



Livre blanc AWS

# Schémas de conception des bonnes pratiques : optimisation des performances Amazon S3



---

# Schémas de conception des bonnes pratiques : optimisation des performances Amazon S3: Livre blanc AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et l'habillage commerciaux d'Amazon ne peuvent pas être utilisés en connexion avec un produit ou un service qui n'est pas celui d'Amazon, d'une manière susceptible de causer de la confusion chez les clients ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon sont la propriété de leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

# Table of Contents

Résumé .....	1
Résumé .....	1
Introduction .....	2
Instructions sur les performances pour Amazon S3 .....	4
Performances des mesures .....	4
Mettre à l'échelle horizontalement les connexions de stockage .....	4
Utiliser les extractions de plages d'octets .....	5
Nouvelle tentative de demandes pour les applications sensibles à la latence .....	5
Combiner Amazon S3 (stockage) et Amazon EC2 (calcul) dans la même région AWS .....	5
Utiliser Amazon S3 Transfer Acceleration pour réduire la latence provoquée par la distance .....	6
Choisissez la version la plus récente des kits AWS SDK. ....	6
Modèles de conception des performances pour Amazon S3 .....	7
Utilisation de la mise en cache pour le contenu fréquemment consulté .....	7
Délais d'expiration et nouvelles tentatives pour les applications sensibles à la latence .....	8
Mise à l'échelle horizontale et mise en parallèle des demandes pour le haut débit .....	9
Utilisation d'Amazon S3 Transfer Acceleration pour accélérer les transferts de données disparates géographiquement .....	11
Participants .....	12
Révisions du document .....	13
Mentions légales .....	14

---

# Schémas de conception des bonnes pratiques : optimisation des performances Amazon S3

Date de publication initiale : Juin 2019 ([Révisions du document](#))

## Résumé

Lors du développement d'applications qui chargent et extraient les données stockées dans Amazon S3, suivez les instructions sur les bonnes pratiques d'AWS afin d'optimiser les performances. AWS propose également des [modèles de conception des performances](#) plus détaillés.

# Introduction

Vos applications peuvent facilement exécuter des milliers de transactions par seconde lors du chargement et de l'extraction du stockage depuis Amazon S3. Amazon S3 se met automatiquement à l'échelle en fonction des taux de demandes très importants. Par exemple, votre application peut atteindre au moins 3 500 demandes PUT/COPY/POST/DELETE et 5 500 demandes GET/HEAD par seconde par préfixe dans un compartiment. Il n'existe aucune limite au nombre de préfixes dans un compartiment. Vous pouvez augmenter vos performances de lecture et d'écriture en effectuant une mise en parallèle des lectures. Par exemple, si vous créez 10 préfixes dans un compartiment Amazon S3 pour paralléliser les lectures, vous pouvez adapter vos performances de lecture à 55 000 demandes de lecture par seconde.

Certaines applications de lacs de données sur Amazon S3 analysent plusieurs milliards d'objets pour les requêtes qui portent sur des pétaoctets de données. Ces applications de lacs de données atteignent des taux de transfert à instance unique qui optimisent l'utilisation de l'interface réseau pour leur instance [Amazon EC2](#), laquelle peut atteindre jusqu'à 100 Gbits/s sur une seule instance. Ces applications regroupent ensuite le débit de plusieurs instances pour parvenir à plusieurs téraoctets par seconde.

D'autres applications sont sensibles à la latence, comme les applications de messagerie des réseaux sociaux. Ces applications peuvent atteindre des latences régulières pour les petits objets (et des latences FBO pour les objets plus grands) d'approximativement 100 à 200 millisecondes.

Les autres services AWS peuvent aussi contribuer à accélérer les performances pour les différentes architectures d'application. Par exemple, si vous souhaitez des taux de transfert plus élevés sur une connexion HTTP unique ou des latences de millisecondes à un chiffre, utilisez [Amazon CloudFront](#) ou [Amazon ElastiCache](#) pour la mise en cache avec Amazon S3.

De plus, afin d'obtenir un transport rapide de données sur de longues distances entre un client et un compartiment S3, utilisez [Amazon S3 Transfer Acceleration](#). Transfer Acceleration utilise les emplacements périphériques distribués à l'échelle mondiale dans CloudFront pour accélérer le transport des données sur des distances géographiques.

Si votre charge de travail Amazon S3 utilise le chiffrement côté serveur avec AWS Key Management Service (SSE-KMS), consultez [Restrictions AWS KMS](#) dans le Guide du développeur AWS Key Management Service pour en savoir plus sur les débits de demandes pris en charge selon le cas d'utilisation.

---

Les rubriques suivantes décrivent les instructions et les modèles de conception des bonnes pratiques pour optimiser les performances des applications qui utilisent Amazon S3.

Ces instructions l'emportent sur toute directive antérieure relative à l'optimisation des performances pour Amazon S3. Par exemple, les directives en matière de performances Amazon S3 antérieures recommandaient de randomiser la dénomination des préfixes avec des caractères hachés afin d'optimiser les performances pour les extractions de données fréquentes. Vous n'avez plus besoin de procéder ainsi pour améliorer les performances et vous pouvez utiliser la dénomination séquentielle basée sur la date pour vos préfixes. Reportez-vous aux Instructions sur les performances et aux Modèles de conception des performances pour consulter les informations les plus récentes sur l'optimisation des performances pour Amazon S3.

# Instructions sur les performances pour Amazon S3

Afin d'optimiser les performances pour votre application sur Amazon S3, AWS recommande ce qui suit.

## Rubriques

- [Performances des mesures](#)
- [Mettre à l'échelle horizontalement les connexions de stockage](#)
- [Utiliser les extractions de plages d'octets](#)
- [Nouvelle tentative de demandes pour les applications sensibles à la latence](#)
- [Combiner Amazon S3 \(stockage\) et Amazon EC2 \(calcul\) dans la même région AWS](#)
- [Utiliser Amazon S3 Transfer Acceleration pour réduire la latence provoquée par la distance](#)
- [Choisissez la version la plus récente des kits AWS SDK.](#)

## Performances des mesures

Lors de l'optimisation des performances, examinez les exigences en matière de débit du réseau, de processeur et de mémoire vive dynamique (DRAM). Selon la combinaison des demandes se rapportant à ces différentes ressources, il peut valoir la peine d'évaluer différents types d'instances [Amazon EC2](#). Pour en savoir plus sur les types d'instances disponibles, consultez [Types d'instances](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Il peut aussi être utile d'examiner le temps de recherche DNS, la latence et la vitesse de transfert des données à l'aide des outils d'analyse HTTP lors de la mesure des performances.

## Mettre à l'échelle horizontalement les connexions de stockage

La répartition des demandes entre plusieurs connexions est un modèle de conception courant pour mettre horizontalement à l'échelle les performances. Lorsque vous développez des applications hautes performances, pensez à Amazon S3 comme à un très grand système réparti, et non comme un simple point de terminaison réseau, tel qu'un serveur de stockage traditionnel. Vous pouvez atteindre les meilleures performances en adressant plusieurs demandes concurrentes à Amazon S3. Répartissez ces demandes sur des connexions distinctes pour maximiser la bande passante accessible à partir d'Amazon S3. Amazon S3 n'a aucune limite pour le nombre de connexions à votre compartiment.

## Utiliser les extractions de plages d'octets

Avec l'en-tête HTTP de plage dans une demande [GET Object](#), vous pouvez extraire une plage d'octets d'un objet, en transférant seulement la portion spécifiée. Vous pouvez utiliser des connexions simultanées vers Amazon S3 pour extraire différentes plages d'octets depuis le même objet.

Vous pouvez ainsi parvenir au regroupement de débits le plus élevé, par opposition à une seule demande d'objet entier. L'extraction des plages les plus petites d'un grand objet permet aussi à votre application d'améliorer l'intervalle des nouvelles tentatives quand les demandes sont interrompues.

Pour en savoir plus, consultez [Obtention d'objets](#).

Les tailles traditionnelles des demandes de plages d'octets sont de 8 Mo ou 16 Mo. Si les objets sont l'objet d'une opération PUT à l'aide d'un chargement en plusieurs parties, une bonne pratique consiste à les soumettre à une opération GET dans les mêmes tailles d'élément (ou au moins alignées sur les frontières d'élément) pour obtenir de meilleures performances. Les demandes GET peuvent directement s'adresser aux éléments individuels ; par exemple, GET ?partNumber=N.

## Nouvelle tentative de demandes pour les applications sensibles à la latence

Les expirations et les nouvelles tentatives agressives permettent d'obtenir une latence cohérente. Étant donnée la large échelle d'Amazon S3, si la première demande est lente, une demande de nouvelle tentative est susceptible d'emprunter un chemin différent et de réussir rapidement. Les kits AWS SDK ont des valeurs configurables de délai d'expiration et de nouvelles tentatives que vous pouvez adapter aux tolérances de votre application.

## Combiner Amazon S3 (stockage) et Amazon EC2 (calcul) dans la même région AWS

Même si les noms de compartiment S3 sont [globalement uniques](#), chaque compartiment est stocké dans une région que vous sélectionnez lorsque vous créez le compartiment. Pour optimiser les performances, nous vous recommandons d'accéder au compartiment depuis les instances Amazon EC2 de la même région AWS chaque fois que possible. Cela permet de réduire la latence réseau et les coûts de transfert des données.

Pour de plus amples informations sur la tarification du transfert des données, veuillez consulter la [Tarification Amazon S3](#).

# Utiliser Amazon S3 Transfer Acceleration pour réduire la latence provoquée par la distance

[Amazon S3 Transfer Acceleration](#) permet un transfert rapide, facile et sécurisé des fichiers sur de longues distances entre votre client et un compartiment S3. Transfer Acceleration tire profit des emplacements périphériques répartis à travers le monde dans [Amazon CloudFront](#). Lorsque les données arrivent dans un emplacement périphérique, elles sont transférées vers Amazon S3 sur un chemin de réseau optimisé. Transfer Acceleration convient parfaitement au transfert régulier de gigaoctets ou téraoctets de données d'un continent à l'autre. Il est aussi utile pour les clients qui chargent sur un compartiment centralisé depuis le monde entier.

Vous pouvez utiliser l'[outil de comparaison de vitesse Amazon S3 Transfer Acceleration](#) pour comparer les vitesses de chargement accéléré et non accéléré dans les régions Amazon S3. L'outil de comparaison de la vitesse utilise les chargements partitionnés pour transférer un fichier à partir de votre navigateur vers différentes régions Amazon S3 avec ou sans Amazon S3 Transfer Acceleration.

## Choisissez la version la plus récente des kits AWS SDK.

Les kits SDK AWS fournissent un support intégré pour la plupart des recommandations en vue de l'optimisation des performances Amazon S3. Les kits SDK fournissent une API plus simple pour tirer parti d'Amazon S3 depuis une application. Ils sont régulièrement mis à jour pour suivre les bonnes pratiques les plus récentes. Par exemple, les kits SDK incluent la logique permettant automatiquement une demande de nouvelle tentative sur les erreurs HTTP 503 et examinent le code pour répondre aux connexions lentes et s'y adapter.

Les kits SDK fournissent aussi [Transfer Manager](#), qui automatise la mise à l'échelle horizontale des connexions pour traiter des milliers de demandes par seconde, à l'aide de demandes de plages d'octets si approprié. Il importe d'utiliser la version la plus récente des kits AWS SDKs pour obtenir les fonctionnalités d'optimisation des performances les plus récentes.

Vous pouvez aussi optimiser les performances lorsque vous utilisez les demandes d'API REST HTTP. Lors de l'utilisation de l'API REST, vous devez suivre les mêmes bonnes pratiques qui font partie des kits SDK. Autorisez les délais d'expiration et les nouvelles tentatives sur les demandes lentes, et multipliez les connexions pour autoriser l'extraction de données d'objet en parallèle. Pour de plus amples informations sur l'utilisation de l'API REST, veuillez consulter la [Référence d'API Amazon Simple Storage Service](#).

# Modèles de conception des performances pour Amazon S3

Lors de la conception d'applications pour charger et extraire les données de stockage dans Amazon S3, utilisez nos modèles de conception des bonnes pratiques afin d'optimiser les performances de votre application. Nous vous proposons aussi de prendre en compte les [Instructions sur les performances](#) lors de la planification de l'architecture de votre application.

Pour optimiser les performances, vous pouvez utiliser les modèles de conception suivants.

## Rubriques

- [Utilisation de la mise en cache pour le contenu accédé fréquemment](#)
- [Délais d'expiration et nouvelles tentatives pour les applications sensibles à la latence](#)
- [Mise à l'échelle horizontale et mise en parallèle des demandes pour le haut débit](#)
- [Utilisation d'Amazon S3 Transfer Acceleration pour accélérer les transferts de données disparates géographiquement](#)

## Utilisation de la mise en cache pour le contenu accédé fréquemment

La plupart des applications qui stockent les données dans Amazon S3 traitent un « ensemble opérationnel » de données, demandé à maintes reprises par les utilisateurs. Si une charge de travail envoie des requêtes GET répétées pour un ensemble commun d'objets, vous pouvez utiliser un cache tel qu'[Amazon CloudFront](#), [Amazon ElastiCache](#), ou [AWS Elemental MediaStore](#) pour optimiser les performances. L'adoption réussie du cache peut se traduire par une latence faible et des taux élevés de transfert des données. Les applications qui utilisent la mise en cache envoient aussi moins de demandes directes à Amazon S3, ce qui peut aider à réduire les coûts des demandes.

Amazon CloudFront est un réseau de diffusion de contenu (CDN) rapide qui met en cache de façon transparente les données depuis Amazon S3 dans un large ensemble de points de présence (PoP) géographiquement répartis. Lorsqu'il est possible d'accéder aux objets depuis plusieurs régions, ou sur Internet, CloudFront permet que les données soient mises en cache à côté des utilisateurs qui accèdent aux objets. Il peut en résulter une diffusion haute performance des contenus Amazon S3 réputés. Pour de plus amples informations sur l'utilisation de CloudFront, veuillez consulter le [Manuel du développeur Amazon CloudFront](#).

Amazon ElastiCache est un cache en mémoire géré. Avec ElastiCache, vous pouvez provisionner des instances Amazon EC2 qui mettent en cache des objets en mémoire. Cette mise en cache se traduit par des ordres de réduction de magnitude dans la latence GET et de substantielles augmentations dans le débit de téléchargement. Pour utiliser ElastiCache, vous modifiez la logique de l'application pour remplir le cache avec des objets sensibles et vérifier la présence d'objets sensibles dans le cache avant de les demander auprès de Amazon S3. Pour des exemples d'utilisation d'ElastiCache pour améliorer les performances GET d'Amazon S3, veuillez consulter le billet de blog [Turbocharge Amazon S3 with Amazon ElastiCache for Redis](#).

AWS Elemental MediaStore est un système de mise en cache et de distribution de contenu spécialement conçu pour les flux de travail vidéo et la diffusion multimédia depuis Amazon S3. MediaStore fournit des API de stockage de bout en bout spécifiquement conçues pour la vidéo. Il est en outre recommandé pour les charges de travail vidéo sensibles aux performances. Pour de plus amples informations sur MediaStore, veuillez consulter le [Guide de l'utilisateur AWS Elemental MediaStore](#).

## Délais d'expiration et nouvelles tentatives pour les applications sensibles à la latence

Dans certaines situations, une application reçoit une réponse d'Amazon S3 indiquant qu'une nouvelle tentative est nécessaire. Amazon S3 mappe les noms de compartiment et d'objet aux données d'objet qui leur sont associées. Si une application génère des taux de demande élevés (généralement des taux soutenus de plus de 5 000 demandes par seconde pour un petit nombre d'objets), elle peut recevoir des réponses HTTP 503 slowdown. Si ces erreurs se produisent, chaque kit AWS SDK implémente une logique de nouvelle tentative automatique à l'aide d'une interruption exponentielle. Si vous n'utilisez pas de kit SDK AWS, vous devez implémenter une logique de nouvelle tentative lors de la réception de l'erreur HTTP 503. Pour de plus amples informations, veuillez consulter [Nouvelles tentatives après erreur et backoff exponentiel dans AWS](#) dans la Référence générale sur Amazon Web Services.

Amazon S3 se met automatiquement à l'échelle en réponse aux taux soutenus de nouvelles demandes, optimisant dynamiquement les performances. Tandis qu'Amazon S3 optimise en interne le taux de nouvelles demandes, vous recevez temporairement des réponses HTTP 503 jusqu'à ce que l'optimisation soit terminée. Après qu'Amazon S3 optimise en interne les performances pour le nouveau taux de demandes, toutes les demandes sont généralement traitées sans nouvelles tentatives.

Pour les applications sensibles à la latence, Amazon S3 conseille un suivi et des opérations de nouvelles tentatives plus lentes. Lorsque vous retentez une demande, nous recommandons l'utilisation d'une nouvelle connexion à Amazon S3 et l'exécution d'une nouvelle recherche DNS.

Lorsque vous effectuez des demandes volumineuses de taille variable (par exemple, plus de 128 Mo), nous conseillons de suivre le débit obtenu et de procéder à de nouvelles tentatives sur les 5 % les plus lents des demandes. Lorsque vous exécutez des demandes plus petites (par exemple, inférieures à 512 Ko), où les latences médianes sont souvent dans une plage de dix millisecondes, une bonne instruction consiste à retenter une opération GET ou PUT après 2 secondes. Si de nouvelles tentatives sont nécessaires, la bonne pratique consiste à se retirer. Par exemple, nous recommandons de n'émettre une nouvelle tentative qu'après 2 secondes et une deuxième nouvelle tentative au bout de 4 secondes supplémentaires.

Si votre application adresse des demandes de taille fixe à Amazon S3, vous devez escompter des temps de réponse plus cohérents pour chacune de ces demandes. Dans ce cas, une stratégie simple consiste à identifier le 1 % de demandes les plus lentes et de les retenter. Même une simple nouvelle tentative est fréquemment efficace pour réduire la latence.

Si vous utilisez AWS Key Management Service (AWS KMS) pour le chiffrement côté serveur, consultez la section [Quotas](#) dans le Guide du développeur AWS Key Management Service afin d'obtenir des informations sur les taux de demandes pris en charge par rapport à votre cas d'utilisation.

## Mise à l'échelle horizontale et mise en parallèle des demandes pour le haut débit

Amazon S3 est un système distribué très large. Pour vous aider à tirer parti de son échelle, nous vous encourageons à horizontalement mettre à l'échelle les demandes parallèles adressées aux points de terminaison du service Amazon S3. En plus de la distribution des demandes au sein d'Amazon S3, ce type d'approche de la mise à l'échelle permet de répartir la charge sur plusieurs chemins d'accès du réseau.

Pour les transferts à haut débit, Amazon S3 conseille l'utilisation d'applications employant plusieurs connexions pour exécuter des opérations GET ou PUT sur les données en parallèle. Par exemple, cet aspect est pris en charge par le [Transfer Manager Amazon S3](#) dans le kit AWS Java SDK, et la plupart des autres kits SDK AWS fournissent des constructions similaires. Pour certaines applications, vous pouvez obtenir des connexions parallèles en lançant plusieurs demandes simultanément dans différents threads d'application ou dans différentes instances d'application. La

meilleure approche à adopter dépend de votre application et de la structure des objets auxquels vous accédez.

Vous pouvez utiliser les kits AWS SDK pour émettre des demandes GET et PUT directement plutôt que d'utiliser la gestion des transferts dans le kit AWS SDK. Cette approche permet d'affiner la charge de travail plus directement, tout en tirant profit du support du kit SDK pour les nouvelles tentatives et la gestion des réponses HTTP 503 qui peuvent se produire. En règle générale, lorsque vous téléchargez des objets volumineux au sein d'une région depuis Amazon S3 vers [Amazon EC2](#), nous vous suggérons d'effectuer des demandes simultanées pour les plages d'octets d'un objet avec la granularité de 8 à 16 Mo. Effectuez une seule demande simultanée pour chaque 85-90 Mbits/s du débit réseau souhaité. Pour saturer une carte NIC 10 Gbits/s, vous pouvez utiliser jusqu'à 15 demandes simultanées sur des connexions distinctes. Vous pouvez augmenter les demandes simultanées sur plusieurs connexions pour saturer les NIC les plus rapides, telles que les NIC 25 Gbits/s ou 100 Gbits/s.

La mesure des performances est importante quand vous réglez le nombre de demandes à émettre simultanément. Il est recommandé de commencer avec une seule demande à la fois. Mesurez la bande passante réseau obtenue et l'utilisation des autres ressources que votre application utilise dans le traitement des données. Vous pouvez alors identifier la ressource dans le goulet d'étranglement (c.-à-d., la ressource avec l'utilisation la plus élevée) et de là le nombre de demandes susceptibles d'être utiles. Par exemple, si le traitement d'une demande à la fois conduit à une utilisation de l'UC de 25 %, cela induit que quatre demandes simultanées au plus peuvent être accueillies.

Les mesures sont essentielles et il vaut la peine de confirmer l'utilisation des ressources tandis que le taux des demandes augmente.

Si votre application adresse directement des demandes à Amazon S3 à l'aide de l'API REST, nous recommandons l'utilisation d'un groupe de connexions HTTP et la réutilisation de chaque connexion pour un ensemble de demandes. Le fait d'éviter la configuration des connexions par demande supprime la nécessité d'exécuter des liaisons TCP à démarrage lent et SSL (Secure Sockets Layer) sur chaque demande. Pour en savoir plus sur l'utilisation de l'API REST, consultez [l'Introduction sur les API REST dans Amazon S3](#).

Enfin, il vaut la peine de prêter attention à DNS et de vérifier que les demandes sont réparties sur un vaste groupe d'adresses IP Amazon S3. Les requêtes DNS pour le Amazon S3 parcourent une large liste de points de terminaison IP. Mais la mise en cache des programmes de résolution ou du code d'application qui réutilise une seule adresse IP ne tire pas parti de la diversité des adresses et de l'équilibrage de charge qui en découle. Les outils d'utilitaire réseau comme l'outil de ligne

de commande netstat peuvent afficher les adresses IP utilisées pour la communication avec Amazon S3 et nous fournissons des instructions sur les configurations DNS à utiliser. Pour en savoir plus sur ces instructions, consultez la section [Routage des demandes](#).

## Utilisation d'Amazon S3 Transfer Acceleration pour accélérer les transferts de données disparates géographiquement

[Amazon S3 Transfer Acceleration](#) est efficace pour réduire ou supprimer la latence provoquée par la distance géographique entre les clients dispersés géographiquement et une application régionale qui utilise Amazon S3. Transfer Acceleration utilise les emplacements périphériques répartis mondialement dans CloudFront pour le transport de données. Le réseau périphérique AWS dispose de points de présence dans plus de 50 emplacements. Aujourd'hui, il permet de diffuser le contenu via CloudFront et de fournir des réponses rapides aux requêtes DNS adressées à [Amazon Route 53](#).

Le réseau périphérique aide également à accélérer les transferts de données dans et en dehors d'Amazon S3. Il convient parfaitement aux applications qui transfèrent les données entre les continents, ont une connexion Internet rapide, utilisent des objets de grande taille ou ont un contenu volumineux à charger. Lorsque les données arrivent dans un emplacement périphérique, elles sont transférées vers Amazon S3 sur un chemin de réseau optimisé. En général, plus vous êtes loin d'une région Amazon S3, plus vous pouvez escompter une amélioration de la vitesse grâce à Transfer Acceleration.

Vous pouvez configurer Transfer Acceleration sur les compartiments nouveaux ou existants. Vous pouvez utiliser un point de terminaison Amazon S3 Transfer Acceleration distinct pour les emplacements périphériques AWS. Le meilleur moyen de tester si Transfer Acceleration améliore les performances des demandes des clients consiste à utiliser l'[Outil de comparaison de la vitesse d'Amazon S3 Transfer Acceleration](#). Les configurations et les états du réseau varient de temps à autre et d'emplacement à emplacement. Par conséquent, seuls les transferts où Amazon S3 Transfer Acceleration peut potentiellement améliorer vos performances de chargement vous sont facturés. Pour en savoir plus sur l'utilisation de Transfer Acceleration avec différents kits SDK AWS, consultez [Exemples Amazon S3 Transfer Acceleration](#).

# Participants

Ont contribué à la préparation du présent document :

- Mai-Lan Tomsen Bukovec, vice-présidente, Amazon S3
- Andy Warfield, ingénieur principal senior, Amazon S3
- Tim Harris, ingénieur principal, Amazon S3

## Révisions du document

Pour recevoir les mises à jour de ce livre blanc, abonnez-vous au flux RSS.

update-history-change

update-history-description

update-history-date

[Mis à jour](#)

Exactitude technique vérifiée

10 mars 2021

[Publication initiale](#)

Publication initiale

1 juin 2019

---

## Mentions légales

Les clients sont responsables de leur propre évaluation indépendante des informations contenues dans ce document. Le présent document : (a) est fourni à titre informatif uniquement, (b) représente les offres et pratiques actuelles de produits AWS, qui sont susceptibles d'être modifiées sans préavis, et (c) ne crée aucun engagement ou assurance de la part d'AWS et de ses affiliés, fournisseurs ou concédants de licences. Les produits ou services AWS sont fournis « en l'état » sans garantie, représentation ou condition, de quelque nature que ce soit, explicite ou implicite. Les responsabilités et obligations d'AWS envers ses clients sont déterminées par les contrats AWS, et le présent document ne fait pas partie d'un contrat entre AWS et ses clients, ni le modifie.

© 2020, Amazon Web Services, Inc. ou ses sociétés apparentées. Tous droits réservés.