Implémentation de l'API REST S3 par StorageGRID

StorageGRID 11.9

NetApp
November 08, 2024

Sommaire

Implémentation de l'API REST S3 par StorageGRID

Requêtes des clients en conflit

Les demandes contradictoires des clients, telles que deux clients qui écrivent sur la même clé, sont résolues sur une base de « derniers-victoires ».

La chronologie de l'évaluation « derniers-victoires » repose sur la date à laquelle le système StorageGRID termine une demande donnée et non sur la date à laquelle les clients S3 commencent une opération.

Valeurs de cohérence

La cohérence assure un équilibre entre la disponibilité des objets et la cohérence de ces objets entre plusieurs 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 nouveaux objets. Tout GET suivant un PUT réussi sera en mesure de lire les données nouvellement écrites. Les écrasements d'objets existants, les mises à jour de métadonnées et les suppressions sont cohérents. La propagation des écrasements ne prend généralement que quelques secondes ou minutes, mais peut prendre jusqu'à 15 jours.

Si vous souhaitez effectuer des opérations d'objet de manière différente, vous pouvez :

- Spécifiez une cohérence pour chaque godet.
- Spécifiez une cohérence pour Chaque opération d'API.
- Modifiez la cohérence par défaut à l'échelle 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` (recherchez **constencyLevel**).

Valeurs de cohérence

La cohérence affecte la façon dont les métadonnées utilisées par StorageGRID pour suivre les objets sont réparties entre les nœuds, et donc la disponibilité des objets pour les requêtes client.

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

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

- **Strong-site** : garantit la cohérence lecture après écriture pour toutes les demandes client au sein d'un site.
- **Read-After-New-write**: (Par défaut) fournit une cohérence lecture-après-écriture pour les nouveaux objets et une cohérence éventuelle pour les mises à jour d'objets. Offre une haute disponibilité et une protection des données garanties. Recommandé dans la plupart des cas.
- **Disponible** : assure la cohérence finale pour les nouveaux objets et les mises à jour d'objets. Pour les compartiments S3, utilisez uniquement si nécessaire (par exemple, pour un compartiment qui contient des valeurs de journal rarement lues ou pour les opérations HEAD ou GET sur des clés qui n'existent pas). Non pris en charge pour les compartiments FabricPool S3.

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 recherche tout d'abord l'objet à partir d'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 « Read-after-New-write » mais que l'objet n'existe pas, la recherche d'objet atteint toujours une cohérence équivalente au comportement pour les opérations de type Strong-global. Cette cohérence exigeant la disponibilité de plusieurs copies des métadonnées d'objet sur chaque site, vous pouvez recevoir un nombre élevé d'erreurs de serveur interne 500 si deux nœuds de stockage ou plus sur le même site sont indisponibles.

À moins que vous ayez besoin de garanties de cohérence similaires à Amazon S3, vous pouvez empêcher 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 finale. Cette opération n'a pas abouti pour accroître la cohérence. Il n'est donc pas nécessaire que plusieurs copies des métadonnées de l'objet soient disponibles.

Indiquez la cohérence du fonctionnement de l'API

Pour définir la cohérence d'une opération d'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 « site fort » 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.

Spécifie la cohérence du compartiment

Pour définir la cohérence du compartiment, vous pouvez utiliser la requête StorageGRID"PRÉSERVER la cohérence du godet". Vous pouvez également "modifier la cohérence d'un compartiment"utiliser le Gestionnaire de locataires.

Lorsque vous définissez la cohérence d'un godet, tenez compte des points suivants :

- La cohérence d'un compartiment détermine la cohérence utilisée pour les opérations S3 exécutées sur les objets du compartiment ou sur la configuration du compartiment. Cela n'affecte pas les opérations du compartiment lui-même.
- La cohérence d'une opération d'API individuelle remplace la cohérence du compartiment.
- En général, les compartiments doivent utiliser la cohérence par défaut, « Read-after-New-write ». Si les demandes ne fonctionnent pas correctement, modifiez le comportement du client d'application si possible. Ou configurez le client de manière à spécifier la cohérence pour chaque requête d'API. Réglez la cohérence au niveau du godet uniquement en dernier recours.

[[comment les contrôles-cohérence-et-règles-ILM-interagissent]]Comment la cohérence et les règles ILM interagissent pour protéger les données

La cohérence et les règles ILM de votre choix affectent la protection des objets. Ces paramètres peuvent interagir.

Par exemple, la cohérence utilisée lorsqu'un objet est stocké affecte le placement initial des métadonnées d'objet, tandis que le comportement d'ingestion sélectionné pour la règle ILM affecte le placement initial des copies d'objet. Comme StorageGRID requiert l'accès aux métadonnées et aux données d'un objet pour répondre aux demandes des clients, le choix de niveaux de protection correspondants pour la cohérence et le comportement d'ingestion permet une meilleure protection initiale des données et des réponses système plus prévisibles.

Les éléments suivants "options d'ingestion" sont disponibles pour les règles ILM :

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

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

Équilibré
: StorageGRID tente de faire toutes les copies spécifiées dans la règle ILM à l'entrée ; si cela n'est pas possible, des copies intermédiaires sont effectuées et le client est renvoyé avec succès. Les copies spécifiées dans la règle ILM sont effectuées lorsque cela est possible.

Exemple d'interaction entre la règle de cohérence et la règle ILM

Supposons que vous disposez d'un grid 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. Utiliser un comportement d'ingestion strict.
- **Cohérence** : fort-global (les métadonnées d'objet sont immédiatement distribuées à tous les sites).

Lorsqu'un client stocke un objet dans la grille, StorageGRID effectue à la fois des copies d'objet et distribue les métadonnées aux deux sites avant de rétablir la réussite du client.

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

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

Si vous avez utilisé la même règle ILM et la même cohérence site forte, le client peut recevoir un message de réussite après la réplication des données de l'objet vers le site distant, mais avant la distribution des métadonnées de l'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 de temps après l'ingestion, les métadonnées d'objet sont perdues. Impossible de récupérer l'objet.

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

Gestion des versions d'objet

Vous pouvez définir l'état de gestion des versions d'un compartiment si vous souhaitez conserver plusieurs versions de chaque objet. L'activation de la gestion des versions pour un compartiment vous protège contre la suppression accidentelle d'objets et vous permet de récupérer et de restaurer des versions antérieures d'un objet.

Le système StorageGRID implémente la gestion des versions avec prise en charge de la plupart des fonctionnalités et avec certaines limites. StorageGRID prend en charge jusqu'à 10,000 versions de chaque objet.

Le contrôle de version d'objets peut être associé à la gestion du cycle de vie des informations (ILM) d'StorageGRID ou à la configuration du cycle de vie des compartiments S3. Vous devez explicitement activer la gestion des versions pour chaque compartiment. Lorsque la gestion des versions est activée pour un compartiment, un ID de version est attribué à chaque objet ajouté au compartiment, qui est généré par le système StorageGRID.

La suppression de l'authentification multifacteur (MFA) n'est pas prise en charge.

> ⓘ Le contrôle de version ne peut être activé que pour les compartiments créés avec StorageGRID version 10.3 ou ultérieure.

ILM et gestion des versions

Les règles ILM sont appliquées à chaque version d'un objet. Un processus d'analyse ILM analyse en continu tous les objets, puis les évalue à nouveau en fonction de la règle ILM actuelle. Toute modification apportée aux règles ILM est appliquée à tous les objets précédemment ingérées. Ceci inclut les versions préalablement ingérées si la gestion des versions est activée. L'analyse ILM applique les modifications de l'ILM aux objets précédemment ingérées.

Pour les objets S3 dans les compartiments avec gestion des versions, la prise en charge de la gestion des versions vous permet de créer des règles ILM qui utilisent « Noncurrent Time » comme heure de référence (sélectionnez **Yes** pour la question « Apply this rule to Older object versions only? » "Étape 1 de l'assistant de création de règles ILM"(Appliquer cette règle aux versions d'objets plus anciennes uniquement ?) dans la section). Lorsqu'un objet est mis à jour, ses versions précédentes deviennent non actuelles. L'utilisation d'un filtre « Noncurrent Time » vous permet de créer des stratégies 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 partitionné, l'heure qui n'est pas à jour pour la version d'origine de l'objet correspond à la création du téléchargement partitionné pour la nouvelle version, et non à la fin du téléchargement partitionné. Dans des cas limités, l'heure non actuelle de la version d'origine peut être des heures ou des jours plus tôt que l'heure de la version actuelle.

Informations associées

- "Suppression d'objets avec version S3"
- "Règles et règles ILM pour les objets avec version S3 (exemple 4)".

Utilisez l'API REST S3 pour configurer le verrouillage objet S3

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

Activation du verrouillage objet S3 pour un compartiment

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

Le verrouillage objet S3 est un paramètre permanent qui ne peut être activé que lorsque vous créez un compartiment. Une fois un compartiment créé, vous ne pouvez ni ajouter ni désactiver le verrouillage objet S3.

Pour activer le verrouillage objet S3 pour un compartiment, utilisez l'une des méthodes suivantes :

- Créez le compartiment à l'aide du Gestionnaire des locataires. Voir "Créer un compartiment S3".
- Créez le compartiment à l'aide d'une demande CreateBucket avec l' `x-amz-bucket-object-lock-enabled`en-tête de la demande. Voir "Opérations sur les compartiments".

Le verrouillage objet S3 requiert la gestion des versions des compartiments, qui est automatiquement activée lors de la création du compartiment. Vous ne pouvez pas suspendre la gestion des versions pour le compartiment. Voir "Gestion des versions d'objet".

Paramètres de conservation par défaut d'un compartiment

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

Mode de rétention par défaut

- En mode CONFORMITÉ :
  - L'objet ne peut pas être supprimé tant que sa date de conservation jusqu'à n'est pas atteinte.
  - La date de conservation de l'objet peut être augmentée, mais elle ne peut pas être réduite.
  - La date de conservation de l'objet jusqu'à ne peut pas être supprimée tant que cette date n'est pas atteinte.

- En mode GOUVERNANCE :
  - Les utilisateurs disposant de l' `s3:BypassGovernanceRetention`autorisation peuvent utiliser l' `x-amz-bypass-governance-retention: true`en-tête de la demande pour contourner les paramètres de rétention.
  - Ces utilisateurs peuvent supprimer une version d'objet avant d'atteindre sa date de conservation jusqu'à.
  - Ces utilisateurs peuvent augmenter, diminuer ou supprimer la date de conservation d'un objet.

Période de conservation par défaut

Une période de conservation par défaut peut être spécifiée en années ou en jours pour chaque compartiment.

Comment définir la conservation par défaut d'un compartiment

Pour définir la rétention par défaut d'un compartiment, utilisez l'une des méthodes suivantes :

- Gérez les paramètres de compartiment depuis le gestionnaire de locataires. Voir "Créer un compartiment S3" et "Mettre à jour la conservation par défaut du verrouillage d'objet S3".
- Exécutez une demande PutObjectLockConfiguration pour que le compartiment indique le mode par défaut et le nombre de jours ou d'années par défaut.

PutObjectLockConfiguration

La demande PutObjectLockConfiguration vous permet de définir et de modifier le mode de rétention par défaut et la période de rétention par défaut pour un compartiment pour lequel S3 Object Lock est activé. Vous pouvez également supprimer les paramètres de conservation par défaut configurés précédemment.

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

Si la période de conservation par défaut est modifiée après l'ingestion d'une version d'objet, la conservation à la date de la version de l'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 root, pour effectuer cette opération.

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

Exemple de demande

Cet exemple active le verrouillage objet S3 pour un compartiment 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 conservation par défaut d'un compartiment

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

- Affichez le compartiment dans le gestionnaire de locataires. Voir "Afficher les compartiments S3".
- Émettre une demande GetObjectLockConfiguration.

GetObjectLockConfiguration

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

Lorsque de nouvelles versions d'objet sont ingérées dans le compartiment, le mode de conservation 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 jusqu'à si `x-amz-object-lock-retain-until-date` n'est pas spécifiée.

Vous devez disposer de l' `s3:GetBucketObjectLockConfiguration`autorisation, ou être root, pour effectuer 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 compartiment lorsque le verrouillage objet S3 est activé peut contenir une combinaison d'objets avec ou sans paramètres de conservation du verrouillage objet S3.

Les paramètres de conservation au niveau objet sont spécifiés à l'aide de l'API REST S3. Les paramètres de conservation d'un objet remplacent les paramètres de conservation par défaut du compartiment.

Vous pouvez spécifier les paramètres suivants pour chaque objet :

- **Mode de conservation** : CONFORMITÉ ou GOUVERNANCE.
- **Conserver-jusqu'à-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 jusqu'à est dans le futur, l'objet peut être récupéré,

mais il ne peut pas être modifié ou supprimé. La date de conservation jusqu'à peut être augmentée, mais cette date ne peut pas être réduite ou supprimée.

  - En mode GOUVERNANCE, les utilisateurs disposant d'une autorisation spéciale peuvent contourner le paramètre conserver jusqu'à la date. Ils peuvent supprimer une version d'objet avant la fin de sa période de conservation. Ils peuvent également augmenter, diminuer ou même supprimer la date de conservation jusqu'à.

- **Mise en garde légale** : l'application d'une mise en garde légale à une version d'objet verrouille immédiatement cet objet. Par exemple, vous devrez peut-être mettre une obligation légale sur un objet lié à une enquête ou à un litige juridique. Une obligation 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 jusqu'à. Si une version d'objet est en attente 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 compartiment, émettez une "PutObject", "Objet de copie"ou "CreateMultipartUpload" une demande.

Vous pouvez utiliser les éléments suivants :

- `x-amz-object-lock-mode`, Qui peut être CONFORMITÉ ou GOUVERNANCE (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 conserver jusqu'à la date doit être au format `2020-08-10T21:46:00Z`. Les secondes fractionnaires sont autorisées, mais seuls 3 chiffres après la virgule sont conservés (précision des millisecondes). Les autres formats ISO 8601 ne sont pas autorisés.
  - La date de conservation doit être ultérieure.
- `x-amz-object-lock-legal-hold`

  Si la conservation légale est ACTIVÉE (sensible à la casse), l'objet est placé sous une obligation légale. Si la mise en attente légale est désactivée, aucune mise en attente légale n'est mise. Toute autre valeur entraîne une erreur 400 Bad Request (InvalidArgument).

Si vous utilisez l'un de ces en-têtes de demande, 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 nécessaire pour CopyObject ou CreateMultipartUpload.
- Si S3 Object Lock n'est pas activé dans le compartiment et qu'un `x-amz-object-lock-*` 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 faire correspondre le comportement AWS. Cependant, lors de l'ingestion d'un objet dans un compartiment lorsque le verrouillage objet S3 est activé, StorageGRID effectue toujours une entrée à double validation.
- Une réponse ultérieure à la version GET ou HeadObject inclura les en-têtes `x-amz-object-lock-mode`, , `x-amz-object-lock-retain-until-date` et `x-amz-object-lock-legal-hold`, s'il est

configuré et si l'expéditeur de la demande a les autorisations correctes `s3:Get*`.

Vous pouvez utiliser la `s3:object-lock-remaining-retention-days` clé de condition de règle pour limiter les périodes de conservation minimale et maximale autorisée 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 conservation d'une version d'objet existante, vous pouvez effectuer les opérations de sous-ressource d'objet suivantes :

- `PutObjectLegalHold`

  Si la nouvelle valeur de conservation légale est ACTIVÉE, l'objet est placé sous une mise en attente légale. Si la valeur de retenue légale est OFF, la suspension légale est levée.

- `PutObjectRetention`
  - La valeur du mode peut être CONFORMITÉ ou GOUVERNANCE (sensible à la casse).
  - La valeur conserver jusqu'à la date doit être au format `2020-08-10T21:46:00Z`. Les secondes fractionnaires sont autorisées, mais seuls 3 chiffres après la virgule sont conservés (précision des millisecondes). Les autres formats ISO 8601 ne sont pas autorisés.
  - Si une version d'objet possède une date de conservation existante, vous pouvez uniquement l'augmenter. La nouvelle valeur doit être future.

Comment utiliser le mode GOUVERNANCE

Les utilisateurs disposant de cette `s3:BypassGovernanceRetention` autorisation peuvent contourner les paramètres de rétention actifs d'un objet qui utilise le mode de GOUVERNANCE. Toutes les opérations de SUPPRESSION ou de PutObjectRetention doivent inclure l' `x-amz-bypass-governance-retention:true`en-tête de la demande. Ces utilisateurs peuvent effectuer les opérations supplémentaires suivantes :

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

  Impossible de supprimer les objets qui sont en attente légale. La mise en attente légale doit être désactivée.

- Exécutez des opérations PutObjectRetention qui changent le mode d'une version d'objet de GOUVERNANCE à CONFORMITÉ avant que la période de conservation de l'objet ne soit écoulée.

  Le passage du mode DE CONFORMITÉ À LA GOUVERNANCE n'est jamais autorisé.

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

Informations associées

- "Gestion des objets avec le verrouillage d'objets S3"
- "Utilisez le verrouillage d'objet S3 pour conserver les objets"
- "Guide de l'utilisateur d'Amazon simple Storage Service : verrouillage d'objets"

Création de la configuration du cycle de vie S3

Vous pouvez créer une configuration de cycle de vie S3 afin de contrôler la suppression d'objets spécifiques du système StorageGRID.

L'exemple simple de cette section illustre la façon dont une configuration du cycle de vie S3 peut contrôler la suppression de certains objets (expirés) dans des compartiments S3 spécifiques. L'exemple de cette section est fourni à titre d'illustration uniquement. Pour plus d'informations sur la création de configurations de cycle de vie S3, reportez-vous à la section "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, mais pas les actions de transition.

La configuration du cycle de vie

La configuration du cycle de vie est un ensemble de règles appliquées aux objets dans des compartiments S3 spécifiques. Chaque règle indique quels objets sont affectés et quand ces objets vont expirer (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 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 partir de l'ingestion de l'objet.
- NonactuelVersionExexpiration : supprimez un objet lorsque le nombre de jours spécifié est atteint, à partir de quand l'objet est devenu non courant.
- Filtre (préfixe, étiquette)
- État
- ID

Chaque objet respecte les paramètres de conservation du cycle de vie d'un compartiment S3 ou une règle ILM. Lorsqu'un cycle de vie d'un compartiment S3 est configuré, les actions d'expiration du cycle de vie remplacent la règle 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 des compartiments utilisent les paramètres de conservation de la règle 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 règle ILM ne sont pas utilisés et les versions d'objet sont conservées indéfiniment. Voir "Exemples de priorités pour le cycle de vie des compartiments S3 et les règles ILM".

Par conséquent, il est possible de supprimer un objet de la grille, même si les instructions de placement d'une règle ILM s'appliquent toujours à l'objet. Il est également possible de conserver un objet dans la grille même après l'expiration des instructions de placement ILM de l'objet. Pour plus de détails, voir "Fonctionnement de ILM tout au long de la vie d'un objet".

> La configuration du cycle de vie des compartiments avec des compartiments dont le verrouillage objet S3 est activé, mais la configuration du cycle de vie des compartiments n'est pas prise en charge pour les compartiments conformes.

StorageGRID prend en charge les opérations suivantes des compartiments pour gérer les configurations du cycle de vie :

- DeleteBuckeLifecycle

- GetBucketLifecycleConfiguration
- PutBucketLifecycleConfiguration

Créer une configuration de cycle de vie

Comme première étape de la création de la configuration du cycle de vie, vous créez un fichier JSON qui inclut 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 `key2` valeur de `tag2`. Le `Expiration` paramètre indique 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 indique que les objets correspondant au filtre expireront 100 jours après leur ingestion.

   > Les règles spécifiant un nombre de jours sont relatives à l'ingestion de l'objet. Si la date actuelle dépasse la date d'ingestion et le nombre de jours, certains objets peuvent être supprimés du compartiment dès que la configuration de cycle de vie est appliquée.

3. La règle 3 s'applique uniquement aux objets qui correspondent au préfixe `category3/`. Le `Expiration` paramètre spécifie que toute version non actuelle des objets correspondants expirera 50 jours après qu'ils ne soient plus à jour.

```
{
    "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"
        }
    ]
}
```

Appliquez 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 demande 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 été correctement appliquée au compartiment, exécutez une demande GetBucketLifecycleConfiguration. Par exemple :

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

Une réponse réussie répertorie la configuration de 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 de 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 mise en correspondance.

> Étant donné que le cycle de vie d'un compartiment remplace ILM, la `expiry-date` date affichée est la date réelle à laquelle l'objet sera supprimé. Pour plus de détails, voir "Méthode de détermination de la conservation des objets".

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

```
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 (01 oct 2020) et qu'il correspond à la règle 2 de la configuration de 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 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 compartiments avec gestion des versions, l' `x-amz-expiration`en-tête de réponse s'applique uniquement aux versions actuelles des objets.

Recommandations pour l'implémentation de l'API REST S3

Suivez ces recommandations lors de l'implémentation de l'API REST S3 pour une utilisation avec StorageGRID.

Recommandations pour les têtes à 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 le "disponible" "la cohérence". Par exemple, vous devez utiliser la cohérence « disponible » si votre application se trouve en tête d'emplacement avant de la METTRE EN PLACE.

Sinon, si l'opération HEAD ne trouve pas l'objet, vous risquez de recevoir un nombre élevé d'erreurs de serveur interne 500 si deux nœuds de stockage ou plus sur le 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 "PRÉSERVER la cohérence du godet" requête ou spécifier la cohérence dans l'en-tête de demande pour une opération d'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 du compartiment.

Compartiments créés dans StorageGRID 11.4 ou version antérieure

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

  ```
  mybucket/mydir/f8e3-image3132.jpg
  ```

  Au lieu de ce format :

  ```
  mybucket/f8e3-image3132.jpg
  ```

Compartiments créés dans StorageGRID 11.4 ou version ultérieure

Il n'est pas nécessaire de restreindre les noms de clés d'objet pour répondre aux bonnes pratiques de performances. Dans la plupart des cas, vous pouvez utiliser des valeurs aléatoires pour les quatre premiers caractères des noms de clé d'objet.

À cela s'exception près un workload 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, il est possible de faire varier la première partie du nom de clé tous les mille objets avec une date comme la date. Supposons par exemple qu'un client S3 écrit généralement 2,000 objets/seconde et que la règle de cycle de vie ILM ou compartiment supprime tous les objets au bout de trois jours. Pour réduire l'impact sur les performances, vous pouvez nommer les clés comme suit : `/mybucket/mydir/`*`yyyymmddhhmmss-random_UUID`*`.jpg`

Recommandations pour les « lectures de plage »

Si "option globale pour compresser les objets stockés"est activé, les applications client S3 doivent éviter d'effectuer des opérations GetObject qui spécifient une plage d'octets. 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 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 demandes client peuvent être en attente.

Si vous devez compresser des objets et que votre application client doit utiliser des lectures de plage, augmentez le délai de lecture de l'application.