypt data at rest and in transit. Data encrypted at rest using AWS KMS for encryption keys. Data encryption in transit uses TLS 1.2
* To access EFS file systems from on-premises, you must have an AWS Direct Connect or AWS VPN connection between your on-premises datacenter and your Amazon VPC.
* EFS file systems can be accessed by Amazon EC2 Linux instances, Amazon ECS, Amazon EKS, AWS Fargate, and AWS Lambda functions via a file system interface such as NFS protocol.
* Amazon EFS supports file system access semantics such as strong consistency and file locking.
* EFS file systems can automatically scale in storage to handle petabytes of data. With Bursting mode, the throughput available to a file system scales as a file system grows. Provisioned Throughput mode allows you to provision a constant file system throughput independent of the amount of data stored.
* EFS file systems can be concurrently accessed by thousands of compute services without sacrificing performance.
* Common use cases for EFS file systems include big data and analytics workloads, media processing workflows, content management, web serving, and home directories.

EFS is a file System whereas FSx are file servers

Amazon FSx : OS specific file servers

Amazon FSx provides highly cost-effective, fully managed, shared cloud file storage for Windows and Linux applications.

2 file system options:

1. Fsx for Windows File server:

Features:

- accessible over SMB protocol.

-data duplication

-end-user file restore

- minimum size 32 Gib

- You can choose the storage type for your file system: SSD or HDD

- FSx for WFS can connect your file system to any VM.

- FSx for WFS is accessible from Windows, Linux, and MacOS compute instances and devices. Thousands of compute instances and devices can access a file system concurrently.

-Each Windows File Server file system can store up to 64 TB of data. You can only manually increase the storage capacity.

- FSx for Windows File Server always encrypts your file system data and your backups at-rest using keys you manage through AWS KMS. Data-in-transit encryption uses SMB Kerberos session keys.

-Common use cases for FSx for WFS include CRM, ERP, custom or .NET applications, home directories, data analytics, media and entertainment workflows, software build environments, and Microsoft SQL Server.

2. FSx for luster :

- it is not specific to Amazon, it is an open-source parallel file system used for high performing computing workloads

- minimum file size to use luster is 1.2 Tb

- very low latency, in sub-milliseconds

- high throughput, 100s of Gib /sec

- up to million IOPs (10 lakh)

- You can choose the storage type for your file system: SSD or HDD

- FSx for Lustre can only be used by Linux-based instances. To access your file system, you first install the open-source Lustre client on that instance. Then you mount your file system using standard Linux commands. Lustre file systems can also be used with Amazon EKS and AWS Batch.

- The Lustre file system is designed for applications that require fast storage. FSx for Lustre file systems can scale to hundreds of GB/s of throughput and millions of IOPS. FSx for Lustre also supports concurrent access to the same file or directory from thousands of compute instances.

- Unlike EFS, storage capacity needs to be manually increased, and only every six hours can you do so.

-FSx for Lustre always encrypts your file system data and your backups at-rest using keys you manage through AWS KMS. FSx encrypts data-in-transit when accessed from supported EC2 instances.

- FSx for Lustre provides two deployment options:

* Scratch file systems are for temporary storage and shorter-term processing of data. Data is not replicated and does not persist if a file server fails.
* Persistent file systems are for longer-term storage and workloads. The file servers are highly available, and data is automatically replicated within the AZ that is associated with the file system.

-Common use cases for Lustre include machine learning, high-performance computing (HPC), video processing, financial modeling, genome sequencing, and electronic design automation (EDA).