Comment StorageGRID implémente l'API REST S3

StorageGRID software

NetApp
July 23, 2026

Sommaire

Comment StorageGRID implémente l'API REST S3

Comment l'API REST StorageGRID S3 résout les requêtes client conflictuelles

Les requêtes conflictuelles des clients, telles que deux clients écrivant sur la même clé, sont résolues selon le principe du « dernier arrivé, dernier servi ».

Le moment de l'évaluation « latest-wins » est basé sur le moment où le système StorageGRID termine une requête donnée, et non sur le moment où les clients S3 commencent une opération.

Comment l'API REST StorageGRID S3 équilibre la disponibilité et la cohérence

La cohérence assure un équilibre entre la disponibilité des objets et leur cohérence sur différents nœuds de stockage et sites. Vous pouvez modifier la cohérence selon les besoins de votre application.

Par défaut, StorageGRID garantit la cohérence en lecture après écriture pour les objets nouvellement créés. Toute requête GET suivant une requête PUT réussie pourra lire les données nouvellement écrites. Les écrasements d'objets existants, les mises à jour de métadonnées et les suppressions sont finalement cohérents.

Si vous souhaitez effectuer des opérations sur les objets avec un niveau de cohérence différent, vous pouvez :

- Spécifiez une cohérence pour chaque compartiment.
- Spécifiez une consistance pour chaque opération API.
- Modifiez la cohérence globale par défaut de la grille en effectuant l'une des tâches suivantes :
  - Dans le Gestionnaire de grille, accédez à **Configuration** > **Système** > **Paramètres de stockage** > **Cohérence par défaut des compartiments**.
  - .

> Une modification de la cohérence globale de la grille s'applique uniquement aux compartiments créés après la modification du paramètre. Pour connaître les détails d'une modification, consultez le journal des audits situé à `/var/local/log` (recherchez **consistencyLevel**).

Valeurs de cohérence

La cohérence influe sur la manière dont les métadonnées que StorageGRID utilise pour le suivi des objets sont distribuées entre les nœuds. La cohérence influe sur la disponibilité des objets pour les requêtes client.

Vous pouvez définir la cohérence d'un compartiment ou d'une opération d'API sur l'une des valeurs suivantes :

- **Tous** : Tous les nœuds reçoivent immédiatement les métadonnées de l'objet ou la requête échouera.

- **Forte-globale** : Garantit la cohérence de lecture après écriture pour toutes les requêtes client sur tous les sites. Lorsque la sémantique de quorum est configurée, les comportements suivants s'appliquent :
  - Permet une tolérance aux pannes de site pour les requêtes client lorsque les grilles comportent trois sites ou plus. Les grilles à deux sites n'ont pas de tolérance aux pannes de site.
  - Les opérations S3 suivantes ne pourront pas aboutir si l'un des sites est hors service :
    - DeleteBucketEncryption
    - PutBucketBranch
    - PutBucketEncryption
    - PutBucketVersioning
    - PutObjectLegalHold
    - PutObjectLockConfiguration
    - PutObjectRetention

  Si nécessaire, vous pouvez "Configurer la sémantique de quorum de StorageGRID pour une cohérence globale forte".
- **Strong-site** : les métadonnées des objets sont immédiatement distribuées aux autres nœuds du site. Garantit la cohérence lecture-après-écriture pour toutes les requêtes client au sein du site.
- **Lecture après nouvelle écriture** : Garantit la cohérence en lecture après écriture pour les nouveaux objets et la cohérence éventuelle pour les mises à jour d'objets. Offre une haute disponibilité et des garanties de protection des données. Recommandé dans la plupart des cas.
- **Disponible** : Garantit la cohérence éventuelle pour les nouveaux objets et les mises à jour d'objets. Pour les compartiments S3, utilisez cette fonctionnalité uniquement en cas de besoin (par exemple, pour un compartiment contenant des valeurs de journaux rarement lues, ou pour les opérations HEAD ou GET sur des clés qui n'existent pas). Non pris en charge pour les compartiments S3 FabricPool.

Utilisez la cohérence « Lecture après nouvelle écriture » et « Disponible »

Lorsqu'une opération HEAD ou GET utilise la cohérence « Read-after-new-write », StorageGRID effectue la recherche en plusieurs étapes, comme suit :

- Il commence par rechercher l'objet avec un niveau de cohérence faible.
- Si cette recherche échoue, elle est répétée à la valeur de cohérence suivante jusqu'à atteindre une cohérence équivalente au comportement pour strong-global.

Si une opération HEAD ou GET utilise la cohérence « Lecture après nouvelle écriture » mais que l'objet n'existe pas, la recherche d'objet atteindra toujours un niveau de cohérence équivalent à celui du comportement pour strong-global. Parce que cette cohérence nécessite que plusieurs copies des métadonnées de l'objet soient disponibles sur chaque site, vous pouvez recevoir un nombre élevé d'erreurs 500 Internal Server Error si deux nœuds de stockage ou plus du même site sont indisponibles.

Sauf si vous exigez des garanties de cohérence similaires à celles d'Amazon S3, vous pouvez éviter ces erreurs pour les opérations HEAD et GET en définissant la cohérence sur « Disponible ». Lorsqu'une opération HEAD ou GET utilise la cohérence « Disponible », StorageGRID assure uniquement une cohérence éventuelle. Il ne retente pas une opération ayant échoué en cas de cohérence croissante, il n'est donc pas nécessaire que plusieurs copies des métadonnées de l'objet soient disponibles.

Spécifiez la cohérence pour l'opération API

Pour définir la cohérence d'une opération API individuelle, les valeurs de cohérence doivent être prises en charge pour l'opération, et vous devez spécifier la cohérence dans l'en-tête de la requête. Cet exemple définit la cohérence sur « Strong-site » pour une opération GetObject.

```
GET /bucket/object HTTP/1.1
Date: date
Authorization: authorization name
Host: host
Consistency-Control: strong-site
```

Vous devez utiliser la même cohérence pour les opérations PutObject et GetObject.

Spécifiez la cohérence du compartiment

Pour définir la cohérence d'un compartiment, vous pouvez utiliser la requête "Cohérence PUT Bucket" StorageGRID. Ou vous pouvez "modifier la cohérence d'un bucket" depuis le Tenant Manager.

Lors du réglage de la consistance d'un bucket, tenez compte des points suivants :

- La configuration de la cohérence d'un compartiment détermine le type de cohérence utilisé pour les opérations S3 effectuées sur les objets contenus dans le compartiment ou sur la configuration du compartiment. Elle n'affecte pas les opérations sur le compartiment lui-même.
- La cohérence d'une opération API individuelle prime sur la cohérence du compartiment.
- En règle générale, les buckets doivent utiliser la cohérence par défaut « Lecture après nouvelle écriture ». Si les requêtes ne fonctionnent pas correctement, modifiez le comportement du client applicatif si possible. Ou, configurez le client pour spécifier la cohérence pour chaque requête API. Définissez la cohérence au niveau du bucket uniquement en dernier recours.

Comment les règles de cohérence et les règles ILM interagissent pour affecter la protection des données

Le niveau de cohérence choisi et la règle ILM influent tous deux sur la manière dont les objets sont protégés. Ces paramètres peuvent interagir.

Par exemple, la cohérence utilisée lors du stockage d'un objet influe sur l'emplacement initial de ses métadonnées, tandis que le comportement d'ingestion sélectionné pour la règle ILM influe sur l'emplacement initial des copies de l'objet. Parce que StorageGRID nécessite l'accès à la fois aux métadonnées d'un objet et à ses données pour répondre aux requêtes client, le choix de niveaux de protection identiques pour la cohérence et le comportement d'ingestion peut offrir une meilleure protection des données initiale et des réponses système plus prévisibles.

Les "options d'ingestion"règles ILM suivantes sont disponibles :

Double engagement

StorageGRID crée immédiatement des copies intermédiaires de l'objet et renvoie la confirmation au client. Les copies spécifiées dans la règle ILM sont effectuées lorsque cela est possible.

Strict

Toutes les copies spécifiées dans la règle ILM doivent être effectuées avant que la réussite ne soit renvoyée au client.

Équilibré

StorageGRID tente de créer toutes les copies spécifiées dans la règle ILM lors de l'ingestion ; si cela s'avère impossible, des copies intermédiaires sont créées et un message de succès est renvoyé au client. Les copies spécifiées dans la règle ILM sont créées dès que possible.

Exemple de la façon dont la cohérence et la règle ILM peuvent interagir

Supposons que vous ayez une grille à trois sites avec la règle ILM suivante et la cohérence suivante :

- **Règle ILM** : Créez trois copies de l'objet, une sur le site local et une sur chaque site distant. Utilisez le comportement d'ingestion strict.
- **Cohérence** : Forte et globale (les métadonnées de l'objet sont immédiatement distribuées à plusieurs sites).

Lorsqu'un client stocke un objet dans la grille, StorageGRID effectue trois copies de l'objet et distribue les métadonnées à plusieurs sites avant de renvoyer un message de succès au client.

L'objet est intégralement protégé contre toute perte dès la confirmation de son ingestion. Par exemple, si le site local est perdu peu après l'ingestion, des copies des données de l'objet et des métadonnées de l'objet existent toujours sur les sites distants. L'objet est entièrement récupérable depuis les autres sites.

Si vous utilisiez la même règle ILM et la cohérence forte des sites, le client pourrait recevoir un message de succès après la réplication des données d'objet sur les sites distants, mais avant la distribution des métadonnées d'objet. Dans ce cas, le niveau de protection des métadonnées d'objet ne correspond pas au niveau de protection des données d'objet. Si le site local est perdu peu après l'ingestion, les métadonnées d'objet sont perdues. L'objet ne peut pas être récupéré.

L'interaction entre la cohérence et les règles ILM peut être complexe. Contactez NetApp si vous avez besoin d'assistance.

Comment StorageGRID utilise le versionnage des objets

Vous pouvez configurer le versionnage d'un compartiment si vous souhaitez conserver plusieurs versions de chaque objet. L'activation du versionnage pour un compartiment peut vous aider à vous prémunir contre la suppression accidentelle d'objets et vous permet de récupérer et de restaurer des versions antérieures d'un objet.

Le système StorageGRID implémente le versionnage et prend en charge la plupart des fonctionnalités, avec toutefois certaines limitations. StorageGRID prend en charge jusqu'à 10 000 versions de chaque objet.

Le versionnage des objets peut être combiné avec la gestion du cycle de vie des informations (ILM) de StorageGRID ou avec la configuration du cycle de vie des compartiments S3. Vous devez activer explicitement le versionnage pour chaque compartiment. Lorsque le versionnage est activé pour un compartiment, chaque objet ajouté au compartiment se voit attribuer un ID de version, qui est généré par le système StorageGRID.

L'utilisation de la suppression MFA (authentification multifacteur) n'est pas prise en charge.

Le versionnage ne peut être activé que sur les compartiments créés avec StorageGRID version 10.3 ou ultérieure.

ILM et gestion des versions

Les politiques ILM sont appliquées à chaque version d'un objet. Un processus d'analyse ILM examine en continu tous les objets et les réévalue par rapport à la politique ILM en vigueur. Toute modification apportée aux politiques ILM est appliquée à tous les objets précédemment importés. Cela inclut les versions précédemment importées si le versionnage est activé. L'analyse ILM applique les nouvelles modifications ILM aux objets précédemment importés.

Pour les objets S3 dans des compartiments avec versionnage activé, la prise en charge du versionnage vous permet de créer des règles ILM qui utilisent « Heure non actuelle » comme heure de référence (sélectionnez **Oui** à la question « Appliquer cette règle uniquement aux anciennes versions de l'objet ? » dans "Étape 1 de l'assistant de création d'une règle ILM"). Lorsqu'un objet est mis à jour, ses versions précédentes deviennent non actuelles. L'utilisation d'un filtre « Heure non actuelle » vous permet de créer des stratégies qui réduisent l'impact sur le stockage des versions précédentes des objets.

Lorsque vous chargez une nouvelle version d'un objet à l'aide d'une opération de chargement en plusieurs parties, l'horodatage de non-actualité de la version originale de l'objet correspond au moment où le chargement en plusieurs parties a été créé pour la nouvelle version, et non au moment où le chargement en plusieurs parties a été terminé. Dans certains cas limités, l'horodatage de non-actualité de la version originale peut être antérieur de plusieurs heures, voire de plusieurs jours, à celui de la version actuelle.

Informations connexes

- "${post_edited_translations.segment}"
- "Règles et politiques ILM pour les objets versionnés S3 (Exemple 4)".

Utilisez l'API REST S3 pour configurer StorageGRID S3 Object Lock

Si le paramètre global S3 Object Lock est activé pour votre système StorageGRID, vous pouvez créer des compartiments avec S3 Object Lock activé. Vous pouvez spécifier une rétention par défaut pour chaque compartiment ou des paramètres de rétention pour chaque version d'objet.

${post_edited_translations.segment}

Si le paramètre de verrouillage global des objets S3 est activé pour votre système StorageGRID, vous pouvez éventuellement activer le verrouillage des objets S3 lors de la création de chaque compartiment.

Le verrouillage d'objet S3 est un paramètre permanent qui ne peut être activé que lors de la création d'un compartiment. Vous ne pouvez pas ajouter ni désactiver le verrouillage d'objet S3 après la création d'un compartiment.

${post_edited_translations.segment}

- Créez le compartiment à l'aide du Tenant Manager. Voir "Créer un compartiment S3".

- Créez le compartiment à l'aide d'une requête CreateBucket avec l' `x-amz-bucket-object-lock-enabled` en-tête de requête. Voir "Opérations sur les compartiments".

S3 Object Lock requiert le versionnement du compartiment, qui est activé automatiquement lors de la création du compartiment. Vous ne pouvez pas suspendre le versionnement pour le compartiment. Voir "Versionnage d'objets".

${post_edited_translations.segment}

Lorsque le verrouillage d'objet S3 est activé pour un compartiment, vous pouvez éventuellement activer la rétention par défaut pour le compartiment et spécifier un mode de rétention par défaut et une période de rétention par défaut.

Mode de rétention par défaut

- En mode CONFORMITÉ :
  - L'objet ne peut pas être supprimé tant que sa date de conservation n'est pas atteinte.
  - La date de conservation de l'objet peut être augmentée, mais elle ne peut pas être diminuée.
  - La date de conservation de l'objet ne peut pas être supprimée avant que cette date ne soit atteinte.
- En mode GOUVERNANCE :
  - Les utilisateurs disposant de l' `s3:BypassGovernanceRetention`autorisation peuvent utiliser l' `x-amz-bypass-governance-retention: true`en-tête de requête pour contourner les paramètres de rétention.
  - Ces utilisateurs peuvent supprimer une version d'objet avant que sa date de conservation ne soit atteinte.
  - Ces utilisateurs peuvent augmenter, diminuer ou supprimer la date de conservation d'un objet.

Période de conservation par défaut

${post_edited_translations.segment}

Comment définir la rétention par défaut pour un compartiment

${post_edited_translations.segment}

- Gérez les paramètres du compartiment depuis le Gestionnaire de locataires. Voir "Créez un compartiment S3" et "Mettre à jour la rétention par défaut du verrouillage d'objet S3".
- Envoyez une requête PutObjectLockConfiguration pour le compartiment afin de spécifier le mode par défaut et le nombre de jours ou d'années par défaut.

PutObjectLockConfiguration

La requête PutObjectLockConfiguration vous permet de définir et de modifier le mode de rétention par défaut et la période de rétention par défaut pour un compartiment ayant le verrouillage d'objets S3 activé. Vous pouvez également supprimer les paramètres de rétention par défaut précédemment configurés.

Lors de l'ingestion de nouvelles versions d'objets dans le compartiment, le mode de rétention par défaut est appliqué si `x-amz-object-lock-mode` et `x-amz-object-lock-retain-until-date` ne sont pas spécifiés. La période de rétention par défaut est utilisée pour calculer la date limite de rétention si `x-amz-object-lock-retain-until-date` n'est pas spécifié.

Si la période de conservation par défaut est modifiée après l'ingestion d'une version d'objet, la date de fin de conservation de la version d'objet reste la même et n'est pas recalculée en utilisant la nouvelle période de conservation par défaut.

Vous devez disposer de l' `s3:PutBucketObjectLockConfiguration`autorisation, ou être account root, pour effectuer cette opération.

L' `Content-MD5`en-tête de requête doit être spécifié dans la requête PUT.

Exemple de requête

Cet exemple active le verrouillage d'objet S3 pour un compartiment et définit le mode de rétention par défaut sur COMPLIANCE et la période de rétention par défaut sur 6 ans.

```
PUT /bucket?object-lock HTTP/1.1
Accept-Encoding: identity
Content-Length: 308
Host: host
Content-MD5: request header
User-Agent: s3sign/1.0.0 requests/2.24.0 python/3.8.2
X-Amz-Date: date
X-Amz-Content-SHA256: authorization-string
Authorization: authorization-string

<ObjectLockConfiguration>
    <ObjectLockEnabled>Enabled</ObjectLockEnabled>
    <Rule>
        <DefaultRetention>
            <Mode>COMPLIANCE</Mode>
            <Years>6</Years>
        </DefaultRetention>
    </Rule>
</ObjectLockConfiguration>
```

${post_edited_translations.segment}

Pour déterminer si le verrouillage d'objet S3 est activé pour un compartiment et pour afficher le mode de rétention et la période de rétention par défaut, utilisez l'une de ces méthodes :

- Consultez le compartiment dans le Gestionnaire de locataires. Voir "Afficher les compartiments S3".
- Envoyez une demande GetObjectLockConfiguration.

GetObjectLockConfiguration

La requête GetObjectLockConfiguration vous permet de déterminer si S3 Object Lock est activé pour un compartiment et, s'il est activé, de voir si un mode de rétention et une période de rétention par défaut sont configurés pour le compartiment.

Lorsque de nouvelles versions d'objets sont ingérées dans le compartiment, le mode de rétention par défaut

est appliqué si `x-amz-object-lock-mode` n'est pas spécifié. La période de rétention par défaut est utilisée pour calculer la date limite de conservation si `x-amz-object-lock-retain-until-date` n'est pas spécifié.

Vous devez disposer de l `s3:GetBucketObjectLockConfiguration`autorisation, ou être account root, pour effectuer cette opération.

Exemple de requête

```
GET /bucket?object-lock HTTP/1.1
Host: host
Accept-Encoding: identity
User-Agent: aws-cli/1.18.106 Python/3.8.2 Linux/4.4.0-18362-Microsoft
botocore/1.17.29
x-amz-date: date
x-amz-content-sha256: authorization-string
Authorization: authorization-string
```

Exemple de réponse

```
HTTP/1.1 200 OK
x-amz-id-2:
iVmcB7OXXJRkRH1FiVq1151/T24gRfpwpuZrEG11Bb9ImOMAAe98oxSpXlknabA0LTvBYJpSIX
k=
x-amz-request-id: B34E94CACB2CEF6D
Date: Fri, 04 Sep 2020 22:47:09 GMT
Transfer-Encoding: chunked
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<ObjectLockConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <ObjectLockEnabled>Enabled</ObjectLockEnabled>
    <Rule>
        <DefaultRetention>
            <Mode>COMPLIANCE</Mode>
            <Years>6</Years>
        </DefaultRetention>
    </Rule>
</ObjectLockConfiguration>
```

${post_edited_translations.segment}

Un compartiment avec le verrouillage d'objets S3 activé peut contenir une combinaison d'objets avec et sans paramètres de rétention de verrouillage d'objets S3.

Les paramètres de rétention au niveau de l'objet sont définis à l'aide de l'API REST S3. Les paramètres de

rétention d'un objet remplacent tous les paramètres de rétention par défaut du compartiment.

Vous pouvez spécifier les paramètres suivants pour chaque objet :

- **Mode de rétention** : soit COMPLIANCE, soit GOVERNANCE.
- **Retain-until-date** : une date spécifiant la durée pendant laquelle la version de l'objet doit être conservée par StorageGRID.
  - En mode COMPLIANCE, si la date de conservation est dans le futur, l'objet peut être récupéré, mais il ne peut pas être modifié ou supprimé. La date de conservation peut être augmentée, mais cette date ne peut pas être diminuée ou supprimée.
  - En mode GOVERNANCE, les utilisateurs disposant d'une autorisation spéciale peuvent contourner le paramètre retain-until-date. Ils peuvent supprimer une version d'objet avant l'expiration de sa période de conservation. Ils peuvent également augmenter, diminuer ou même supprimer le retain-until-date.
- **Conservation légale** : L'application d'une conservation légale à une version d'objet verrouille immédiatement cet objet. Par exemple, vous pourriez avoir besoin d'appliquer une conservation légale à un objet lié à une enquête ou à un litige. Une conservation légale n'a pas de date d'expiration, mais reste en place jusqu'à ce qu'elle soit explicitement levée.

  Le paramètre de conservation légale d'un objet est indépendant du mode de conservation et de la date limite de conservation. Si une version d'un objet est soumise à une conservation légale, personne ne peut supprimer cette version.

Pour spécifier les paramètres de S3 Object Lock lors de l'ajout d'une version d'objet à un compartiment, émettez une requête "PutObject", "CopyObject" ou "CreateMultipartUpload".

${post_edited_translations.segment}

- `x-amz-object-lock-mode`, qui peut être COMPLIANCE ou GOVERNANCE (sensible à la casse).

  > Si vous spécifiez `x-amz-object-lock-mode`, vous devez également spécifier `x-amz-object-lock-retain-until-date`.

- `x-amz-object-lock-retain-until-date`
  - La valeur de conservation jusqu'à la date doit être au format `2020-08-10T21:46:00Z`. Les fractions de seconde sont autorisées, mais seules trois décimales sont conservées (précision à la milliseconde). Les autres formats ISO 8601 ne sont pas autorisés.
  - ${post_edited_translations.segment}
- `x-amz-object-lock-legal-hold`

  Si la conservation légale est ON (sensible à la casse), l'objet est placé sous conservation légale. Si la conservation légale est OFF, aucune conservation légale n'est appliquée. Toute autre valeur entraîne une erreur 400 Bad Request (InvalidArgument).

Si vous utilisez l'un de ces en-têtes de requête, tenez compte des restrictions suivantes :

- L' `Content-MD5`en-tête de requête est requis si un `x-amz-object-lock-*`en-tête de requête est présent dans la requête PutObject. `Content-MD5`n'est pas requis pour CopyObject ou CreateMultipartUpload.
- Si le compartiment n'a pas activé S3 Object Lock et qu'un en-tête de requête `x-amz-object-lock-*` est présent, une erreur 400 Bad Request (InvalidRequest) est renvoyée.

- La requête PutObject prend en charge l'utilisation de `x-amz-storage-class: REDUCED_REDUNDANCY` pour correspondre au comportement d'AWS. Toutefois, lorsqu'un objet est ingéré dans un compartiment avec S3 Object Lock activé, StorageGRID effectue toujours une ingestion dual-commit.
- Une réponse GET ou HeadObject de version ultérieure inclura les en-têtes `x-amz-object-lock-mode`, `x-amz-object-lock-retain-until-date` et `x-amz-object-lock-legal-hold`, s'ils sont configurés et si l'expéditeur de la requête dispose des autorisations `s3:Get*` correctes.

Vous pouvez utiliser la clé de condition de politique `s3:object-lock-remaining-retention-days` pour limiter les périodes de rétention minimales et maximales autorisées pour vos objets.

${post_edited_translations.segment}

${post_edited_translations.segment}

- `PutObjectLegalHold`

  Si la nouvelle valeur de conservation légale est ON, l'objet est placé sous conservation légale. Si la valeur de conservation légale est OFF, la conservation légale est levée.

- `PutObjectRetention`
  - La valeur du mode peut être COMPLIANCE ou GOVERNANCE (sensible à la casse).
  - La valeur de conservation jusqu'à la date doit être au format `2020-08-10T21:46:00Z`. Les fractions de seconde sont autorisées, mais seules trois décimales sont conservées (précision à la milliseconde). Les autres formats ISO 8601 ne sont pas autorisés.
  - Si une version d'objet possède déjà une date de conservation (retain-until), vous pouvez uniquement l'augmenter. La nouvelle valeur doit être dans le futur.

Comment utiliser le mode GOVERNANCE

Les utilisateurs qui disposent de l'autorisation `s3:BypassGovernanceRetention` peuvent contourner les paramètres de rétention active d'un objet qui utilise le mode GOVERNANCE. Toutes les opérations DELETE ou PutObjectRetention doivent inclure l'en-tête de requête `x-amz-bypass-governance-retention:true`. Ces utilisateurs peuvent effectuer ces opérations supplémentaires :

- Exécutez les opérations DeleteObject ou DeleteObjects pour supprimer une version d'objet avant que sa période de rétention ne soit écoulée.

  Les objets soumis à une conservation légale ne peuvent pas être supprimés. La conservation légale doit être désactivée.

- Effectuez des opérations PutObjectRetention qui modifient le mode d'une version d'objet de GOVERNANCE à COMPLIANCE avant que la période de rétention de l'objet ne soit écoulée.

  Il est strictement interdit de passer du mode COMPLIANCE au mode GOVERNANCE.

- Effectuez des opérations PutObjectRetention pour augmenter, diminuer ou supprimer la période de conservation d'une version d'objet.

Informations connexes

- "Gérez les objets avec S3 Object Lock"
- "${post_edited_translations.segment}"

- "Guide de l'utilisateur Amazon Simple Storage Service : Verrouillage des objets"

Découvrez la configuration du cycle de vie S3 pour StorageGRID

Vous pouvez créer une configuration de cycle de vie S3 pour contrôler le moment où des objets spécifiques sont supprimés du système StorageGRID.

L'exemple simple présenté dans cette section illustre comment une configuration de cycle de vie S3 peut contrôler la suppression (expiration) de certains objets dans des compartiments S3 spécifiques. L'exemple de cette section est fourni uniquement à des fins d'illustration. Pour des informations complètes sur la création de configurations de cycle de vie S3, consultez "Guide de l'utilisateur Amazon Simple Storage Service : gestion du cycle de vie des objets". Notez que StorageGRID prend uniquement en charge les actions d'expiration ; il ne prend pas en charge les actions de transition.

Ce qu'est la configuration du cycle de vie

Une configuration de cycle de vie est un ensemble de règles appliquées aux objets dans des compartiments S3 spécifiques. Chaque règle spécifie quels objets sont concernés et quand ces objets expireront (à une date précise ou après un certain nombre de jours).

StorageGRID prend en charge jusqu'à 1 000 règles de cycle de vie dans une configuration de cycle de vie. Chaque règle peut inclure les éléments XML suivants :

- Expiration : Supprimer un objet lorsqu'une date spécifiée est atteinte ou lorsqu'un nombre de jours spécifié est atteint, à compter de la date d'ingestion de l'objet.
- NoncurrentVersionExpiration : Supprimez un objet lorsqu'un nombre de jours spécifié est atteint, à compter du moment où l'objet est devenu non actuel.
- Filtrer (Préfixe, Tag)
- Statut
- ID

Chaque objet suit les paramètres de rétention définis soit par le cycle de vie d'un compartiment S3, soit par une stratégie ILM. Lorsqu'un cycle de vie de compartiment S3 est configuré, les actions d'expiration de ce cycle de vie prévalent sur la stratégie ILM pour les objets correspondant au filtre de cycle de vie du compartiment. Les objets qui ne correspondent pas au filtre de cycle de vie du compartiment utilisent les paramètres de rétention de la stratégie ILM. Si un objet correspond à un filtre de cycle de vie de compartiment et qu'aucune action d'expiration n'est explicitement spécifiée, les paramètres de rétention de la stratégie ILM ne sont pas utilisés et il est sous-entendu que les versions de l'objet sont conservées indéfiniment. Voir "Exemples de priorités pour le cycle de vie d'un compartiment S3 et la politique ILM".

Par conséquent, un objet peut être retiré de la grille même si les instructions de placement d'une règle ILM s'appliquent toujours à cet objet. Ou, un objet peut être conservé sur la grille même après l'expiration de toutes les instructions de placement ILM le concernant. Pour plus de détails, voir "Comment l'ILM fonctionne tout au long du cycle de vie d'un objet".

> La configuration du cycle de vie des compartiments peut être utilisée avec les compartiments dont le S3 Object Lock est activé, mais la configuration du cycle de vie des compartiments n'est pas prise en charge pour les anciens compartiments Compliant.

StorageGRID prend en charge l'utilisation des opérations de compartiment suivantes pour gérer les

configurations de cycle de vie :

- DeleteBucketLifecycle
- GetBucketLifecycleConfiguration
- PutBucketLifecycleConfiguration

Créer une configuration de cycle de vie

La première étape de la création d'une configuration de cycle de vie consiste à créer un fichier JSON contenant une ou plusieurs règles. Par exemple, ce fichier JSON contient trois règles, comme suit :

1. La règle 1 s'applique uniquement aux objets qui correspondent au préfixe `category1/` et qui ont une valeur de `key2 tag2`. Le paramètre `Expiration` spécifie que les objets correspondant au filtre expireront à minuit le 22 août 2020.
2. La règle 2 s'applique uniquement aux objets qui correspondent au préfixe `category2/`. Le `Expiration` paramètre spécifie que les objets correspondant au filtre expireront 100 jours après leur ingestion.

   > Les règles qui spécifient un nombre de jours sont relatives à la date d'ingestion de l'objet. Si la date actuelle est postérieure à la date d'ingestion plus le nombre de jours, certains objets peuvent être supprimés du compartiment dès l'application de la configuration du cycle de vie.

3. La règle 3 s'applique uniquement aux objets qui correspondent au préfixe `category3/`. Le `Expiration` paramètre spécifie que toutes les versions non actuelles des objets correspondants expireront 50 jours après qu'elles soient devenues non actuelles.

```
{
    "Rules": [
        {
            "ID": "rule1",
            "Filter": {
                "And": {
                    "Prefix": "category1/",
                    "Tags": [
                        {
                            "Key": "key2",
                            "Value": "tag2"
                        }
                    ]
                }
            },
            "Expiration": {
                "Date": "2020-08-22T00:00:00Z"
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule2",
            "Filter": {
                "Prefix": "category2/"
            },
            "Expiration": {
                "Days": 100
            },
            "Status": "Enabled"
        },
        {
            "ID": "rule3",
            "Filter": {
                "Prefix": "category3/"
            },
            "NoncurrentVersionExpiration": {
                "NoncurrentDays": 50
            },
            "Status": "Enabled"
        }
    ]
}
```

Appliquer la configuration du cycle de vie au compartiment

Après avoir créé le fichier de configuration du cycle de vie, vous l'appliquez à un compartiment en émettant une requête PutBucketLifecycleConfiguration.

Cette requête applique la configuration de cycle de vie du fichier d'exemple aux objets d'un compartiment nommé `testbucket`.

```
aws s3api --endpoint-url <StorageGRID endpoint> put-bucket-lifecycle-
configuration
--bucket testbucket --lifecycle-configuration file://bktjson.json
```

Pour vérifier qu'une configuration de cycle de vie a bien été appliquée au compartiment, envoyez une requête GetBucketLifecycleConfiguration. Par exemple :

```
aws s3api --endpoint-url <StorageGRID endpoint> get-bucket-lifecycle-
configuration
 --bucket testbucket
```

Une réponse réussie répertorie la configuration du cycle de vie que vous venez d'appliquer.

Vérifiez que l'expiration du cycle de vie du compartiment s'applique à l'objet

Vous pouvez déterminer si une règle d'expiration dans la configuration du cycle de vie s'applique à un objet spécifique lors de l'émission d'une requête PutObject, HeadObject ou GetObject. Si une règle s'applique, la réponse inclut un `Expiration` paramètre qui indique quand l'objet expire et quelle règle d'expiration a été appliquée.

> Étant donné que le cycle de vie du bucket remplace ILM, la `expiry-date`date affichée correspond à la date réelle à laquelle l'objet sera supprimé. Pour plus de détails, consultez "Comment la rétention d'objet est déterminée".

Par exemple, cette requête PutObject a été émise le 22 juin 2020 et place un objet dans le `testbucket` bucket.

```
aws s3api --endpoint-url <StorageGRID endpoint> put-object
--bucket testbucket --key obj2test2 --body bktjson.json
```

La réponse positive indique que l'objet expirera dans 100 jours (01 Oct 2020) et qu'il correspond à la règle 2 de la configuration du cycle de vie.

```
{
      *"Expiration": "expiry-date=\"Thu, 01 Oct 2020 09:07:49 GMT\", rule-
id=\"rule2\"",
      "ETag": "\"9762f8a803bc34f5340579d4446076f7\""
}
```

Par exemple, cette requête HeadObject a été utilisée pour obtenir les métadonnées du même objet dans le compartiment testbucket.

```
aws s3api --endpoint-url <StorageGRID endpoint> head-object
--bucket testbucket --key obj2test2
```

La réponse de succès inclut les métadonnées de l'objet et indique que l'objet expirera dans 100 jours et qu'il correspond à la Rule 2.

```
{
      "AcceptRanges": "bytes",
      *"Expiration": "expiry-date=\"Thu, 01 Oct 2020 09:07:48 GMT\", rule-
id=\"rule2\"",
      "LastModified": "2020-06-23T09:07:48+00:00",
      "ContentLength": 921,
      "ETag": "\"9762f8a803bc34f5340579d4446076f7\""
      "ContentType": "binary/octet-stream",
      "Metadata": {}
}
```

Pour les compartiments compatibles avec le versionnage, l' `x-amz-expiration`en-tête de réponse s'applique uniquement aux versions actuelles des objets.

Recommandations pour la mise en œuvre de l'API REST S3 avec StorageGRID

${post_edited_translations.segment}

${post_edited_translations.segment}

Si votre application vérifie régulièrement si un objet existe dans un chemin où vous ne vous attendez pas à ce que l'objet existe réellement, vous devez utiliser le niveau « Available » "cohérence". Par exemple, vous devez utiliser la cohérence « Available » si votre application effectue une requête HEAD sur un emplacement avant d'y effectuer un PUT.

Sinon, si l'opération HEAD ne trouve pas l'objet, vous pourriez recevoir un grand nombre d'erreurs 500 Internal Server errors si deux ou plusieurs Storage Nodes du même site sont indisponibles ou si un site distant est inaccessible.

Vous pouvez définir la cohérence « Disponible » pour chaque compartiment à l'aide de la requête "Cohérence PUT Bucket", ou vous pouvez spécifier la cohérence dans l'en-tête de requête pour une opération API individuelle.

Recommandations concernant les clés d'objet

Suivez ces recommandations pour les noms de clés d'objet, en fonction du moment où le compartiment a été créé pour la première fois.

Compartiments créés dans StorageGRID 11.4 ou une version antérieure

- N'utilisez pas de valeurs aléatoires comme premiers quatre caractères des clés d'objet. Ceci contraste avec l'ancienne recommandation d'AWS concernant les préfixes de clés. Utilisez plutôt des préfixes non aléatoires et non uniques, tels que `image`.
- Si vous suivez l'ancienne recommandation d'AWS d'utiliser des caractères aléatoires et uniques dans les préfixes de clés, préfixez les clés d'objet avec un nom de répertoire. C'est-à-dire, utilisez ce format :

  `mybucket/mydir/f8e3-image3132.jpg`

  Au lieu de ce format :

  `mybucket/f8e3-image3132.jpg`

${post_edited_translations.segment}

Il n'est pas nécessaire de restreindre les noms de clés d'objets pour respecter les bonnes pratiques en matière de performances. Dans la plupart des cas, vous pouvez utiliser des valeurs aléatoires pour les quatre premiers caractères des noms de clés d'objets.

Une exception à cela est une charge de travail S3 qui supprime continuellement tous les objets après une courte période. Pour minimiser l'impact sur les performances pour ce cas d'utilisation, faites varier la partie initiale du nom de la clé tous les quelques milliers d'objets avec un élément tel que la date. Par exemple, supposez qu'un client S3 écrive généralement 2 000 objets/seconde et que la politique ILM ou de cycle de vie du compartiment supprime tous les objets après trois jours. Pour minimiser l'impact sur les performances, vous pouvez nommer les clés en utilisant un modèle comme celui-ci : `/mybucket/mydir/`*`yyyymmddhhmmss-random_UUID`*`.jpg`

${post_edited_translations.segment}

Si la "option globale pour compresser les objets stockés" est activée, les applications clientes S3 doivent éviter d'effectuer des opérations GetObject qui spécifient le retour d'une plage d'octets. Ces opérations de « lecture par plage » sont inefficaces car StorageGRID doit effectivement décompresser les objets pour accéder aux octets demandés. Les opérations GetObject qui demandent une petite plage d'octets à partir d'un objet très volumineux sont particulièrement inefficaces ; par exemple, il est inefficace de lire une plage de 10 Mo à partir d'un objet compressé de 50 Go.

Si les plages sont lues à partir d'objets compressés, les requêtes client peuvent expirer.

Si vous devez compresser des objets et que votre application cliente doit utiliser des lectures par plage, augmentez le délai d'expiration de lecture pour l'application.