URLs pré-assinadas

Conceda acesso temporário a um objeto privado

Pré-requisitos

Conheça a distinção entre conta do cliente, função IAM e chave de objeto. O laboratório usa Node.js 24 e o SDK JavaScript da AWS. O endereço .example e as credenciais fictícias permitem assinar localmente sem chamar o S3.

Entenda a capacidade concedida [1]

A URL pré-assinada inclui parâmetros de autenticação para uma operação específica sobre um objeto. O download usa as permissões da identidade que assinou. Quem tiver a URL pode usá-la enquanto ela for válida e as permissões se mantiverem. Evite publicar a URL em logs, mensagens compartilhadas ou registros de análise.

Autorize na emissão

A API carrega o comprovante pela reserva e verifica se a conta pode ler aquela revisão. A chave final vem do registro do servidor. Em caso de recusa, a API encerra o pedido sem assinar. Em produção, obtenha as credenciais temporárias da função do serviço e conceda leitura só no prefixo necessário.

Escolha um prazo compatível com a jornada [1]

Se a URL for emitida às 14h00 para 120 segundos, o prazo nominal termina às 14h02. Se a credencial usada expirar às 14h01, a URL termina antes. Em um download iniciado antes da expiração, uma conexão ainda aberta pode continuar. Uma nova requisição após a expiração precisa de autorização e emissão novamente.

Prefira operação e caminho explícitos [2]

GetObjectCommand descreve uma leitura. Uma URL de download não concede escrita. Para upload, use uma operação separada, chaves controladas e regras de gravação e verificação de conteúdo.

URLs pré-assinadas

Exercício resolvido • Autorize a revisão certa

Enunciado e dados de partida

O comprovante v1 de R45 pertence a Lia. Beto deve receber recusa. Gere uma URL temporária de leitura para Lia e confira caminho e prazo sem imprimir o link.

1. Salve acesso.mjs

```
import assert from 'node:assert/strict';
import { S3Client, GetObjectCommand } from '@aws-sdk/client-s3';
import { getSignedUrl } from '@aws-sdk/s3-request-presigner';

const s3 = new S3Client({
  region: 'us-east-1',
  endpoint: 'https://objetos.example',
  forcePathStyle: true,
  credentials: { accessKeyId: 'EXEMPLO', secretAccessKey: 'ficticia' },
});

async function emitir(conta, comprovante) {
  if (conta !== comprovante.titular) return null;
  const comando = new GetObjectCommand({
    Bucket: 'agenda-lab',
    Key: comprovante.chave,
  });
  return getSignedUrl(s3, comando, { expiresIn: 120 });
}

const pdf = { titular: 'Lia', chave: 'comprovantes/U12/R45/v1.pdf' };
assert.equal(await emitir('Beto', pdf), null);
const url = new URL(await emitir('Lia', pdf));
assert.equal(url.pathname, '/agenda-lab/comprovantes/U12/R45/v1.pdf');
assert.equal(url.searchParams.get('X-Amz-Expires'), '120');
assert.ok(url.searchParams.has('X-Amz-Signature'));
console.log('Autorizacao, objeto e prazo conferidos');
```

Leia os parâmetros

emitir recebe dados já carregados pelo servidor e retorna null para outro titular. getSignedUrl calcula a assinatura para o comando e as credenciais apresentados. O endpoint fictício permite verificar parâmetros sem tentar baixar o objeto.

URLs pré-assinadas

Execute localmente e confira a validade

2. Rode no PowerShell, na pasta do arquivo

```
npm.cmd init -y
npm.cmd install @aws-sdk/client-s3 @aws-sdk/s3-request-presigner
node acesso.mjs
```

Resultado esperado

A saída é 'Autorizacao, objeto e prazo conferidos'. O teste confere a presença dos campos necessários, o objeto e a recusa local. Ele não valida a assinatura contra a AWS nem a existência do comprovante.

Caso de erro • Assine antes de autorizar

Se retirar a condição da titularidade, a primeira asserção falha porque Beto também recebe uma URL. Restaure a verificação. O serviço de produção precisa aplicar a política completa da reserva e considerar o estado corrente do documento.

Variação • A URL expirou

Se o navegador voltar a pedir o comprovante após o prazo, a aplicação pode buscar uma nova URL mediante nova autorização. Para verificar em teste integrado, use credenciais de função e um bucket de teste. Compare leitura antes do prazo, nova requisição após a expiração e emissão negada para outra conta.

Critérios para conferir

A recusa ocorre antes da assinatura. A URL é para o objeto esperado, com expiração declarada de 120 segundos. O laboratório não transmite credenciais, arquivos nem URLs ao vivo. A validação da política efetiva do bucket exige teste na AWS.

Referências primárias

[1] AWS • Segurança e expiração de URLs pré-assinadas no S3

[2] AWS • Exemplos do S3 com SDK JavaScript v3

URLs pré-assinadas

Trate a URL como acesso temporário ao objeto

Autorize antes de assinar

Antes de gerar a URL, valide a reserva e a permissão da sessão. Use a chave prevista no fluxo e prazo curto. Uma URL assinada pode ser reutilizada enquanto válida, então mantenha a confirmação do upload idempotente e valide o objeto recebido.

Exercício adicional

O upload demorou para começar e a URL expirou. O cliente deve criar outro anexo?

Resolução

Peça uma nova assinatura para a tentativa existente após revalidar o acesso. Se o objeto já tiver sido recebido, valide-o e finalize o vínculo. Assim a repetição não multiplica registros.

Ao assistir ao vídeo

Separe a emissão pelo backend da transferência feita pelo navegador.

Vídeo complementar

Amazon S3 Presigned URLs Uploads and Downloads (em inglês)