Cómo StorageGRID implementa la API REST de S3

StorageGRID software

NetApp
December 03, 2025

Tabla de contenidos

Cómo StorageGRID implementa la API REST de S3

Solicitudes de clientes conflictivas

Las solicitudes de clientes conflictivas, como dos clientes que escriben en la misma clave, se resuelven según el criterio del "último que gana".

El momento de la evaluación de "últimas victorias" se basa en cuándo el sistema StorageGRID completa una solicitud determinada y no en cuándo los clientes S3 comienzan una operación.

Valores de consistencia

La consistencia proporciona un equilibrio entre la disponibilidad de los objetos y la consistencia de esos objetos en diferentes nodos de almacenamiento y sitios. Puede cambiar la consistencia según lo requiera su aplicación.

De forma predeterminada, StorageGRID garantiza la consistencia de lectura tras escritura para los objetos recién creados. Cualquier operación GET posterior a una operación PUT completada con éxito podrá leer los datos recién escritos. Las sobrescrituras de objetos existentes, las actualizaciones de metadatos y las eliminaciones son eventualmente consistentes. Las sobrescrituras generalmente tardan segundos o minutos en propagarse, pero pueden demorar hasta 15 días.

Si desea realizar operaciones de objetos con una consistencia diferente, puede:

- Especificar una consistencia paracada cubo .
- Especificar una consistencia paracada operación de API .
- Cambie la consistencia predeterminada de toda la cuadrícula realizando una de las siguientes tareas:
  - En el Administrador de cuadrícula, vaya a **CONFIGURACIÓN** > **Sistema** > **Configuración de almacenamiento** > **Consistencia predeterminada**.
  - .

> Un cambio en la consistencia de toda la cuadrícula se aplica solo a los depósitos creados después de que se modificó la configuración. Para determinar los detalles de un cambio, consulte el registro de auditoría ubicado en `/var/local/log` (buscar **consistencyLevel**).

Valores de consistencia

La consistencia afecta cómo se distribuyen los metadatos que StorageGRID utiliza para rastrear objetos entre los nodos y, por lo tanto, la disponibilidad de los objetos para las solicitudes de los clientes.

Puede establecer la consistencia de un depósito o una operación de API en uno de los siguientes valores:

- **Todos**: Todos los nodos reciben los datos inmediatamente o la solicitud fallará.
- **Strong-global**: garantiza la consistencia de lectura después de escritura para todas las solicitudes de clientes en todos los sitios.

- **Sitio fuerte**: garantiza la consistencia de lectura después de escritura para todas las solicitudes de clientes dentro de un sitio.
- **Lectura después de nueva escritura**: (predeterminado) proporciona consistencia de lectura después de escritura para objetos nuevos y consistencia eventual para actualizaciones de objetos. Ofrece alta disponibilidad y garantías de protección de datos. Recomendado para la mayoría de los casos.
- **Disponible**: Proporciona consistencia eventual tanto para objetos nuevos como para actualizaciones de objetos. Para los buckets S3, úselo solo cuando sea necesario (por ejemplo, para un bucket que contiene valores de registro que rara vez se leen, o para operaciones HEAD o GET en claves que no existen). No compatible con depósitos S3 FabricPool .

Utilice la consistencia "Lectura después de nueva escritura" y "Disponible"

Cuando una operación HEAD o GET utiliza la consistencia "Lectura después de nueva escritura", StorageGRID realiza la búsqueda en varios pasos, de la siguiente manera:

- Primero busca el objeto utilizando una consistencia baja.
- Si esa búsqueda falla, se repite la búsqueda en el siguiente valor de consistencia hasta que alcanza una consistencia equivalente al comportamiento de strong-global.

Si una operación HEAD o GET utiliza la consistencia "Lectura después de nueva escritura" pero el objeto no existe, la búsqueda del objeto siempre alcanzará una consistencia equivalente al comportamiento para una globalización fuerte. Debido a que esta consistencia requiere que varias copias de los metadatos del objeto estén disponibles en cada sitio, puede recibir una gran cantidad de errores internos del servidor 500 si dos o más nodos de almacenamiento en el mismo sitio no están disponibles.

A menos que necesite garantías de consistencia similares a Amazon S3, puede evitar estos errores para las operaciones HEAD y GET configurando la consistencia en "Disponible". Cuando una operación HEAD o GET utiliza la consistencia "Disponible", StorageGRID solo proporciona consistencia eventual. No vuelve a intentar una operación fallida con una consistencia creciente, por lo que no requiere que haya varias copias disponibles de los metadatos del objeto.

Especificar la consistencia para la operación de la API

Para establecer la consistencia de una operación de API individual, los valores de consistencia deben ser compatibles con la operación y debe especificar la consistencia en el encabezado de la solicitud. Este ejemplo establece la consistencia en "Strong-site" para una operación GetObject.

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

Debe utilizar la misma consistencia para las operaciones PutObject y GetObject.

Especificar la consistencia para el depósito

Para establecer la consistencia del bucket, puede utilizar StorageGRID"Consistencia del depósito PUT" pedido. O puedes"cambiar la consistencia de un cubo" del administrador de inquilinos.

Al configurar la consistencia de un depósito, tenga en cuenta lo siguiente:

- La configuración de la consistencia de un depósito determina qué consistencia se utiliza para las operaciones S3 realizadas en los objetos del depósito o en la configuración del depósito. No afecta las operaciones en el bucket en sí.
- La consistencia de una operación de API individual anula la consistencia del depósito.
- En general, los buckets deben usar la consistencia predeterminada: "Lectura después de nueva escritura". Si las solicitudes no funcionan correctamente, cambie el comportamiento del cliente de la aplicación si es posible. O bien, configure el cliente para especificar la consistencia para cada solicitud de API. Establezca la consistencia a nivel de depósito sólo como último recurso.

Cómo interactúan los controles de consistencia y las reglas ILM para afectar la protección de datos

Tanto su elección de consistencia como su regla ILM afectan cómo se protegen los objetos. Estas configuraciones pueden interactuar.

Por ejemplo, la consistencia utilizada cuando se almacena un objeto afecta la ubicación inicial de los metadatos del objeto, mientras que el comportamiento de ingesta seleccionado para la regla ILM afecta la ubicación inicial de las copias del objeto. Debido a que StorageGRID requiere acceso tanto a los metadatos de un objeto como a sus datos para cumplir con las solicitudes de los clientes, seleccionar niveles de protección coincidentes para la consistencia y el comportamiento de ingesta puede brindar una mejor protección de datos inicial y respuestas del sistema más predecibles.

La siguiente"opciones de ingesta" Están disponibles para las reglas ILM:

Compromiso dual
: StorageGRID realiza inmediatamente copias provisionales del objeto y devuelve el éxito al cliente. Cuando sea posible se realizarán las copias especificadas en la regla ILM.

Estricto
: Se deben realizar todas las copias especificadas en la regla ILM antes de devolver el éxito al cliente.

Equilibrado
: StorageGRID intenta hacer todas las copias especificadas en la regla ILM durante la ingesta; si esto no es posible, se hacen copias provisionales y se devuelve el resultado exitoso al cliente. Las copias especificadas en la regla ILM se realizan cuando es posible.

Ejemplo de cómo la consistencia y la regla ILM pueden interactuar

Supongamos que tiene una cuadrícula de dos sitios con la siguiente regla ILM y la siguiente consistencia:

- **Regla ILM**: Crea dos copias de objetos, una en el sitio local y otra en un sitio remoto. Utilice el comportamiento de ingesta estricto.
- **Consistencia**: Fuerte-global (los metadatos del objeto se distribuyen inmediatamente a todos los sitios).

Cuando un cliente almacena un objeto en la red, StorageGRID realiza copias de los objetos y distribuye metadatos a ambos sitios antes de devolver el éxito al cliente.

El objeto está completamente protegido contra pérdida en el momento del mensaje de ingesta exitosa. Por ejemplo, si el sitio local se pierde poco después de la ingesta, aún existen copias de los datos del objeto y de los metadatos del objeto en el sitio remoto. El objeto es completamente recuperable.

Si, en cambio, utilizara la misma regla ILM y la consistencia del sitio fuerte, el cliente podría recibir un mensaje de éxito después de que los datos del objeto se repliquen en el sitio remoto pero antes de que los metadatos del objeto se distribuyan allí. En este caso, el nivel de protección de los metadatos del objeto no coincide con el nivel de protección de los datos del objeto. Si el sitio local se pierde poco después de la ingesta, se pierden los metadatos del objeto. No se puede recuperar el objeto.

La interrelación entre la consistencia y las reglas ILM puede ser compleja. Comuníquese con NetApp si necesita ayuda.

Control de versiones de objetos

Puede establecer el estado de control de versiones de un depósito si desea conservar varias versiones de cada objeto. Habilitar el control de versiones de un bucket puede ayudar a proteger contra la eliminación accidental de objetos y le permite recuperar y restaurar versiones anteriores de un objeto.

El sistema StorageGRID implementa control de versiones con soporte para la mayoría de las funciones y con algunas limitaciones. StorageGRID admite hasta 10 000 versiones de cada objeto.

El control de versiones de objetos se puede combinar con la gestión del ciclo de vida de la información (ILM) de StorageGRID o con la configuración del ciclo de vida del bucket S3. Debe habilitar explícitamente el control de versiones para cada depósito. Cuando se habilita el control de versiones para un depósito, a cada objeto agregado al depósito se le asigna un ID de versión, que es generado por el sistema StorageGRID .

No se admite el uso de MFA (autenticación multifactor).

> ⓘ El control de versiones solo se puede habilitar en depósitos creados con StorageGRID versión 10.3 o posterior.

ILM y control de versiones

Las políticas ILM se aplican a cada versión de un objeto. Un proceso de escaneo ILM escanea continuamente todos los objetos y los reevalúa en función de la política ILM actual. Cualquier cambio que realice en las políticas de ILM se aplicará a todos los objetos ingeridos previamente. Esto incluye versiones ingeridas previamente si el control de versiones está habilitado. El escaneo ILM aplica nuevos cambios ILM a objetos ingeridos previamente.

Para los objetos S3 en depósitos habilitados para control de versiones, la compatibilidad con control de versiones le permite crear reglas ILM que usan "Hora no actual" como Hora de referencia (seleccione **Sí** para la pregunta "¿Aplicar esta regla solo a versiones de objetos anteriores?" en"Paso 1 del asistente para crear una regla de ILM"). Cuando se actualiza un objeto, sus versiones anteriores dejan de ser actuales. El uso de un filtro de "Tiempo no actual" le permite crear políticas que reducen el impacto del almacenamiento de versiones anteriores de los objetos.

> ⓘ Cuando se carga una nueva versión de un objeto mediante una operación de carga multiparte, el tiempo no actual de la versión original del objeto refleja cuándo se creó la carga multiparte para la nueva versión, no cuándo se completó la carga multiparte. En casos limitados, la hora no actual de la versión original puede ser horas o días anterior a la hora de la versión actual.

Información relacionada

- "Cómo se eliminan los objetos versionados de S3"

- "Reglas y políticas de ILM para objetos versionados de S3 (Ejemplo 4)" .

Utilice la API REST de S3 para configurar el bloqueo de objetos de S3

Si la configuración global de Bloqueo de objetos S3 está habilitada para su sistema StorageGRID , puede crear depósitos con el Bloqueo de objetos S3 habilitado. Puede especificar la retención predeterminada para cada depósito o configuraciones de retención para cada versión de objeto.

Cómo habilitar el bloqueo de objetos S3 para un bucket

Si la configuración global de Bloqueo de objetos S3 está habilitada para su sistema StorageGRID , puede habilitar opcionalmente el Bloqueo de objetos S3 cuando cree cada depósito.

El bloqueo de objetos S3 es una configuración permanente que solo se puede habilitar cuando se crea un depósito. No es posible agregar ni deshabilitar el bloqueo de objetos S3 después de crear un depósito.

Para habilitar el bloqueo de objetos S3 para un depósito, utilice cualquiera de estos métodos:

- Cree el depósito mediante el Administrador de inquilinos. Ver "Crear un depósito S3" .
- Cree el depósito mediante una solicitud CreateBucket con el `x-amz-bucket-object-lock-enabled` encabezado de solicitud. Ver "Operaciones en buckets" .

El bloqueo de objetos S3 requiere control de versiones del depósito, que se habilita automáticamente cuando se crea el depósito. No se puede suspender el control de versiones del depósito. Ver "Control de versiones de objetos" .

Configuración de retención predeterminada para un depósito

Cuando el bloqueo de objetos S3 está habilitado para un depósito, puede habilitar opcionalmente la retención predeterminada para el depósito y especificar un modo de retención predeterminado y un período de retención predeterminado.

Modo de retención predeterminado

- En modo CUMPLIMIENTO:
  - El objeto no se puede eliminar hasta que se alcance su fecha de conservación.
  - La fecha de conservación del objeto se puede aumentar, pero no se puede disminuir.
  - La fecha de retención del objeto no se puede eliminar hasta que se alcance esa fecha.
- En modo GOBERNANZA:
  - Usuarios con la `s3:BypassGovernanceRetention` El permiso puede utilizar el `x-amz-bypass-governance-retention: true` encabezado de solicitud para omitir la configuración de retención.
  - Estos usuarios pueden eliminar una versión de un objeto antes de que se alcance su fecha de conservación.
  - Estos usuarios pueden aumentar, disminuir o eliminar la fecha de conservación de un objeto.

Período de retención predeterminado

Cada depósito puede tener un período de retención predeterminado especificado en años o días.

Cómo configurar la retención predeterminada para un depósito

Para establecer la retención predeterminada para un depósito, utilice cualquiera de estos métodos:

- Administre la configuración del depósito desde el Administrador de inquilinos. Ver"Crear un bucket S3" y"Actualizar la retención predeterminada de bloqueo de objetos S3" .
- Emita una solicitud PutObjectLockConfiguration para el depósito para especificar el modo predeterminado y el número predeterminado de días o años.

Configuración de bloqueo de objeto de colocación

La solicitud PutObjectLockConfiguration le permite establecer y modificar el modo de retención predeterminado y el período de retención predeterminado para un depósito que tiene habilitado el bloqueo de objetos S3. También puede eliminar las configuraciones de retención predeterminadas configuradas previamente.

Cuando se incorporan nuevas versiones de objetos al depósito, se aplica el modo de retención predeterminado si `x-amz-object-lock-mode` y `x-amz-object-lock-retain-until-date` no están especificados El período de retención predeterminado se utiliza para calcular la fecha de retención hasta si `x-amz-object-lock-retain-until-date` No está especificado.

Si el período de retención predeterminado se modifica después de la ingesta de una versión de objeto, la fecha de retención de la versión del objeto permanece igual y no se vuelve a calcular utilizando el nuevo período de retención predeterminado.

Debes tener el `s3:PutBucketObjectLockConfiguration` permiso, o ser la cuenta root, para completar esta operación.

El `Content-MD5` El encabezado de la solicitud debe especificarse en la solicitud PUT.

Ejemplo de solicitud

Este ejemplo habilita el bloqueo de objetos S3 para un depósito y establece el modo de retención predeterminado en CUMPLIMIENTO y el período de retención predeterminado en 6 años.

```
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>
```

Cómo determinar la retención predeterminada para un bucket

Para determinar si S3 Object Lock está habilitado para un bucket y ver el modo de retención predeterminado y el período de retención, utilice cualquiera de estos métodos:

- Ver el depósito en el Administrador de inquilinos. Ver "Ver depósitos S3" .
- Emite una solicitud GetObjectLockConfiguration.

Obtener configuración de bloqueo de objeto

La solicitud GetObjectLockConfiguration le permite determinar si S3 Object Lock está habilitado para un depósito y, si está habilitado, ver si hay un modo de retención predeterminado y un período de retención configurados para el depósito.

Cuando se incorporan nuevas versiones de objetos al depósito, se aplica el modo de retención predeterminado si `x-amz-object-lock-mode` No está especificado. El período de retención predeterminado se utiliza para calcular la fecha de retención hasta si `x-amz-object-lock-retain-until-date` No está especificado.

Debes tener el `s3:GetBucketObjectLockConfiguration` permiso, o ser la cuenta root, para completar esta operación.

Ejemplo de solicitud

```
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
```

Ejemplo de respuesta

```
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>
```

Cómo especificar la configuración de retención para un objeto

Un depósito con el bloqueo de objetos S3 habilitado puede contener una combinación de objetos con y sin configuraciones de retención de bloqueo de objetos S3.

Las configuraciones de retención a nivel de objeto se especifican mediante la API REST de S3. La configuración de retención de un objeto anula cualquier configuración de retención predeterminada para el depósito.

Puede especificar las siguientes configuraciones para cada objeto:

- **Modo de retención**: CUMPLIMIENTO o GOBERNANZA.
- **Retain-until-date**: una fecha que especifica durante cuánto tiempo StorageGRID debe conservar la versión del objeto.

    - En el modo CUMPLIMIENTO, si la fecha de retención hasta está en el futuro, el objeto se puede recuperar, pero no se puede modificar ni eliminar. La fecha de conservación hasta se puede aumentar, pero esta fecha no se puede disminuir ni eliminar.
    - En el modo GOBERNANZA, los usuarios con permiso especial pueden omitir la configuración de conservar hasta la fecha. Pueden eliminar una versión de un objeto antes de que transcurra su período de retención. También pueden aumentar, disminuir o incluso eliminar la fecha de conservación.
- **Retención legal**: al aplicar una retención legal a una versión de un objeto, se bloquea inmediatamente ese objeto. Por ejemplo, es posible que necesites colocar una retención legal en un objeto que esté relacionado con una investigación o una disputa legal. Una retención legal no tiene fecha de vencimiento, sino que permanece vigente hasta que se elimina explícitamente.

  La configuración de retención legal de un objeto es independiente del modo de retención y de la fecha de retención. Si una versión de un objeto está bajo retención legal, nadie puede eliminar esa versión.

Para especificar la configuración de bloqueo de objetos S3 al agregar una versión de objeto a un depósito, emita un"PonerObjeto" ,"Copiar objeto" , o"Crear carga de varias partes" pedido.

Puedes utilizar lo siguiente:

- `x-amz-object-lock-mode`, que puede ser CUMPLIMIENTO o GOBERNANZA (sensible a mayúsculas y minúsculas).

  > Si lo especifica `x-amz-object-lock-mode` , también debe especificar `x-amz-object-lock-retain-until-date` .

- `x-amz-object-lock-retain-until-date`
    - El valor de conservación hasta la fecha debe tener el formato `2020-08-10T21:46:00Z` . Se permiten fracciones de segundo, pero solo se conservan 3 dígitos decimales (precisión de milisegundos). No se permiten otros formatos ISO 8601.
    - La fecha de conservación debe ser en el futuro.
- `x-amz-object-lock-legal-hold`

  Si la retención legal está activada (distingue entre mayúsculas y minúsculas), el objeto se coloca bajo una retención legal. Si la retención legal está desactivada, no se aplica ninguna retención legal. Cualquier otro valor generará un error 400 Solicitud incorrecta (argumento inválido).

Si utiliza alguno de estos encabezados de solicitud, tenga en cuenta estas restricciones:

- El `Content-MD5` El encabezado de solicitud es obligatorio si lo hay `x-amz-object-lock-*` El encabezado de solicitud está presente en la solicitud PutObject. `Content-MD5` no es necesario para CopyObject o CreateMultipartUpload.
- Si el depósito no tiene habilitado el bloqueo de objetos S3 y un `x-amz-object-lock-*` Si el encabezado de solicitud está presente, se devuelve un error 400 Solicitud incorrecta (InvalidRequest).
- La solicitud PutObject admite el uso de `x-amz-storage-class: REDUCED_REDUNDANCY` para que coincida con el comportamiento de AWS. Sin embargo, cuando se ingiere un objeto en un bucket con el bloqueo de objetos S3 habilitado, StorageGRID siempre realizará una ingesta de confirmación dual.
- Una respuesta de versión GET o HeadObject posterior incluirá los encabezados `x-amz-object-lock-mode` , `x-amz-object-lock-retain-until-date` , y `x-amz-object-lock-legal-hold` , si está

configurado y si el remitente de la solicitud tiene la información correcta `s3:Get*` permisos.

Puedes utilizar el `s3:object-lock-remaining-retention-days` Clave de condición de política para limitar los períodos de retención mínimos y máximos permitidos para sus objetos.

Cómo actualizar la configuración de retención de un objeto

Si necesita actualizar la configuración de retención o retención legal para una versión de objeto existente, puede realizar las siguientes operaciones de subrecurso de objeto:

- `PutObjectLegalHold`

  Si el nuevo valor de retención legal está activado, el objeto se coloca bajo una retención legal. Si el valor de retención legal está DESACTIVADO, se levanta la retención legal.

- `PutObjectRetention`
  - El valor del modo puede ser CUMPLIMIENTO o GOBERNANZA (distingue entre mayúsculas y minúsculas).
  - El valor de conservación hasta la fecha debe tener el formato `2020-08-10T21:46:00Z` . Se permiten fracciones de segundo, pero solo se conservan 3 dígitos decimales (precisión de milisegundos). No se permiten otros formatos ISO 8601.
  - Si una versión de objeto tiene una fecha de conservación existente, solo puedes aumentarla. El nuevo valor debe estar en el futuro.

Cómo utilizar el modo GOBERNANZA

Los usuarios que tengan la `s3:BypassGovernanceRetention` El permiso puede omitir la configuración de retención activa de un objeto que utiliza el modo GOBERNANZA. Cualquier operación DELETE o PutObjectRetention debe incluir el `x-amz-bypass-governance-retention:true` encabezado de solicitud. Estos usuarios pueden realizar estas operaciones adicionales:

- Realice las operaciones DeleteObject o DeleteObjects para eliminar una versión de un objeto antes de que transcurra su período de retención.

  Los objetos que están bajo retención legal no se pueden eliminar. La retención legal debe estar DESACTIVADA.

- Realice operaciones PutObjectRetention que cambien el modo de la versión de un objeto de GOBERNANZA a CUMPLIMIENTO antes de que transcurra el período de retención del objeto.

  Nunca se permite cambiar el modo de CUMPLIMIENTO a GOBERNANZA.

- Realice operaciones PutObjectRetention para aumentar, disminuir o eliminar el período de retención de una versión de objeto.

Información relacionada

- "Administrar objetos con S3 Object Lock"
- "Utilice S3 Object Lock para retener objetos"
- "Guía del usuario de Amazon Simple Storage Service: Bloqueo de objetos"

Crear la configuración del ciclo de vida de S3

Puede crear una configuración de ciclo de vida S3 para controlar cuándo se eliminan objetos específicos del sistema StorageGRID .

El ejemplo simple de esta sección ilustra cómo una configuración del ciclo de vida de S3 puede controlar cuándo se eliminan (caducan) determinados objetos de depósitos S3 específicos. El ejemplo de esta sección es sólo para fines ilustrativos. Para obtener detalles completos sobre la creación de configuraciones del ciclo de vida de S3, consulte "Guía del usuario de Amazon Simple Storage Service: Gestión del ciclo de vida de los objetos" . Tenga en cuenta que StorageGRID solo admite acciones de vencimiento; no admite acciones de transición.

¿Qué es la configuración del ciclo de vida?

Una configuración de ciclo de vida es un conjunto de reglas que se aplican a los objetos en depósitos S3 específicos. Cada regla especifica qué objetos se ven afectados y cuándo expirarán esos objetos (en una fecha específica o después de una cierta cantidad de días).

StorageGRID admite hasta 1000 reglas de ciclo de vida en una configuración de ciclo de vida. Cada regla puede incluir los siguientes elementos XML:

- Vencimiento: elimina un objeto cuando se alcanza una fecha específica o cuando se alcanza una cantidad específica de días, a partir del momento en que se ingirió el objeto.
- NoncurrentVersionExpiration: elimina un objeto cuando se alcanza una cantidad específica de días, a partir del momento en que el objeto dejó de ser actual.
- Filtro (Prefijo, Etiqueta)
- Estado
- IDENTIFICACIÓN

Cada objeto sigue la configuración de retención de un ciclo de vida de un bucket S3 o de una política ILM. Cuando se configura un ciclo de vida de un depósito S3, las acciones de vencimiento del ciclo de vida anulan la política de ILM para los objetos que coinciden con el filtro del ciclo de vida del depósito. Los objetos que no coinciden con el filtro del ciclo de vida del depósito utilizan la configuración de retención de la política de ILM. Si un objeto coincide con un filtro de ciclo de vida de depósito y no se especifican explícitamente acciones de vencimiento, no se utilizan las configuraciones de retención de la política ILM y se implica que las versiones del objeto se conservan para siempre. Ver "Prioridades de ejemplo para el ciclo de vida del depósito S3 y la política de ILM" .

Como resultado, es posible que se elimine un objeto de la cuadrícula aunque las instrucciones de ubicación de una regla ILM todavía se apliquen al objeto. O bien, un objeto podría conservarse en la cuadrícula incluso después de que hayan transcurrido las instrucciones de ubicación de ILM para el objeto. Para obtener más información, consulte "Cómo funciona ILM a lo largo de la vida de un objeto" .

> La configuración del ciclo de vida del bucket se puede usar con buckets que tienen habilitado el bloqueo de objetos S3, pero la configuración del ciclo de vida del bucket no es compatible con buckets compatibles heredados.

StorageGRID admite el uso de las siguientes operaciones de depósito para administrar las configuraciones del ciclo de vida:

- Eliminar ciclo de vida del cubo

- Obtener configuración del ciclo de vida del cubo
- Configuración del ciclo de vida de PutBucket

Crear configuración de ciclo de vida

Como primer paso para crear una configuración de ciclo de vida, crea un archivo JSON que incluye una o más reglas. Por ejemplo, este archivo JSON incluye tres reglas, como sigue:

1. La regla 1 se aplica únicamente a los objetos que coinciden con el prefijo `category1` / y que tienen una `key2` valor de `tag2` . El `Expiration` El parámetro especifica que los objetos que coincidan con el filtro caducarán a la medianoche del 22 de agosto de 2020.
2. La regla 2 se aplica únicamente a los objetos que coinciden con el prefijo `category2` /. El `Expiration` El parámetro especifica que los objetos que coinciden con el filtro caducarán 100 días después de su ingesta.

   > Las reglas que especifican un número de días son relativas al momento en que se ingirió el objeto. Si la fecha actual excede la fecha de ingesta más la cantidad de días, es posible que se eliminen algunos objetos del depósito tan pronto como se aplique la configuración del ciclo de vida.

3. La regla 3 se aplica únicamente a los objetos que coinciden con el prefijo `category3` /. El `Expiration` El parámetro especifica que cualquier versión no actual de los objetos coincidentes expirará 50 días después de que dejen de estar actuales.

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

Aplicar la configuración del ciclo de vida al depósito

Después de haber creado el archivo de configuración del ciclo de vida, debe aplicarlo a un depósito emitiendo una solicitud PutBucketLifecycleConfiguration.

Esta solicitud aplica la configuración del ciclo de vida en el archivo de ejemplo a los objetos en un depósito llamado `testbucket` .

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

Para validar que una configuración de ciclo de vida se aplicó correctamente al depósito, emita una solicitud GetBucketLifecycleConfiguration. Por ejemplo:

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

Una respuesta exitosa enumera la configuración del ciclo de vida que acaba de aplicar.

Validar que la expiración del ciclo de vida del bucket se aplique al objeto

Puede determinar si una regla de expiración en la configuración del ciclo de vida se aplica a un objeto específico al emitir una solicitud PutObject, HeadObject o GetObject. Si se aplica una regla, la respuesta incluye una `Expiration` parámetro que indica cuándo expira el objeto y qué regla de expiración coincidió.

> Debido a que el ciclo de vida del bucket anula ILM, `expiry-date` Se muestra la fecha real en la que se eliminará el objeto. Para obtener más información, consulte "Cómo se determina la retención de objetos" .

Por ejemplo, esta solicitud PutObject se emitió el 22 de junio de 2020 y coloca un objeto en el `testbucket` balde.

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

La respuesta de éxito indica que el objeto caducará en 100 días (1 de octubre de 2020) y que coincidió con la Regla 2 de la configuración del ciclo de vida.

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

Por ejemplo, esta solicitud HeadObject se utilizó para obtener metadatos para el mismo objeto en el depósito testbucket.

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

La respuesta de éxito incluye los metadatos del objeto e indica que el objeto caducará en 100 días y que coincidió con la Regla 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": {}
}
```

Para los depósitos con control de versiones habilitado, el `x-amz-expiration` El encabezado de respuesta se aplica solo a las versiones actuales de los objetos.

Recomendaciones para implementar la API REST de S3

Debe seguir estas recomendaciones al implementar la API REST S3 para su uso con StorageGRID.

Recomendaciones para HEADs a objetos inexistentes

Si su aplicación verifica rutinariamente si existe un objeto en una ruta en la que no espera que el objeto exista realmente, debe usar la opción "Disponible"."consistencia" . Por ejemplo, debe utilizar la consistencia "Disponible" si su aplicación encabeza una ubicación antes de PUT en ella.

De lo contrario, si la operación HEAD no encuentra el objeto, es posible que reciba una gran cantidad de errores internos del servidor 500 si dos o más nodos de almacenamiento en el mismo sitio no están disponibles o si no se puede acceder a un sitio remoto.

Puede establecer la consistencia "Disponible" para cada depósito utilizando el"Consistencia del depósito PUT" solicitud, o puede especificar la consistencia en el encabezado de la solicitud para una operación de API individual.

Recomendaciones para claves de objeto

Siga estas recomendaciones para los nombres de claves de objeto, según el momento en que se creó el depósito por primera vez.

Cubos creados en StorageGRID 11.4 o anterior

- No utilice valores aleatorios como los primeros cuatro caracteres de las claves de objeto. Esto contrasta con la recomendación anterior de AWS para prefijos clave. En su lugar, utilice prefijos no aleatorios ni únicos, como `image` .
- Si sigue la recomendación anterior de AWS de utilizar caracteres aleatorios y únicos en los prefijos de clave, anteponga a las claves de objeto un nombre de directorio. Es decir, utiliza este formato:

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

  En lugar de este formato:

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

Cubos creados en StorageGRID 11.4 o posterior

No es necesario restringir los nombres de claves de objeto para cumplir con las mejores prácticas de rendimiento. En la mayoría de los casos, puede utilizar valores aleatorios para los primeros cuatro caracteres de los nombres de claves de objeto.

> Una excepción a esto es una carga de trabajo S3 que elimina continuamente todos los objetos después de un corto período de tiempo. Para minimizar el impacto en el rendimiento de este caso de uso, varíe una parte inicial del nombre de la clave cada varios miles de objetos con algo como la fecha. Por ejemplo, supongamos que un cliente S3 normalmente escribe 2000 objetos por segundo y la política de ciclo de vida del ILM o del bucket elimina todos los objetos después de tres días. Para minimizar el impacto en el rendimiento, puedes nombrar las claves utilizando un patrón como este: `/mybucket/mydir/`*`yyyymmddhhmmss-random_UUID`*`.jpg`

Recomendaciones para "lecturas de rango"

Si el"Opción global para comprimir objetos almacenados" Si está habilitado, las aplicaciones cliente S3 deben evitar realizar operaciones GetObject que especifiquen un rango de bytes que se devolverán. Estas operaciones de "lectura de rango" son ineficientes porque StorageGRID debe descomprimir efectivamente los objetos para acceder a los bytes solicitados. Las operaciones GetObject que solicitan un rango pequeño de bytes de un objeto muy grande son especialmente ineficientes; por ejemplo, es ineficiente leer un rango de 10 MB de un objeto comprimido de 50 GB.

Si se leen rangos de objetos comprimidos, las solicitudes del cliente pueden expirar.

> Si necesita comprimir objetos y su aplicación cliente debe utilizar lecturas de rango, aumente el tiempo de espera de lectura de la aplicación.