Comment StorageGRID implémente l'API REST S3

StorageGRID software

NetApp
December 03, 2025

Sommaire

Comment StorageGRID implémente l'API REST S3

Demandes clients conflictuelles

Les demandes client conflictuelles, telles que deux clients écrivant sur la même clé, sont résolues sur la base des « derniers gagnants ».

Le moment de l'évaluation des « dernières victoires » est basé sur le moment où le système StorageGRID termine une demande donnée, et non sur le moment où les clients S3 commencent une opération.

Valeurs de cohérence

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

Par défaut, StorageGRID garantit la cohérence de lecture après écriture pour les objets nouvellement créés. Tout GET suivant un PUT terminé avec succès pourra lire les données nouvellement écrites. Les écrasements d'objets existants, les mises à jour de métadonnées et les suppressions sont finalement cohérents. Les écrasements prennent généralement quelques secondes ou minutes pour se propager, mais peuvent prendre jusqu'à 15 jours.

Si vous souhaitez effectuer des opérations sur les objets avec une cohérence différente, vous pouvez :

- Spécifier une consistance pourchaque seau .
- Spécifier une consistance pourchaque opération API .
- Modifiez la cohérence par défaut de l'ensemble 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**.
  - .

> Une modification de la cohérence à l'échelle de la grille s'applique uniquement aux compartiments créés après la modification du paramètre. Pour déterminer les détails d'une modification, consultez le journal d'audit situé à l'adresse `/var/local/log` (rechercher **consistencyLevel**).

Valeurs de cohérence

La cohérence affecte la manière dont les métadonnées utilisées par StorageGRID pour suivre les objets sont distribuées entre les nœuds et, par conséquent, la disponibilité des objets pour les demandes des clients.

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

- **Tous** : Tous les nœuds reçoivent les données immédiatement, sinon la demande échouera.
- **Strong-global** : garantit la cohérence de lecture après écriture pour toutes les demandes client sur tous les sites.

- **Strong-site** : garantit la cohérence de lecture après écriture pour toutes les requêtes client au sein d'un site.
- **Lecture après nouvelle écriture** : (par défaut) Fournit une cohérence de lecture après écriture pour les nouveaux objets et une cohérence éventuelle pour les mises à jour d'objets. Offre des garanties de haute disponibilité et de protection des données. Recommandé dans la plupart des cas.
- **Disponible** : Fournit une cohérence éventuelle pour les nouveaux objets et les mises à jour d'objets. Pour les buckets S3, utilisez-les uniquement si nécessaire (par exemple, pour un bucket contenant des valeurs de journal rarement lues ou pour des opérations HEAD ou GET sur des clés qui n'existent pas). Non pris en charge pour les buckets S3 FabricPool .

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

Lorsqu'une opération HEAD ou GET utilise la cohérence « Lecture après nouvelle écriture », StorageGRID effectue la recherche en plusieurs étapes, comme suit :

- Il recherche d'abord l'objet en utilisant une faible cohérence.
- Si cette recherche échoue, elle répète la recherche à la valeur de cohérence suivante jusqu'à ce qu'elle atteigne une cohérence équivalente au comportement de 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 une cohérence équivalente au comportement de strong-global. Étant donné 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 de serveur interne 500 si deux ou plusieurs nœuds de stockage sur le même site ne sont pas disponibles.

À moins que vous n'ayez besoin de 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 fournit uniquement la cohérence éventuelle. Il ne réessaye pas une opération ayant échoué en augmentant la cohérence, il ne nécessite donc pas que plusieurs copies des métadonnées de l'objet soient disponibles.

Spécifier la cohérence pour le fonctionnement de l'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 demande. 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.

Spécifier la cohérence du bucket

Pour définir la cohérence du bucket, vous pouvez utiliser le StorageGRID"Cohérence du seau PUT" demande. Ou vous pouvez"changer la consistance d'un seau" du gestionnaire locataire.

Lorsque vous définissez la cohérence d'un bucket, tenez compte des éléments suivants :

- La définition de la cohérence d'un bucket détermine la cohérence utilisée pour les opérations S3 effectuées sur les objets du bucket ou sur la configuration du bucket. Cela n'affecte pas les opérations sur le bucket lui-même.
- La cohérence d'une opération API individuelle remplace la cohérence du bucket.
- En général, les buckets doivent utiliser la cohérence par défaut, « Lecture après nouvelle écriture ». Si les requêtes ne fonctionnent pas correctement, modifiez le comportement du client de l'application si possible. Ou configurez le client pour spécifier la cohérence de chaque demande d'API. Définissez la consistance au niveau du seau uniquement en dernier recours.

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

Votre choix de cohérence et votre règle ILM affectent la manière dont les objets sont protégés. Ces paramètres peuvent interagir.

Par exemple, la cohérence utilisée lors du stockage d'un objet affecte le placement initial des métadonnées de l'objet, tandis que le comportement d'ingestion sélectionné pour la règle ILM affecte le placement initial des copies de l'objet. Étant donné que StorageGRID nécessite l'accès aux métadonnées d'un objet et à ses données pour répondre aux demandes des clients, la sélection de niveaux de protection correspondants pour la cohérence et le comportement d'ingestion peut fournir une meilleure protection initiale des données et des réponses système plus prévisibles.

Ce qui suit"options d'ingestion" sont disponibles pour les règles ILM :

Double engagement
: StorageGRID effectue immédiatement des copies intermédiaires de l'objet et renvoie le succès au client. Les copies spécifiées dans la règle ILM sont réalisées lorsque cela est possible.

Strict
: Toutes les copies spécifiées dans la règle ILM doivent être effectuées avant que le succès ne soit renvoyé au client.

Équilibré
: StorageGRID tente de réaliser toutes les copies spécifiées dans la règle ILM lors de l'ingestion ; si cela n'est pas possible, des copies intermédiaires sont réalisées et le succès est renvoyé au client. Les copies spécifiées dans la règle ILM sont réalisées lorsque cela est possible.

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

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

- **Règle ILM** : Créez deux copies d'objet, une sur le site local et une sur un site distant. Adoptez un comportement d'ingestion strict.
- **cohérence** : Global fort (les métadonnées de l'objet sont immédiatement distribuées à tous les sites).

Lorsqu'un client stocke un objet dans la grille, StorageGRID effectue les deux copies de l'objet et distribue les métadonnées aux deux sites avant de renvoyer le succès au client.

L'objet est entièrement protégé contre la perte au moment de l'ingestion réussie du message. Par exemple, si le site local est perdu peu de temps après l'ingestion, des copies des données d'objet et des métadonnées

d'objet existent toujours sur le site distant. L'objet est entièrement récupérable.

Si vous avez utilisé la même règle ILM et la cohérence de site forte, le client peut recevoir un message de réussite après la réplication des données d'objet sur le site distant, mais avant que les métadonnées d'objet y soient distribuées. Dans ce cas, le niveau de protection des métadonnées de l'objet ne correspond pas au niveau de protection des données de l'objet. Si le site local est perdu peu de temps après l'ingestion, les métadonnées de l'objet sont perdues. L'objet ne peut pas être récupéré.

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

Versionnage d'objet

Vous pouvez définir l'état de version d'un bucket si vous souhaitez conserver plusieurs versions de chaque objet. L'activation du contrôle de version pour un bucket peut aider à protéger 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 contrôle de version avec prise en charge de la plupart des fonctionnalités et avec certaines limitations. StorageGRID prend en charge jusqu'à 10 000 versions de chaque objet.

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

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

> ⓘ Le contrôle de version ne peut être activé que sur les buckets 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 analyse en continu tous les objets et les réévalue par rapport à la politique ILM actuelle. Toutes les modifications que vous apportez aux stratégies ILM sont appliquées à tous les objets précédemment ingérés. Cela inclut les versions précédemment ingérées si le contrôle de version est activé. L'analyse ILM applique de nouvelles modifications ILM aux objets précédemment ingérés.

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

Lorsque vous téléchargez une nouvelle version d'un objet à l'aide d'une opération de téléchargement en plusieurs parties, l'heure non actuelle de la version d'origine de l'objet reflète le moment où le téléchargement en plusieurs parties a été créé pour la nouvelle version, et non le moment où le téléchargement en plusieurs parties a été terminé. Dans des cas limités, l'heure non actuelle de la version originale peut être antérieure de plusieurs heures ou jours à l'heure de la version actuelle.

Informations connexes

- "Comment les objets versionnés S3 sont supprimés"
- "Règles et politiques ILM pour les objets versionnés S3 (exemple 4)" .

Utiliser l'API REST S3 pour configurer le verrouillage d'objet S3

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

Comment activer le verrouillage d'objet S3 pour un bucket

Si le paramètre global de verrouillage d'objet S3 est activé pour votre système StorageGRID , vous pouvez éventuellement activer le verrouillage d'objet S3 lorsque vous créez chaque bucket.

Le verrouillage d'objet S3 est un paramètre permanent qui ne peut être activé que lorsque vous créez un bucket. Vous ne pouvez pas ajouter ou désactiver le verrouillage d'objet S3 après la création d'un bucket.

Pour activer le verrouillage d'objet S3 pour un bucket, utilisez l'une de ces méthodes :

- Créez le bucket à l'aide du gestionnaire de locataires. Voir "Créer un bucket S3" .
- Créez le bucket à l'aide d'une requête CreateBucket avec le `x-amz-bucket-object-lock-enabled` en-tête de requête. Voir "Opérations sur les godets" .

S3 Object Lock nécessite le contrôle de version du bucket, qui est activé automatiquement lors de la création du bucket. Vous ne pouvez pas suspendre le contrôle de version du bucket. Voir "Versionnage d'objet" .

Paramètres de conservation par défaut pour un bucket

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 tant que cette date n'est pas atteinte.

- En mode GOUVERNANCE :
  - Les utilisateurs avec le `s3:BypassGovernanceRetention` l'autorisation peut utiliser le `x-amz-bypass-governance-retention: true` en-tête de demande pour contourner les paramètres de conservation.
  - 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

Chaque bucket peut avoir une période de conservation par défaut spécifiée en années ou en jours.

Comment définir la rétention par défaut pour un bucket

Pour définir la rétention par défaut d'un bucket, utilisez l'une de ces méthodes :

- Gérez les paramètres du bucket à partir du gestionnaire de locataires. Voir"Créer un bucket S3" et"Mettre à jour la conservation par défaut du verrouillage des objets S3" .
- Émettez une demande PutObjectLockConfiguration pour le bucket afin de spécifier le mode par défaut et le nombre de jours ou d'années par défaut.

Configuration de PutObjectLock

La demande PutObjectLockConfiguration vous permet de définir et de modifier le mode de conservation par défaut et la période de conservation par défaut pour un bucket sur lequel le verrouillage d'objet S3 est activé. Vous pouvez également supprimer les paramètres de conservation par défaut précédemment configurés.

Lorsque de nouvelles versions d'objet sont ingérées dans le bucket, 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 conservation par défaut est utilisée pour calculer la date de conservation 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 conservation de la version d'objet reste la même et n'est pas recalculée à l'aide de la nouvelle période de conservation par défaut.

Vous devez avoir le `s3:PutBucketObjectLockConfiguration` autorisation, ou être un compte root, pour terminer cette opération.

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

Exemple de demande

Cet exemple active le verrouillage d'objet S3 pour un bucket et définit le mode de conservation par défaut sur CONFORMITÉ et la période de conservation 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>
```

Comment déterminer la rétention par défaut d'un bucket

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

- Afficher le bucket dans le gestionnaire de locataires. Voir "Afficher les buckets S3" .
- Émettez une demande GetObjectLockConfiguration.

Obtenir la configuration du verrouillage de l'objet

La demande GetObjectLockConfiguration vous permet de déterminer si le verrouillage d'objet S3 est activé pour un compartiment et, s'il est activé, de voir s'il existe un mode de conservation par défaut et une période de conservation configurés pour le compartiment.

Lorsque de nouvelles versions d'objet sont ingérées dans le bucket, le mode de rétention par défaut est appliqué si `x-amz-object-lock-mode` n'est pas spécifié. La période de conservation par défaut est utilisée pour calculer la date de conservation si `x-amz-object-lock-retain-until-date` n'est pas spécifié.

Vous devez avoir le `s3:GetBucketObjectLockConfiguration` autorisation, ou être un compte root, pour terminer cette opération.

Exemple de demande

```
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>
```

Comment spécifier les paramètres de conservation d'un objet

Un bucket avec S3 Object Lock activé peut contenir une combinaison d'objets avec et sans paramètres de conservation S3 Object Lock.

Les paramètres de conservation au niveau de l'objet sont spécifiés 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 conservation** : Soit CONFORMITÉ, soit GOUVERNANCE.
- **Retain-until-date** : une date spécifiant la durée pendant laquelle la version de l'objet doit être conservée par StorageGRID.
    - En mode CONFORMITÉ, 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 GOUVERNANCE, les utilisateurs disposant d'une autorisation spéciale peuvent contourner le paramètre de conservation jusqu'à la date. Ils peuvent supprimer une version d'objet avant l'expiration de sa période de conservation. Ils peuvent également augmenter, diminuer ou même supprimer la date de conservation.

- **Conservation légale** : L'application d'une conservation légale à une version d'objet verrouille immédiatement cet objet. Par exemple, vous pourriez avoir besoin de suspendre légalement un objet lié à une enquête ou à un litige juridique. Une conservation légale n'a pas de date d'expiration, mais reste en place jusqu'à ce qu'elle soit explicitement supprimée.

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

Pour spécifier les paramètres de verrouillage d'objet S3 lors de l'ajout d'une version d'objet à un bucket, émettez un"Mettre l'objet" ,"Copier l'objet" , ou"Créer un téléchargement multi-parties" demande.

Vous pouvez utiliser les éléments suivants :

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

  > Si vous précisez `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 secondes sont autorisées, mais seuls 3 chiffres décimaux sont conservés (précision en millisecondes). Les autres formats ISO 8601 ne sont pas autorisés.
    - La date de conservation doit être dans le futur.
- `x-amz-object-lock-legal-hold`

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

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

- Le `Content-MD5` l'en-tête de la demande est requis le cas échéant `x-amz-object-lock-*` l'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 de verrouillage d'objet S3 activé et qu'un `x-amz-object-lock-*` l'en-tête de requête 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. Cependant, lorsqu'un objet est ingéré dans un bucket avec S3 Object Lock activé, StorageGRID effectuera toujours une ingestion à double validation.
- Une réponse de version GET ou HeadObject ultérieure inclura les en-têtes `x-amz-object-lock-mode` , `x-amz-object-lock-retain-until-date` , et `x-amz-object-lock-legal-hold` , si configuré et si l'expéditeur de la requête a le bon `s3:Get*` autorisations.

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

Comment mettre à jour les paramètres de conservation d'un objet

Si vous devez mettre à jour les paramètres de conservation légale ou de rétention d'une version d'objet existante, vous pouvez effectuer les opérations de sous-ressources d'objet suivantes :

- `PutObjectLegalHold`

  Si la nouvelle valeur de conservation légale est activée, l'objet est placé sous une conservation légale. Si la valeur de maintien légal est OFF, le maintien légal est levé.

- `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 secondes sont autorisées, mais seuls 3 chiffres décimaux sont conservés (précision en millisecondes). Les autres formats ISO 8601 ne sont pas autorisés.
  - Si une version d'objet possède une date de conservation existante, vous ne pouvez que l'augmenter. La nouvelle valeur doit être dans le futur.

Comment utiliser le mode GOUVERNANCE

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

- Exécutez les opérations DeleteObject ou DeleteObjects pour supprimer une version d'objet avant l'expiration de sa période de conservation.

  Les objets faisant l'objet d'une suspension légale ne peuvent pas être supprimés. La retenue légale doit être désactivée.

- Exécutez des opérations PutObjectRetention qui modifient le mode de version d'un objet de GOUVERNANCE à CONFORMITÉ avant l'expiration de la période de conservation de l'objet.

  Le passage du mode CONFORMITÉ au mode GOUVERNANCE n'est jamais autorisé.

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

Informations connexes

- "Gérer les objets avec S3 Object Lock"
- "Utilisez S3 Object Lock pour conserver les objets"
- "Guide de l'utilisateur d'Amazon Simple Storage Service : Verrouillage d'objets"

Créer une configuration du cycle de vie S3

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 de cette section illustre comment une configuration de cycle de vie S3 peut contrôler le moment où certains objets sont supprimés (expirés) de compartiments S3 spécifiques. L'exemple dans cette section est fourni à titre d'illustration uniquement. Pour plus de détails sur la création de configurations de cycle de vie S3, consultez "Guide de l'utilisateur d'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.

Quelle 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 spécifique 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 : supprimez un objet lorsqu'une date spécifiée est atteinte ou lorsqu'un nombre de jours spécifié est atteint, à compter du moment où l'objet a été ingéré.
- NoncurrentVersionExpiration : supprimez un objet lorsqu'un nombre de jours spécifié est atteint, à compter du moment où l'objet est devenu non actuel.
- Filtre (préfixe, balise)
- Statut
- ID

Chaque objet suit les paramètres de conservation d'un cycle de vie de compartiment S3 ou d'une politique ILM. Lorsqu'un cycle de vie de compartiment S3 est configuré, les actions d'expiration du cycle de vie remplacent 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 bucket utilisent les paramètres de conservation 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 conservation de la stratégie ILM ne sont pas utilisés et il est implicite que les versions d'objet sont conservées pour toujours. Voir "Exemples de priorités pour le cycle de vie du bucket S3 et la politique ILM" .

Par conséquent, un objet peut être supprimé de la grille même si les instructions de placement d'une règle ILM s'appliquent toujours à l'objet. Ou bien, un objet peut être conservé sur la grille même après l'expiration des instructions de placement ILM pour l'objet. Pour plus de détails, consultez la section "Comment ILM fonctionne tout au long de la vie d'un objet" .

> La configuration du cycle de vie du bucket peut être utilisée avec les buckets pour lesquels le verrouillage d'objet S3 est activé, mais la configuration du cycle de vie du bucket n'est pas prise en charge pour les buckets conformes hérités.

StorageGRID prend en charge l'utilisation des opérations de bucket suivantes pour gérer les configurations du cycle de vie :

- Supprimer le cycle de vie du bucket
- GetBucketLifecycleConfiguration
- Configuration du cycle de vie de PutBucket

Créer une configuration du cycle de vie

Comme première étape de la création d'une configuration de cycle de vie, vous créez un fichier JSON qui inclut une ou plusieurs règles. Par exemple, ce fichier JSON comprend trois règles, comme suit :

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

   > Les règles qui spécifient un nombre de jours sont relatives au moment où l'objet a été ingéré. Si la date actuelle dépasse la date d'ingestion plus le nombre de jours, certains objets peuvent être supprimés du bucket dès que la configuration du cycle de vie est appliquée.

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

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

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

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

Pour valider qu'une configuration de cycle de vie a été correctement appliquée au bucket, émettez une demande 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.

Valider que l'expiration du cycle de vie du bucket 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 demande PutObject, HeadObject ou GetObject. Si une règle s'applique, la réponse comprend une `Expiration` paramètre qui indique quand l'objet expire et quelle règle d'expiration a été respectée.

> Étant donné que le cycle de vie du bucket remplace ILM, le `expiry-date` la date réelle à laquelle l'objet sera supprimé est indiquée. Pour plus de détails, consultez la section "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` seau.

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

La réponse de réussite indique que l'objet expirera dans 100 jours (1er octobre 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 des métadonnées pour le même objet dans le bucket testbucket.

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

La réponse de réussite inclut les métadonnées de l'objet et indique que l'objet expirera dans 100 jours et qu'il correspond à la règle 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 buckets avec contrôle de version activé, le `x-amz-expiration` l'en-tête de réponse s'applique uniquement aux versions actuelles des objets.

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

Vous devez suivre ces recommandations lors de l'implémentation de l'API REST S3 à utiliser avec StorageGRID.

Recommandations pour les HEADs vers des objets inexistants

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 l'option « Disponible »."cohérence" . Par exemple, vous devez utiliser la cohérence « Disponible » si votre application HEAD un emplacement avant d'y effectuer un PUT.

Dans le cas contraire, si l'opération HEAD ne trouve pas l'objet, vous risquez de recevoir un nombre élevé d'erreurs de serveur interne 500 si deux ou plusieurs nœuds de stockage sur le même site ne sont pas disponibles ou si un site distant est inaccessible.

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

Recommandations pour les clés d'objet

Suivez ces recommandations pour les noms de clés d'objet, en fonction de la date de création initiale du bucket.

Buckets créés dans StorageGRID 11.4 ou version antérieure

- N'utilisez pas de valeurs aléatoires comme quatre premiers caractères des clés d'objet. Ceci est en contraste avec l'ancienne recommandation AWS pour les préfixes de clé. Utilisez plutôt des préfixes non aléatoires et non uniques, tels que `image` .
- Si vous suivez l'ancienne recommandation AWS d'utiliser des caractères aléatoires et uniques dans les préfixes de clé, 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
  ```

Buckets créés dans StorageGRID 11.4 ou version ultérieure

Il n'est pas nécessaire de restreindre les noms de clés d'objet pour respecter les meilleures 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'objet.

> Une exception à cette règle est une charge de travail S3 qui supprime en continu tous les objets après une courte période de temps. Pour minimiser l'impact sur les performances de ce cas d'utilisation, faites varier une partie initiale du nom de la clé tous les plusieurs milliers d'objets avec quelque chose comme la date. Par exemple, supposons qu'un client S3 écrit généralement 2 000 objets/seconde et que la politique de cycle de vie ILM ou bucket 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`

Recommandations pour les « lectures de plage »

Si le"option globale pour compresser les objets stockés" est activé, les applications clientes S3 doivent éviter d'effectuer des opérations GetObject qui spécifient une plage d'octets à renvoyer. Ces opérations de « lecture de plage » sont inefficaces car StorageGRID doit décompresser efficacement les objets pour accéder aux octets demandés. Les opérations GetObject qui demandent une petite plage d'octets à partir d'un très grand objet 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 demandes des clients peuvent expirer.

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