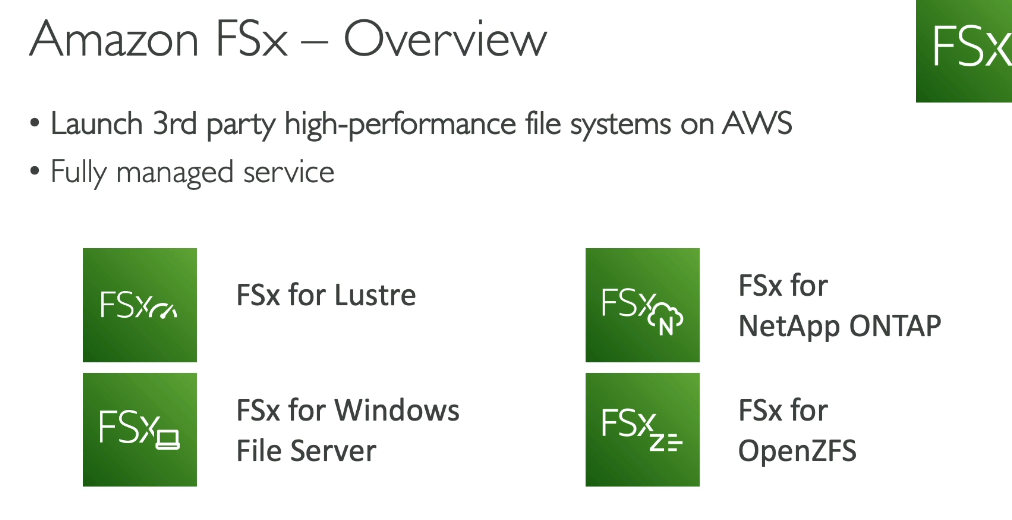
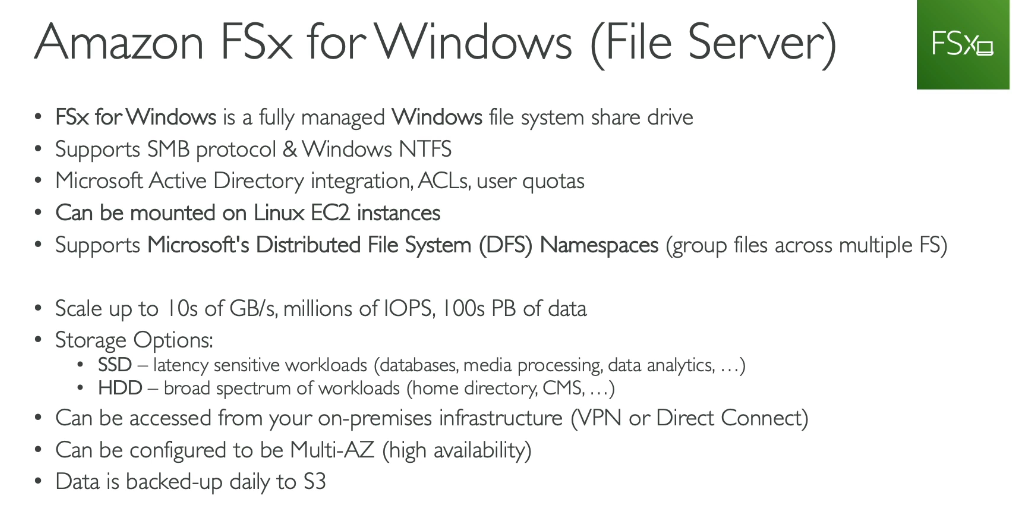


**FSx**

**All FSx file systems can be mounted on Linux EC2 instances, not just FSx for Windows File Server.**



**DFS is not a storage system - it's a namespace and organization layer that sits on top of existing storage.**

What DFS actually is:

* A logical mapping service that creates a unified view of multiple file shares
* A directory structure that points to actual storage locations on different servers
* A referral mechanism that directs users to the physical storage where files are located

What DFS is NOT:

* Not a storage system itself - it doesn't store any file data
* Not a file server - it just organizes and presents existing file shares

**DFS (Distributed File System) Use Case:**

A large enterprise with multiple branch offices needs to provide employees access to shared departmental files. They deploy Windows file servers in each office location and use DFS to:

* Create a unified namespace (e.g., \company.com\departments) that presents all branch file shares as a single directory structure
* Enable DFS Replication to automatically sync files between offices, ensuring employees access local copies for faster performance
* Provide automatic failover - if one server goes down, users are transparently redirected to another server with replicated data
* Manage the underlying Windows servers, including patching, backups, and capacity planning

**Amazon FSx Use Case:**

A media production company migrating to AWS needs shared storage for video editing workflows:

* Deploy FSx for Windows File Server with Multi-AZ configuration for high availability
* Editors access the file system from EC2 Windows instances using SMB protocol with Active Directory authentication
* FSx automatically handles backups, patching, and scaling storage capacity as project files grow
* Use DFS Namespaces (supported by FSx) to scale performance across multiple file systems up to tens of GBps
* Lambda functions automatically process completed videos and move them to department shares
* No server management required - AWS handles all infrastructure maintenance

**Amazon FSx for Windows File Server can be mounted on Linux instances using the SMB protocol.**

How it works:

* Linux instances use the CIFS (Common Internet File System) client to mount FSx for Windows File Server file shares
* You can mount from both Active Directory-joined and non-joined Linux instances
* For non-AD-joined instances, you mount using the file system's private IP address with NTLM authentication

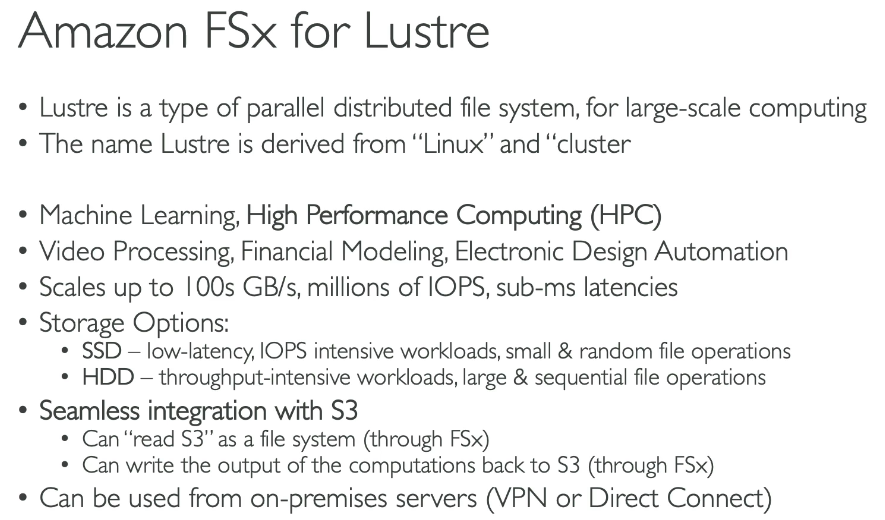
Basic mounting steps:

1. Install the CIFS utilities package on your Linux instance
2. Create a credentials file with your Active Directory username, password, and domain
3. Use the mount command with the -t cifs option to mount the file share

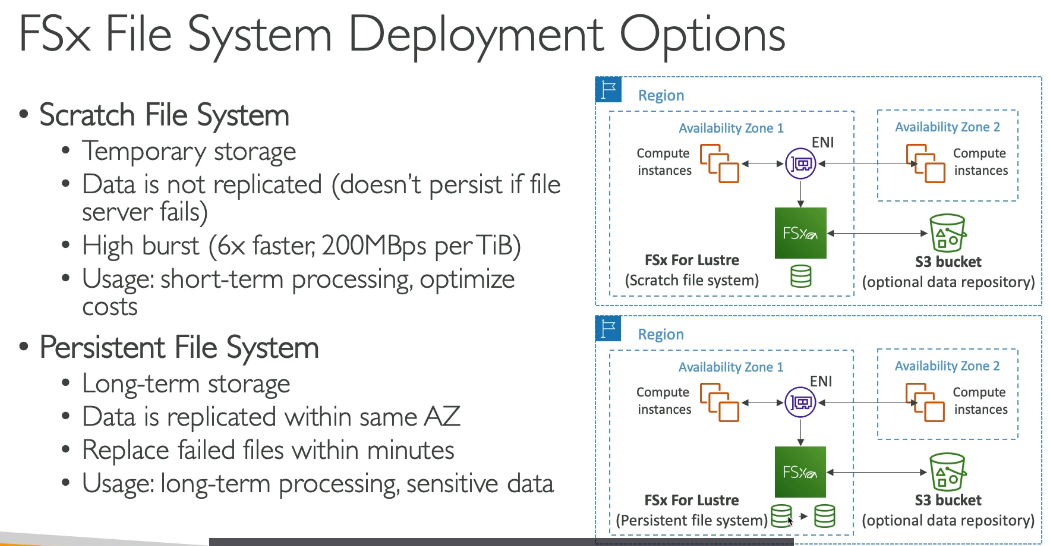
Example mount command: *[Code generation is disabled. Use Amazon Q Developer in the AWS console to generate code.]*

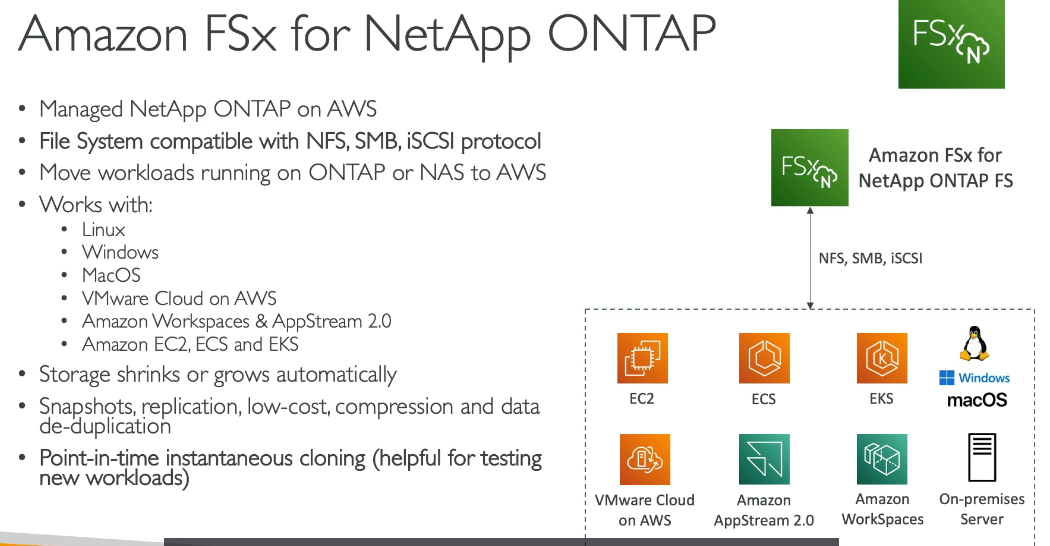
Important notes:

* Linux clients support SMB protocol versions 2.0 and later
* Linux clients do not support automatic DNS-based failover (unlike Windows clients)
* Performance can vary based on SMB protocol version, caching settings, and Linux kernel version This cross-platform capability makes FSx for Windows File Server suitable for mixed Windows and Linux environments.

**Amazon FSx for Lustre uses a custom, POSIX-compliant protocol that is optimized for performance.**

Lustre Deployment options





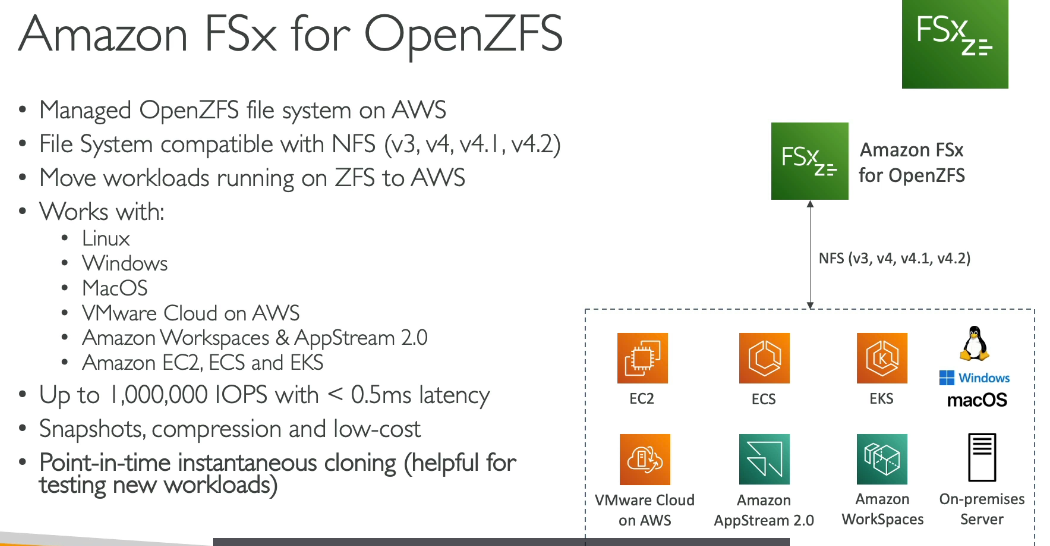
**Data deduplication** is a storage optimization technique that identifies and eliminates redundant data by storing duplicate portions of a dataset only once, rather than multiple times.

How it works:

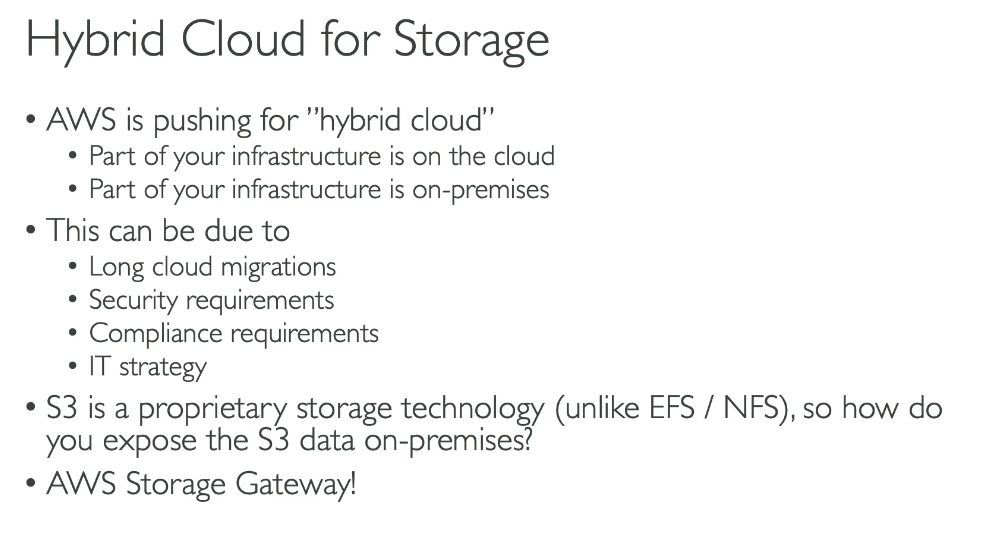
* Scans files and breaks them into smaller chunks
* Identifies duplicate chunks across different files
* Stores each unique chunk only once
* Uses pointers to reference duplicates instead of storing them again
* Often includes compression for additional space savings

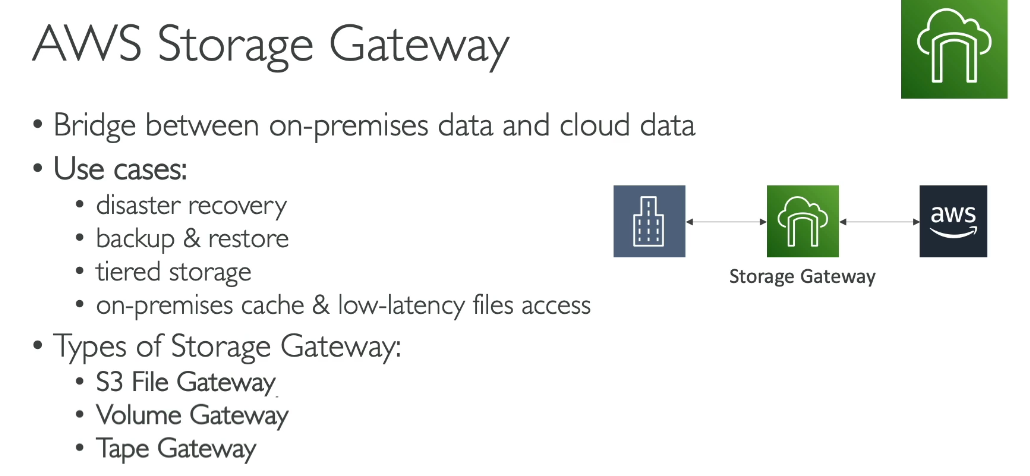
Benefits:

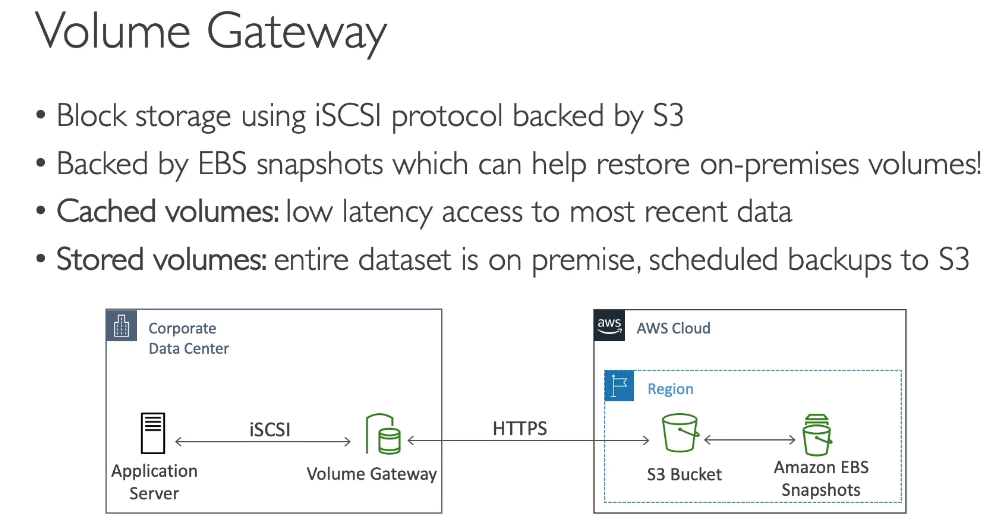
* Reduces storage costs - Typical savings average 50-60% for general-purpose file shares
* Optimizes capacity - More data can be stored in the same physical space
* Maintains data integrity - Original files remain accessible and unchanged to users

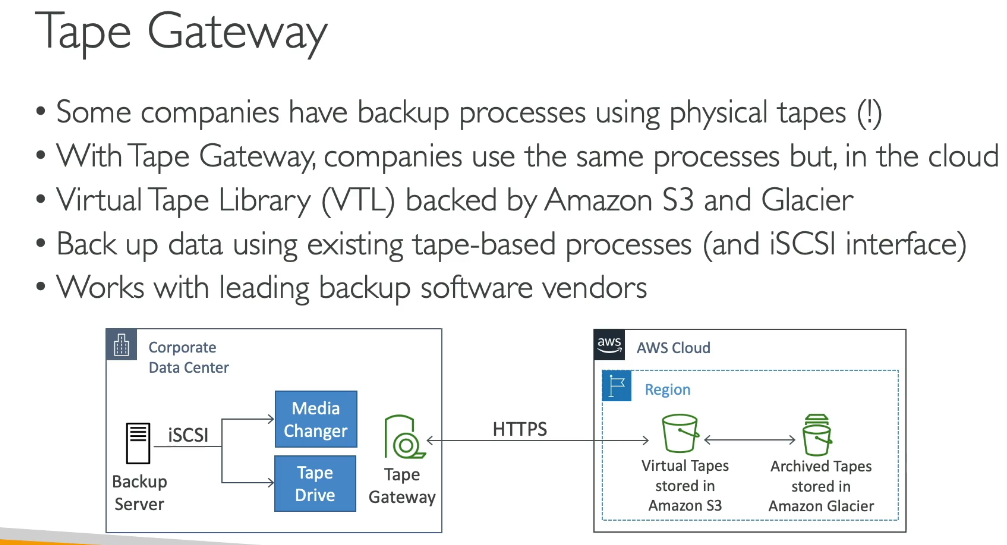


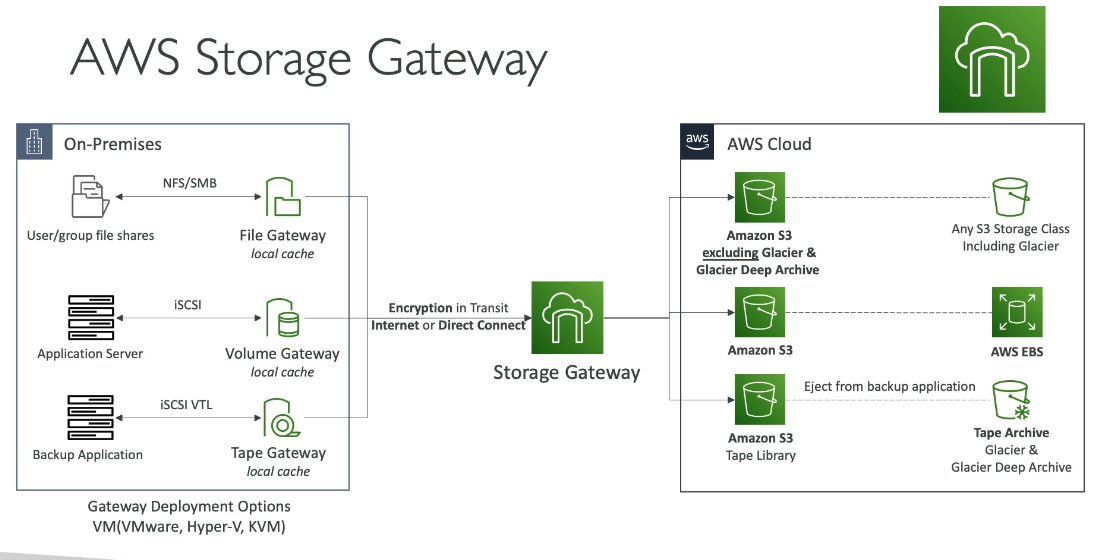
**OpenZFS does not support De-Duplication**

**Storage Gateway:**  










**AWS Storage Gateway is a hybrid cloud storage service with three main use cases:**

1. Moving backups to the cloud:

* Back up on-premises workloads to AWS
* Move tape-based archives to the cloud
* Store long-term retention data in the cloud
* Reduce data center infrastructure and storage administration costs

2. Cloud-backed file shares for on-premises applications:

* Provide easy-to-use, cost-effective, and scalable file storage
* Enable on-premises applications to access cloud storage without modification
* Present virtually unlimited storage to applications while paying only for what you consume

3. Low-latency access to cloud data:

* Cache frequently accessed data locally for minimal latency
* Provide on-premises applications with fast access to data stored in AWS
* Optimize data transfers by compressing data and fetching only what's needed

How it works: Storage Gateway connects your on-premises applications to AWS cloud storage using standard protocols (NFS, SMB, iSCSI) without requiring application changes. It caches frequently used data locally while storing all data durably in AWS, giving you the best of both worlds - local performance with cloud scalability.

**AWS Storage Gateway offers three types because different applications use different storage protocols and have distinct use cases:**

**1. File Gateway (NFS/SMB protocols):**

* Use case: File sharing and collaboration
* Example: A media company stores video files in S3 but editors need to access them via standard file shares. File Gateway presents S3 as an NFS/SMB share, allowing editing applications to work seamlessly without modification.
* Best for: Unstructured data, content repositories, home directories

**2. Volume Gateway (iSCSI block storage):**

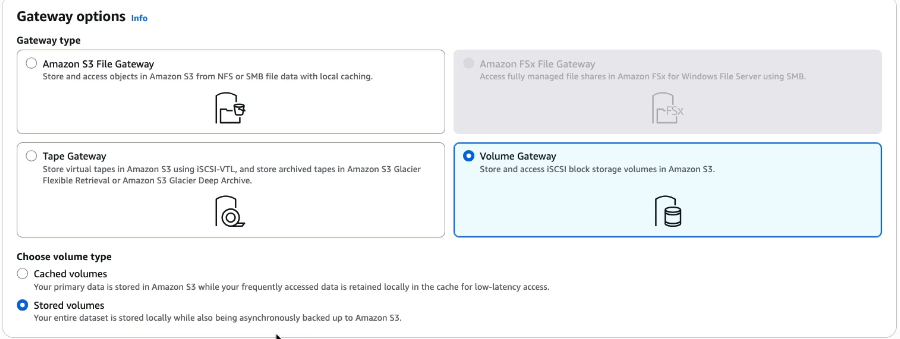
* Use case: Block-based applications and databases
* Example: A company runs an on-premises SQL Server database that needs cloud backup. Volume Gateway presents cloud-backed iSCSI volumes to the database server, with point-in-time snapshots stored as EBS snapshots in AWS.
* Best for: Databases, enterprise applications requiring block storage, disaster recovery

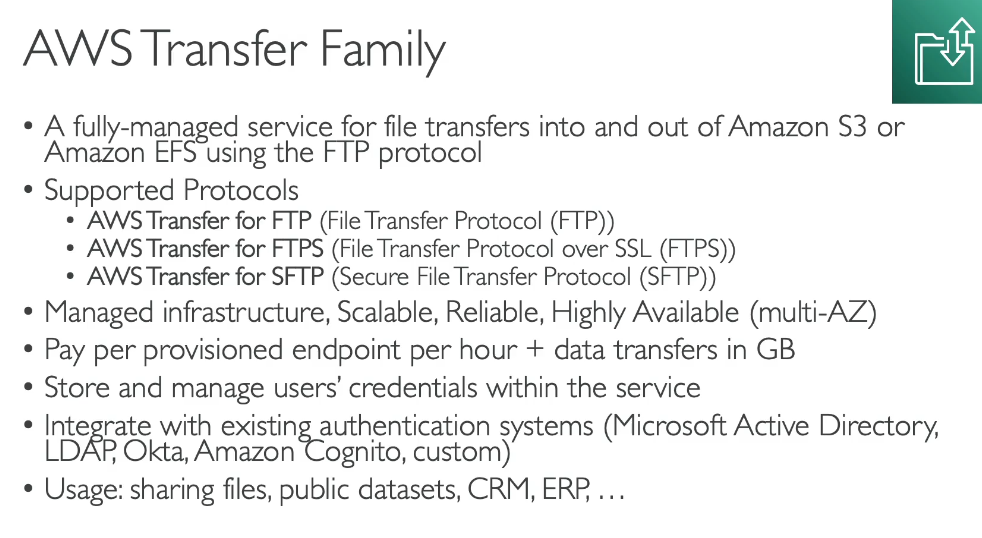
**3. Tape Gateway (Virtual Tape Library):**

* Use case: Backup and archival with existing tape-based workflows
* Example: An organization uses backup software like Veeam or Veritas that writes to tape libraries. Tape Gateway emulates a physical tape library, allowing the same backup software to write to virtual tapes stored in S3 and S3 Glacier.
* Best for: Long-term archival, replacing physical tape infrastructure, compliance retention

**Online migration with Tape Gateway:**

* Deploy Tape Gateway in your data center (as VM or hardware appliance)
* Read data from physical tapes using your tape drives
* Write data to virtual tapes through Tape Gateway
* Data is automatically uploaded to S3 over your internet connection
* Best for: Smaller volumes, when you have sufficient bandwidth



AS2 (Applicability Statement 2) - messaging protocol for exchanging business-to-business data

**AWS Transfer Family** is ideal for secure file exchange with external partners and third parties without changing their existing workflows.

**Real-world use case - Healthcare Provider:**

A hospital network needs to receive patient records, lab results, and medical imaging from hundreds of external clinics, laboratories, and insurance companies. Each partner uses different SFTP clients and has established file transfer processes.

**Without Transfer Family:**

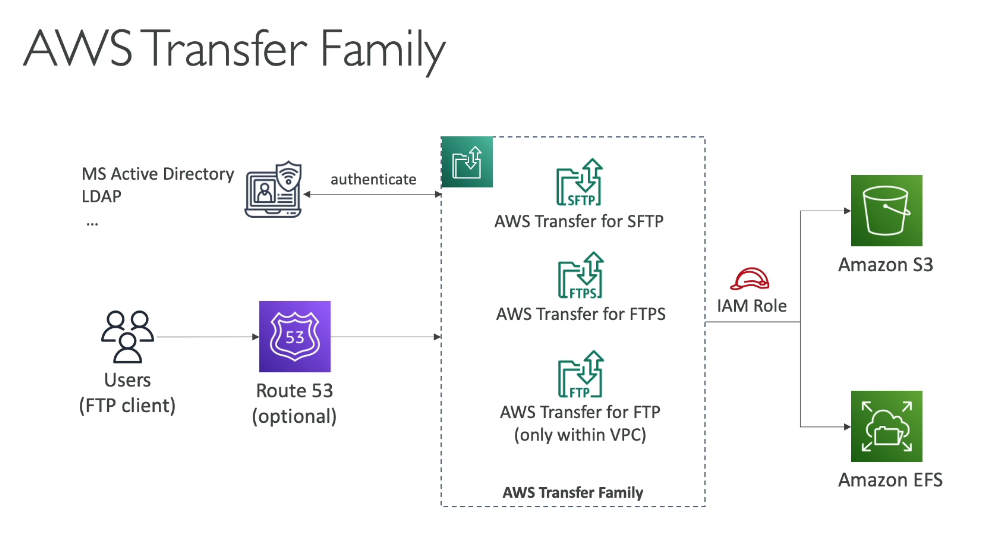
* Hospital must maintain on-premises SFTP servers
* Manage server patches, security updates, and scaling
* Handle user authentication and access controls manually
* Ensure high availability and disaster recovery
* Significant IT overhead and infrastructure costs

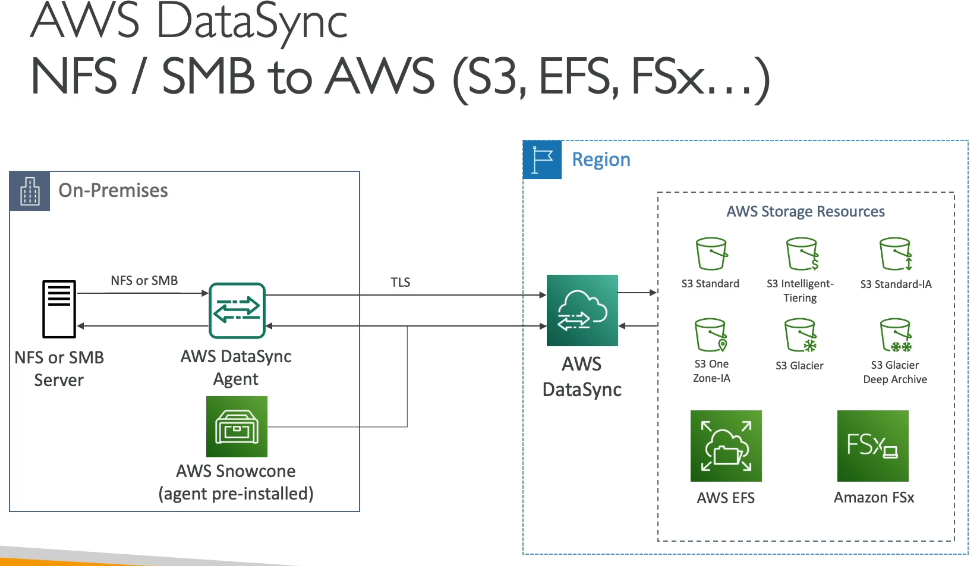
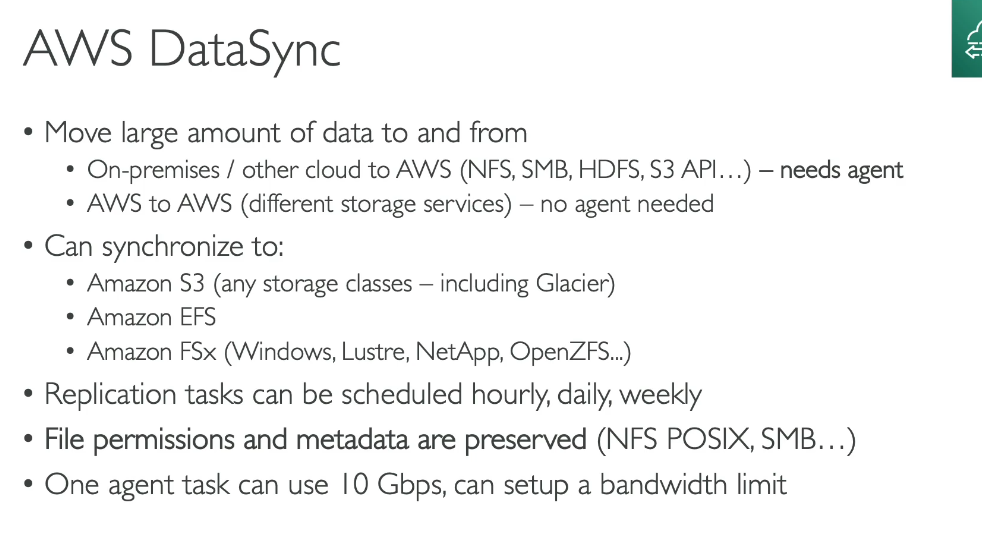
**With Transfer Family:**

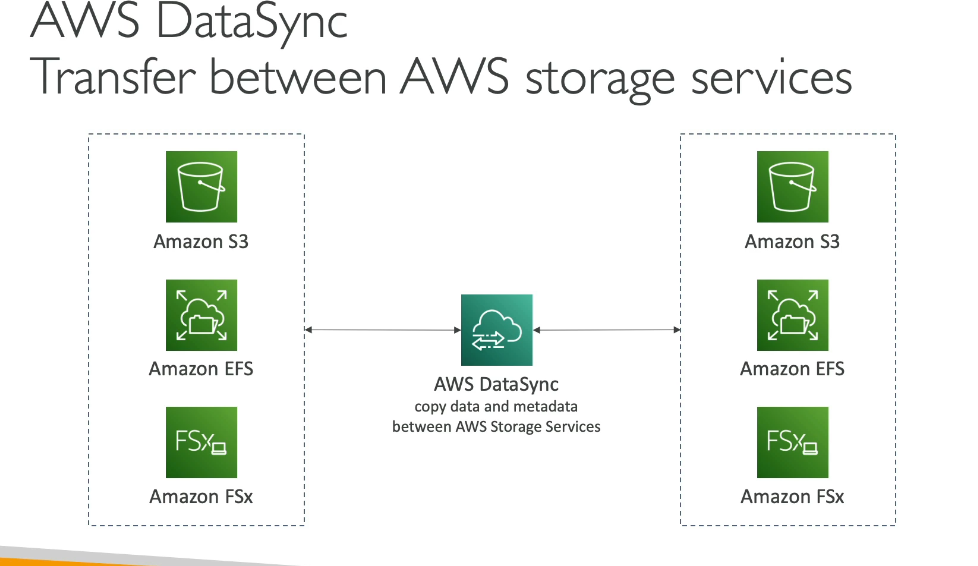
* Hospital creates a Transfer Family server with SFTP enabled
* Each partner gets credentials and connects using their existing SFTP client
* Files automatically land in designated S3 buckets organized by partner
* Integrates with AWS Step Functions to trigger automated workflows:
  + Validate file formats
  + Extract and process data
  + Update electronic health records
  + Send confirmation notifications
* Partners experience no change - same SFTP process they've always used
* Hospital eliminates server management and gains automatic scaling

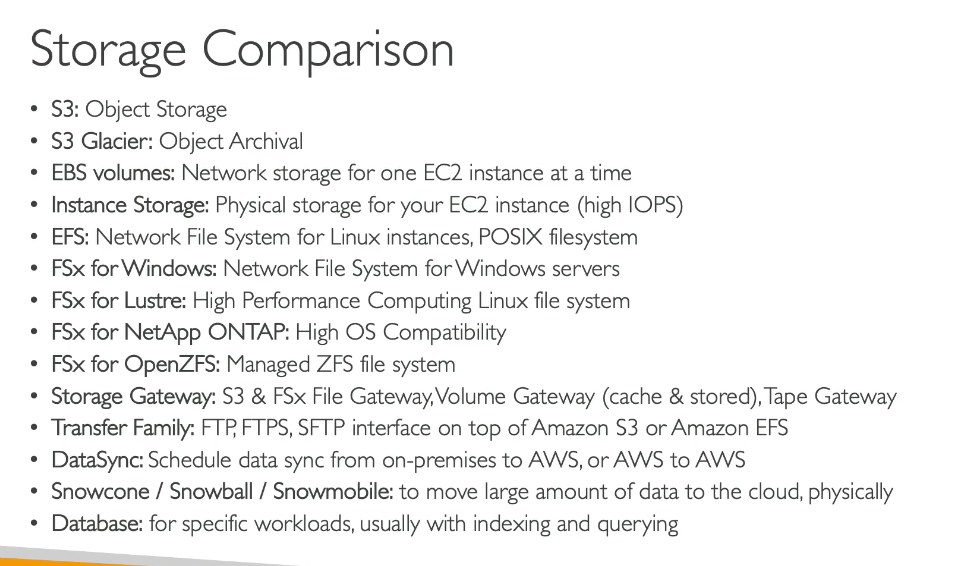
**Key benefits:**

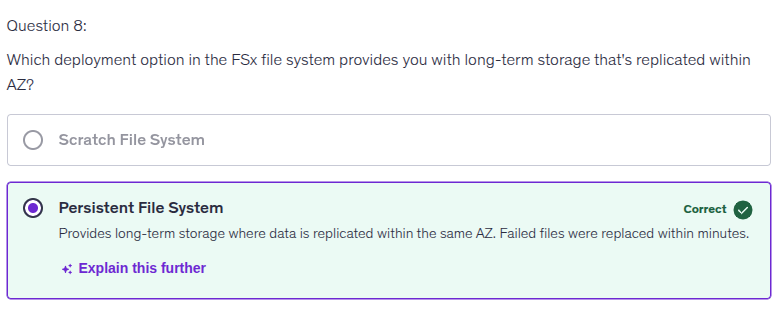
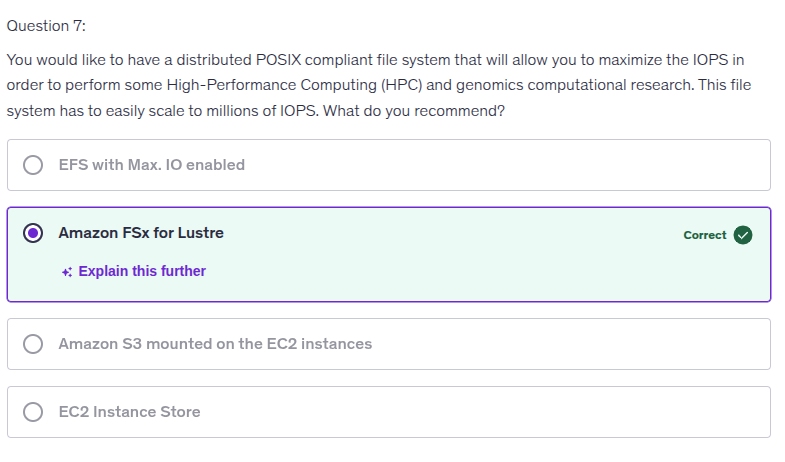
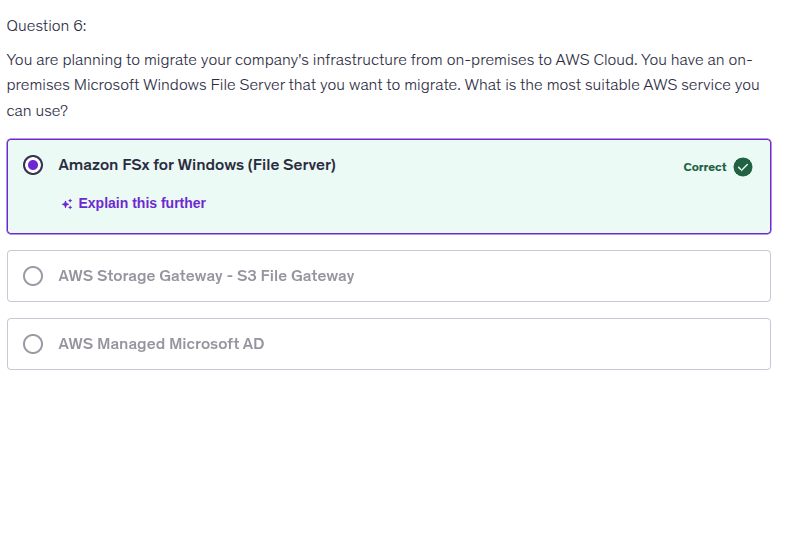
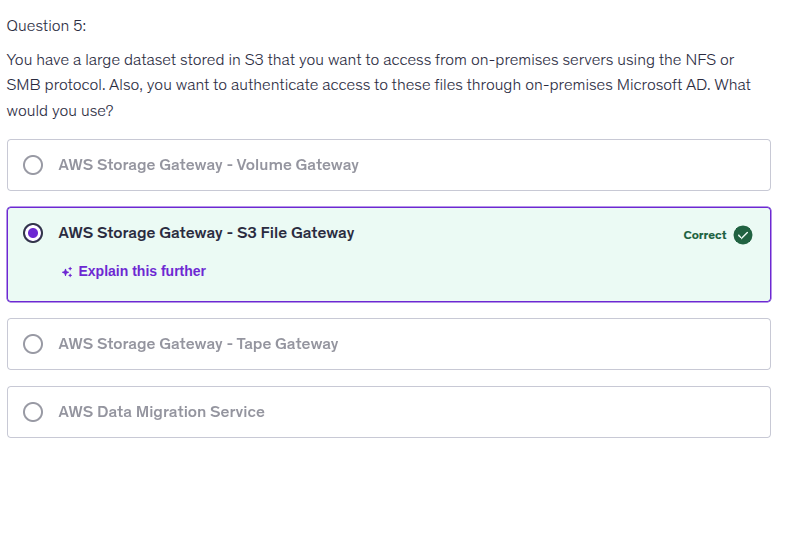
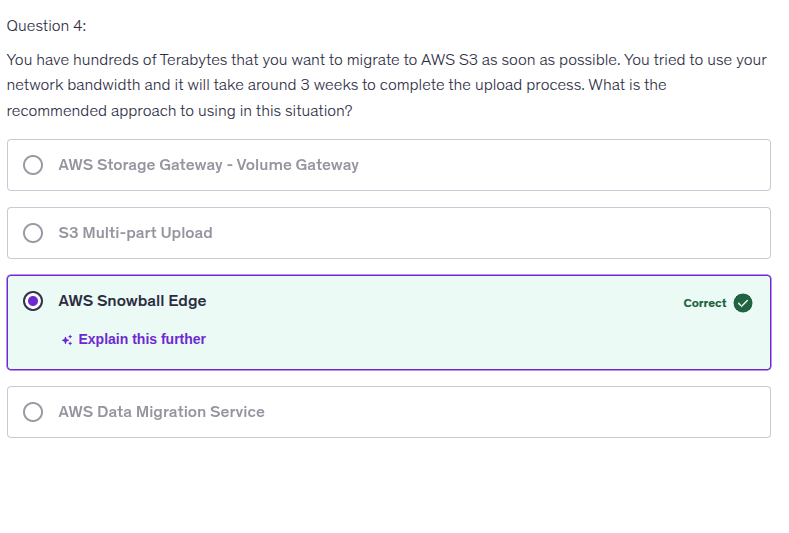
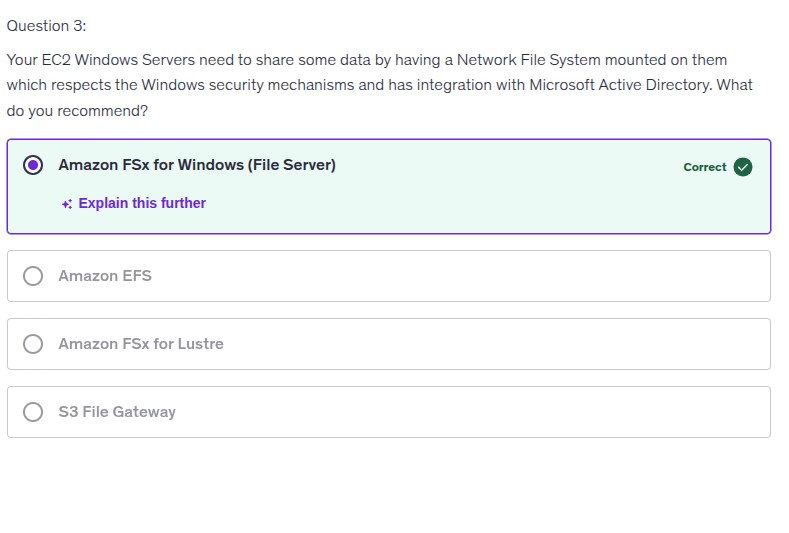
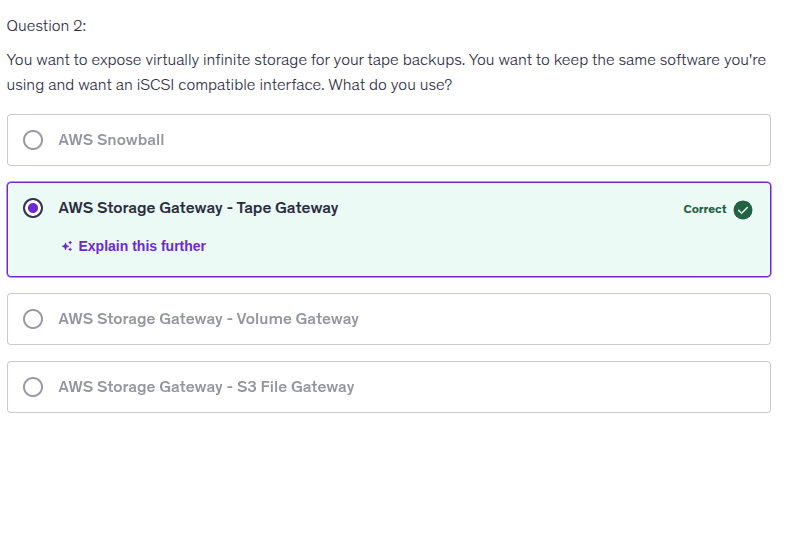
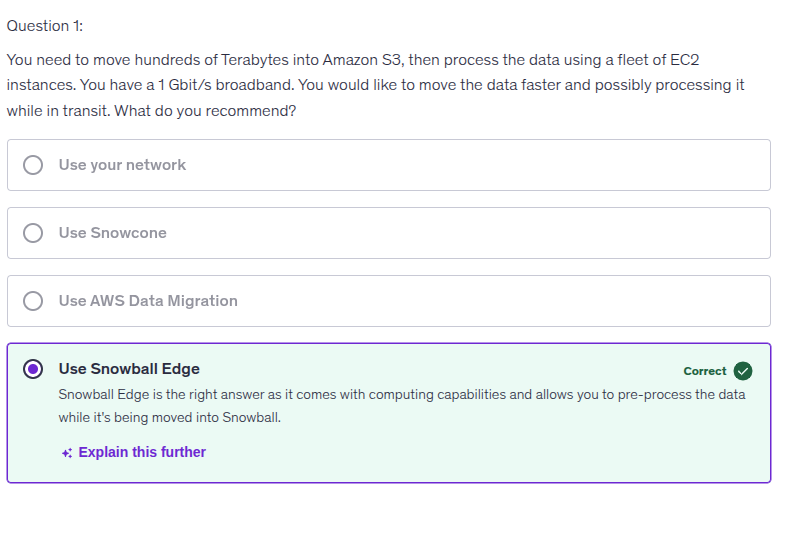
* Partners don't need to change their systems or learn new tools
* Fully managed - no infrastructure to maintain
* Scales automatically during peak periods
* Secure, compliant file transfers with audit logging
* Seamless integration with AWS services for automated processing

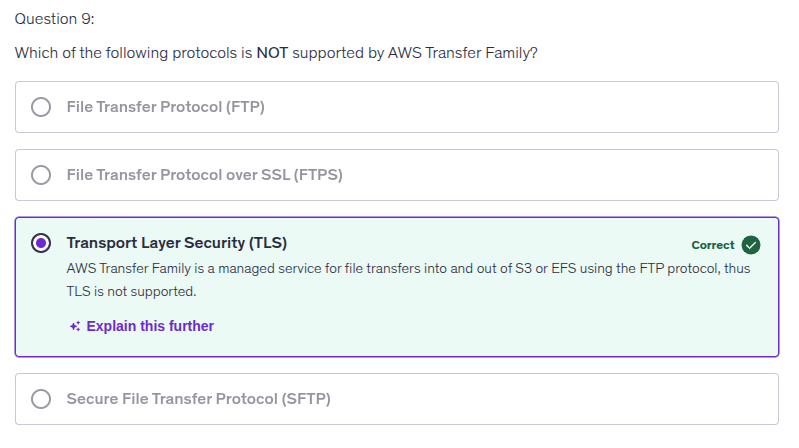


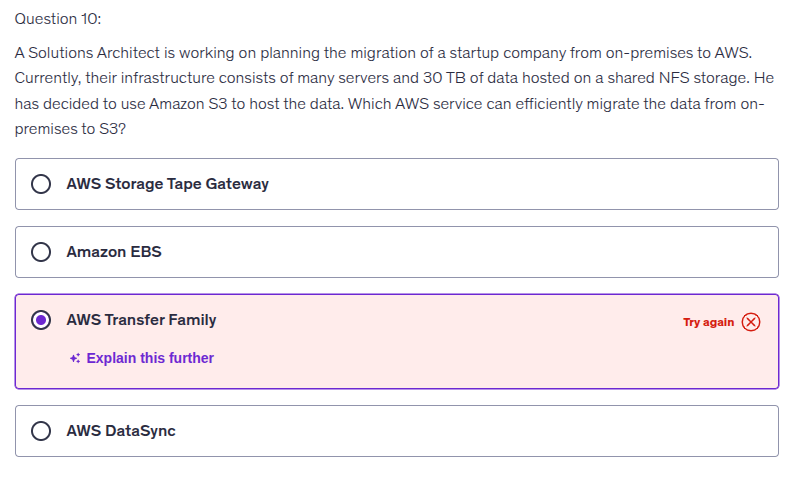










**Why NOT AWS Transfer Family**

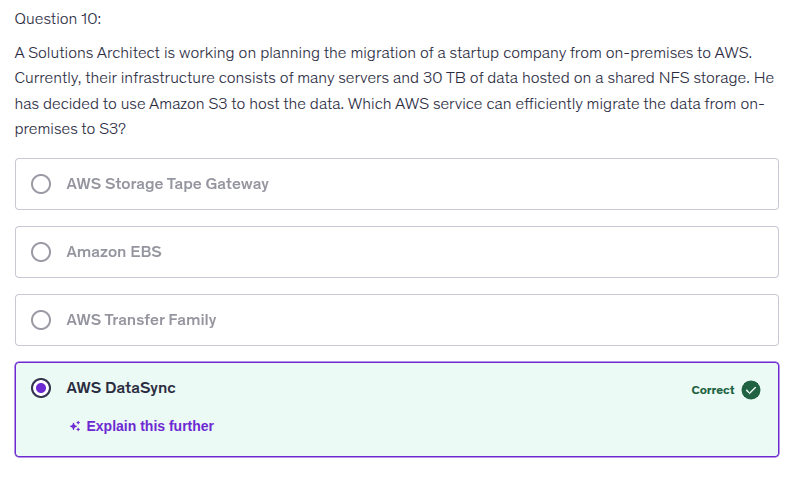
AWS Transfer Family is **just a managed SFTP/FTPS/FTP endpoint**. That’s it.

It’s designed for:

* External partners uploading/downloading files
* Light, user-driven transfers
* Protocol-based access (SFTP, FTPS, FTP)

It is **NOT** designed for:

* Large-scale bulk migration (30 TB is not “small”)
* High-throughput parallel transfers
* NFS-based shared storage
* Automated, continuous syncs

AWS DataSync is **built specifically for this exact scenario**.

### **What DataSync does right:**

* ✅ **Native NFS support**
* ✅ **Optimized, parallel, high-throughput transfers**
* ✅ **Automatic retries, checksums, and integrity validation**
* ✅ **Incremental syncs** (only moves changed data)
* ✅ **Minimal operational effort**
* ✅ **Direct on-prem → S3 pipeline**

