AWS Storage Extras

- AWS Snowball
- AWS FSx
- AWS Storage Gateway
- AWS Transfer Family : La porte d'entrée protocoles historiques
- AWS DataSync : L'accélérateur de migration de données

AWS Snowball

1. Famille AWS Snowball : Le défi des données massives et isolées

Le besoin global

Imagine que tu as 100 To de données sur site (On-Premise) à transférer vers AWS. Avec une très bonne connexion internet dédiée de 100 Mbps, cela prendrait environ **100 jours** en continu, sans compter la saturation du réseau.

- **Le besoin :** Transférer des pétaoctets de données rapidement, de manière sécurisée, sans saturer la bande passante internet.
- **Le contexte :** Sites distants, navires, usines, ou simplement des centres de données avec une connectivité réseau limitée ou inexistante.

Les variantes de Snowball

Snowball n'est pas juste une grosse clé USB, c'est une valise ultra-sécurisée et durcie qui se décline en deux versions selon le besoin :

- **Snowball Edge Storage Optimized (Optimisé pour le stockage) :**
  - **Besoin :** Migrer massivement des données (jusqu'à 80 To par appareil).
  - **Contexte :** Tu as des téraoctets de vidéos, de sauvegardes ou de logs à envoyer vers Amazon S3. Tu commandes la valise, tu branches en local, tu copies, et tu la renvoies par transporteur.
- **Snowball Edge Compute Optimized (Optimisé pour le calcul) :**
  - **Besoin :** Traiter de la donnée *avant* ou *pendant* son stockage dans un lieu sans internet.
  - **Contexte :** Une plateforme pétrolière en mer doit analyser des données sismiques en temps réel (via des instances EC2 embarquées ou AWS Lambda) avant de stocker le résultat final. L'appareil offre une puissance de calcul (CPU/GPU) bien supérieure.

---

2. Snowball vers Amazon S3 Glacier

Le besoin & Le contexte

Tu veux archiver des années de sauvegardes physiques (bandes magnétiques) vers la solution la moins chère d'AWS (**Glacier**), mais tu ne peux pas les uploader par internet.

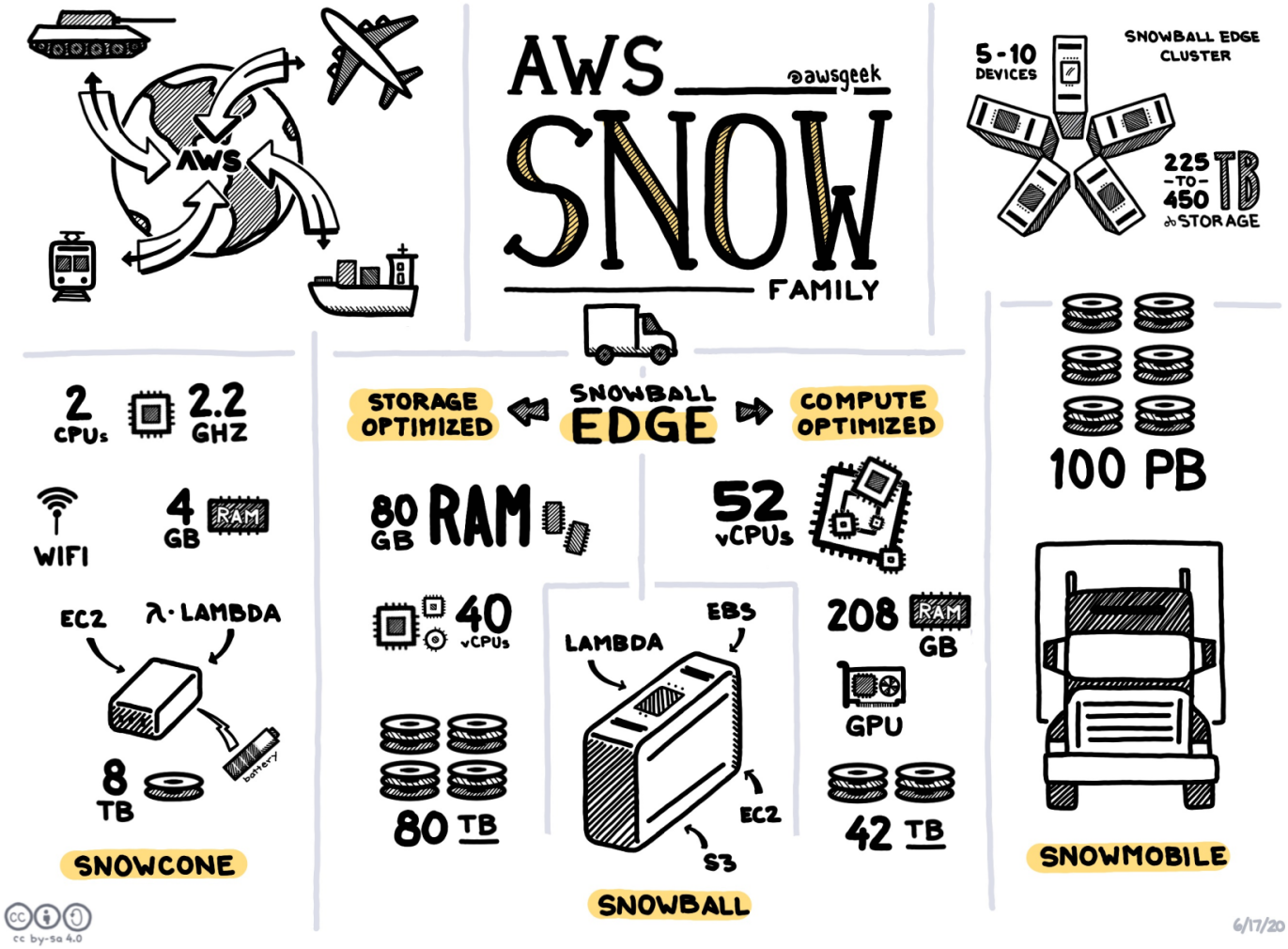
> ⚠ **Point clé pour l'examen :** Tu ne peux **pas** envoyer un Snowball directement dans un coffre-fort Glacier. Le flux physique impose une étape intermédiaire.

Le workflow exact

1. Les données du Snowball sont déchargées par AWS dans un bucket **Amazon S3 Standard** (ou S3 Standard-IA).
2. Tu configures une **S3 Lifecycle Policy (Politique de cycle de vie)** sur ce bucket.
3. Cette politique va automatiquement et immédiatement (par exemple, après 0 jour) migrer (transférer) les données de S3 vers **Amazon S3 Glacier Flexible or Deep Archive**.

CheatSheet

Source : https://www.awsgeek.com/AWS-Snow-Family/



AWS FSx

AWS FSx : Les systèmes de fichiers partagés hautes performances

Le besoin global

Amazon EFS est excellent, mais il ne gère que le protocole POSIX (Linux). Amazon S3 n'est pas un système de fichiers (c'est du stockage objet).

- **Le besoin :** Tu as des applications existantes (legacy) qui ont besoin d'un "vrai" réseau de fichiers partagés (un lecteur réseau `Z:\` sous Windows ou un point de montage spécifique sous Linux) avec des performances extrêmes, sans réécrire le code de l'application.

Est-ce que AWS FSx permet de délocaliser mon système de fichier vers S3 ?

Non ! Quand tu crées un système AWS FSx (que ce soit pour Windows, Lustre, ONTAP ou OpenZFS), AWS provisionne des serveurs de stockage dédiés (généralement des disques SSD ou HDD ultra-rapides) cachés derrière le service.

- **Ce n'est pas du S3.** S3 est un stockage "Objet" (via des clés/valeurs et des requêtes HTTP). FSx est un stockage "Fichiers" traditionnel (avec des dossiers, des sous-dossiers et des protocoles réseau comme SMB ou NFS).
- **Le cas de la VM :** Si tu as une machine virtuelle (EC2), tu détaches ton stockage local ou tu crées simplement un partage réseau, et tu le connectes à FSx. Ta VM écrit directement sur les disques de FSx, pas sur S3.

La confusion vient souvent de **FSx for Lustre**, qui possède une fonctionnalité unique de liaison avec S3 :

- **Le besoin :** Tu as des téraoctets de données brutes stockées à bas coût dans un bucket S3 (par exemple, des images satellites). Tu veux faire un calcul intensif dessus (du Machine Learning).
- **Le fonctionnement :** Tu crées un FSx for Lustre et tu lui dis de se "lier" à ton bucket S3. FSx va alors indexer le bucket S3. Pour tes machines virtuelles, le système FSx semblera contenir tous les fichiers. Dès qu'une VM demandera un fichier, FSx for Lustre ira le chercher de manière transparente dans S3, le copiera sur ses disques SSD ultra-rapides pour que la VM puisse travailler dessus à la vitesse de l'éclair. Une fois le calcul fini, FSx peut renvoyer les résultats dans S3.

A) FSx for Windows File Server

- **Le contexte :** Tu migres une application d'entreprise basée sur Windows (comme un CRM maison, des profils utilisateurs).
- **Points clés SAA-C03 :**
  - **NTFS, SMB, AD :** Utilise le système Windows natif (NTFS), le protocole SMB, et s'intègre obligatoirement avec **Microsoft Active Directory** pour la gestion des permissions.
  - **Montage sur Linux EC2 : Oui, c'est possible !** Bien que conçu pour Windows, Linux gère le protocole SMB. Si l'examen te parle d'un stockage partagé entre instances Windows *et* Linux, FSx for Windows est une option.
  - **Microsoft DFS (Distributed File System) :** Supporté. Permet de regrouper plusieurs serveurs de fichiers en un seul chemin logique et de faire de la réplication.
  - **SSD / HDD :** SSD pour les bases de données et la haute performance (IOPS élevés), HDD pour les partages de fichiers classiques ou l'archivage économique.
  - **Accès depuis l'On-Premise :** Accessible depuis ton centre de données local via un VPN AWS ou AWS Direct Connect.

B) FSx for Lustre

- **Le besoin :** Le High Performance Computing (HPC). Tu as besoin de traiter des volumes gigantesques de données à une vitesse phénoménale (des millions d'IOPS).
- **Le contexte :** Machine Learning (entraînement de modèles), modélisation financière, rendu 3D, analyse génomique.
- **Points clés SAA-C03 :**
  - **Intégration transparente avec S3 :** C'est sa force. Il peut se lier à un bucket S3. FSx for Lustre va "lire" les données de S3 comme s'il s'agissait de fichiers locaux, faire le calcul ultra-rapide, et réécrire le résultat directement dans S3.
  - **On-Premises :** Utilisable depuis le local via Direct Connect pour du calcul hybride.

Quand tu crées du FSx for Lustre, tu dois choisir entre ces deux modes :

Caractéristique	Scratch File System (Stockage temporaire)	Persistent File System (Stockage long terme)
Le Besoin	Coût minimal pour du calcul brut à court terme.	Données hautement disponibles qui doivent durer dans le temps.
Le Contexte	Tu lances un calcul de 4 heures pour du Machine Learning. Si un serveur crash, tu relances le job, ce n'est pas grave.	Tu as un pipeline de données sensible qui tourne en continu sur plusieurs semaines ou mois.
Comportement	**Pas de réplication.** Si le matériel lâche, les données non sauvegardées sur S3 sont **perdues**.	**Réplication automatique** au sein de la même Zone de Disponibilité (AZ). Remplacement automatique en cas de panne.

C) FSx for NetApp ONTAP

- **Le besoin :** Tu utilises déjà la technologie de stockage **NetApp** dans ton propre centre de données et tu veux migrer vers AWS sans changer tes habitudes ni tes scripts d'administration.

- **Le contexte :** Environnements d'entreprise hautement spécialisés et multi-protocoles.
- **Points clés SAA-C03 :**
  - **Multi-protocole :** Compatible simultanément avec NFS (Linux), SMB (Windows) et iSCSI (stockage en bloc pour les VM).
  - **Fonctionnalités avancées :** Supporte la déduplication et la compression (low cost), les **Snapshots** instantanés, et le **Hot Cloning** (cloner un volume de plusieurs To en 2 secondes pour les environnements de dev ou de gaming sans dupliquer l'espace disque).

D) FSx for OpenZFS

- **Le besoin :** Tu migres des serveurs de fichiers basés sur le système de fichiers open-source ZFS vers un service managé.
- **Le contexte :** Souvent utilisé pour remplacer des serveurs de fichiers Linux customisés gourmands en bande passante.
- **Points clés SAA-C03 :**
  - Compatible **NFS** (optimisé pour Linux/Unix).
  - Offre des performances de type "milliseconde" (très faible latence).
  - Permet le **Point-in-time cloning** (pratique pour copier instantanément des données pour tester un patch d'application sans impacter la prod).

Exemples

Scénario 1 : Le Cloud 100% Natif (Partage de fichiers entre VMs)

C'est le cas d'usage le plus classique. Tu as migré tes applications sur AWS, elles tournent sur des instances EC2, et elles ont besoin d'accéder *en même temps* aux mêmes fichiers.

- **Le besoin :** Tu as une application web Windows déployée sur 10 instances EC2 différentes (pour la haute disponibilité). Ces instances doivent partager un dossier unique pour stocker les images de profil des utilisateurs.
- **La solution FSx :** Tu crées un **FSx for Windows File Server**. Les 10 instances EC2 connectent ce lecteur réseau (ex: le lecteur `Z:\`). Quand l'instance n°1 écrit une image, les 9 autres la voient instantanément.
- **Le contexte :** Cloud natif. Pas de serveur On-Premise ici.

Scénario 2 : Le "Lift-and-Shift" (Migration d'applications d'entreprise)

Tu as une application lourde dans ton centre de données local et ton patron te dit : "*On déménage tout dans le cloud ce week-end, mais on n'a pas le temps de réécrire le code*".

- **Le besoin :** Ton application utilise des protocoles spécifiques (comme SMB pour Windows ou des bases de données SQL Server qui ont besoin de stockage partagé, ou des configurations complexes NetApp ONTAP).
- **La solution FSx :** Au lieu de recréer des serveurs de fichiers complexes sur AWS (ce qui demanderait de gérer des VMs, des disques, de la réplication), tu crées un service **FSx** correspondant. Tu migres tes données du local vers FSx (avec DataSync par exemple), et tu branches ton application dessus. L'application ne voit aucune différence, elle pense être toujours sur son stockage d'origine.
- **Le contexte :** Migration de l'On-Premise vers le Cloud.

Scénario 3 : L'Hybride (Extension du On-Premise)

C'est le scénario où tes serveurs restent en local, mais tu utilises la puissance d'AWS.

- **Le besoin :** Tu as une équipe de chercheurs dans ton laboratoire local. Ils génèrent des données mais tes serveurs locaux n'ont pas la puissance de calcul pour les analyser.
- **La solution FSx (Lustre ou ONTAP) :** Tu crées un système FSx sur AWS connecté à ton centre de données via un câble dédié (**AWS Direct Connect**). Tes chercheurs peuvent écrire directement dans FSx à travers le réseau. Ensuite, des dizaines d'instances EC2 sur AWS récupèrent ces fichiers sur FSx, lancent des calculs massifs d'Intelligence Artificielle, et renvoient le résultat.
- **Le contexte :** Hybride (On-Prem + Cloud).

AWS Storage Gateway

Le besoin global

Ton entreprise possède des serveurs locaux (On-Premise) qui manquent d'espace disque, ou tu veux sauvegarder tes données locales dans le cloud de manière transparente, **sans que tes applications locales ne s'en rendent compte**. Tu veux une extension de ton centre de données dans AWS.

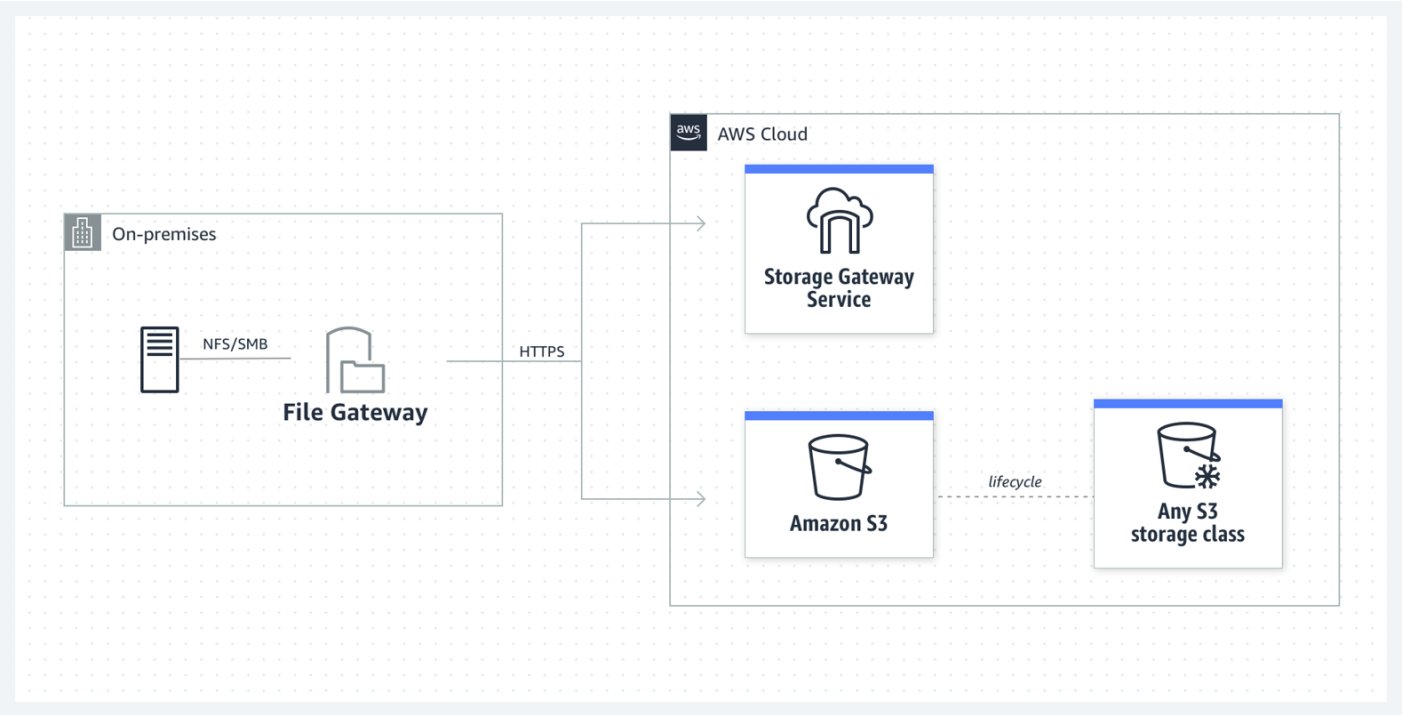
Le contexte architectural

C'est le service phare du **Cloud Hybride**. Pour l'utiliser, **tu as obligatoirement besoin d'installer une VM** (sur VMware ESXi, Microsoft Hyper-V ou KVM) directement *dans ton centre de données local*. C'est cette VM (l'appliance Storage Gateway) qui va faire office de traducteur entre tes protocoles locaux et les API d'AWS.

AWS décline cette passerelle en 3 modes selon le type de stockage (Fichier, Bloc, ou Objet) :

A) S3 File Gateway (Mode Fichier)

- **Le besoin :** Tes applications locales partagent des fichiers via des dossiers réseau classiques, et tu veux que ces fichiers soient stockés directement dans **Amazon S3** pour un coût réduit et une durabilité maximale.
- **Point clé SAA-C03 :** Il y a une correspondance 1:1 entre un fichier local et un objet dans S3. La VM locale garde un **cache local** des fichiers les plus récents pour que l'accès soit ultra-rapide (faible latence).



B) Volume Gateway (Mode Bloc - iSCSI)

- **Le besoin :** Tes serveurs locaux ont besoin de disques durs virtuels (blocs) supplémentaires ou de sauvegarder leurs disques (EBS-like) dans le cloud.
- **Le fonctionnement :** La VM locale présente des volumes via le protocole **iSCSI**. Tes serveurs locaux les voient comme des disques durs physiques locaux.
- **Deux sous-modes à connaître :**
  - **Stored Volumes :** Toutes les données sont gardées en local, et des sauvegardes (snapshots) sont envoyées asynchrones vers S3 (idéal pour la reprise après sinistre / *disaster recovery* avec une latence locale de 0).
  - **Cached Volumes :** Seules les données récentes sont en local (cache). Le reste (100%) est sur AWS S3. Économique si tu as un énorme volume de données mais peu d'espace local.

C) Tape Gateway (Mode Sauvegarde - VTL)

- **Le besoin :** Tu as une vieille infrastructure de sauvegarde qui écrit sur des **bandes magnétiques physiques** (scripts existants, logiciels comme NetBackup, Veeam, etc.) et tu veux t'en débarrasser.
- **Le fonctionnement :** La VM émule une **VTL (Virtual Tape Library)** via iSCSI. Ton logiciel de sauvegarde croit qu'il écrit sur des vraies bandes, mais en réalité, la Gateway envoie les sauvegardes vers **S3 Glacier** ou **Glacier Deep Archive** pour un coût dérisoire.

AWS Transfer Family : La porte d'entrée protocoles historiques

---

Introduction

Le besoin

Tu as des clients, des partenaires ou des applications externes qui doivent t'envoyer des fichiers (ex: des rapports financiers tous les soirs), mais ils refusent d'installer l'AWS CLI ou d'utiliser des clés d'API AWS. Ils veulent utiliser des protocoles standards et sécurisés de transfert de fichiers.

Le contexte (SAA-C03)

- **Protocoles supportés :** FTP, FTPS (FTP sécurisé via SSL), et **SFTP** (FTP sécurisé via SSH).
- **Où vont les données ?** AWS Transfer Family est un point d'accès managé (Serverless). Dès qu'un utilisateur dépose un fichier en SFTP, le fichier atterrit directement et de manière transparente dans **Amazon S3** ou **Amazon EFS**.
- **Astuce examen :** Si la question mentionne "*Migrer un endpoint SFTP existant sans changer les habitudes des clients externes*", la réponse est obligatoirement **AWS Transfer Family**.

Exemple

? Le Scénario : L'intégration des factures fournisseurs

Tu es architecte cloud pour une grande chaîne de supermarchés. Tous les soirs, **150 fournisseurs différents** (qui ont des systèmes informatiques anciens) doivent t'envoyer leur catalogue de prix et leurs factures sous forme de fichiers CSV ou XML.

- **Leur contrainte :** Ils ne connaissent pas AWS, n'ont pas le droit d'installer d'outils AWS, et veulent un bête serveur **SFTP** avec un identifiant et un mot de passe pour y déposer leurs fichiers, comme ils le font depuis 20 ans.
- **Ton objectif :** Tu veux que ces fichiers atterrissent directement dans un bucket **Amazon S3** pour qu'une fonction AWS Lambda se déclenche automatiquement, analyse les factures et mette à jour ta base de données.

?? Comment ça se passe en pratique (Étape par Étape)

Voici comment tu configures et comment fonctionne AWS Transfer Family pour répondre à ce besoin :

Étape 1 : La création du point d'accès (Endpoint)

Dans la console AWS, tu lances le service **AWS Transfer Family** et tu crées un serveur en quelques clics :

1. Tu sélectionnes le protocole : **SFTP**.
2. Tu choisis l'hébergement : **Service managed** (AWS gère l'infrastructure, la haute disponibilité et le scaling à ta place, c'est du *Serverless*).
3. Tu lui associes une adresse publique (ex: `sftp.mon-supermarche.com`).

Étape 2 : La gestion des utilisateurs (Fournisseurs)

Tu crées un accès pour ton premier fournisseur, par exemple "Fournisseur_Alpha" :

- Tu lui génères un identifiant/mot de passe (ou tu lui demandes sa clé publique SSH).
- **Le mapping magique :** Tu configures le service pour lui dire : "*Quand Fournisseur_Alpha se connecte, son dossier racine* `/` *correspond en réalité au bucket S3* `s3://mon-bucket-factures/alpha/`".

Étape 3 : Le flux de données en direct

Le soir venu, le script automatisé du fournisseur s'exécute :

1. Le serveur du fournisseur se connecte via son client SFTP standard (comme FileZilla ou un script WinSCP) à `sftp.mon-supermarche.com`.
2. Il transfère le fichier `facture_04_06_2026.csv`.
3. **Côté AWS :** Le service Transfer Family reçoit le flux SFTP "à la volée", traduit le protocole en requêtes HTTPS sécurisées pour S3, et écrit le fichier directement dans ton bucket.
4. Le fournisseur reçoit un message "Transfert réussi" de la part de son client SFTP.

> **Le gros avantage :** Pour le fournisseur, il s'est connecté à un serveur de fichiers Linux classique. En réalité, **il n'y a aucun serveur Linux à gérer pour toi**, pas de disque dur à vider, pas de patch de sécurité à installer. Tout va directement dans le stockage illimité de S3.

AWS DataSync : L'accélérateur de migration de données

Introduction

Le besoin

Tu as des millions de fichiers (des téraoctets ou pétaoctets) en local et tu veux les **migrer massivement et rapidement** vers AWS, ou mettre en place une synchronisation régulière (ex: toutes les nuits).

Le contexte & Fonctionnement

DataSync est un outil de **transfert de données à grande vitesse**, optimisé pour saturer proprement ta bande passante réseau et aller 10 fois plus vite que des outils classiques (comme rsync ou un simple script de copie).

- **On-Prem vers AWS :** Tu as **besoin d'installer un Agent** (sous forme de VM) dans ton centre de données local pour lire les fichiers et les compresser/chiffrer avant l'envoi.
- **AWS vers AWS :** Si tu synchronises des données entre deux services AWS (ex: d'un compte AWS à un autre, ou d'une région à une autre), **aucun agent n'est requis**, AWS gère tout en interne.
- **Vers quels services AWS peut-il synchroniser ?**
  - Amazon S3
  - Amazon EFS
  - **AWS FSx** (toutes les versions : Windows, Lustre, ONTAP, OpenZFS).
- **Préservation des métadonnées :** C'est capital pour l'examen. DataSync préserve les permissions des fichiers, les dates de modification, les ownerships (UID/GID pour Linux, ACL pour Windows).

Exemple

? Le Scénario : La sauvegarde et migration d'un studio d'animation 3D

Tu es l'architecte cloud d'un studio de cinéma d'animation. Vos graphistes travaillent en local sur des serveurs de stockage ultra-rapides (des NAS de marque NetApp ou des serveurs de fichiers Linux Windows). Chaque nuit, les artistes génèrent **20 To de nouvelles images et vidéos haute**

définition (des millions de petits fichiers de rendu).

- **Le besoin :** Tu veux répliquer automatiquement ces 20 To de données chaque nuit vers AWS (dans un système **AWS FSx for Windows** ou **Amazon S3**) pour que l'équipe de production à l'autre bout du monde puisse travailler dessus le lendemain.
- **Le défi :** Un simple script de copie (`rsync` ou `robocopy`) mettrait trop de temps, saturerait la ligne internet, et perdrait les droits d'accès (permissions Windows ACL / Linux) des fichiers.

?? Comment ça se passe en pratique (Étape par Étape)

Voici comment tu mets en place **AWS DataSync** pour automatiser cela :

Étape 1 : Installer l'Agent en local

Dans ton centre de données, tu télécharges l'**Agent DataSync** fourni par AWS sous forme de machine virtuelle (VM) et tu l'installes sur tes serveurs (VMware ou Hyper-V).

- Cet agent va s'occuper de "scanner" ton stockage local très rapidement, de compresser les données et de les chiffrer avant l'envoi.

Étape 2 : Configurer la "Tâche" (Task) dans AWS

Dans la console AWS, tu crées une **Task DataSync** :

1. **Source :** Tu indiques le chemin de ton NAS local (ex: via le protocole NFS ou SMB).
2. **Destination :** Tu choisis ta cible AWS (ex: un système de fichiers **AWS FSx for NetApp ONTAP** ou un bucket **S3**).
3. **Fréquence :** Tu planifies l'exécution (ex: tous les soirs à 1h du matin).

Étape 3 : La synchronisation intelligente en action

À 1h du matin, la tâche démarre :

1. L'agent local analyse les fichiers. Contrairement à un outil classique, DataSync utilise un protocole réseau propriétaire d'AWS ultra-optimisé qui **parallélise le transfert**. Il s'adapte pour saturer au maximum (mais proprement) ta bande passante.
2. **Transfert incrémental :** DataSync compare la source et la destination. Si un fichier vidéo de 5 Go n'a pas changé, il ne le renvoie pas. Il n'envoie que les nouveautés ou les modifications.
3. **Préservation des métadonnées :** Si le fichier appartenait à l'utilisateur "Graphiste_Jean" avec des droits de lecture spécifiques en local, DataSync réapplique exactement les mêmes droits une fois le fichier arrivé dans AWS FSx.
4. **Vérification de l'intégrité :** À la fin du transfert, DataSync calcule une clé de vérification (checksum) pour chaque fichier pour s'assurer qu'aucun octet n'a été corrompu pendant le voyage sur internet.

? Ce qu'il faut retenir pour l'examen SAA-C03 (La différence clé)

On confond souvent **Storage Gateway** et **DataSync**. Voici le moyen infaillible de ne plus les mélanger :

- **AWS Storage Gateway est un PONT (Temps réel) :** Tes applications locales écrivent dedans *au fil de l'eau* tout au long de la journée. C'est une extension de ton stockage qui garde un cache local.
- **AWS DataSync est un CAMION DE DÉMÉNAGEMENT (Planifié / Batch) :** Il est là pour **déplacer ou synchroniser** de gros volumes de fichiers existants d'un point A à un point B, le plus vite possible, de manière planifiée (ex: une fois par jour, ou pour une migration définitive).