

## □ AWS S3 – Simple Storage Service

### 🔍 Introduction

Amazon S3 (Simple Storage Service) est un service de stockage d'objets **hautement disponible, scalable, sécurisé et économique**. Il est souvent considéré comme **la pierre angulaire du succès d'AWS**, notamment grâce à sa **durabilité de 99.9999999999%** (les fameux "11 9s").

---

### ☑ Qu'est-ce qu'un stockage d'objets ?

Contrairement au stockage bloc (EBS) ou fichier (EFS), S3 permet de stocker des **objets** dans des **buckets**. Chaque objet contient :

- **Les données** elles-mêmes (ex. une image, une vidéo, un JSON...),
  - **Une clé unique** (le nom de l'objet),
  - **Les métadonnées** associées.
- 

### 📦 Cas d'utilisation classiques d'AWS S3

- Stockage de **fichiers statiques** pour un site web
  - Sauvegarde et archivage
  - Stockage de données pour le Big Data / Machine Learning
  - Hosting d'applications front-end
  - Intégration CI/CD (stocker des artefacts de build)
  - Streaming ou vidéos embarquées
- 

### 🌐 Architecture simplifiée d'accès

```
plaintext
CopyEdit
Client (navigateur ou app)
  ↓ HTTP/HTTPS
  S3 Bucket (mon-site.com)
    └─ Objet (ex: /images/logo.png)
```

Tous les objets stockés dans S3 sont accessibles **via une URL unique** du type :

```
php-template
CopyEdit
https://<bucket-name>.s3.<region>.amazonaws.com/<object-key>
```

⚠ Mais l'accès **est restreint par défaut** — vous devez configurer les bonnes politiques d'accès.

---

## ☐ Avantages clés de S3

Avantage	Description
🔒 <b>Sécurité</b>	Contrôle d'accès via IAM, Bucket Policy, ACL, chiffrement (SSE-S3, SSE-KMS...)
📈 <b>Scalabilité</b>	Gère des <b>milliers de requêtes par seconde</b> sans config manuelle
💰 <b>Coût optimisé</b>	Plusieurs classes de stockage (S3 Standard, S3 IA, Glacier) selon l'usage
☐ <b>Haute intégration</b>	Compatible avec la majorité des services AWS
⚙️ <b>Fiabilité &amp; durabilité</b>	Les fameux <b>11 9s</b> de durabilité
📁 <b>Versioning</b>	Garde l'historique de tous les objets
🌐 <b>Globalement disponible</b>	Peut être accédé dans le monde entier avec faible latence

---

## 📁 Exemple de noms de buckets professionnels

Nom standardisé pour les environnements :

```
lua
CopyEdit
app1-payments-production-example.com
app2-chat-staging-company.io
```

---

## 🔒 Exemple de politique : Bloquer tout le monde sauf le propriétaire

```
json
CopyEdit
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BlockAllExceptOwner",
      "Effect": "Deny",
```

```

    "Principal": "*",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::your-bucket-name/*",
    "Condition": {
      "StringNotEquals": {
        "aws:userid": "BUCKET_OWNER_CANONICAL_ID"
      }
    }
  }
]
}

```

➡ Remplace `your-bucket-name` par le nom réel, et `BUCKET_OWNER_CANONICAL_ID` par ton ID utilisateur AWS (disponible dans la console S3 dans les détails du propriétaire).

---

## Héberger un site web statique sur AWS S3

### Étapes :

1. Créer un bucket avec un nom **exactement égal au nom de domaine** (ex: `www.monsite.com`)
2. Activer **l'hébergement statique**
3. Uploader les fichiers HTML/CSS/JS
4. Rendre le bucket **public** via une politique

### Exemple de politique pour site statique :

```

json
CopyEdit
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::www.monsite.com/*"
    }
  ]
}

```

⚠ Attention : ceci rend les objets publics — n'utilisez cette configuration que pour des fichiers statiques publics (ex: site web).

---

## Classes de stockage (à choisir selon l'usage)

Classe	Utilisation	Coût 💰
S3 Standard	Accès fréquent	\$ \$
S3 Intelligent-Tiering	Accès variable	\$
S3 Standard-IA (Infrequent Access)	Accès rare mais rapide	\$
S3 Glacier	Archivage long terme	📦 minime
S3 One Zone-IA	Pas de redondance multi-AZ	▽ plus risqué mais moins cher

---

## 🔑 Limites importantes

- ☒ Taille max d'un objet : **5 TB**
  - ☒ Upload possible en multipart pour fichiers > 100MB
  - ☒ Nombre illimité d'objets
  - ☒ Pas de structure hiérarchique réelle (mais vous pouvez utiliser des **prefixes logiques**)
- 

## ❏ Exemples concrets en CLI

### Créer un bucket :

```
bash
CopyEdit
aws s3 mb s3://my-bucket-name
```

### Uploader un fichier :

```
bash
CopyEdit
aws s3 cp index.html s3://my-bucket-name/index.html
```

### Synchroniser un dossier :

```
bash
CopyEdit
aws s3 sync ./dist s3://my-bucket-name
```

### Voir les objets :

```
bash
CopyEdit
aws s3 ls s3://my-bucket-name
```

---

## Sécurité recommandée pour DevOps

- Utilisez **SSE-KMS** pour chiffrer les objets
  - Activez **versioning** pour récupérer des suppressions accidentelles
  - Mettez des **lifecycle policies** pour déplacer vers Glacier
  - Limitez les accès via **IAM Roles**
  - Activez **MFA Delete** pour plus de protection
- 



## Conclusion

Amazon S3 est simple à apprendre mais puissant et critique dans tout environnement Cloud. Il peut servir aussi bien de backend statique pour votre front-end, de CDN, de backup, de stockage d'artefacts CI/CD ou de base pour l'archivage.

---



## Bonus : Pour aller plus loin

- [Amazon S3 Official Docs](#)
- [Best Practices for S3](#)
- [S3 + CloudFront static hosting](#)