Como o StorageGRID implementa a API REST do S3

StorageGRID

NetApp
March 12, 2025

Índice

Como o StorageGRID implementa a API REST do S3 . . . 1
- Solicitações de cliente conflitantes . . . 1
- Controles de consistência . . . 1
  - Controles de consistência . . . 1
  - Use controles de consistência "read-after-new-write" e "available" . . . 2
  - Especifique o controle de consistência para a operação da API . . . 2
  - Especifique o controle de consistência para o balde . . . 2
    - como os controles de consistência e as regras ILM interagem para afetar a proteção de dados . . . 3
  - Exemplo de como o controle de consistência e a regra ILM podem interagir . . . 3
- Como as regras do StorageGRID ILM gerenciam objetos . . . 4
- Controle de versão de objetos . . . 5
  - ILM e versionamento . . . 5
- Use a API REST do S3 para configurar o bloqueio de objetos do S3 . . . 6
  - Como ativar o bloqueio de objetos S3D para um balde . . . 6
  - Configurações de retenção padrão para um balde . . . 6
  - Como definir a retenção padrão para um balde . . . 7
  - Como determinar a retenção padrão para um balde . . . 8
  - Como especificar configurações de retenção para um objeto . . . 9
  - Como atualizar as configurações de retenção para um objeto . . . 11
  - Como usar o modo DE GOVERNANÇA . . . 11
- Crie a configuração do ciclo de vida do S3 . . . 12
  - Qual é a configuração do ciclo de vida . . . 12
  - Criar configuração do ciclo de vida . . . 13
  - Aplique a configuração do ciclo de vida ao bucket . . . 15
  - Valide que a expiração do ciclo de vida do bucket se aplica ao objeto . . . 15
- Recomendações para a implementação da API REST do S3 . . . 16
  - Recomendações para heads to non-existent objects . . . 16
  - Recomendações para chaves de objeto . . . 17
  - Recomendações para "leituras de intervalo" . . . 17

Como o StorageGRID implementa a API REST do S3

Solicitações de cliente conflitantes

As solicitações de cliente conflitantes, como dois clientes escrevendo para a mesma chave, são resolvidas com base em "vitórias mais recentes".

O tempo para a avaliação "últimos ganhos" é baseado em quando o sistema StorageGRID completa uma determinada solicitação e não em quando os clientes S3 começam uma operação.

Controles de consistência

Os controles de consistência fornecem um equilíbrio entre a disponibilidade dos objetos e a consistência desses objetos em diferentes nós de storage e locais, conforme necessário pela aplicação.

Por padrão, o StorageGRID garante consistência de leitura após gravação para objetos recém-criados. Qualquer GET seguindo um PUT concluído com sucesso será capaz de ler os dados recém-escritos. As substituições de objetos existentes, atualizações de metadados e exclusões são, eventualmente, consistentes. As substituições geralmente levam segundos ou minutos para se propagar, mas podem levar até 15 dias.

Se você quiser executar operações de objeto em um nível de consistência diferente, você pode especificar um controle de consistência para cada bucket ou para cada operação de API.

Controles de consistência

O controle de consistência afeta como os metadados que o StorageGRID usa para rastrear objetos são distribuídos entre nós e, portanto, a disponibilidade de objetos para solicitações de clientes.

Você pode definir o controle de consistência para um bucket ou uma operação de API para um dos seguintes valores:

- **Todos**: Todos os nós recebem os dados imediatamente, ou a solicitação falhará.
- **Strong-global**: Garante consistência de leitura após gravação para todas as solicitações de clientes em todos os sites.
- * Strong-site*: Garante consistência de leitura-após-gravação para todas as solicitações de clientes dentro de um site.
- **Read-after-novo-write**: (Padrão) fornece consistência de leitura-após-gravação para novos objetos e eventual consistência para atualizações de objetos. Oferece alta disponibilidade e garantias de proteção de dados. Recomendado para a maioria dos casos.
- **Disponível**: Fornece consistência eventual para novos objetos e atualizações de objetos. Para buckets do S3, use somente conforme necessário (por exemplo, para um bucket que contém valores de log raramente lidos, ou para operações HEAD ou GET em chaves que não existem). Não compatível com buckets do FabricPool S3.

Use controles de consistência "read-after-new-write" e "available"

Quando uma operação HEAD ou GET usa o controle de consistência "read-after-new-write", o StorageGRID executa a pesquisa em várias etapas, como segue:

- Ele primeiro procura o objeto usando uma baixa consistência.
- Se essa pesquisa falhar, ela repete a pesquisa no próximo nível de consistência até atingir um nível de consistência equivalente ao comportamento para strong-global.

Se uma operação HEAD ou GET usar o controle de consistência "read-after-novo-write", mas o objeto não existir, a pesquisa de objetos sempre alcançará um nível de consistência equivalente ao comportamento para strong-global. Como esse nível de consistência exige que várias cópias dos metadados de objetos estejam disponíveis em cada local, você pode receber um número alto de erros de servidor interno do 500 se dois ou mais nós de storage no mesmo local não estiverem disponíveis.

A menos que você precise de garantias de consistência semelhantes ao Amazon S3, você pode evitar esses erros para operações HEAD and GET definindo o controle de consistência como "disponível". Quando uma operação HEAD ou GET usa o controle de consistência "disponível", o StorageGRID fornece consistência eventual apenas. Ele não tenta novamente uma operação com falha em níveis crescentes de consistência, portanto, não requer que várias cópias dos metadados do objeto estejam disponíveis.

Especifique o controle de consistência para a operação da API

Para definir o controle de consistência para uma operação de API individual, os controles de consistência devem ser suportados para a operação e você deve especificar o controle de consistência no cabeçalho da solicitação. Este exemplo define o controle de consistência como "local-trong" para uma operação GET Object.

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

> Você deve usar o mesmo controle de consistência para as operações COLOCAR Objeto e OBTER Objeto.

Especifique o controle de consistência para o balde

Para definir o controle de consistência para o bucket, você pode usar a solicitação de consistência do bucket do StorageGRID PUT e a solicitação DE consistência do bucket do GET. Ou você pode usar o Gerenciador do Locatário ou a API de Gerenciamento do Locatário.

Ao definir os controlos de consistência para um balde, tenha em atenção o seguinte:

- Definir o controle de consistência para um balde determina qual controle de consistência é usado para operações S3D realizadas nos objetos no balde ou na configuração do balde. Não afeta as operações no próprio balde.
- O controle de consistência para uma operação de API individual substitui o controle de consistência para o bucket.

- Em geral, os buckets devem usar o controle de consistência padrão, "read-after-new-write". Se as solicitações não estiverem funcionando corretamente, altere o comportamento do cliente do aplicativo, se possível. Ou configure o cliente para especificar o controle de consistência para cada solicitação de API. Defina o controle de consistência no nível do balde apenas como último recurso.

como os controles de consistência e as regras ILM interagem para afetar a proteção de dados

Tanto a sua escolha de controle de consistência quanto a sua regra ILM afetam a forma como os objetos são protegidos. Essas configurações podem interagir.

Por exemplo, o controle de consistência usado quando um objeto é armazenado afeta o posicionamento inicial dos metadados do objeto, enquanto o comportamento de ingestão selecionado para a regra ILM afeta o posicionamento inicial das cópias do objeto. Como o StorageGRID exige acesso aos metadados de um objeto e aos dados para atender às solicitações do cliente, selecionar níveis de proteção correspondentes para o nível de consistência e comportamento de ingestão pode fornecer melhor proteção inicial de dados e respostas do sistema mais previsíveis.

Os seguintes comportamentos de ingestão estão disponíveis para regras ILM:

- * Commit duplo*: O StorageGRID faz imediatamente cópias provisórias do objeto e retorna sucesso ao cliente. Cópias especificadas na regra ILM são feitas quando possível.
- **Strict**: Todas as cópias especificadas na regra ILM devem ser feitas antes que o sucesso seja devolvido ao cliente.
- **Balanced**: O StorageGRID tenta fazer todas as cópias especificadas na regra ILM no ingest; se isso não for possível, cópias provisórias são feitas e o sucesso é retornado ao cliente. As cópias especificadas na regra ILM são feitas quando possível.

> Antes de selecionar o comportamento de ingestão para uma regra ILM, leia a descrição completa dessas configurações nas instruções para gerenciar objetos com gerenciamento do ciclo de vida das informações.

Exemplo de como o controle de consistência e a regra ILM podem interagir

Suponha que você tenha uma grade de dois locais com a seguinte regra ILM e a seguinte configuração de nível de consistência:

- **Regra ILM**: Crie duas cópias de objeto, uma no local e outra em um local remoto. O comportamento de ingestão estrita é selecionado.
- **Nível de consistência**: "Trong-global" (metadados de objetos são imediatamente distribuídos para todos os sites.)

Quando um cliente armazena um objeto na grade, o StorageGRID faz cópias de objeto e distribui metadados para ambos os sites antes de retornar sucesso ao cliente.

O objeto é totalmente protegido contra perda no momento da mensagem de ingestão bem-sucedida. Por exemplo, se o local for perdido logo após a ingestão, cópias dos dados do objeto e dos metadados do objeto ainda existem no local remoto. O objeto é totalmente recuperável.

Se, em vez disso, você usou a mesma regra ILM e o nível de consistência "site-trong", o cliente poderá receber uma mensagem de sucesso depois que os dados do objeto forem replicados para o site remoto, mas antes que os metadados do objeto sejam distribuídos lá. Nesse caso, o nível de proteção dos metadados de

objetos não corresponde ao nível de proteção dos dados de objeto. Se o site local for perdido logo após a ingestão, os metadados do objeto serão perdidos. O objeto não pode ser recuperado.

A inter-relação entre níveis de consistência e regras de ILM pode ser complexa. Contacte a NetApp se necessitar de assistência.

Informações relacionadas

"Gerenciar objetos com ILM"

"OBTER consistência de balde"

"COLOQUE a consistência do balde"

Como as regras do StorageGRID ILM gerenciam objetos

O administrador da grade cria regras de gerenciamento do ciclo de vida das informações (ILM) para gerenciar dados de objetos ingeridos no sistema StorageGRID a partir de aplicativos clientes da API REST do S3. Essas regras são então adicionadas à política ILM para determinar como e onde os dados do objeto são armazenados ao longo do tempo.

As configurações de ILM determinam os seguintes aspetos de um objeto:

- **Geografia**

  O local dos dados de um objeto, seja no sistema StorageGRID (pool de storage) ou em um pool de storage de nuvem.

- **Grau de armazenamento**

  O tipo de storage usado para armazenar dados de objetos: Por exemplo, flash ou disco giratório.

- * Proteção contra perdas*

  Quantas cópias são feitas e os tipos de cópias criadas: Replicação, codificação de apagamento ou ambos.

- **Retenção**

  As mudanças ao longo do tempo para como os dados de um objeto são gerenciados, onde são armazenados e como eles são protegidos contra perda.

- **Proteção durante o consumo**

  O método usado para proteger dados de objetos durante a ingestão: Colocação síncrona (usando as opções balanceadas ou rigorosas para o comportamento de ingestão) ou fazendo cópias provisórias (usando a opção de confirmação dupla).

As regras do ILM podem filtrar e selecionar objetos. Para objetos ingeridos usando S3, as regras do ILM podem filtrar objetos com base nos seguintes metadados:

- Conta de locatário
- Nome do intervalo

- Tempo de ingestão
- Chave
- Último tempo de acesso

> Por padrão, as atualizações para o último tempo de acesso são desativadas para todos os buckets do S3. Se o seu sistema StorageGRID incluir uma regra ILM que use a opção último tempo de acesso, você deverá habilitar as atualizações para o último tempo de acesso para os buckets do S3 especificados nessa regra. Use a solicitação de última hora de acesso do PUT Bucket, o Gerenciador do locatário (consulte "Ative ou desative as atualizações da última hora de acesso") ou a API de gerenciamento do locatário. Ao ativar as atualizações da última hora de acesso, esteja ciente de que o desempenho do StorageGRID pode ser reduzido, especialmente em sistemas com objetos pequenos.

- Restrição de localização
- Tamanho do objeto
- Metadados do usuário
- Etiqueta do objeto

Informações relacionadas

"Use uma conta de locatário"

"Gerenciar objetos com ILM"

"COLOQUE o último tempo de acesso do balde"

Controle de versão de objetos

Você pode usar o controle de versão para reter várias versões de um objeto, o que protege contra a exclusão acidental de objetos e permite recuperar e restaurar versões anteriores de um objeto.

O sistema StorageGRID implementa o controle de versão com suporte para a maioria dos recursos, e com algumas limitações. O StorageGRID suporta até 1.000 versões de cada objeto.

O controle de versão de objetos pode ser combinado com o gerenciamento do ciclo de vida das informações do StorageGRID (ILM) ou com a configuração do ciclo de vida do bucket do S3. Você deve habilitar explicitamente o controle de versão para cada bucket para ativar essa funcionalidade para o bucket. Cada objeto no seu bucket recebe um ID de versão, que é gerado pelo sistema StorageGRID.

O uso de MFA (autenticação multifator) Excluir não é compatível.

> O controle de versão pode ser ativado somente em buckets criados com o StorageGRID versão 10,3 ou posterior.

ILM e versionamento

As políticas de ILM são aplicadas a cada versão de um objeto. Um processo de digitalização ILM verifica continuamente todos os objetos e os reavalia em relação à política ILM atual. Quaisquer alterações feitas às políticas ILM são aplicadas a todos os objetos ingeridos anteriormente. Isso inclui versões ingeridas anteriormente se o controle de versão estiver ativado. A digitalização ILM aplica novas alterações ILM a

objetos ingeridos anteriormente.

Para objetos S3 em buckets habilitados para versionamento, o suporte para versionamento permite que você crie regras ILM que usam "'hora não atual'" como tempo de referência (selecione **Sim** para a pergunta, "'aplicar esta regra apenas para versões de objetos mais antigas?'" no "Etapa 1 do assistente criar uma regra ILM"). Quando um objeto é atualizado, suas versões anteriores se tornam não atuais. O uso de um filtro "tempo não atual" permite criar políticas que reduzam o impactos de armazenamento de versões anteriores de objetos.

> Quando você carrega uma nova versão de um objeto usando uma operação de upload multipart, o tempo não atual para a versão original do objeto reflete quando o upload multipart foi criado para a nova versão, não quando o upload multipart foi concluído. Em casos limitados, o tempo não atual para a versão original pode ser horas ou dias antes do tempo para a versão atual.

"Regras e políticas do ILM para objetos com versão S3 (exemplo 4)"Consulte .

Use a API REST do S3 para configurar o bloqueio de objetos do S3

Se a configuração global de bloqueio de objetos S3 estiver ativada para o seu sistema StorageGRID, você poderá criar buckets com o bloqueio de objetos S3 ativado. Você pode especificar a retenção padrão para cada bucket ou configurações de retenção para cada versão do objeto.

Como ativar o bloqueio de objetos S3D para um balde

Se a configuração global de bloqueio de objetos S3 estiver ativada para o seu sistema StorageGRID, você poderá ativar opcionalmente o bloqueio de objetos S3 quando criar cada bucket.

S3 Object Lock é uma configuração permanente que só pode ser ativada quando você cria um bucket. Não é possível adicionar ou desativar o bloqueio de objetos S3 após a criação de um bucket.

Para ativar o bloqueio de objetos S3D para um bucket, use um destes métodos:

- Crie o bucket usando o Gerenciador do locatário. "Crie um balde S3D."Consulte .
- Crie o bucket usando uma solicitação DE COLOCAR balde com o `x-amz-bucket-object-lock-enabled` cabeçalho de solicitação. "Operações em baldes"Consulte .

O bloqueio de objetos S3 requer o controle de versão do bucket, que é ativado automaticamente quando o bucket é criado. Não é possível suspender o controle de versão para o bucket. "Controle de versão de objetos"Consulte .

Configurações de retenção padrão para um balde

Quando o bloqueio de objetos S3D está ativado para um bucket, você pode opcionalmente habilitar a retenção padrão para o bucket e especificar um modo de retenção padrão e um período de retenção padrão.

Modo de retenção predefinido

- No modo DE CONFORMIDADE:

- O objeto não pode ser excluído até que sua data de retenção seja alcançada.
- O retent-until-date do objeto pode ser aumentado, mas não pode ser diminuído.
- A data de retenção do objeto não pode ser removida até que essa data seja atingida.

- No MODO DE GOVERNANÇA:
  - Os usuários com `s3:BypassGovernanceRetention` permissão podem usar o `x-amz-bypass-governance-retention: true` cabeçalho de solicitação para ignorar as configurações de retenção.
  - Esses usuários podem excluir uma versão de objeto antes de sua data de retenção ser alcançada.
  - Esses usuários podem aumentar, diminuir ou remover a data de retenção até um objeto.

Período de retenção predefinido

Cada bucket pode ter um período de retenção padrão especificado em anos ou dias.

Como definir a retenção padrão para um balde

Para definir a retenção padrão para um bucket, use um destes métodos:

- Gerencie as configurações do balde a partir do Gerenciador do Locatário. "Crie um bucket do S3"Consulte e "Atualização S3 retenção padrão bloqueio Objeto".
- Emita uma solicitação DE configuração de bloqueio de objeto PUT para que o bucket especifique o modo padrão e o número padrão de dias ou anos.

COLOCAR Configuração bloqueio Objeto

A solicitação de configuração de bloqueio de objeto PUT permite que você defina e modifique o modo de retenção padrão e o período de retenção padrão para um bucket que tenha o bloqueio de objeto S3 ativado. Você também pode remover as configurações de retenção padrão configuradas anteriormente.

Quando novas versões de objetos são ingeridas para o bucket, o modo de retenção padrão é aplicado se `x-amz-object-lock-mode` e `x-amz-object-lock-retain-until-date` não forem especificados. O período de retenção padrão é usado para calcular a data de retenção até se `x-amz-object-lock-retain-until-date` não for especificado.

Se o período de retenção padrão for modificado após a ingestão de uma versão de objeto, a data de retenção até a versão do objeto permanecerá a mesma e não será recalculada usando o novo período de retenção padrão.

Você deve ter a `s3:PutBucketObjectLockConfiguration` permissão, ou ser raiz da conta, para concluir esta operação.

O `Content-MD5` cabeçalho da solicitação deve ser especificado na solicitação DE COLOCAÇÃO.

Exemplo de solicitação

Este exemplo habilita o bloqueio de objetos S3 para um bucket e define o modo de retenção padrão para CONFORMIDADE e o período de retenção padrão para 6 anos.

```
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>
```

Como determinar a retenção padrão para um balde

Para determinar se o bloqueio de objeto S3 está ativado para um bucket e para ver o modo de retenção e o período de retenção padrão, use um destes métodos:

- Veja o bucket no Gerenciador do Locatário. "Veja os baldes do S3"Consulte .
- Emitir um pedido DE configuração GET Object Lock.

OBTER Configuração bloqueio Objeto

A solicitação DE configuração GET Object Lock permite que você determine se o bloqueio de objeto S3 está ativado para um bucket e, se ele está ativado, veja se há um modo de retenção padrão e período de retenção configurados para o bucket.

Quando novas versões de objetos são ingeridas para o bucket, o modo de retenção padrão é aplicado se `x-amz-object-lock-mode` não for especificado. O período de retenção padrão é usado para calcular a data de retenção até se `x-amz-object-lock-retain-until-date` não for especificado.

Você deve ter a `s3:GetBucketObjectLockConfiguration` permissão, ou ser raiz da conta, para concluir esta operação.

Exemplo de solicitação

```
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
```

Exemplo de resposta

```
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>
```

Como especificar configurações de retenção para um objeto

Um bucket com o bloqueio de objetos S3 ativado pode conter uma combinação de objetos com e sem as configurações de retenção do bloqueio de objetos S3.

As configurações de retenção no nível do objeto são especificadas usando a API REST do S3. As configurações de retenção de um objeto substituem quaisquer configurações de retenção padrão para o bucket.

Você pode especificar as seguintes configurações para cada objeto:

- **Modo de retenção**: CONFORMIDADE ou GOVERNANÇA.
- **Retent-until-date**: Uma data especificando quanto tempo a versão do objeto deve ser mantida pelo StorageGRID.

    - No modo DE CONFORMIDADE, se a data de retenção estiver no futuro, o objeto pode ser recuperado, mas não pode ser modificado ou excluído. A data de retenção até pode ser aumentada, mas esta data não pode ser diminuída ou removida.
    - No MODO DE GOVERNANÇA, os usuários com permissão especial podem ignorar a configuração reter até a data. Eles podem excluir uma versão de objeto antes que seu período de retenção tenha decorrido. Eles também podem aumentar, diminuir ou até mesmo remover a data de retenção.
- **Retenção legal**: Aplicar uma retenção legal a uma versão de objeto bloqueia imediatamente esse objeto. Por exemplo, você pode precisar colocar uma retenção legal em um objeto relacionado a uma investigação ou disputa legal. Uma retenção legal não tem data de expiração, mas permanece em vigor até que seja explicitamente removida.

  A configuração de retenção legal para um objeto é independente do modo de retenção e da data de retenção. Se uma versão de objeto estiver sob uma retenção legal, ninguém poderá excluir essa versão.

Para especificar as configurações de bloqueio de objetos do S3 ao adicionar uma versão de objeto a um bucket, emita uma solicitação"Objeto PUT" , "COLOCAR Objeto - Copiar"ou "Inicie o carregamento de várias peças".

Você pode usar o seguinte:

- `x-amz-object-lock-mode`, Que pode ser CONFORMIDADE ou GOVERNANÇA (diferencia maiúsculas de minúsculas).

  > Se você especificar `x-amz-object-lock-mode`, você também deve especificar `x-amz-object-lock-retain-until-date`.

- `x-amz-object-lock-retain-until-date`
    - O valor reter-até-data deve estar no formato `2020-08-10T21:46:00Z`. Segundos fracionários são permitidos, mas apenas 3 dígitos decimais são preservados (precisão de milissegundos). Outros formatos ISO 8601 não são permitidos.
    - A data de retenção deve ser no futuro.
- `x-amz-object-lock-legal-hold`

  Se a retenção legal estiver ATIVADA (sensível a maiúsculas e minúsculas), o objeto é colocado sob uma retenção legal. Se a retenção legal estiver DESLIGADA, nenhuma retenção legal será colocada. Qualquer outro valor resulta em um erro de 400 Bad Request (InvalidArgument).

Se você usar qualquer um desses cabeçalhos de solicitação, esteja ciente dessas restrições:

- O `Content-MD5` cabeçalho de solicitação é necessário se qualquer `x-amz-object-lock-*` cabeçalho de solicitação estiver presente na solicitação DE Objeto PUT. `Content-MD5` Não é necessário para COLOCAR Objeto - Copiar ou iniciar carregamento Multipart.
- Se o bucket não tiver o bloqueio de objeto S3 ativado e um `x-amz-object-lock-*` cabeçalho de solicitação estiver presente, um erro de solicitação incorreta 400 (InvalidRequest) será retornado.
- A solicitação put Object suporta o uso do `x-amz-storage-class: REDUCED_REDUNDANCY` para corresponder ao comportamento da AWS. No entanto, quando um objeto é ingerido em um bucket com o bloqueio de objeto S3 ativado, o StorageGRID sempre realizará uma ingestão de confirmação dupla.
- Uma resposta DE versão DE GET ou HEAD Object posterior incluirá os cabeçalhos `x-amz-object-lock-mode`, `x-amz-object-lock-retain-until-date`, e `x-amz-object-lock-legal-hold`, se

configurado e se o remetente da solicitação tiver as permissões corretas `s3:Get*`.

Você pode usar a `s3:object-lock-remaining-retention-days` chave de condição de política para limitar os períodos de retenção mínimo e máximo permitidos para seus objetos.

Como atualizar as configurações de retenção para um objeto

Se você precisar atualizar as configurações de retenção legal ou retenção para uma versão de objeto existente, poderá executar as seguintes operações de subrecursos de objeto:

- `PUT Object legal-hold`

  Se o novo valor de retenção legal estiver ATIVADO, o objeto será colocado sob uma retenção legal. Se o valor de retenção legal estiver DESLIGADO, a retenção legal é levantada.

- `PUT Object retention`
  - O valor do modo pode ser CONFORMIDADE ou GOVERNANÇA (sensível a maiúsculas e minúsculas).
  - O valor reter-até-data deve estar no formato `2020-08-10T21:46:00Z`. Segundos fracionários são permitidos, mas apenas 3 dígitos decimais são preservados (precisão de milissegundos). Outros formatos ISO 8601 não são permitidos.
  - Se uma versão de objeto tiver uma data retida-até-data existente, você só poderá aumentá-la. O novo valor deve estar no futuro.

Como usar o modo DE GOVERNANÇA

Os usuários que têm a `s3:BypassGovernanceRetention` permissão podem ignorar as configurações de retenção ativa de um objeto que usa o modo DE GOVERNANÇA. Qualquer operação de retenção de objetos de EXCLUSÃO ou DE COLOCAÇÃO deve incluir o `x-amz-bypass-governance-retention:true` cabeçalho de solicitação. Esses usuários podem executar essas operações adicionais:

- Execute as operações DELETE Object ou DELETE Multiple Objects para excluir uma versão do objeto antes de seu período de retenção ter decorrido.

  Os objetos que estão sob uma retenção legal não podem ser excluídos. A retenção legal deve estar DESLIGADA.

- Execute operações de retenção de objetos que alteram o modo de uma versão DE objeto DE GOVERNANÇA para CONFORMIDADE antes que o período de retenção do objeto tenha decorrido.

  Alterar o modo DE CONFORMIDADE para GOVERNANÇA nunca é permitido.

- Execute operações DE retenção de objetos PUT para aumentar, diminuir ou remover o período de retenção de uma versão de objeto.

Informações relacionadas

- "Gerencie objetos com o S3 Object Lock"
- "Use o bloqueio de objetos S3D para reter objetos"
- "Guia do usuário do Amazon Simple Storage Service: Usando o bloqueio de objeto S3"

Crie a configuração do ciclo de vida do S3

Você pode criar uma configuração de ciclo de vida do S3 para controlar quando objetos específicos são excluídos do sistema StorageGRID.

O exemplo simples nesta seção ilustra como uma configuração do ciclo de vida do S3 pode controlar quando certos objetos são excluídos (expirados) de buckets específicos do S3. O exemplo nesta seção é apenas para fins ilustrativos. Para obter detalhes completos sobre como criar configurações de ciclo de vida do S3, "Amazon Simple Storage Service Developer Guide: Gerenciamento do ciclo de vida do objeto" consulte . Observe que o StorageGRID suporta apenas ações de expiração; ele não oferece suporte a ações de transição.

Qual é a configuração do ciclo de vida

Uma configuração de ciclo de vida é um conjunto de regras que são aplicadas aos objetos em buckets específicos do S3. Cada regra especifica quais objetos são afetados e quando esses objetos expirarão (em uma data específica ou após algum número de dias).

O StorageGRID dá suporte a até 1.000 regras de ciclo de vida em uma configuração de ciclo de vida. Cada regra pode incluir os seguintes elementos XML:

- Expiração: Exclua um objeto quando uma data especificada é atingida ou quando um número especificado de dias é atingido, a partir de quando o objeto foi ingerido.
- NoncurrentVersionExpiration: Exclua um objeto quando um número especificado de dias é atingido, a partir de quando o objeto se tornou inatual.
- Filtro (prefixo, Tag)
- Estado
- ID

Se você aplicar uma configuração de ciclo de vida a um bucket, as configurações de ciclo de vida do bucket sempre substituem as configurações de ILM do StorageGRID. O StorageGRID usa as configurações de expiração para o bucket, não o ILM, para determinar se deseja excluir ou reter objetos específicos.

Como resultado, um objeto pode ser removido da grade, mesmo que as instruções de colocação em uma regra ILM ainda se apliquem ao objeto. Ou, um objeto pode ser retido na grade mesmo depois que quaisquer instruções de colocação de ILM para o objeto tiverem expirado. Para obter detalhes, "Como o ILM opera ao longo da vida de um objeto"consulte .

> A configuração do ciclo de vida do bucket pode ser usada com buckets que têm o S3 Object Lock ativado, mas a configuração do ciclo de vida do bucket não é suportada para buckets compatíveis com legado.

O StorageGRID dá suporte ao uso das seguintes operações de bucket para gerenciar configurações do ciclo de vida:

- ELIMINAR ciclo de vida do balde
- OBTENHA o ciclo de vida do Bucket
- COLOQUE o ciclo de vida do balde

Criar configuração do ciclo de vida

Como primeira etapa na criação de uma configuração de ciclo de vida, você cria um arquivo JSON que inclui uma ou mais regras. Por exemplo, este arquivo JSON inclui três regras, como segue:

1. A regra 1 aplica-se apenas a objetos que correspondam ao prefixo `category1/` e que tenham um `key2` valor `tag2` de . O `Expiration` parâmetro especifica que os objetos correspondentes ao filtro expirarão à meia-noite de 22 de agosto de 2020.
2. A regra 2 aplica-se apenas a objetos que correspondam ao prefixo `category2/`. O `Expiration` parâmetro especifica que os objetos correspondentes ao filtro expirarão 100 dias após serem ingeridos.

   > As regras que especificam um número de dias são relativas a quando o objeto foi ingerido. Se a data atual exceder a data de ingestão mais o número de dias, alguns objetos podem ser removidos do intervalo assim que a configuração do ciclo de vida for aplicada.

3. A regra 3 aplica-se apenas a objetos que correspondam ao prefixo `category3/`. O `Expiration` parâmetro especifica que quaisquer versões não atuais de objetos correspondentes expirarão 50 dias após se tornarem não atuais.

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

Aplique a configuração do ciclo de vida ao bucket

Depois de criar o arquivo de configuração do ciclo de vida, aplique-o a um bucket emitindo uma solicitação DE ciclo de vida do PUT Bucket.

Essa solicitação aplica a configuração do ciclo de vida no arquivo de exemplo a objetos em um bucket `testbucket` chamado .

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

Para validar que uma configuração de ciclo de vida foi aplicada com sucesso ao bucket, emita uma solicitação DE ciclo de vida do GET Bucket. Por exemplo:

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

Uma resposta bem-sucedida lista a configuração do ciclo de vida que você acabou de aplicar.

Valide que a expiração do ciclo de vida do bucket se aplica ao objeto

É possível determinar se uma regra de expiração na configuração do ciclo de vida se aplica a um objeto específico ao emitir uma SOLICITAÇÃO PUT Object, HEAD Object ou GET Object. Se uma regra se aplicar, a resposta inclui um `Expiration` parâmetro que indica quando o objeto expira e qual regra de expiração foi correspondida.

> Como o ciclo de vida do bucket substitui o ILM, a `expiry-date` mostrada é a data real em que o objeto será excluído. Para obter detalhes, "Como a retenção de objetos é determinada"consulte .

Por exemplo, essa SOLICITAÇÃO PUT Object foi emitida em 22 de junho de 2020 e coloca um objeto no `testbucket` intervalo.

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

A resposta de sucesso indica que o objeto expirará em 100 dias (01 de outubro de 2020) e que correspondia à regra 2 da configuração do ciclo de vida.

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

Por exemplo, essa solicitação de objeto PRINCIPAL foi usada para obter metadados para o mesmo objeto no bucket do testbucket.

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

A resposta de sucesso inclui os metadados do objeto e indica que o objeto expirará em 100 dias e que correspondia à regra 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": {}
}
```

Recomendações para a implementação da API REST do S3

Você deve seguir estas recomendações ao implementar a API REST do S3 para uso com o StorageGRID.

Recomendações para heads to non-existent objects

Se seu aplicativo verificar rotineiramente para ver se um objeto existe em um caminho onde você não espera que o objeto realmente exista, você deve usar o controle de consistência ""disponível"". Por exemplo, você deve usar o controle de consistência "disponível" se seu aplicativo dirigir um local antes DE COLOCÁ-lo.

Caso contrário, se a operação PRINCIPAL não encontrar o objeto, você poderá receber um número alto de 500 erros de servidor interno se um ou mais nós de storage não estiverem disponíveis.

Você pode definir o controle de consistência "'disponível'" para cada bucket usando a solicitação de consistência do PUT Bucket, ou você pode especificar o controle de consistência no cabeçalho da solicitação para uma operação de API individual.

Recomendações para chaves de objeto

Siga estas recomendações para nomes de chave de objeto, com base em quando o intervalo foi criado pela primeira vez.

Buckets criados no StorageGRID 11,4 ou anterior

- Não use valores aleatórios como os primeiros quatro carateres de chaves de objeto. Isso contrasta com a antiga recomendação da AWS para prefixos-chave. Em vez disso, use prefixos não aleatórios e não exclusivos, como `image` .
- Se você seguir a antiga recomendação da AWS para usar carateres aleatórios e exclusivos em prefixos de chave, prefix as chaves de objeto com um nome de diretório. Ou seja, use este formato:

  `mybucket/mydir/f8e3-image3132.jpg`

  Em vez deste formato:

  `mybucket/f8e3-image3132.jpg`

Buckets criados no StorageGRID 11,4 ou posterior

Não é necessário restringir nomes de chaves de objeto para atender às práticas recomendadas de desempenho. Na maioria dos casos, você pode usar valores aleatórios para os primeiros quatro carateres de nomes de chave de objeto.

> Uma exceção a isso é uma carga de trabalho S3 que remove continuamente todos os objetos após um curto período de tempo. Para minimizar o impacto no desempenho desse caso de uso, varie uma parte principal do nome da chave a cada milhares de objetos com algo como a data. Por exemplo, suponha que um cliente S3 normalmente grava 2.000 objetos/segundo e que a política de ciclo de vida ILM ou bucket remove todos os objetos após três dias. Para minimizar o impactos no desempenho, você pode nomear chaves usando um padrão como este: `/mybucket/mydir/yyyymmddhhmmss-random_UUID.jpg`

Recomendações para "leituras de intervalo"

Se o "opção global para comprimir objetos armazenados" estiver ativado, os aplicativos cliente S3 devem evitar executar operações GET Object que especificem um intervalo de bytes que sejam retornados. Essas operações de leitura de intervalo são ineficientes porque o StorageGRID deve descompactar efetivamente os objetos para acessar os bytes solicitados. As operações GET Object que solicitam um pequeno intervalo de bytes de um objeto muito grande são especialmente ineficientes; por exemplo, é ineficiente ler um intervalo de 10 MB a partir de um objeto compactado de 50 GB.

Se os intervalos forem lidos a partir de objetos compatados, as solicitações do cliente podem expirar.

> Se você precisar compactar objetos e seu aplicativo cliente precisar usar leituras de intervalo, aumente o tempo limite de leitura para o aplicativo.

Informações relacionadas

- "Controles de consistência"
- "COLOQUE a consistência do balde"
- "Administrar o StorageGRID"

Informações sobre direitos autorais

Informações sobre marcas comerciais