Metadados

Registre o que descreve o objeto e sua emissão

Pré-requisitos

Conheça objetos S3, cabeçalhos HTTP e chaves de comprovante. O laboratório usa Python 3, sem dependências, e uma sequência de bytes fictícia. Os dados de negócio são registrados separadamente dos bytes.

Escolha onde guardar cada informação [1]

Content-Type e Content-Disposition descrevem como o servidor entrega o objeto. Metadados personalizados, como x-amz-meta-revisao, podem facilitar diagnóstico. Tags servem a regras próprias do armazenamento, como classificação e ciclo de vida. Titular, unidade, revisão atual e autorização de leitura pertencem ao registro da aplicação.

Use o registro da aplicação para comparar

Na emissão da revisão 2 de R45, registre chave, tipo esperado, tamanho em bytes e hash. Marque a revisão como disponível somente depois de conferir a escrita. Leia metadados do objeto quando precisar inspecionar seu tamanho e tipo. Para conferir criptograficamente bytes recebidos, calcule o agregado sobre o conteúdo ou use o mecanismo de checksum do armazenamento de acordo com seu contrato.

Escolha a evidência de integridade [2]

Tamanho identifica truncamentos e complementa o hash. Arquivos diferentes podem ter o mesmo comprimento. Um checksum calculado segundo um algoritmo acordado detecta alterações no conteúdo. ETag identifica uma representação e seu valor depende da forma de armazenamento. Use o campo de checksum adequado quando precisar de uma verificação específica de algoritmo.

Preserve revisão e rastreabilidade

No S3, alterar metadados definidos no upload exige copiar o objeto com substituição dos metadados. Se a mudança faz parte de uma nova emissão, considere uma chave de revisão nova. Evite guardar nome de cliente, contato ou segredos nos metadados e nomes de objeto.

Metadados

Exercício resolvido • Confira a revisão 2

Enunciado e dados de partida

R45 tem um comprovante associado à revisão 2. Registre o tamanho e SHA-256 do conteúdo esperado. Teste a leitura original, um tipo declarado incorreto, um truncamento e uma alteração de um byte com o mesmo tamanho.

1. Salve metadados.py

```
from hashlib import sha256

original = b'comprovante ficticio R45 revisao 2'
registro = {
    'chave': 'comprovantes/U12/R45/v2.pdf',
    'tipo': 'application/pdf',
    'tamanho': len(original),
    'sha256': sha256(original).hexdigest(),
}

def conferir(conteudo, tipo):
    return (
        tipo == registro['tipo']
        and len(conteudo) == registro['tamanho']
        and sha256(conteudo).hexdigest() == registro['sha256']
    )

assert conferir(original, 'application/pdf')
assert not conferir(original, 'text/plain')
assert not conferir(original[:-1], 'application/pdf')
alterado = original[:-1] + b'1'
assert len(alterado) == len(original)
assert not conferir(alterado, 'application/pdf')
print('Tipo, tamanho e integridade conferidos')
```

2. Execute python metadados.py

A saída é ‘Tipo, tamanho e integridade conferidos’. A sequência de bytes é uma amostra didática. O teste compara o tipo informado com o tipo esperado e verifica os bytes. Um validador de formato PDF seria uma etapa própria.

Metadados

Interprete recusas e cuide da recuperação

3. Resolva os quatro casos

O original coincide com tipo, tamanho e hash. text/plain falha pelo tipo. O truncamento falha pelo tamanho. A troca do último byte preserva o comprimento e falha pelo SHA-256. Compare com o hash armazenado na emissão via origem confiável. Recalcular o esperado a partir do próprio download eliminaria essa comparação independente.

Caso de erro • Use apenas tamanho

Remova a condição SHA-256 e ajuste a expressão para terminar na comparação de tamanho. A última asserção falha, pois a alteração tem o mesmo número de bytes. Restaure a condição e confira o resultado.

Variação • Ajuste o nome exibido no download

Se só o nome sugerido mudar, Content-Disposition pode ser ajustado pela estratégia de entrega ou em novos metadados do objeto. Preserve a identidade do comprovante e verifique se o contrato exige manter o histórico do nome anterior.

Critérios para conferir

Identifique a revisão, a chave, o tipo de mídia e a origem da expectativa de integridade. Confirme que um registro divergente fica indisponível para a jornada até ser investigado. O teste é local. No bucket de teste, confira também os cabeçalhos efetivamente entregues e a opção de checksum usada pela integração.

Referências primárias

[1] AWS • Metadados de objetos S3

[2] AWS • Verificação de integridade de objetos

Metadados

Sirva anexos com metadados coerentes

Separe tipo declarado de tipo validado

Content-Type orienta o cliente sobre a interpretação do conteúdo. O valor enviado no upload é uma declaração do cliente. Antes de disponibilizar um PDF, confira o tipo real do arquivo e defina como ele será entregue, inclusive Content-Disposition quando a intenção for download.

Exercício adicional

Um PDF chega ao navegador como application/octet-stream. O que revisar?

Resolução

Confira o tipo armazenado e valide o arquivo. Corrija o metadado ao regravar ou copiar o objeto com os metadados desejados. Se houver CDN, confira o cabeçalho também após expirar ou invalidar a cópia em cache.

Ao assistir ao vídeo

Observe quando os metadados são definidos no envio do objeto.

Vídeo complementar

AWS S3 Object Metadata (em inglês)