

01. S3 Overview

1. L'origine et Le besoin

Lancé en **2006**, Amazon S3 est l'un des tout premiers services proposés par AWS. L'idée d'Amazon était de créer un système de stockage "infini", ultra-disponible, et accessible de n'importe où via de simples requêtes HTTP (via une API Web).

Avant le stockage objet, on gérait des serveurs de fichiers traditionnels (NAS/SAN) ou des disques durs (stockage de blocs comme AWS EBS). Les problèmes ? C'est difficile à scaler, cher, et limité en capacité. S3 répond au besoin de **stocker des volumes massifs de données non structurées** (images, vidéos, fichiers de logs, sauvegardes, fichiers statiques de sites web) de manière :

- **Hautelement disponible et durable** (conçu pour une durabilité de 99,999999999 % - les fameux "11 neuf").
- **Scalable horizontalement** : Tu n'as jamais besoin de prévoir la taille du disque. S3 grossit à l'infini avec tes données.
- **Économique** : Tu ne payes que ce que tu consommes réellement.

2. Différence entre Objets et Buckets

S3 est un service de **stockage d'objets** (et non un système de fichiers classique).

- **Bucket (Le conteneur)** : C'est l'équivalent d'un dossier racine ou d'un "lecteur". Tout objet dans S3 doit résider dans un bucket. C'est au niveau du bucket que l'on configure la région AWS, la sécurité (chiffrement, politiques d'accès), et les options comme le versioning.
- **Objet (La donnée)** : C'est le fichier que tu déposes. Contrairement à un fichier sur un disque dur, un objet est une entité autonome qui contient :
 - La donnée elle-même (la séquence de bits du fichier).
 - Des métadonnées (taille, type, date de création, etc.).
 - Un identifiant unique (la clé).

3. Le Fullpath : Préfixe vs Nom d'objet

Dans S3, **les dossiers n'existent pas physiquement**. C'est une architecture totalement plate (flat). L'illusion des dossiers est créée par la structure du nom de la clé (le *Key Name*).

Prenons l'URL (ou fullpath) suivante : `s3://mon-bucket-examen/images/2026/photo.jpg`

- **Bucket** : `mon-bucket-examen`

- **Key (Nom complet de l'objet) :** `images/2026/photo.jpg`
- **Prefix (Préfixe) :** `images/2026/`

La console AWS va utiliser les barres obliques (/) comme délimiteurs pour t'afficher une interface avec des dossiers cliquables, mais pour S3, il s'agit simplement d'une chaîne de caractères unique.

4. Le Naming : Global vs Account-Regional Namespace

Là encore, bravo ! C'est une nouveauté majeure introduite par AWS début 2026. Tu as désormais le choix entre deux modèles d'espace de nommage (*namespaces*) :

A. Le modèle Historique : Shared / Global Namespace

C'est le mode par défaut que tout le monde connaît. Le nom du bucket doit être **unique au monde**, tous comptes AWS confondus. Si quelqu'un a pris `mon-bucket-data`, tu ne peux pas l'utiliser.

B. La Nouveauté 2026 : Account-Regional Namespace

Pour éviter la "guerre" des noms de buckets et simplifier l'automatisation (Infrastructure as Code), AWS permet maintenant de créer des buckets dans un espace de nommage propre à ton compte et à ta région.

- **Le principe :** S3 ajoute automatiquement un suffixe obligatoire à la fin de ton nom de bucket comprenant ton **ID de compte AWS** et la **Région** (ex: `mon-bucket-123456789012-us-east-1-an`).
- **L'avantage :** Grâce à ce suffixe géré par AWS, tu as la garantie que le nom est disponible et aucun autre compte AWS externe ne pourra te "voler" ce nom ou utiliser ce suffixe. Tu peux ainsi réutiliser la même structure de nom de manière prédictible d'un environnement à l'autre (Dev, Staging, Prod).

5. Taille maximale d'un objet (Max object size)

- La taille totale d'un bucket est **illimitée**.
- La taille maximale d'un **seul objet** dans S3 est de **50 To (Terabytes)**.
- La taille maximale pour un upload classique en une seule requête HTTP (Single PUT) est de **5 Go (Gigabytes)**.

6. Multi-part Upload (Téléchargement en plusieurs parties)

Si tu veux uploader un fichier qui dépasse 5 Go, tu **dois** utiliser le Multi-part Upload. Cependant, AWS recommande fortement de l'utiliser pour tous les fichiers dès qu'ils dépassent **100 Mo**.

Comment ça marche ?

Le fichier est découpé en plusieurs morceaux (*parts*) qui sont envoyés en parallèle vers S3. Une fois toutes les parties reçues, S3 les réassemble pour recréer l'objet final.

Les avantages pour l'examen :

1. **Meilleure bande passante** : Les morceaux sont envoyés en parallèle.
2. **Résilience aux pannes réseau** : Si l'upload du morceau 4 échoue sur un fichier de 50 Go, tu as juste à renvoyer le morceau 4, pas tout le fichier.
3. **Pause et reprise** : Tu peux mettre l'upload en pause et le reprendre plus tard.

“ ⚠ **Alerte SAA-C03** : Les morceaux non réassemblés (en cas d'upload abandonné) restent stockés dans S3 et te sont facturés ! Pour éviter cela, la bonne pratique est de configurer une **S3 Lifecycle Policy** (politique de cycle de vie) pour supprimer automatiquement les *Incomplete Multipart Uploads* après quelques jours.

7. Les Métadonnées (Metadata)

Chaque objet possède des métadonnées sous forme de paires **Clé/Valeur**. On en distingue deux types :

- **System Metadata (Métadonnées système)** : Gérées par AWS (ex: `Date de création`, `Taille`, `Chiffrement`, `Content-Type` comme `image/jpeg`).
- **User-Defined Metadata (Métadonnées utilisateur)** : Tu peux ajouter tes propres informations lors de l'upload (ex: `statut: confidentiel`, `application: CRM`). Elles doivent obligatoirement commencer par le préfixe `x-amz-meta-`. Elles ne peuvent plus être modifiées après l'upload (il faut réécrire l'objet pour les changer).

8. Les Tags (Étiquettes)

Similaires aux métadonnées, les tags sont des paires Clé/Valeur (jusqu'à 10 par objet), mais ils ont un rôle très différent et dynamique.

Différence cruciale avec les métadonnées :

- Les tags peuvent être modifiés **sans modifier ou réuploader l'objet**.
 - **À quoi ça sert ?**
 1. **Gestion des coûts** : Analyser la facturation (ex: tag `Project: Apollo`).
 2. **Sécurité (ABAC - Attribute-Based Access Control)** : Autoriser un utilisateur à lire un objet uniquement si l'objet possède le tag `Security: Public`.
 3. **Cycle de vie** : Dire à S3 "Supprime tous les objets qui ont le tag `Temporary: True` après 30 jours".
-

9. Le Version ID (L'identifiant de version)

Par défaut, le **Versioning** est désactivé sur un bucket. Une fois activé, il ne peut plus être désactivé, seulement suspendu (*Suspended*).

- **Le principe** : Chaque fois que tu uploades un objet avec le même nom (la même clé), S3 ne l'écrase pas. Il crée une nouvelle version et lui attribue un **Version ID** unique.
- **Protection contre les erreurs** : Si tu supprimes un objet par mégarde, S3 ajoute simplement un **Delete Marker** (un marqueur de suppression) par-dessus. L'objet semble avoir disparu, mais il suffit de supprimer ce marqueur pour récupérer le fichier. Pour supprimer définitivement, il faut spécifier le *Version ID* exact.
- **Facturation** : Tu payes pour le stockage de **toutes** les versions. Si tu as 5 versions d'un fichier de 1 Go, tu payes pour 5 Go.

Revision #3

Created 2026-05-28 09:36:45 UTC by Victor

Updated 2026-05-28 10:02:04 UTC by Victor