Summary: Introduction to Storage

This lesson introduces the fundamental concept that different types of data require different types of storage solutions. Just as a coffee shop uses specific containers for coffee beans, paperwork, and secret recipes, AWS provides distinct storage categories to efficiently manage various data needs.

The Three Core Types of Cloud Storage

1. Block Storage

- **How it works:** Data is broken down into fixed-size pieces called **blocks**. These blocks are stored with a unique identifier but have no other context or metadata. It functions like a physical hard drive attached to a server.
- **Key Characteristics:**
  - **Fast and Efficient:** Provides low-latency access, ideal for high-performance needs.
  - **Granular Updates:** You can modify a small part of a file (a single block) without having to rewrite the entire file.
- **Best for:** Databases, transactional applications, or any workload requiring frequent and rapid updates.
- **Primary AWS Services:**
  - **Amazon Elastic Block Store (EBS):** A *managed* service providing persistent, durable block storage volumes for EC2 instances.
  - **EC2 Instance Store:** An *unmanaged*, non-persistent (temporary) block storage directly attached to an EC2 instance.

2. Object Storage

- **How it works:** Data is stored as self-contained units called **objects**. Each object bundles the data itself, a unique ID, and descriptive **metadata**. Objects are stored in a flat structure called a **bucket**, not a traditional folder hierarchy.
- **Key Characteristics:**
  - **Massively Scalable:** Offers virtually unlimited scalability for vast amounts of unstructured data.
  - **Durable and Available:** Designed for high durability and accessibility from anywhere.
  - **Immutable Updates:** To change an object, you must rewrite and upload the entire object again.
- **Best for:** Files that don't change frequently, such as images, videos, backups, archives, and log files.
- **Primary AWS Service:**
  - **Amazon Simple Storage Service (S3):** A *fully managed*, highly scalable object storage service.

3. File Storage

- **How it works:** Data is stored in a familiar hierarchical structure of files and folders. It

operates like a shared network drive that multiple users or applications can access simultaneously.

- **Key Characteristics:**
  - **Shared Access:** Allows multiple clients (like EC2 instances) to mount and access the same file system at the same time.
  - **Standard Protocols:** Uses common file system protocols like NFS (Network File System).
- **Best for:** Content management systems, shared corporate directories, and applications that require a shared file system.
- **Primary AWS Services:**
  - **Amazon Elastic File System (EFS):** A *fully managed*, scalable file system for Linux-based workloads (NFS).
  - **Amazon FSx:** A *fully managed* service for launching popular third-party file systems like Windows File Server, Lustre, and NetApp ONTAP.

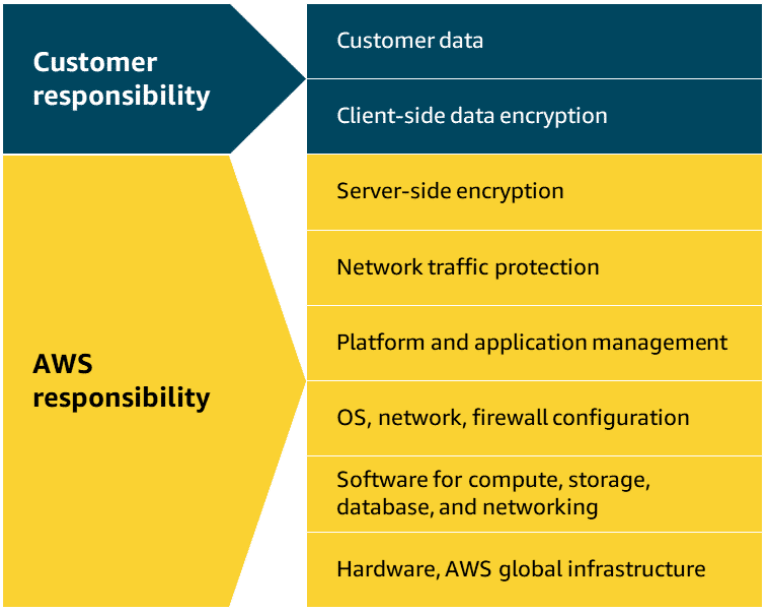
Additional Storage Services

- **AWS Storage Gateway:** A hybrid-cloud service that connects your on-premises applications to virtually unlimited cloud storage.
- **AWS Elastic Disaster Recovery:** A service that helps you recover your physical, virtual, and cloud-based servers into AWS after a disaster.

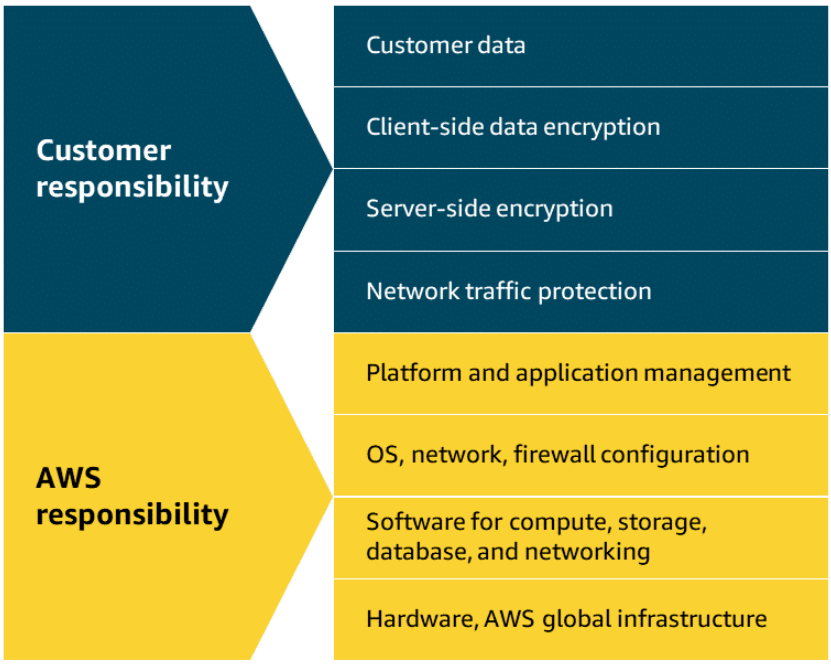
Storage and the Shared Responsibility Model

The level of responsibility you have depends on the type of storage service:

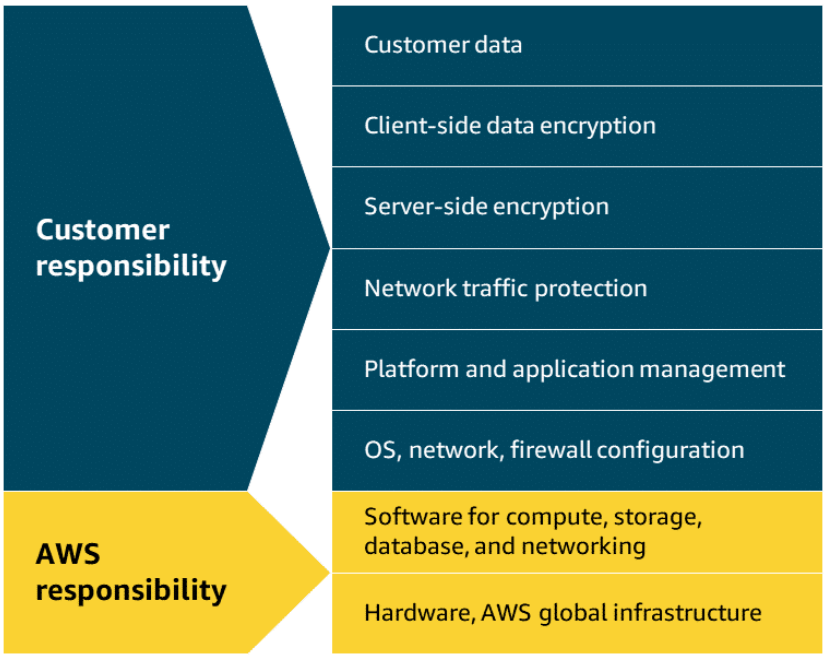
- **Fully Managed Services (e.g., Amazon S3):** AWS manages the entire storage stack, including hardware, availability, and durability. Your responsibility is limited to managing your data and configuring access controls.



- **Managed Services (e.g., Amazon EBS, EFS):** AWS manages the underlying infrastructure and replication. You are responsible for data backups, encryption configuration, and performance/capacity planning.



- **Unmanaged Services (e.g., EC2 Instance Store):** AWS manages only the physical hardware. You are responsible for everything else, including data management, durability, backups, and encryption.



Summary: EC2 Instance Store and Amazon Elastic Block Store (Amazon EBS)

This lesson compares the two primary types of block-level storage available for Amazon EC2 instances. Block storage functions like a traditional hard drive, storing data in chunks or "blocks," and is essential for running applications and operating systems.

1. Amazon EC2 Instance Store

- **What It Is:** A temporary, block-level storage that is **physically attached** to the host computer of an EC2 instance. It's like the built-in, local hard drive of the server.
- **Key Characteristic - Non-Persistent (Ephemeral):** This is the most crucial concept. All data on an instance store is **permanently deleted** if the associated EC2 instance is **stopped** or **terminated**. Because the instance may start on a different physical host later, the local data cannot be reattached.
- **Use Cases:** Ideal for temporary data that doesn't need to be retained long-term.
    - Buffers and caches
    - Scratch space for calculations or temporary file processing
    - Data that is replicated across a fleet of instances (like a distributed database)
- **Benefits:**
    - **High Performance:** Offers extremely low latency and high I/O performance because the storage is physically connected to the instance.
    - **Cost-Effective:** The cost of the instance store is included in the price of the EC2 instance.

2. Amazon Elastic Block Store (Amazon EBS)

- **What It Is:** A durable, high-performance block storage service that provides persistent virtual hard drives (called **EBS volumes**) that are attached to EC2 instances over the network. They function like external, plug-and-play hard drives.
- **Key Characteristic - Persistent:** Data stored on an EBS volume **remains available** even if the EC2 instance is stopped or terminated. The volume exists independently of the instance and can be reattached to the same or another instance later.
- **Use Cases:** The standard and recommended choice for most workloads where data must be retained.
    - Hosting databases and file systems
    - Running enterprise applications
    - Serving as the "boot volume" (operating system drive) for EC2 instances
- **Benefits:**
    - **Data Persistence:** Your data is safe and durable, surviving instance stops and terminations.
    - **Data Portability & Migration:** Volumes can be detached from one EC2

instance and reattached to another. You can also create **snapshots** (point-in-time backups) to migrate data across Availability Zones or create new volumes.
    - **Disaster Recovery:** Snapshots provide a reliable backup mechanism for disaster recovery.
    - **Flexibility & Performance Tuning:** You can change the size and performance characteristics (volume type) of an EBS volume on the fly to match your application's changing demands without downtime.

Key Distinction

The primary difference to remember is **persistence**:

- **EC2 Instance Store** = **Temporary** storage.
- **Amazon EBS** = **Permanent** and durable storage.

Summary: Amazon Elastic Block Store (Amazon EBS) Data Lifecycle

This lesson focuses on how to manage and protect the data stored on EBS volumes throughout its lifecycle, emphasizing the critical role of backups and automation.

The Challenge: Customer Responsibility for Data

According to the AWS Shared Responsibility Model, while AWS manages the physical infrastructure for EBS, **you, the customer, are responsible for managing your own data**. This includes implementing a strategy for:

- Provisioning and de-provisioning volumes.
- Backing up critical data.
- Managing the lifecycle of those backups.

The Solution: Amazon EBS Snapshots

EBS Snapshots are the primary tool for backing up EBS volumes.

- **What They Are:** A point-in-time copy (backup) of an EBS volume. Snapshots are stored durably in Amazon S3, separate from the volume itself.
- **How They Work (Incremental Backups):**
    - **Initial Snapshot:** The first snapshot you take of a volume is a **full copy** of all the data.
    - **Subsequent Snapshots:** All following snapshots are **incremental**. This means they only save the data blocks that have **changed** since the last snapshot was

taken.
- **Key Benefits of Incremental Backups:**
    - **Cost-Effective:** You are not charged for storing redundant data, significantly reducing storage costs.
    - **Fast:** Snapshots complete much faster because less data needs to be copied.
- **Use Cases:**
    - **Data Protection and Recovery:** Restore a volume to a specific point in time in case of data corruption or system failure.
    - **Operational Flexibility:** Create new, identical EBS volumes from a snapshot to quickly launch consistent development or test environments.
    - **Data Migration:** Copy snapshots across different AWS Regions to migrate data.

The Automation Tool: Amazon Data Lifecycle Manager (DLM)

Manually creating and deleting snapshots for hundreds or thousands of volumes is time-consuming and prone to human error. **Amazon Data Lifecycle Manager (DLM)** solves this problem.

- **What It Is:** A fully managed service that allows you to **automate the lifecycle of your EBS snapshots**.
- **How It Works:**
    1. You create a **lifecycle policy**.
    2. In the policy, you define **when** to create snapshots (e.g., every day at 2 AM).
    3. You also define a **retention rule** that specifies **how long** to keep the snapshots (e.g., keep daily snapshots for 7 days, then delete them).
    4. DLM then automatically executes this policy, managing the creation and deletion of snapshots for you.

Shared Responsibility for Snapshots

Even with automation, the customer's responsibility remains:

- **You must define and implement the backup strategy** using tools like DLM.
- You are responsible for ensuring sensitive data within snapshots is encrypted.
- You should regularly test your restoration procedures to verify snapshot integrity.

Summary: Amazon Simple Storage Service (Amazon S3)

This lesson introduces Amazon S3, a foundational and highly versatile object storage service designed to store and retrieve virtually unlimited amounts of data.

What is Amazon S3?

Amazon S3 is a **fully managed object storage service**. Instead of a traditional file system with folders, S3 stores data as objects within containers called buckets. It is designed for high durability, availability, and scalability, making it suitable for a vast range of use cases, from simple backups to complex data lakes.

Core Elements of Amazon S3

1. S3 Buckets

- **Definition:** A bucket is a container for objects stored in S3.
- **Key Feature:** Bucket names must be **globally unique** across all of AWS.
- **Function:** They serve as the top-level organizer for your data and are the primary unit for applying access control policies.

2. S3 Objects

- **Definition:** An object is the fundamental unit of storage in S3, representing any type of file (image, video, document, etc.).
- **Components:** An object consists of three main parts:
    1. **Data:** The actual content of the file.
    2. **Key:** The unique name or identifier for the object within the bucket (like a filename).
    3. **Metadata:** Information about the object, such as its size, last modified date, and content type.
- **Size:** An individual object can be up to 5 terabytes in size.

Key Benefits and Features

- **Durability and Redundancy:** S3 is designed for **99.999999999% (11 nines) of durability**. It automatically stores copies of your objects across multiple physical locations (Availability Zones) to protect against data loss.
- **Virtually Unlimited Storage:** While individual objects have a size limit, there is no limit to the total amount of data you can store in a bucket. S3 scales automatically to meet your needs.
- **Object Lifecycle Management:** You can create automated policies to manage your data throughout its lifecycle. For example, you can automatically move infrequently accessed data to a cheaper storage class or delete old log files after a certain period to optimize costs.
- **Versioning:** You can enable versioning on a bucket to keep a history of all versions of an object. This protects against accidental deletions or overwrites, as you can always restore a previous version.
- **Broad Range of Use Cases:** S3 is extremely versatile and is commonly used for:
    - Backups and data archiving.

    - Hosting static websites.
    - Content storage and distribution for media files (images, videos).
    - Storing application data and log files.
    - Powering data lakes for big data analytics.

Security and Privacy Management

Security is a cornerstone of S3, built on the principle that all data is **private by default**. You must explicitly grant access permissions.

- **Block Public Access:** A key security feature, enabled by default at the account and bucket level, that prevents accidental public exposure of your data.
- **Bucket Policies:** Resource-based policies written in JSON that are attached directly to a bucket. They allow you to grant broad permissions to all objects within that bucket (e.g., "allow public read access to all objects").
- **Identity-based Policies (IAM):** You can attach IAM policies to users, groups, or roles to grant them specific permissions to S3 buckets and objects (e.g., "allow the marketing team to upload files to the `marketing-assets` bucket").
- **Encryption:** S3 supports robust encryption to protect your data:
    - **Encryption in Transit:** Secures data as it travels to and from S3 (using SSL/TLS).
    - **Encryption at Rest:** Secures data while it is stored in S3 buckets.

Summary: Amazon S3 Storage Classes and S3 Lifecycle

This lesson explains how to optimize storage costs in Amazon S3 by selecting the appropriate storage class for your data based on its access patterns and by automating data movement using S3 Lifecycle policies.

Why Multiple Storage Classes?

Not all data is accessed with the same frequency. Amazon S3 provides a range of storage classes, each designed to offer the best balance of performance, access speed, and cost for different use cases. You can use multiple storage classes within a single bucket to match the right tier to the right data.

Amazon S3 Storage Classes

For Frequently Accessed Data:

- **S3 Standard:** The default, general-purpose storage class. Ideal for frequently accessed data like dynamic websites, content distribution, and mobile/gaming applications. Offers high durability, availability, and low-latency performance.
- **S3 Express One Zone:** The highest performance object storage class, purpose-built

for latency-sensitive applications. Delivers single-digit millisecond data access by storing data in a single Availability Zone.

For Automatically Optimizing Costs:

- **S3 Intelligent-Tiering:** Perfect for data with unknown or changing access patterns. It automatically moves objects between frequent, infrequent, and archive access tiers based on usage, optimizing costs without manual intervention.

For Infrequently Accessed Data:

- **S3 Standard-IA (Infrequent Access):** For data that is accessed less frequently but requires rapid access when needed. Cheaper storage price than S3 Standard, but with a per-GB retrieval fee. Ideal for long-term backups and disaster recovery files.
- **S3 One Zone-IA:** Similar to Standard-IA but stores data in a **single Availability Zone**. This makes it about 20% cheaper and is suitable for non-critical, easily recreatable data like secondary backups.

For Archival Data:

- **S3 Glacier Instant Retrieval:** For long-lived archive data that is rarely accessed (e.g., once a quarter) but requires millisecond retrieval when needed. Ideal for medical images or media archives.
- **S3 Glacier Flexible Retrieval:** A low-cost option for archives accessed 1-2 times a year where retrieval times of minutes to hours are acceptable. Offers free bulk retrievals. Great for backups and disaster recovery.
- **S3 Glacier Deep Archive:** The **lowest-cost** storage class in AWS, designed for long-term data retention (7-10 years) for compliance or digital preservation. Data retrieval takes up to 12 hours.

For On-Premises Data:

- **S3 on Outposts:** Delivers S3 object storage to your on-premises AWS Outposts environment, addressing local data residency and low-latency processing needs.

Automating Data Management with S3 Lifecycle

Manually moving objects between storage classes is inefficient. **S3 Lifecycle policies** automate this process.

- **What It Is:** A set of rules you define for a bucket that tells S3 how to manage objects over their lifetime.
- **Lifecycle Actions:**
    1. **Transition Actions:** Define when to move objects to a different (usually cheaper) storage class. For example, "Move objects to S3 Standard-IA 30 days

after creation."
2. **Expiration Actions:** Define when objects should be permanently deleted. For example, "Delete objects 365 days after they are created."

Example Lifecycle Policy:

1. An object is uploaded to **S3 Standard**.
2. **After 30 days** of no access, it automatically transitions to **S3 Standard-IA**.
3. **After another 60 days** (90 days total), it transitions to **S3 Glacier Instant Retrieval**.
4. **After 365 days**, the object is **permanently deleted**.

This automated process ensures that you are always using the most cost-effective storage tier for your data as its access patterns change over time, without requiring manual effort.

Summary: Amazon S3 Demonstration

This lesson provides a practical, step-by-step walkthrough of the fundamental features of Amazon S3 using the AWS Management Console. The demonstration covers creating a bucket, uploading data, and configuring various bucket and object-level settings.

Step 1: Creating an S3 Bucket

The first step is to create a container for your objects.

1. **Navigate to the S3 Console.**
2. **Provide a Globally Unique Bucket Name:** The name must be unique across all AWS accounts worldwide (e.g., `morgan-bucket-2025`).
3. **Select an AWS Region:** Choose the geographic location where your bucket will reside.
4. **Configure Public Access Settings:** By default, **"Block all public access"** is enabled. This is a crucial security feature that prevents accidental data exposure. For the initial setup, this setting is left enabled.

Step 2: Organizing and Uploading Objects

Once the bucket is created, you can add folders and files.

- **Creating Folders:** You can create folders within the bucket to organize your objects logically.
- **Uploading Files:**
    - **Individual Files:** Use the **"Add files"** button to select and upload single files from your local computer.
    - **Entire Folders:** The easiest way to upload a complete folder while preserving its internal structure is to **drag and drop** it directly from your local file system

into the S3 console window.

Step 3: Configuring Object and Bucket Properties

The demo explores the various configuration tabs available at both the bucket and object levels.

A. Bucket-Level Configurations:

- **Properties Tab:**
  - **Bucket Versioning:** Can be enabled to protect against accidental deletions.
  - **Tags:** Apply custom labels for cost tracking or organization.
  - **Default Encryption:** Set a default encryption method for all new objects.
  - **Static Website Hosting:** Configure the bucket to serve a static website.
- **Permissions Tab:**
  - **Block Public Access:** The primary control for preventing public access.
  - **Bucket Policy:** Attach a JSON document to define fine-grained access permissions. The demo shows an example of a policy granting another AWS account permission to upload objects (`s3:PutObject`).
  - **Object Ownership & CORS:** Other advanced permission settings.
- **Management Tab:**
  - **Lifecycle Rules:** Configure rules to automatically transition or delete objects.
  - **Replication:** Set up automatic, asynchronous copying of objects to another bucket.

B. Object-Level Configurations:

By selecting an individual object, you can view and manage its specific properties.

- **Object Overview:** Provides basic information and the **Object URL**, which is used to access the file directly (if permissions allow).
- **Metadata:**
  - **System-defined:** Includes details like creation date, size, and storage class.
  - **User-defined:** Allows you to add your own custom key-value tags for categorization or application-specific needs.
- **Storage Class:** View or change the storage class for the specific object.
- **Versions Tab:** If versioning is enabled on the bucket, this tab allows you to see and manage the complete version history of the object.

Summary: Amazon Elastic File System (Amazon EFS)

This lesson introduces Amazon EFS, a fully managed, scalable, and shared file storage service designed for Linux-based workloads on AWS and on-premises.

What is Amazon EFS?

Amazon EFS is a **managed network file system (NFS)** that provides simple, scalable file storage. Unlike block storage (like EBS) which attaches to a single instance, an EFS file system can be mounted and accessed by **thousands of EC2 instances simultaneously**, from different Availability Zones within a Region.

It is "elastic" because it **automatically grows and shrinks** as you add and remove files, eliminating the need to provision storage capacity in advance. You only pay for the storage you use.

Key Benefits of Amazon EFS

- **Shared Access:** It is the primary solution for workloads that require a shared file system, allowing multiple servers to read and write to the same data concurrently. This is ideal for content management systems, shared code repositories, and data science notebooks.
- **Elastic and Scalable:** The file system scales from gigabytes to petabytes automatically without disrupting applications, ensuring you always have the capacity you need.
- **Multi-AZ Redundancy and Availability:** EFS Standard storage classes automatically replicate your data across multiple Availability Zones (AZs) within a Region, providing high durability and availability.
- **Fully Managed:** AWS handles all the heavy lifting of managing file servers, patching, and maintenance, allowing you to focus on your applications.

Key Distinction: EFS vs. EBS

It is crucial to understand the difference between EFS and EBS:

Feature	Amazon EBS (Elastic Block Store)	Amazon EFS (Elastic File System)
Storage Type	Block Storage (like a hard drive)	File Storage (a true file system)
Access Model	Attaches to a **single** EC2 instance in the **same Availability Zone**.	Can be accessed by **many** EC2 instances across **multiple Availability Zones**.

Scalability	You provision a fixed size. Does **not** scale automatically.	**Scales automatically** as you add/remove files.
Protocol	N/A (acts as a block device)	Network File System (NFS) for Linux

Amazon EFS Storage Classes and Lifecycle Management

To optimize costs, EFS offers different storage classes and automates data movement between them.

EFS Storage Classes:

- **Standard Classes (Multi-AZ):**
  - **EFS Standard:** For frequently accessed files.
  - **EFS Standard-Infrequent Access (Standard-IA):** A lower-cost tier for files that are not accessed every day.
- **One Zone Classes (Single-AZ):**
  - **EFS One Zone** and **EFS One Zone-IA:** These classes store data in a single Availability Zone, offering a lower price point at the cost of Multi-AZ resilience. They are suitable for workloads where data can be easily recreated.
- **Archive Class:**
  - **EFS Archive:** The most cost-effective tier, optimized for long-term storage of cold data that is accessed only a few times a year.

EFS Lifecycle Management: You can create lifecycle policies to automatically manage storage costs. These policies can:

- **Transition to IA:** Move files from Standard to Infrequent Access after a specified period of no access (e.g., 30 days).
- **Transition to Archive:** Move files from Standard to Archive after a longer period of no access (e.g., 90 days).
- **Transition to Standard:** (Optional) Move a file back to the Standard class if it is accessed while in an IA or Archive tier.

Summary: Amazon FSx

This lesson introduces Amazon FSx, a fully managed service that allows you to launch and run popular, feature-rich, and high-performance file systems in the cloud. It is designed for a wide range of workloads and simplifies migrating existing file storage to AWS.

What is Amazon FSx?

Amazon FSx is a file storage service that provides compatibility with several popular commercial and open-source file systems. Unlike Amazon EFS, which is specifically for Linux-based workloads using the NFS protocol, Amazon FSx supports multiple file system types, making it a versatile choice for various application needs.

As a **fully managed service**, Amazon FSx handles the underlying hardware provisioning, software patching, and backups, allowing you to focus on your applications rather than infrastructure management.

Key Benefits of Amazon FSx

- **File System Integration:** Supports industry-standard file system protocols, allowing for seamless integration with existing applications and workflows without code changes.
- **Fully Managed Infrastructure:** Reduces operational complexity by automating administrative tasks.
- **Scalable and High-Performance:** Built on the latest AWS compute and storage technologies, FSx can dynamically scale resources to meet performance and capacity demands.
- **Cost-Effective:** You only pay for the resources you use, and features like data deduplication and automated tiering help optimize costs.

Amazon FSx File System Options

Amazon FSx offers four main file system types, each tailored for specific use cases:

1. Amazon FSx for Windows File Server

- **Description:** Provides fully managed shared storage built on Windows Server, supporting the **Server Message Block (SMB)** protocol.
- **Use Cases:**
    - Migrating Windows-based applications and file servers to AWS.
    - Microsoft SQL Server deployments.
    - Virtual desktops and user home directories.

2. Amazon FSx for NetApp ONTAP

- **Description:** Offers fully managed shared storage with the popular data access and management capabilities of NetApp's ONTAP file system.
- **Use Cases:**
    - Seamlessly migrating workloads that rely on ONTAP features.
    - Building modern applications with enterprise-grade storage features.
    - Streamlining business continuity and disaster recovery.

3. Amazon FSx for OpenZFS

- **Description:** Provides fully managed shared file storage built on the popular OpenZFS file system, accessible via the **NFS protocol**.
- **Use Cases:**
    - Migrating on-premises ZFS or other Linux-based file servers.
    - High-performance data analytics and content management workloads.
    - Accelerating development and testing pipelines.

4. Amazon FSx for Lustre

- **Description:** A fully managed file system optimized for speed, offering the scalability and performance of the Lustre parallel file system.
- **Use Cases:**
    - **High-Performance Computing (HPC):** Ideal for compute-intensive workloads.
    - **Machine Learning (ML):** Accelerating training jobs that require fast access to large datasets.
    - Big data analytics and media processing workflows.

Summary: AWS Storage Gateway

This lesson introduces AWS Storage Gateway, a **hybrid cloud storage service** that seamlessly connects an on-premises software appliance with cloud-based storage. It acts as a bridge, allowing organizations to leverage the scalability and durability of AWS storage within their existing on-premises environments with minimal disruption.

What is AWS Storage Gateway?

AWS Storage Gateway provides on-premises applications with access to virtually unlimited cloud storage. It helps streamline storage management for hybrid cloud use cases, such as moving backups to the cloud, using on-premises file shares backed by cloud storage, and providing low-latency access to data in AWS for local applications.

Key Benefits of Storage Gateway

- **Seamless Integration:** Connects on-premises applications to AWS storage using standard storage protocols (NFS, SMB, iSCSI), preserving existing workflows.
- **Improved Data Management:** Provides a centralized way to manage hybrid storage environments, enhancing data accessibility, security, and compliance.
- **Local Caching:** Maintains a local cache of frequently accessed data on-premises, ensuring low-latency performance for critical applications.
- **Cost Optimization:** Reduces the need for on-premises storage infrastructure by using cost-effective AWS Cloud storage for backups, archives, and primary data.

Storage Gateway Types

Storage Gateway offers three distinct types of gateways to meet different hybrid storage needs:

1. Amazon S3 File Gateway

- **What it is:** Presents a file interface (NFS and SMB) that allows you to store and retrieve objects in Amazon S3 as if they were files on a local file server.
- **How it works:** Files written to the gateway are automatically uploaded to an S3 bucket. A local cache keeps recently used data on-premises for low-latency access.
- **Use Case:** Ideal for on-premises applications that need file-based access to S3, cloud-backed file shares, and migrating file data to the cloud.

2. Volume Gateway

- **What it is:** Provides block storage volumes over the iSCSI protocol. On-premises applications can mount these as local block devices, while the data is stored in AWS. It asynchronously backs up data as Amazon EBS snapshots.
- **It operates in two modes:**
    - **Cached Volume Mode:** The primary data is stored in Amazon S3, and a cache of frequently accessed data is retained locally. This minimizes the need for on-premises storage.
    - **Stored Volume Mode:** The entire dataset is stored on-premises for low-latency access, while asynchronously backing up that data to Amazon S3. This provides durable off-site backups.
- **Use Case:** Perfect for backing up local applications and for disaster recovery strategies.

3. Tape Gateway

- **What it is:** Modernizes backups by replacing physical tape libraries with a virtual tape library (VTL) in AWS. It integrates with existing tape-based backup software.
- **How it works:** Your existing backup application writes data to virtual tapes. These virtual tapes are stored in Amazon S3 and can be archived to lower-cost storage tiers like S3 Glacier Deep Archive.
- **Use Case:** Designed for organizations looking to move away from physical tape backups for archiving and long-term data retention without changing their existing backup workflows.

Summary: AWS Elastic Disaster Recovery

This lesson introduces AWS Elastic Disaster Recovery (DRS), a fully managed service designed to minimize downtime and data loss by enabling rapid and reliable recovery of physical,

virtual, and cloud-based servers into AWS.

What is AWS Elastic Disaster Recovery?

Elastic Disaster Recovery is a service that continuously replicates your critical workloads from a source environment (like an on-premises data center or another cloud) to a low-cost staging area in your AWS account. It uses **continuous, block-level data replication** to keep an up-to-date copy of your servers' entire state, including the operating system, applications, and data.

In the event of a disaster, you can quickly launch thousands of recovery instances on AWS in their fully provisioned state within minutes, significantly reducing recovery time objectives (RTO).

Key Benefits

- **Business Resilience:** Provides robust business continuity by maintaining an exact replica of your production servers. The continuous replication ensures that recovery points are seconds behind the source, minimizing data loss (RPO).
- **Streamlined Disaster Recovery:** Simplifies and automates the entire disaster recovery (DR) process through an intuitive console. This reduces the complexity of manual configurations and the risk of human error during a crisis.
- **Cost Optimization:** Eliminates the enormous expense of building and maintaining a secondary physical data center for disaster recovery. With DRS, you pay a minimal amount for the staging area resources and only pay for the full recovery instances when you actually launch them during a drill or an actual disaster.
- **Non-Disruptive Testing:** Allows you to conduct frequent and non-disruptive disaster recovery drills, ensuring your recovery procedures are effective and meet compliance requirements without impacting your production environment.

Use Cases

Elastic Disaster Recovery is crucial for industries where system availability and data integrity are critical.

- **Healthcare Data Protection:**
    - Hospitals can replicate on-premises servers containing electronic patient records to AWS. This ensures that critical medical data remains accessible during system outages, helping to maintain compliance with regulations like HIPAA.
- **Financial Services Continuity:**
    - A bank can use DRS to protect its core banking and transaction processing systems. If the primary data center fails, the bank can quickly failover to AWS, maintaining customer trust and meeting regulatory requirements for business

continuity.

- **Manufacturing Operations Recovery:**
    - A global manufacturer can protect its production planning and supply chain management systems. Replicating these servers to AWS ensures that a disaster at one site does not halt global operations.

Summary: Comparing Storage Services

This "Cloud in Real Life" lesson compares Amazon S3, Amazon EBS, and Amazon EFS by applying them to three distinct real-world business scenarios to clarify their ideal use cases and key differences.

Scenario 1: Hosting a Static Website

- **Business Challenge:** A coffee shop needs to host its company website. The solution must be simple to manage, automatically scalable to handle customer traffic, and cost-effective.
- **AWS Storage Solution: Amazon S3**
- **Why it's the right fit:**
    - **Simplicity:** By enabling the **static website hosting** feature on an S3 bucket, you can serve HTML, CSS, JavaScript, and image files directly to users without needing a web server.
    - **Scalability:** S3 is a managed service that automatically scales to handle any amount of traffic, from a few visitors to millions, without any manual intervention.
    - **Cost-Effectiveness:** You only pay for the storage you use and the data transferred out to your visitors, making it a very economical choice for static content.

Scenario 2: High-Performance Database for a Mobile App

- **Business Challenge:** A fitness center's mobile app for booking classes is slow, and the database running on an EC2 instance is identified as the bottleneck. The application is performance-critical and requires consistent, low-latency storage for rapid database read and write operations.
- **AWS Storage Solution: Amazon EBS** (specifically a high-performance volume type like Provisioned IOPS SSD).
- **Why it's the right fit:**
    - **Block-Level Access:** Databases require **block storage** for rapid and continuous read/write operations on transactional data. S3 (object storage) is not designed for this use case.
    - **High Performance:** EBS acts as a high-performance virtual hard drive directly attached to a single EC2 instance. To solve the latency issue, the business can

choose a more performant volume type, such as a Provisioned IOPS SSD volume, which is optimized for mission-critical, IOPS-intensive database workloads.
    - **Persistence:** Data on EBS volumes persists independently of the EC2 instance's lifecycle, which is essential for a database.

Scenario 3: Collaborative Platform for a Multi-Location Business

- **Business Challenge:** A chain of automotive repair shops needs a centralized, shared storage platform. Mechanics from multiple locations need to access and collaborate on the same set of large files—such as high-resolution images, videos, and technical diagrams—simultaneously.
- **AWS Storage Solution: Amazon EFS**
- **Why it's the right fit:**
    - **Shared File System:** EFS is a managed, **shared file system** (using the NFS protocol) that can be mounted and accessed by thousands of EC2 instances at the same time, even across different Availability Zones.
    - **Elasticity:** EFS automatically scales its storage capacity up or down as files are added or removed, without requiring you to provision storage in advance.
    - **Performance for Media:** It provides the low latency and high throughput required for media processing workflows and collaborative environments where large files are frequently accessed.

Conclusion: Key Differences at a Glance

- **Amazon S3:** Use for **object storage**. Best for web assets, backups, archives, and unstructured data that can be accessed over the internet.
- **Amazon EBS:** Use for **block storage**. Acts as a hard drive for a *single* EC2 instance. Essential for operating systems and databases.
- **Amazon EFS:** Use for **file storage**. A shared file system that can be accessed by *multiple* EC2 instances. Perfect for shared datasets, content management systems, and collaborative applications.