



Cómo StorageGRID implementa la API DE REST de S3

StorageGRID

NetApp
November 04, 2025

Tabla de contenidos

Cómo StorageGRID implementa la API DE REST de S3	1
Solicitudes de clientes en conflicto	1
Controles de consistencia	1
Controles de consistencia	1
Utilizar controles de coherencia «reescritura tras escritura» y «disponibles»	2
Especifique el control de coherencia para la operación de API	2
Especifique el control de coherencia para el bloque	2
Cómo interactúan los controles de coherencia y las reglas de ILM para afectar a la protección de datos	3
Ejemplo de cómo puede interactuar el control de consistencia y la regla de ILM	3
Cómo gestionan las reglas de ILM de StorageGRID los objetos	4
Control de versiones de objetos	5
ILM y versiones	5
Use la API REST DE S3 para configurar el bloqueo de objetos de S3	6
Cómo habilitar S3 Object Lock para un bucket	6
Configuración de retención predeterminada para un bloque	6
Cómo establecer la retención predeterminada para un depósito	7
Cómo determinar la retención predeterminada de un depósito	8
Cómo especificar la configuración de retención para un objeto	9
Cómo actualizar la configuración de retención de un objeto	11
Cómo utilizar el modo de GOBIERNO	11
Cree una configuración del ciclo de vida de S3	12
Qué es la configuración del ciclo de vida	12
Cree la configuración del ciclo de vida	13
Aplicar la configuración del ciclo de vida al bloque	15
Validar que la caducidad del ciclo de vida del bloque se aplica al objeto	15
Recomendaciones para implementar la API REST de S3	16
Recomendaciones para las cabezas a los objetos no existentes	16
Recomendaciones para las claves de objeto	17
Recomendaciones para «lecturas de rango»	17

Cómo StorageGRID implementa la API DE REST de S3

Solicitudes de clientes en conflicto

Las solicitudes de clientes en conflicto, como dos clientes que escriben en la misma clave, se resuelven en función de las "últimas victorias".

El plazo para la evaluación de "logros más recientes" se basa en cuándo el sistema StorageGRID completa una solicitud determinada, y no en cuándo los clientes de S3 comienzan una operación.

Controles de consistencia

Los controles de consistencia proporcionan un equilibrio entre la disponibilidad de los objetos y la coherencia de dichos objetos en diferentes nodos de almacenamiento y sitios, según lo requiera la aplicación.

De forma predeterminada, StorageGRID garantiza la coherencia de lectura tras escritura de los objetos recién creados. Cualquier OBTENER después de un PUESTO completado correctamente podrá leer los datos recién escritos. Las sobrescrituras de objetos existentes, actualizaciones de metadatos y eliminaciones son coherentes en la actualidad. Por lo general, las sobrescrituras tardan segundos o minutos en propagarse, pero pueden tardar hasta 15 días.

Si desea realizar operaciones de objetos en un nivel de coherencia diferente, puede especificar un control de coherencia para cada bloque o para cada operación de API.

Controles de consistencia

El control de consistencia afecta a cómo los metadatos que utiliza StorageGRID para realizar un seguimiento de los objetos se distribuyen entre los nodos y, por lo tanto, la disponibilidad de los objetos para las solicitudes del cliente.

Puede establecer el control de coherencia de un bloque o una operación API en uno de los siguientes valores:

- **Todos:** Todos los nodos reciben los datos inmediatamente, o la solicitud fallará.
- **Strong-global:** Garantiza la coherencia de lectura tras escritura para todas las solicitudes de clientes en todos los sitios.
- **Strong-site:** Garantiza la coherencia de lectura después de escritura para todas las solicitudes de cliente dentro de un sitio.
- **Read-after-new-write:** (Predeterminado) proporciona consistencia de lectura después de escritura para nuevos objetos y consistencia eventual para actualizaciones de objetos. Ofrece garantías de alta disponibilidad y protección de datos. Recomendado para la mayoría de los casos.
- **Disponible:** Proporciona consistencia eventual tanto para nuevos objetos como para actualizaciones de objetos. Para los cubos S3, utilice solo según sea necesario (por ejemplo, para un depósito que contiene valores de registro que rara vez se leen, o para operaciones HEAD u GET en claves que no existen). No se admite para bloques de FabricPool S3.

Utilizar controles de coherencia «reescritura tras escritura» y «disponibles»

Cuando una OPERACIÓN HEAD u GET utiliza el control de consistencia «relativamente una vez que una nueva escritura», StorageGRID realiza la búsqueda en varios pasos, como se indica a continuación:

- Primero busca el objeto con una baja consistencia.
- Si esa búsqueda falla, repite la búsqueda en el siguiente nivel de coherencia hasta que alcanza un nivel de coherencia equivalente al comportamiento de la búsqueda global.

Si una operación HEAD u GET utiliza el control de consistencia "READ-after-new-write", pero el objeto no existe, la búsqueda de objetos siempre alcanzará un nivel de consistencia equivalente al comportamiento de strong-global. Debido a que este nivel de coherencia requiere que haya varias copias de los metadatos del objeto disponibles en cada sitio, puede recibir un número elevado de errores del servidor interno 500 si no están disponibles dos o más nodos de almacenamiento en el mismo sitio.

A menos que necesite garantías de coherencia similares a las de Amazon S3, podrá evitar estos errores en LAS operaciones DE CABEZA y OBTENER al establecer el control de coherencia en "Available". Cuando un CABEZAL o UNA operación DE OBTENER utiliza el control de consistencia "disponible", StorageGRID sólo proporciona consistencia eventual. No vuelve a intentar una operación fallida en los niveles de coherencia crecientes, por lo que no es necesario que haya disponibles varias copias de los metadatos de objeto.

Especifique el control de coherencia para la operación de API

Para configurar el control de coherencia para una operación de API individual, deben ser compatibles los controles de coherencia para la operación y debe especificar el control de coherencia en el encabezado de la solicitud. En este ejemplo se establece el control de coherencia en «punto de referencia» para una operación GET Object.

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



Debe usar el mismo control de coherencia para las operaciones PUT Object y GET Object.

Especifique el control de coherencia para el bloque

Para establecer el control de consistencia para el bloque, puede utilizar StorageGRID la solicitud de consistencia PUT Bucket y LA solicitud DE consistencia GET Bucket. También puede usar el Administrador de inquilinos o la API de gestión de inquilinos.

Cuando configure los controles de coherencia para un cucharón, tenga en cuenta lo siguiente:

- La configuración del control de coherencia para un bloque determina el control de coherencia que se utiliza para las operaciones de S3 realizadas en los objetos del bloque o en la configuración de bloques. No afecta a las operaciones del propio cucharón.
- El control de coherencia de una operación API individual anula el control de coherencia del bloque.
- En general, los depósitos deben usar el control de consistencia predeterminado, «read-after-new-write». Si las solicitudes no funcionan correctamente, cambie el comportamiento del cliente de aplicación si es

posible. O bien, configure el cliente para especificar el control de consistencia de cada solicitud API. Establecer el control de consistencia a nivel de cucharón únicamente como último recurso.

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

Tanto la elección del control de coherencia como la regla de ILM afectan a la forma en que se protegen los objetos. Estos ajustes pueden interactuar.

Por ejemplo, el control de consistencia usado cuando se almacena un objeto afecta a la colocación inicial de los metadatos de objetos, mientras que el comportamiento de procesamiento seleccionado para la regla de ILM afecta a la colocación inicial de las copias de objetos. Dado que StorageGRID requiere acceso tanto a los metadatos de un objeto como a sus datos para cumplir con las solicitudes de los clientes, seleccionar los niveles de protección correspondientes para el nivel de coherencia y el comportamiento de ingesta puede proporcionar una mejor protección de datos inicial y respuestas más predecibles del sistema.

Para las reglas de ILM hay disponibles los siguientes comportamientos de consumo:

- **Confirmación doble:** StorageGRID realiza inmediatamente copias provisionales del objeto y devuelve el éxito al cliente. Las copias especificadas en la regla ILM se realizan cuando es posible.
- **Estricto:** Todas las copias especificadas en la regla ILM deben hacerse antes de que el éxito se devuelva al cliente.
- **Balanceado:** StorageGRID intenta hacer todas las copias especificadas en la regla ILM en la ingesta; si esto no es posible, se hacen copias provisionales y se devuelve éxito al cliente. Las copias especificadas en la regla ILM se realizan cuando es posible.



Antes de seleccionar el comportamiento de procesamiento de una regla de ILM, lea la descripción completa de estos ajustes en las instrucciones para gestionar objetos con gestión del ciclo de vida de la información.

Ejemplo de cómo puede interactuar el control de consistencia y la regla de ILM

Suponga que tiene una cuadrícula de dos sitios con la siguiente regla de ILM y la siguiente configuración de nivel de coherencia:

- **Norma ILM:** Cree dos copias de objetos, una en el sitio local y otra en un sitio remoto. Se ha seleccionado el comportamiento de procesamiento estricto.
- **Nivel de coherencia:** "Strong-global" (los metadatos de objetos se distribuyen inmediatamente a todos los sitios).

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

El objeto está completamente protegido contra la pérdida en el momento del mensaje de procesamiento correcto. Por ejemplo, si el sitio local se pierde poco después del procesamiento, seguirán existiendo copias de los datos del objeto y los metadatos del objeto en el sitio remoto. El objeto se puede recuperar completamente.

Si en su lugar usa la misma regla de ILM y el nivel de consistencia de «otrong-site», es posible que el cliente reciba un mensaje de éxito después de replicar los datos del objeto 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 de objetos no coincide con el nivel de protección de los datos de objetos. Si el sitio local se pierde poco después del procesamiento, se pierden los metadatos del objeto. No se puede recuperar el objeto.

La interrelación entre los niveles de coherencia y las reglas del ILM puede ser compleja. Póngase en contacto con NetApp si necesita ayuda.

Información relacionada

["Gestión de objetos con ILM"](#)

["OBTENGA coherencia de bloques"](#)

["PONGA la consistencia del cucharón"](#)

Cómo gestionan las reglas de ILM de StorageGRID los objetos

El administrador de grid crea reglas de gestión del ciclo de vida de la información (ILM) para gestionar los datos de los objetos que se ingieren en el sistema StorageGRID desde aplicaciones cliente de la API REST S3. A continuación, estas reglas se añaden a la política de ILM para determinar cómo y dónde se almacenan los datos de objetos con el tiempo.

La configuración de ILM determina los siguientes aspectos de un objeto:

- **Geografía**

La ubicación de los datos de un objeto, ya sea en el sistema StorageGRID (pool de almacenamiento) o en un pool de almacenamiento en el cloud.

- **Grado de almacenamiento**

El tipo de almacenamiento utilizado para almacenar datos de objetos, como la tecnología flash o el disco giratorio.

- **Protección contra pérdidas**

Cuántas copias se hacen y los tipos de copias que se crean: Replicación, codificación de borrado o ambos.

- **Retención**

Los cambios se producen a lo largo del tiempo en el modo en que se gestionan los datos de un objeto, dónde se almacenan y cómo se protegen de pérdidas.

- **Protección durante la ingesta**

El método utilizado para proteger los datos de objetos durante el procesamiento: Colocación síncrona (utilizando las opciones equilibradas o estrictas para el comportamiento de ingesta) o creación de copias provisionales (mediante la opción Dual Commit).

Las reglas de ILM pueden filtrar y seleccionar objetos. Para los objetos ingeridos mediante S3, las reglas de ILM pueden filtrar objetos en función de los siguientes metadatos:

- Cuenta de inquilino
- Nombre del bloque

- Tiempo de ingesta
- Clave
- Hora del último acceso



De forma predeterminada, las actualizaciones del último tiempo de acceso se deshabilitan para todos los bloques S3. Si su sistema StorageGRID incluye una regla de ILM que utiliza la opción de hora de último acceso, debe habilitar las actualizaciones a la hora del último acceso para los bloques S3 especificados en esa regla. Utilice la solicitud de hora de último acceso a PUT Bucket, el Gestor de inquilinos (consulte "[Activar o desactivar las actualizaciones de la hora del último acceso](#)") O la API de gestión de inquilinos. Al habilitar las actualizaciones del último tiempo de acceso, tenga en cuenta que el rendimiento de StorageGRID puede reducirse, especialmente en sistemas con objetos pequeños.

- Restricción de ubicación
- Tamaño del objeto
- Metadatos del usuario
- Etiqueta de objeto

Información relacionada

["Usar una cuenta de inquilino"](#)

["Gestión de objetos con ILM"](#)

["PUT Bucket última hora de acceso"](#)

Control de versiones de objetos

Puede utilizar el control de versiones para conservar varias versiones de un objeto, lo que protege contra la eliminación accidental de objetos y le permite recuperar y restaurar versiones anteriores de un objeto.

El sistema StorageGRID implementa versiones con compatibilidad para la mayoría de las funciones y con algunas limitaciones. StorageGRID admite hasta 1,000 versiones de cada objeto.

El control de versiones de objetos puede combinarse con la gestión del ciclo de vida de la información (ILM) de StorageGRID o con la configuración del ciclo de vida de bloques de S3. Debe habilitar explícitamente el control de versiones para cada segmento a fin de activar esta funcionalidad para el bloque. A cada objeto de su bloque se le asigna un ID de versión, que genera el sistema StorageGRID.

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



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

ILM y versiones

Las políticas de ILM se aplican a cada versión de un objeto. Un proceso de análisis de ILM analiza continuamente todos los objetos y los vuelve a evaluar en relación con la política actual de ILM. Todos los cambios realizados en las políticas de ILM se aplican a todos los objetos procesados anteriormente. Esto incluye versiones que se han ingerido previamente si la versión está activada. El análisis de ILM aplica nuevos

cambios de ILM a los objetos procesados previamente.

Para los objetos S3 en bloques con control de versiones, el soporte de control de versiones le permite crear reglas de ILM que usen “Tiempo no corriente” como tiempo de referencia (seleccione **Sí** para la pregunta, “¿Aplicar esta regla solo a versiones de objetos anteriores?”) pulg ["Paso 1 del asistente Crear una regla de ILM"](#)). Cuando se actualiza un objeto, sus versiones anteriores se vuelven no actuales. El uso de un filtro de tiempo no corriente permite crear políticas que reducen el impacto en el almacenamiento de las versiones anteriores de objetos.



Cuando se carga una nueva versión de un objeto mediante una operación de carga de varias partes, la hora no actual de la versión original del objeto se refleja cuando se creó la carga de varias partes para la nueva versión, no cuando se completó la carga de varias partes. En casos limitados, la hora no actual de la versión original puede ser horas o días antes de la hora de la versión actual.

Consulte ["Reglas de ILM y políticas para objetos con versiones de S3 \(ejemplo 4\)"](#).

Use la API REST DE S3 para configurar el bloqueo de objetos de S3

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

Cómo habilitar S3 Object Lock para un bucket

Si la opción de configuración global de bloqueo de objetos S3 se encuentra habilitada para el sistema StorageGRID, también puede habilitar el bloqueo de objetos S3 al crear cada bloque.

S3 Bloqueo de objetos es un ajuste permanente que solo se puede activar cuando se crea un depósito. No puede agregar o deshabilitar S3 Object Lock después de crear un bucket.

Para activar el bloqueo de objetos S3 para un depósito, utilice uno de estos métodos:

- Cree el bloque con el Administrador de arrendatarios. Consulte ["Crear bloque de S3"](#).
- Cree el segmento mediante una solicitud PUT Bucket con el `x-amz-bucket-object-lock-enabled` solicite el encabezado. Consulte ["Operaciones en bloques"](#).

S3 Object Lock requiere el control de versiones de bloque, que se habilita automáticamente cuando se crea el bloque. No puede suspender el control de versiones del depósito. Consulte ["Control de versiones de objetos"](#).

Configuración de retención predeterminada para un bloque

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

Modo de retención predeterminado

- En modo de CUMPLIMIENTO:

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

Período de retención predeterminado

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

Cómo establecer la retención predeterminada para un depósito

Para definir la retención predeterminada de un depósito, utilice uno de estos métodos:

- Gestione la configuración de bloques desde el Gestor de inquilinos. Consulte ["Cree un bloque de S3"](#) y.. ["Actualizar S3 Retención predeterminada de bloqueo de objetos"](#).
- Emita una solicitud de configuración de bloqueo de objeto PUT para que el depósito especifique el modo por defecto y el número por defecto de días o años.

PONER configuración de bloqueo de objeto

La solicitud de configuración PUT Object Lock permite establecer y modificar el modo de retención predeterminado y el período de retención predeterminado para un depósito que tiene S3 Object Lock activado. También es posible eliminar los ajustes de retención predeterminados previamente configurados.

Cuando se ingieren nuevas versiones de objetos en el bloque, se aplica el modo de retención predeterminado si `x-amz-object-lock-mode` y.. `x-amz-object-lock-retain-until-date` no se han especificado. El período de retención predeterminado se utiliza para calcular el valor de retener hasta la fecha if `x-amz-object-lock-retain-until-date` no se ha especificado.

Si el período de retención predeterminado se modifica tras recibir una versión de objeto, la fecha de retención hasta la de la versión del objeto sigue siendo la misma y no se vuelve a calcular con el nuevo período de retención predeterminado.

Debe tener la `s3:PutBucketObjectLockConfiguration` permiso, o be account root, para completar esta operación.

La `Content-MD5` La cabecera de la solicitud se debe especificar 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 DE NORMATIVAS 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 de un depósito

Para determinar si S3 Object Lock está activado para un depósito y para ver el modo de retención y el período de retención predeterminados, utilice uno de estos métodos:

- Ver el depósito en el Gestor de inquilinos. Consulte ["Ver S3 cubos"](#).
- Emita una solicitud de configuración OBTENER bloqueo de objeto.

OBTENER configuración de bloqueo de objeto

La solicitud OBTENER configuración de bloqueo de objeto le permite determinar si el bloqueo de objeto S3 está activado para un depósito y, si está activado, consulte si hay un modo de retención predeterminado y un período de retención configurado para el depósito.

Cuando se ingieren nuevas versiones de objetos en el bloque, se aplica el modo de retención predeterminado si `x-amz-object-lock-mode` no se ha especificado. El período de retención predeterminado se utiliza para calcular el valor de retener hasta la fecha si `x-amz-object-lock-retain-until-date` no se ha especificado.

Debe tener la `s3:GetBucketObjectLockConfiguration` permiso, o `be account 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 bucket con S3 Object Lock habilitado puede contener una combinación de objetos con y sin la configuración de retención de S3 Object Lock.

La configuración de retención en el nivel de objeto se especifica mediante la API DE REST S3. La configuración de retención de un objeto anula cualquier configuración de retención predeterminada del bloque.

Puede especificar los siguientes ajustes para cada objeto:

- **Modo de retención:** Ya sea CUMPLIMIENTO o GOBIERNO.
- **Retain-until-date:** Una fecha que especifica cuánto tiempo la versión del objeto debe ser retenida por StorageGRID.
 - En el modo de CUMPLIMIENTO DE NORMATIVAS, si la fecha de retención hasta la fecha es

posterior, el objeto se puede recuperar, pero no se puede modificar ni eliminar. Se puede aumentar la fecha de retención hasta la fecha, pero esta fecha no se puede reducir ni eliminar.

- En el modo de GOBIERNO, los usuarios con permiso especial pueden omitir la configuración Retener hasta la fecha. Pueden eliminar una versión de objeto antes de que haya transcurrido su período de retención. También pueden aumentar, disminuir o incluso eliminar la fecha de retención hasta la fecha.
- **Retención legal:** La aplicación de una retención legal a una versión de objeto bloquea inmediatamente ese objeto. Por ejemplo, es posible que necesite poner una retención legal en un objeto relacionado con una investigación o una disputa legal. Una retención legal no tiene fecha de vencimiento, pero permanece en su lugar hasta que se elimina explícitamente.

La configuración de conservación legal de un objeto es independiente del modo de retención y la retención hasta la fecha. Si una versión de objeto está bajo una conservación 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 "OBJETO PUT", "PONER objeto: Copiar", o "Inicie la carga de varias partes" solicitud.

Puede utilizar lo siguiente:

- `x-amz-object-lock-mode`, Que puede ser CUMPLIMIENTO o GOBERNANZA (distingue entre mayúsculas y minúsculas).



Si 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 retener hasta la fecha debe tener el formato `2020-08-10T21:46:00Z`. Se permiten segundos fraccionarios, pero sólo se conservan 3 dígitos decimales (precisión de milisegundos). No se permiten otros formatos ISO 8601.
 - La fecha de retención debe ser futura.
- `x-amz-object-lock-legal-hold`

Si la conservación legal está ACTIVADA (distingue entre mayúsculas y minúsculas), el objeto se colocará bajo una retención legal. Si se HA DESACTIVADO la retención legal, no se ha colocado ningún tipo de retención legal. Cualquier otro valor produce un error 400 Bad Request (InvalidArgument).

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

- La `Content-MD5` la cabecera de la solicitud es necesaria si la hay `x-amz-object-lock-*` El encabezado de la solicitud está presente en LA solicitud PUT Object. `Content-MD5` No es necesario PARA PONER objeto: Copiar o iniciar carga de varias partes.
- Si el bloque no tiene habilitado el bloqueo de objetos S3 y un `x-amz-object-lock-*` El encabezado de la solicitud está presente, se devuelve un error de solicitud incorrecta 400 (InvalidRequest).
- La solicitud PUT Object admite el uso de `x-amz-storage-class: REDUCED_REDUNDANCY` Para igualar el comportamiento de AWS. Sin embargo, cuando un objeto se procesa en un bucket con el bloqueo de objetos S3 habilitado, StorageGRID siempre ejecuta un procesamiento de compromiso doble.
- Una respuesta posterior A LA versión GET o HEAD Object 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 el correcto `s3:Get*` permisos.

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

Cómo actualizar la configuración de retención de un objeto

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

- `PUT Object legal-hold`

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

- `PUT Object retention`
 - El valor de modo puede ser CUMPLIMIENTO o GOBIERNO (distingue entre mayúsculas y minúsculas).
 - El valor retener hasta la fecha debe tener el formato `2020-08-10T21:46:00Z`. Se permiten segundos fraccionarios, pero sólo se conservan 3 dígitos decimales (precisión de milisegundos). No se permiten otros formatos ISO 8601.
 - Si una versión de objeto tiene una fecha de retención existente, sólo puede aumentarla. El nuevo valor debe ser el futuro.

Cómo utilizar el modo de GOBIERNO

Los usuarios que tienen el `s3:BypassGovernanceRetention` El permiso puede omitir la configuración de retención activa de un objeto que utiliza el modo de GOBIERNO. Todas las operaciones de retención DELETE o PUT Object deben incluir `x-amz-bypass-governance-retention:true` solicite el encabezado. Estos usuarios pueden realizar las siguientes operaciones adicionales:

- Realice operaciones de SUPRESIÓN DE objetos o SUPRESIÓN DE varios objetos para suprimir una versión de objeto antes de que haya transcurrido su período de retención.

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

- Realice operaciones de retención de objetos PUT que cambian el modo de una versión de objeto de GOBIERNO a CUMPLIMIENTO antes de que haya transcurrido el período de retención del objeto.

Cambiar el modo de CUMPLIMIENTO a GOBIERNO nunca está permitido.

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

Información relacionada

- ["Gestione objetos con S3 Object Lock"](#)
- ["Utilice Bloqueo de objetos S3 para retener objetos"](#)
- ["Guía del usuario de Amazon simple Storage Service: Uso del bloqueo de objetos de S3"](#)

Cree una configuración del ciclo de vida de S3

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

El ejemplo sencillo de esta sección muestra cómo puede controlar una configuración del ciclo de vida de S3 cuando se eliminan ciertos objetos (caducados) de bloques S3 específicos. El ejemplo de esta sección es solo con fines ilustrativos. Para obtener información completa sobre la creación de configuraciones del ciclo de vida de S3, consulte ["Guía para desarrolladores de Amazon simple Storage Service: Gestión del ciclo de vida de los objetos"](#). Tenga en cuenta que StorageGRID solo admite acciones de caducidad, 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 bloques de S3 específicos. Cada regla especifica qué objetos se ven afectados y cuándo caducarán dichos objetos (en una fecha específica o después de un número determinado de días).

StorageGRID admite hasta 1,000 reglas de ciclo de vida en una configuración del ciclo de vida. Cada regla puede incluir los siguientes elementos XML:

- Caducidad: Elimine un objeto cuando se alcance una fecha especificada o cuando se alcance un número especificado de días, empezando desde el momento en que se ingirió el objeto.
- NoncurrentVersionExpiration: Elimine un objeto cuando se alcance un número especificado de días, empezando desde el momento en que el objeto se volvió no actual.
- Filtro (prefijo, etiqueta)
- Estado
- ID

Si aplica una configuración del ciclo de vida a un bloque, la configuración del ciclo de vida del bloque siempre anula la configuración de ILM de StorageGRID. StorageGRID utiliza la configuración de caducidad del bloque, no de ILM, para determinar si se deben eliminar o conservar objetos específicos.

Como resultado, es posible que se elimine un objeto de la cuadrícula aunque las instrucciones de colocación de una regla de ILM aún se apliquen al objeto. O bien, es posible que un objeto se conserve en la cuadrícula incluso después de que hayan transcurrido las instrucciones de colocación de ILM para el objeto. Para obtener más información, consulte ["Cómo funciona ILM durante la vida de un objeto"](#).



La configuración del ciclo de vida de bloques se puede usar con bloques que tienen habilitado el bloqueo de objetos S3, pero la configuración del ciclo de vida de bloques no se admite para bloques compatibles con versiones anteriores.

StorageGRID admite el uso de las siguientes operaciones de bloques para gestionar las configuraciones del ciclo de vida:

- ELIMINAR ciclo de vida de bloque
- OBTENGA el ciclo de vida de la cuchara
- CICLO de vida DE la cuchara

Cree la configuración del ciclo de vida

Como primer paso en la creación de una configuración de ciclo de vida, se crea un archivo JSON que incluye una o varias reglas. Por ejemplo, este archivo JSON incluye tres reglas, de la siguiente manera:

1. La regla 1 sólo se aplica a los objetos que coinciden con el prefijo `category1/` y que tienen un `key2` valor de `tag2`. La `Expiration` Parámetro especifica que los objetos que coinciden con el filtro caducarán a medianoche el 22 de agosto de 2020.
2. La regla 2 sólo se aplica a los objetos que coinciden con el prefijo `category2/`. La `Expiration` el parámetro especifica que los objetos que coinciden con el filtro caducarán 100 días después de que se ingieran.



Las reglas que especifican un número de días son relativas al momento en que se ingirió el objeto. Si la fecha actual supera la fecha de ingesta más el número de días, es posible que algunos objetos se eliminen del bloque en cuanto se aplique la configuración del ciclo de vida.

3. La regla 3 sólo se aplica a los objetos que coinciden con el prefijo `category3/`. La `Expiration` parámetro especifica que cualquier versión no actual de objetos coincidentes caducará 50 días después de que se conviertan en no actualizados.

```

{
  "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 bloque

Después de crear el archivo de configuración del ciclo de vida, se aplica a un bloque emitiendo una solicitud PUT Bucket Lifecycle.

Esta solicitud aplica la configuración del ciclo de vida del archivo de ejemplo a los objetos de un bloque denominado `testbucket`.

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

Para validar que se ha aplicado correctamente una configuración del ciclo de vida al bloque, emita una solicitud GET Bucket Lifecycle. Por ejemplo:

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

Una respuesta correcta muestra la configuración del ciclo de vida que acaba de aplicar.

Validar que la caducidad del ciclo de vida del bloque se aplica al objeto

Puede determinar si una regla de caducidad en la configuración del ciclo de vida se aplica a un objeto específico al emitir una solicitud PUT Object, HEAD Object o GET Object. Si se aplica una regla, la respuesta incluye una `Expiration` parámetro que indica cuándo caduca el objeto y qué regla de caducidad se ha coincido.



Dado que el ciclo de vida de los bloques anula la gestión del ciclo de vida de `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 PUT Object fue emitida el 22 de junio de 2020 y coloca un objeto en el `testbucket` cucharón.

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

La respuesta correcta indica que el objeto caducará en 100 días (01 de octubre de 2020) y que coincide 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 DE OBJETO HEAD se utilizó para obtener metadatos para el mismo objeto en el bloque testbucket.

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

La respuesta correcta incluye los metadatos del objeto e indica que el objeto caducará en 100 días y que coincide 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": {}
}
```

Recomendaciones para implementar la API REST de S3

Debe seguir estas recomendaciones al implementar la API DE REST de S3 para usar con StorageGRID.

Recomendaciones para las cabezas a los objetos no existentes

Si tu aplicación comprueba de forma rutinaria si existe un objeto en una ruta en la que no esperas que exista el objeto, debes usar el control de consistencia "disponible". Por ejemplo, debe utilizar el control de coherencia "disponible" si su aplicación dirige una ubicación antes DE PONERLA en práctica.

De lo contrario, si la operación HEAD no encuentra el objeto, es posible que reciba un número elevado de 500 errores internos de Server si uno o más nodos de almacenamiento no están disponibles.

Puede establecer el control de consistencia "Available" para cada bloque mediante LA solicitud DE consistencia PUT Bucket, o bien puede especificar el control de consistencia en el encabezado de solicitud para una operación de API individual.

Recomendaciones para las claves de objeto

Siga estas recomendaciones para los nombres de clave del objeto, según cuándo se creó el bloque por primera vez.

Bloques creados en StorageGRID 11,4 o versiones anteriores

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

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

En lugar de este formato:

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

Bloques creados en StorageGRID 11,4 o versiones posteriores

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



Una excepción a esto es una carga de trabajo S3 que elimina continuamente todos los objetos después de un breve periodo 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 similar a la fecha. Por ejemplo, suponga que un cliente S3 normalmente escribe 2.000 objetos por segundo y la política de ciclo de vida de la gestión de la vida útil de la información o del bloque elimina los objetos al cabo de tres días. Para minimizar el impacto en el rendimiento, puede asignar un nombre a las claves utilizando un patrón como el siguiente:

```
/mybucket/mydir/yyyymmddhhmmss-random_UUID.jpg
```

Recomendaciones para «lecturas de rango»

Si la ["opción global para comprimir objetos almacenados"](#) está activado, las aplicaciones cliente S3 deben evitar realizar operaciones GET Object que especifiquen un rango de bytes devueltos. Estas operaciones de «lectura de rango» son ineficientes, ya que StorageGRID debe descomprimir de forma efectiva los objetos para acceder a los bytes solicitados. LAS operaciones GET Object 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 tener un tiempo de espera.



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

Información relacionada

- ["Controles de consistencia"](#)
- ["PONGA la consistencia del cucharón"](#)
- ["Administre StorageGRID"](#)

Información de copyright

Copyright © 2025 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPTIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.