Comment StorageGRID implémente l'API REST S3

StorageGRID 11.5

NetApp
April 11, 2024

Sommaire

Comment StorageGRID implémente l'API REST S3

Une application client peut utiliser des appels d'API REST S3 pour se connecter à StorageGRID pour créer, supprimer et modifier des compartiments, ainsi que pour stocker et récupérer des objets.

- "Requêtes des clients en conflit"
- "Contrôles de cohérence"
- "Gestion des objets par les règles StorageGRID ILM"
- "Gestion des versions d'objet"
- "Recommandations pour l'implémentation de l'API REST S3"

Requêtes des clients en conflit

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

Le calendrier de l'évaluation « derniers-victoires » est basé sur le moment où le système StorageGRID remplit une demande donnée et non sur le moment où les clients S3 commencent une opération.

Contrôles de cohérence

Les contrôles de cohérence assurent la reprise entre la disponibilité des objets et la cohérence de ces objets sur différents nœuds et sites de stockage, 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.

Pour effectuer des opérations d'objet à un niveau de cohérence différent, vous pouvez définir un contrôle de cohérence pour chaque compartiment ou pour chaque opération d'API.

Contrôles de cohérence

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

Vous pouvez définir le contrôle de cohérence pour une opération de compartiment ou API sur l'une des valeurs suivantes :

Contrôle de cohérence	Description
tous	Tous les nœuds reçoivent les données immédiatement, sinon la requête échoue.

Contrôle de cohérence	Description
forte croissance mondiale	Garantit une cohérence de lecture après écriture pour toutes les demandes client sur tous les sites.
site fort	Garantit la cohérence de lecture après écriture pour toutes les demandes client dans un site.
lecture-après-nouvelle-écriture	(Valeur par défaut) assure la cohérence en lecture après écriture des nouveaux objets et la cohérence des mises à jour des objets. Offre une haute disponibilité et une protection des données garanties. Correspondance avec les garanties de cohérence Amazon S3. **Remarque :** si votre application utilise des demandes HEAD sur des objets qui n'existent pas, vous pouvez recevoir un nombre élevé de 500 erreurs de serveur interne si un ou plusieurs nœuds de stockage ne sont pas disponibles. Pour éviter ces erreurs, définissez le contrôle de cohérence sur « disponible », sauf si vous avez besoin de garanties de cohérence similaires à Amazon S3.
Disponible (cohérence possible pour les opérations DE TÊTE)	Se comporte de la même manière que le niveau de cohérence « entre la date et la nouvelle écriture », mais n'assure qu'une cohérence éventuelle pour les opérations DE TÊTE. Niveaux de disponibilité supérieurs à ceux de la « nouvelle écriture » en cas d'indisponibilité des nœuds de stockage Diffère des garanties de cohérence Amazon S3 pour les opérations HEAD uniquement.

Utilisation des contrôles de cohérence « en cas de nouvelle écriture » et « disponibles »

Lorsqu'une OPÉRATION EN TÊTE ou GET utilise le contrôle de cohérence « en cas de nouvelle écriture » ou QU'une opération GET utilise le contrôle de cohérence « disponible », 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 au niveau de cohérence suivant jusqu'à ce qu'elle atteigne le niveau de cohérence le plus élevé, « tous », qui nécessite la disponibilité de toutes les copies des métadonnées de l'objet.

Si une OPÉRATION HEAD ou GET utilise le contrôle de cohérence « read-after-New-write » mais que l'objet n'existe pas, la recherche d'objet atteint toujours le niveau de cohérence « All ». Comme ce niveau de cohérence requiert la disponibilité de toutes les copies des métadonnées de l'objet, vous pouvez recevoir un nombre élevé de 500 erreurs de serveur interne si un ou plusieurs nœuds de stockage sont indisponibles.

Sauf si vous avez besoin de garanties de cohérence similaires à Amazon S3, vous pouvez empêcher ces erreurs pour les opérations HEAD en définissant le contrôle de cohérence sur « disponible ». Lorsqu'une

opération DE TÊTE utilise le contrôle de cohérence « disponible », StorageGRID n'offre qu'une cohérence éventuelle. Il ne réessaie pas l'échec d'une opération tant qu'elle n'atteint pas le niveau de cohérence « tous ». Il n'est donc pas nécessaire que toutes les copies des métadonnées de l'objet soient disponibles.

Spécification du contrôle de cohérence pour une opération d'API

Pour définir le contrôle de cohérence pour une opération API individuelle, les contrôles de cohérence doivent être pris en charge pour l'opération, et vous devez spécifier le contrôle de cohérence dans l'en-tête de la demande. Cet exemple définit le contrôle de cohérence sur "site de segmentation" pour une opération D'OBTENTION d'objet.

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

> Vous devez utiliser le même contrôle de cohérence pour les opérations PLACER l'objet et OBTENIR l'objet.

Spécification du contrôle de cohérence pour un compartiment

Pour définir le contrôle de cohérence du compartiment, vous pouvez utiliser la demande de cohérence StorageGRID PUT bucket et la demande DE cohérence GET bucket. Vous pouvez également utiliser le Gestionnaire de locataires ou l'API de gestion des locataires.

Lors du réglage des commandes de cohérence pour un godet, tenez compte des éléments suivants :

- La configuration du contrôle de cohérence d'un compartiment détermine quel contrôle de cohérence est utilisé pour les opérations S3 effectuées sur les objets dans le compartiment ou sur la configuration du compartiment. Cela n'affecte pas les opérations du compartiment lui-même.
- Le contrôle de cohérence d'une opération API individuelle remplace le contrôle de cohérence du compartiment.
- En général, les compartiments doivent utiliser le contrôle de cohérence par défaut, « en cas d'écriture ultérieure ». Si les demandes ne fonctionnent pas correctement, modifiez le comportement du client de l'application si possible. Ou configurez le client afin de spécifier le contrôle de cohérence pour chaque requête d'API. Réglez le contrôle de cohérence au niveau du godet uniquement en dernier recours.

Interaction des contrôles de cohérence et des règles ILM pour la protection des données

Le contrôle de cohérence et la règle ILM de votre choix affectent la protection des objets. Ces paramètres peuvent interagir.

Par exemple, le contrôle de cohérence utilisé 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. Étant donné que StorageGRID nécessite l'accès aux métadonnées d'un objet et à ses données pour répondre aux demandes client, la sélection de niveaux de protection correspondant au niveau de cohérence et au comportement d'ingestion permet d'améliorer la protection des données initiale et de mieux prévoir les réponses du système.

Les comportements d'ingestion suivants sont disponibles pour les règles ILM :

- **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 faire toutes les copies spécifiées dans la règle ILM à l'entrée; si ce n'est pas possible, des copies intermédiaires sont faites et le succès est renvoyé au client. Les copies spécifiées dans la règle ILM sont effectuées lorsque cela est possible.
- **Double commit**: StorageGRID effectue immédiatement des copies intermédiaires de l'objet et retourne le succès au client. Les copies spécifiées dans la règle ILM sont effectuées lorsque cela est possible.

> Avant de sélectionner le comportement d'entrée d'une règle ILM, lisez la description complète de ces paramètres dans les instructions de gestion des objets avec la gestion du cycle de vie des informations.

Exemple d'interaction du contrôle de cohérence et de la règle ILM

Supposons que vous disposez d'une grille à deux sites avec la règle ILM suivante et le paramètre de niveau de cohérence suivant :

- **Règle ILM** : créez deux copies d'objet, une sur le site local et une sur un site distant. Le comportement d'entrée strict est sélectionné.
- **Niveau de cohérence**: "Sept-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 utilisez à la place la même règle ILM et le niveau de cohérence « sept-site », 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 ne soient distribuées sur ce site. 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. L'objet ne peut pas être récupéré.

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

Informations associées

"Gestion des objets avec ILM"

"DEMANDE de cohérence des compartiments"

"PUT Bucket Consistency demandée"

Gestion des objets par les règles StorageGRID ILM

L'administrateur du grid crée des règles de gestion du cycle de vie des informations pour gérer les données d'objet ingérées sur le système StorageGRID à partir des applications

client de l'API REST S3. Ces règles sont ensuite ajoutées à la règle ILM pour déterminer la façon dont et l'emplacement de stockage des données d'objet au fil du temps.

Les paramètres ILM déterminent les aspects suivants d'un objet :

- **Géographie**

  L'emplacement des données d'un objet, dans le système StorageGRID (pool de stockage) ou dans un pool de stockage cloud.

- **Grade de stockage**

  Type de stockage utilisé pour stocker les données d'objet : par exemple, Flash ou disque rotatif.

- * Protection contre les pertes*

  Le nombre de copies effectuées et les types de copies créées : réplication, code d'effacement, ou les deux.

- * Rétention*

  Évolution au fil du temps de la gestion des données d'un objet, de leur emplacement de stockage et de leur protection contre la perte.

- * Protection pendant l'ingestion*

  Méthode de protection des données d'objet lors de l'ingestion : placement synchrone (avec options équilibrées ou strictes pour le comportement d'ingestion) ou copies intermédiaires (avec l'option de double validation).

Les règles ILM peuvent filtrer et sélectionner des objets. Pour les objets ingérées à l'aide du protocole S3, les règles ILM peuvent filtrer les objets en fonction des métadonnées suivantes :

- Compte de locataire
- Nom du compartiment
- Temps d'ingestion
- Clé
- Heure du dernier accès

> Par défaut, les mises à jour de l'heure du dernier accès sont désactivées pour tous les compartiments S3. Si votre système StorageGRID inclut une règle ILM utilisant l'option heure du dernier accès, vous devez activer les mises à jour de l'heure du dernier accès pour les compartiments S3 spécifiés dans cette règle. Vous pouvez activer les dernières mises à jour des temps d'accès à l'aide de la demande D'heure du dernier accès AU compartiment, de la case à cocher **S3** > **seaux** > **configurer le dernier accès** dans le Gestionnaire de locataires ou à l'aide de l'API de gestion des locataires. Lors de l'activation des mises à jour du dernier accès, notez que les performances du StorageGRID peuvent être réduites, notamment dans les systèmes dotés d'objets de petite taille.

- Contrainte d'emplacement
- Taille de l'objet

- Métadonnées utilisateur
- Balise d'objet

Pour plus d'informations sur ILM, reportez-vous aux instructions de gestion des objets avec des informations relatives à la gestion du cycle de vie.

Informations associées

"Utilisez un compte de locataire"

"Gestion des objets avec ILM"

"DEMANDE de temps de dernier accès au compartiment"

Gestion des versions d'objet

Vous pouvez utiliser la gestion des versions pour conserver plusieurs versions d'un objet, ce qui vous protège contre la suppression accidentelle d'objets et vous permet d'extraire et de restaurer les 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 1,000 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 activer explicitement la gestion des versions pour chaque compartiment pour activer cette fonctionnalité. Chaque objet du compartiment est associé à un ID de version, 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 des compartiments activés pour la gestion des versions, la prise en charge du contrôle de version vous permet de créer des règles ILM qui utilisent l'heure actuelle non sélectionnée comme heure de référence. Lorsqu'un objet est mis à jour, ses versions précédentes deviennent non actuelles. L'utilisation d'un filtre de temps non actuel vous permet de créer des règles qui réduisent l'impact sur le stockage des versions précédentes d'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.

Pour obtenir des informations sur la gestion du cycle de vie des objets avec la gestion du cycle de vie des informations, consultez les instructions de gestion des objets avec version S3.

Informations associées

"Gestion des objets avec ILM"

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 sur un chemin où vous ne vous attendez pas à ce que l'objet existe réellement, vous devez utiliser le contrôle de cohérence « disponible ». Par exemple, vous devez utiliser le contrôle de cohérence « disponible » si votre application dirige un emplacement avant DE LE PLACER.

Sinon, si l'opération HEAD ne trouve pas l'objet, vous pouvez recevoir un nombre élevé de 500 erreurs de serveur interne si un ou plusieurs nœuds de stockage ne sont pas disponibles.

Vous pouvez définir le contrôle de cohérence « disponible » pour chaque compartiment à l'aide de la demande DE cohérence PUT bucket, ou spécifier le contrôle de cohérence dans l'en-tête de demande pour une opération API individuelle.

Recommandations pour les clés d'objet

Pour les compartiments créés dans StorageGRID 11.4 ou version ultérieure, il n'est plus nécessaire de limiter les noms de clés d'objet afin de respecter les meilleures pratiques en matière de performances. Par exemple, vous pouvez maintenant utiliser des valeurs aléatoires pour les quatre premiers caractères des noms de clés d'objet.

Pour les compartiments créés dans les versions antérieures à StorageGRID 11.4, suivez les recommandations suivantes pour les noms de clés d'objet :

- Vous ne devez pas utiliser de valeurs aléatoires comme les quatre premiers caractères des clés d'objet. Cela contraste avec l'ancienne recommandation AWS pour les préfixes de clés. Au lieu de cela, vous devez utiliser des préfixes non aléatoires et non uniques, tels que `image`.
- Si vous suivez l'ancienne recommandation AWS pour utiliser des caractères aléatoires et uniques dans les préfixes de clés, vous devez préfixer les clés d'objet avec 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
```

Recommandations pour « plages de lectures »

Si l'option **Compress emmagased Objects** est sélectionnée (**Configuration** > **Grid Options**), les applications clientes S3 doivent éviter d'effectuer des opérations GET Object qui indiquent une plage d'octets à renvoyer. Ces opérations de « lecture à plage » sont inefficaces, car StorageGRID doit décompresser efficacement les objets pour accéder aux octets demandés. LES opérations GET Object qui demandent une petite plage d'octets provenant d'un objet très volumineux sont particulièrement inefficaces. Par exemple, il est très 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.

Informations associées

"Contrôles de cohérence"

"PUT Bucket Consistency demandée"

"Administrer StorageGRID"