MODULE 7 · STUDY NOTES

Storage

Block, object, file, and archive storage — Amazon EBS, Amazon S3, Amazon EFS, and Amazon S3 Glacier, and how to choose among them

Cloud storage is typically more reliable, scalable, and secure than on-premises storage — and it underpins everything else in the cloud: big data analytics, data warehouses, IoT, databases, and backup and archive all sit on some storage architecture. AWS splits the job across four core services, each matched to an access pattern: **Amazon EBS** for block volumes bolted to a single EC2 instance, **Amazon S3** for objects fetched from anywhere by URL, **Amazon EFS** for a file system shared by thousands of instances, and **Amazon S3 Glacier** for archives you touch once a year. The certification exam — and real architecture work — comes down to reading a workload's access pattern and picking the right one.

LEARNING OBJECTIVES

- Identify the different types of storage
- Explain Amazon S3
- Identify the functionality in Amazon S3
- Explain Amazon EBS
- Identify the functionality in Amazon EBS
- Perform functions in Amazon EBS to build an Amazon EC2 storage solution
- Explain Amazon EFS
- Identify the functionality in Amazon EFS
- Explain Amazon S3 Glacier
- Identify the functionality in Amazon S3 Glacier
- Differentiate between Amazon EBS, Amazon S3, Amazon EFS, and Amazon S3 Glacier

7.1 The AWS storage portfolio

Storage is a **core AWS service category**, alongside compute (EC2), networking (VPC), and identity (IAM). Five broad options cover the spectrum, and each is defined by *how data is accessed and how long it lives*:

TABLE 7.1 The five storage categories

CATEGORY	WHAT IT IS	DEFINING ACCESS FACTS
Instance store	Ephemeral (temporary) storage added to an EC2 instance	Data does not survive the instance — scratch space only
Amazon EBS	Persistent, mountable block storage	Mounts as a device to **one** EC2 instance at a time, **within the same Availability Zone** only
Amazon EFS	Shared file system	**Multiple** EC2 instances mount it simultaneously
Amazon S3	Persistent object storage	Each file becomes an **object** with a **URL**; accessible from anywhere
Amazon S3 Glacier	Cold storage	Data not accessed frequently — long-term archival and compliance retention

The module works through each service in turn, with four recorded console demos, a hands-on lab (**Lab 4: Working with Amazon EBS**), and a storage-solution case-study activity that asks you to match businesses to services — exactly the skill the exam tests.

7.2 Amazon Elastic Block Store (Amazon EBS)

Amazon EBS provides **persistent block storage volumes** for EC2 instances — *persistent* (non-volatile) meaning the data survives power-off, unlike the instance store. Each volume is **automatically replicated within its Availability Zone** to protect against component failure, delivering the consistent, low-latency performance workloads need. You can scale usage up or down **within minutes** and pay a low price for only what you **provision**.

First, the fundamental distinction. What happens if you change one character in a 1-GB file? With **block storage**, only the block containing that character is rewritten. With **object storage**, the *entire file* must be re-uploaded. That single difference drives throughput, latency, and cost: block solutions are typically faster and use less bandwidth, but cost more than object storage.

BLOCK STORAGE – AMAZON EBS	OBJECT STORAGE – AMAZON S3
change only the block that changed	*the whole file is re-uploaded*
▪ Edits rewrite individual blocks, not files	▪ Any change replaces the entire object
▪ Typically faster, less bandwidth	▪ Accessed over HTTP(S) by URL
▪ Typically higher cost per GB	▪ Typically lower cost per GB
▪ Feels like a local disk to the instance	▪ Built for scale-out access from anywhere

An EBS volume behaves like a **durable, detachable external hard drive** for an instance: directly attached, so latency between storage and compute stays low — which is why databases run happily on EC2 + EBS. Volumes are captured as part of **Amazon Machine Images (AMIs)**, which are stored in S3 and reused to launch new instances. A volume backup is a **snapshot**: the first is the **baseline**, and every snapshot after it captures only what changed since the previous one.

EBS USE CASES

- **Boot volumes** and storage for EC2 instances
- Data storage with a file system
- **Database hosts** requiring frequent updates and low latency
- Enterprise applications and throughput-intensive continuous disk scans

TABLE 7.2 EBS volume types

TYPE	BOOT VOLUME?	CHARACTER & WHEN TO CHOOSE IT
General Purpose SSD	Yes	Usually sufficient when your application doesn't need extreme performance — the sensible default
Provisioned IOPS SSD	Yes	Highest performance available; choose only when the workload will actually use it
Magnetic (HDD)	No	Lower-cost option for additional storage and non-boot use cases

Matching the correct technology to the workload is the stated best practice for reducing storage cost — don't pay for Provisioned IOPS an application won't use. Note the hard rule: **only SSD volumes can serve as EC2 boot volumes**.

THREE HEADLINE FEATURES

- **Snapshots** — point-in-time backups; re-create a new volume from one at any time; share snapshots or **copy them to other Regions** for disaster recovery (e.g., encrypt and copy from Virginia to Tokyo).
- **Encryption** — encrypted volumes at **no additional cost**; data moving between the instance and the volume inside AWS data centers is encrypted in transit.
- **Elasticity** — increase capacity or change type (HDD → SSD, 50 GB → 16 TB) **dynamically, without stopping the instance**.

TABLE 7.3 What EBS costs

COST DIMENSION	HOW YOU'RE CHARGED
Volumes	All types: by the amount **provisioned** in GB per month, until the storage is released
IOPS — General Purpose SSD	I/O **included** in the volume price
IOPS — Magnetic	Charged by the **number of requests** to the volume
IOPS — Provisioned IOPS SSD	Also charged by IOPS provisioned × the percentage of days provisioned for the month
Snapshots	Added cost **per GB-month** of data stored in S3
Data transfer	**Inbound is free**; copying snapshots across Regions is charged for the transfer, then standard snapshot storage rates apply in the destination Region

COMMON PITFALL

Two attachment rules trip students constantly: an EBS volume mounts to **one instance at a time**, and instance + volume must sit in the **same Availability Zone**. If a scenario needs many instances reading one data set, the answer is EFS (or S3) — never EBS.

LAB 4 – WORKING WITH AMAZON EBS

The module's lab walks the full volume lifecycle: **create** an EBS volume → **attach** it to an EC2 instance → configure the instance to use the virtual disk → **snapshot** the volume → **restore** the snapshot into a new volume. The recorded demo covers the same ground in the console, including mounting the volume from the command line.

7.3 Amazon Simple Storage Service (Amazon S3)

Amazon S3 is **managed, object-level cloud storage** built to collect, store, and analyze data at massive scale — websites, mobile apps, corporate applications, IoT sensor data. Data lives as **objects** inside **buckets**. Storage is virtually unlimited (S3 holds trillions of objects and peaks at millions of requests per second), a single object can be up to **5 TB**, and the service is **designed for 11 9s of durability**. Because S3 is object storage, changing part of a file means re-uploading the entire modified file.

Your data is tied to no particular server and you manage no infrastructure. By default S3 stores data **redundantly across multiple facilities and multiple devices in each facility** within your chosen Region — designed to survive concurrent data loss in **two** AWS facilities. Storage behind a bucket scales automatically as data grows: nothing to provision, and you're billed only for what you use.

ACCESS AND SECURITY

- **None of your data is shared publicly by default.**
- Fine-grained control via **IAM policies**, **S3 bucket policies**, and **per-object ACLs**; access can also stay private through a **VPC endpoint**.
- Encrypt data **in transit** and optionally enable **server-side encryption** on objects.
- Reach S3 through the console, **AWS CLI**, **SDKs**, or REST endpoints over HTTP/HTTPS.
- **Event notifications** fire when objects are uploaded or deleted — and can trigger processes like AWS Lambda functions.
- **Storage class analysis** watches access patterns and identifies the right lifecycle policy for moving cooler data to Standard-IA; results visualize daily in the console or export for BI tools like Amazon QuickSight.

Buckets and URLs. You create a bucket in a chosen AWS Region and upload nearly any number of objects. Bucket names must be **globally unique across all of S3** and **DNS-compliant** (object keys should stick to URL-safe characters), because every object is URL-addressable. Two URL styles refer to the same bucket in, say, Tokyo (`ap-northeast-1`): path-style `https://s3.ap-northeast-1.amazonaws.com/bucket-name` and virtual-hosted-style `https://bucket-name.s3-ap-northeast-1.amazonaws.com`. An object is **data plus metadata** describing it, and permissions can be set on both at upload time.

TABLE 7.4 The six S3 storage classes

CLASS	DESIGNED FOR	FACTS TO RETAIN
S3 Standard	Frequently accessed data — cloud apps, dynamic websites, content distribution, mobile/gaming, big data analytics	11 9s durability, **four 9s availability**; low latency, high throughput
S3 Intelligent-Tiering	Long-lived data with **unknown or unpredictable** access patterns	Auto-moves objects untouched for 30 days to the infrequent tier, back on access; **no retrieval fees**, small per-object monitoring/ automation fee
S3 Standard-IA	Less-frequent access that must be **rapid** when it happens — long-term storage, backups, DR files	Standard's durability/throughput/latency with lower GB price + **per-GB retrieval fee**; **three 9s availability**
S3 One Zone-IA	Infrequent access where you can tolerate less resilience — secondary backup copies, easily re-creatable data, cross-Region replication targets	Data in a **single AZ** (other classes use ≥ 3); costs less than Standard-IA
S3 Glacier	Secure, durable, low-cost archiving	Three retrieval options (minutes to hours); upload directly or via lifecycle policies
S3 Glacier Deep Archive	Lowest-cost storage — data accessed once or twice a year, retained **7–10+ years** for regulatory compliance; tape replacement	Restore **within 12 hours**; replicated across ≥ 3 geographically dispersed AZs; 11 9s durability

COMMON USE CASES AND SCENARIOS

- **Application assets** — a shared location any application instance (EC2 or traditional) can reach; clients can fetch content directly from S3, offloading your servers.
- **Static web hosting** — serve HTML, CSS, JavaScript straight from a bucket.
- **Backup and disaster recovery** — high durability plus **cross-Region replication** for automatic geographic redundancy.
- **Staging area for big data** before analysis.
- Business scenarios: backup and storage services, application hosting, media hosting (video/photo/music up- and downloads), and software delivery.

TABLE 7.5 S3 pricing – what you pay for and what's free

YOU PAY FOR	YOU DO NOT PAY FOR
GBs per month stored (rate varies by storage class)	Data transfer **IN** to Amazon S3
Transfer OUT to other Regions	Transfer OUT from S3 to **Amazon CloudFront** in the same Region
Requests — PUT, COPY, POST, LIST, GET (GET is priced at different rates than the others)	Transfer OUT from S3 to **EC2 in the same Region**

The request verbs matter beyond billing: **GET** retrieves an object (needs READ access), **PUT** adds an object to a bucket (needs WRITE), and **COPY** duplicates an object already in S3 — internally the same as a GET followed by a PUT. The rule of thumb: *you pay only for transfers that cross your Region's boundary outbound.* The recorded demo shows bucket creation, uploads, folders, and the common bucket settings.

> **COMMON PITFALL**
>
> Keep **durability** and **availability** straight. Every class discussed here is designed for 11 9s of *durability* (your data isn't lost). *Availability* — can you reach it right now — is where classes differ: Standard four 9s, Standard-IA three 9s. An exam answer swapping those numbers is wrong.

7.4 Amazon Elastic File System (Amazon EFS)

Amazon EFS is a **fully managed, shared file system** — file storage over a network, implemented with the **NFS protocol (versions 4.0 and 4.1)** and compatible with **all Linux-based AMIs** for EC2. It provides the storage semantics real applications expect from a file system — strong consistency and file locking, via standard OS file I/O APIs — and it is built to scale on demand **without disrupting applications**, growing and shrinking automatically as you add and remove files. There's no minimum fee or setup cost: you pay only for the storage you use.

> **EFS CHARACTERISTICS**
>
> - **Petabyte-scale, low-latency** file system — scales from gigabytes to petabytes with no provisioning.
> - **Shared storage**: thousands of EC2 instances can access one file system at the same time, with consistent performance to each.
> - **Elastic capacity** — grows and shrinks as files come and go.
> - Ideal workloads: **big data and analytics, media processing workflows, content management, web serving, and home directories**.
> - Accessible from the console, an API, or the CLI; designed to be highly durable and highly available.

Architecture. You create the file system inside a VPC and reach it through **mount targets** — network interfaces created in subnets, **one mount target per Availability Zone** (even if the AZ has multiple subnets, the mount target lives in just one). EC2 instances running in multiple AZs of the same Region can all access the file system, so many users share one common data source; best practice is to mount from the **target in your own AZ**. The mount command can use either the target's **IP address or DNS name**.

TABLE 7.6 EFS resources

RESOURCE	ROLE	PROPERTIES WORTH KNOWING
File system	The **primary** resource	ID, creation token, creation time, size in bytes, number of mount targets, state
Mount target	Access point for the file system	Mount target ID, **subnet ID**, file system ID, **IP address** for mounting, state; secured by security groups; must be in the same VPC
Tags	Organizational metadata	Key-value pairs you assign per file system

Mount targets and tags are **subresources** — they exist only in association with a file system.

IMPLEMENTATION – FIVE STEPS, IN ORDER

- **1.** Create your EC2 resources and launch your instance (create a key pair first if you don't have one).
- **2.** Create your Amazon EFS file system.
- **3.** Create mount targets in the appropriate subnets.
- **4.** Connect your EC2 instances to the mount targets.
- **5.** Verify the resources and protect your AWS account (clean up).

COMMON PITFALL

EBS vs. EFS in one line: EBS is a disk for **one** instance in **one** AZ; EFS is a file system for **many** instances across the AZs of a Region. Any scenario with the words *shared*, *concurrent*, or *multiple instances* pointing at the same files is an EFS answer.

7.5 Amazon S3 Glacier

Amazon S3 Glacier is a **data archiving service designed for security, durability, and extremely low cost** — cheaper even than S3 — with the trade-off that **you cannot retrieve data immediately**. Retrieval can take hours, which is exactly why it suits archives and long-term backup. Glacier is designed for **11 9s of durability**, supports **SSL/TLS encryption in transit**, encrypts data **at rest by default**, and its **Vault Lock** feature enforces compliance through policy. Pricing is based on Region.

KEY TERMS

Archive
Any object — photo, video, file, document — stored in Glacier; the base unit of storage. Each archive gets a unique ID and can carry a description.

Vault
A container for storing archives; you name it and pick its Region at creation.

Vault access policy
Determines who can and cannot access a vault's data and which operations they may perform — one per vault.

Vault lock policy
Ensures a vault **cannot be altered** — each vault can have one access policy and one lock policy attached.

TABLE 7.7 Glacier retrieval options

OPTION	TYPICAL AVAILABILITY	COST POSITION
Expedited	1–5 minutes	Highest cost
Standard	3–5 hours	Between expedited and bulk
Bulk	5–12 hours	Lowest cost

Think of the options like choosing a shipping speed for a package — pay for exactly the urgency you need.

WHERE GLACIER EARNS ITS KEEP

- **Media asset archiving** — petabytes of video and news footage, moved back to S3 for distribution when needed.
- **Healthcare information archiving** — patient records (LIS, PACS, EHR) retained for decades to meet regulation.
- **Regulatory and compliance archiving** — Vault Lock supports controls such as **SEC Rule 17a-4(f)**.
- **Scientific data archiving** — sheds hardware, facility, and capacity-planning burdens.
- **Digital preservation** — regular, systematic data-integrity checks; designed to be automatically self-healing.
- **Magnetic tape replacement** — no upfront investment or specialized library maintenance.

Working with Glacier. The console handles only a few operations — creating and deleting vaults, and creating and managing archive policies. Nearly everything else goes through the **Glacier REST APIs, the AWS Java or .NET SDKs, or the AWS CLI**. The other route into Glacier is automatic: **S3 lifecycle policies** transfer data between the active storage classes (Standard, Intelligent-Tiering, Standard-IA, One Zone-IA) and Glacier, cycling data to cheaper tiers at regular intervals so you pay less as data loses urgency. Rules can be set **per object or per bucket**.

WORKED LIFECYCLE SCENARIO

A user uploads a video; your app generates a thumbnail preview and stores it in **S3 Standard** (likely to be viewed right away). Usage data shows most previews go untouched after 30 days, so a lifecycle policy moves them to **Standard-IA at day 30**. Another 30 days later they move to **S3 Glacier**, sit there for **1 year**, then are **deleted** — all of it automatic, no operator involved.

AMAZON S3	AMAZON S3 GLACIER
frequent, low-latency access	*low-cost, long-term, infrequent access*
▪ Maximum item size **5 TB**	▪ Items up to **40 TB**
▪ Higher storage cost per GB	▪ Extremely low storage cost per GB
▪ Request charges: PUT, COPY, POST, LIST, GET	▪ Request charges: **UPLOAD and retrieval** — each retrieval costs more than an S3 request
▪ Server-side encryption is **opt-in** — your application enables it	▪ Data archived in Glacier is **encrypted by default**

TABLE 7.8 S3 server-side encryption options (protecting data at rest)

OPTION	WHO PROVIDES KEYS	HOW IT WORKS
SSE-S3	Amazon S3	Strong multi-factor encryption: each object encrypted with a **unique key**, that key encrypted with a regularly rotated master key; **AES-256**
SSE-C	You (the customer)	You send your own encryption key with the request; S3 performs encryption on write and decryption on access
SSE-KMS	AWS KMS	**Customer Master Keys (CMKs)** managed in AWS KMS; centrally create keys, define usage policies, and **audit key usage** — via the IAM console's Encryption Keys section or the API

Glacier security: by default only you can access your data. Access is enabled and controlled with **IAM policies**, data is encrypted with **AES-256**, and **Glacier manages the keys for you**. Both S3 and Glacier transfer data securely over HTTPS.

7.6 Choosing a storage solution

The module closes with a case-study activity: three companies, three architectures, one question — *which storage service fits?* Case 1 is a travel-analytics firm storing **billions of customer events per day**; case 2 is a collaboration-software company processing email for 250+ enterprises and 500,000+ users, storing **petabytes**; case 3 is a data-protection company ingesting large volumes of customer data under **compliance requirements** (EC2 for compute, DynamoDB for metadata lookups). In every case the method is the same: read the access pattern, then map it to a service.

TABLE 7.9 Decision shorthand

THE SCENARIO SAYS...	REACH FOR	BECAUSE
Boot volume, database on EC2, low-latency disk for one instance	Amazon EBS	Persistent block storage attached to a single instance in the same AZ
Many instances share the same files; home directories; content management	Amazon EFS	NFS file system mounted concurrently across AZs in a Region
Store/retrieve any amount from anywhere; static site; backup target; big data staging	Amazon S3	Object storage by URL, 11 9s durability, virtually unlimited
"Not frequently accessed," "archive," "compliance retention," "cost-effective"	S3 Glacier	Extremely low cost; retrieval in minutes to hours is acceptable
Temporary scratch data that can vanish with the instance	Instance store	Ephemeral by design

SAMPLE EXAM QUESTION, WORKED

A company wants to store data that is not frequently accessed. What is the best and cost-effective solution? Options: AWS Storage Gateway, Amazon S3 Glacier, Amazon EBS, Amazon S3. Strategy: rule out choices from the keywords — **"not frequently accessed"** and **"cost-effective"** point directly at archival storage. EBS and S3 serve active data; Storage Gateway is a hybrid connector, not an archive tier. Answer: **B, Amazon S3 Glacier**.

COMMON PITFALL

Don't let *"best"* pull you toward the most capable service — the exam pairs it with a qualifier like *cost-effective* or *not frequently accessed*, and that qualifier decides. S3 would *work* for cold data; Glacier is the **correct** answer because the access pattern doesn't justify S3's price.

Module summary

- Five storage flavors: instance store (ephemeral), EBS (block, one instance, one AZ), EFS (shared NFS file system), S3 (objects by URL from anywhere), S3 Glacier (cold archive).
- Block storage rewrites only changed blocks; object storage re-uploads the whole file — the root of the throughput, latency, and cost differences.

- EBS: replicated within its AZ, snapshot backups to S3 (baseline + incrementals), free encryption, elastic resizing without stopping the instance; three volume types — General Purpose SSD, Provisioned IOPS SSD, Magnetic (only SSDs boot).
- S3: 11 9s durability, objects to 5 TB, globally unique DNS-compliant bucket names, private by default; pay for GB-month, requests, and cross-Region transfer out — transfer in is free.
- Six S3 classes trade access speed for price: Standard, Intelligent-Tiering, Standard-IA, One Zone-IA (single AZ!), Glacier, Glacier Deep Archive (restore within 12 hours).
- EFS: petabyte-scale NFS v4.0/4.1 file system for Linux AMIs, mounted concurrently by thousands of instances via one mount target per AZ; grows and shrinks automatically.
- Glacier: archives up to 40 TB in vaults, encrypted at rest by default, Vault Lock for compliance; retrievals — expedited 1–5 min, standard 3–5 h, bulk 5–12 h.
- Lifecycle policies automate aging: Standard → Standard-IA → Glacier → delete; storage class analysis tells you when; match service to access pattern to control cost.

Review questions

1. A database on one EC2 instance needs fast, durable disk storage. Which service, and what two attachment constraints apply?

Answer: Amazon EBS — it provides low-latency, persistent block storage. The volume can be attached to only one instance at a time, and the instance and volume must be in the same Availability Zone.

2. Why are EBS snapshots after the first one cheap and fast?

Answer: They're incremental: the baseline snapshot captures the full volume, and each later snapshot stores only the blocks that changed since the previous snapshot. Snapshots live in S3 and are billed per GB-month.

3. How does S3 keep 11 9s of durability inside a Region?

Answer: Data is stored redundantly across multiple facilities and multiple devices within each facility, designed to survive concurrent data loss in two facilities. Only One Zone-IA gives this up by using a single AZ.

4. When would S3 Intelligent-Tiering beat Standard-IA?

Answer: When access patterns are unknown or unpredictable. Intelligent-Tiering monitors each object and moves it to the infrequent tier after 30 idle days — and back on access — with no retrieval fees, only a small per-object monitoring fee. Standard-IA suits data you already know is infrequently accessed.

5. What makes EFS the answer for a fleet of web servers sharing content?

Answer: It's a managed NFS (v4.0/4.1) file system that thousands of EC2 instances can mount simultaneously across AZs in a Region, with strong consistency and file locking, elastic capacity, and pay-for-what-you-use pricing — no single-instance limit like EBS.

6. Name the three Glacier retrieval options and when you'd pick each.

Answer: Expedited (1–5 minutes) when you must have the data now and will pay for it; standard (3–5 hours) for routine restores; bulk (5–12 hours) when cost matters most — like choosing a shipping speed.

7. How does encryption at rest differ between S3 and Glacier?

Answer: Glacier encrypts everything at rest by default with AES-256 and manages keys for you. In S3 your application must enable server-side encryption, choosing SSE-S3 (S3-managed keys), SSE-C (keys you supply per request), or SSE-KMS (KMS CMKs with central policies and usage auditing).

8. Design the lifecycle for video thumbnails rarely viewed after a month and worthless after about 14 months.

Answer: A lifecycle policy: store in S3 Standard on upload, transition to Standard-IA after 30 days, to S3 Glacier 30 days later, and delete after a year in the archive — all automatic, paying less at each step as the data cools.