

Static Web

Transcrição

Agora que já sabemos quais são os princípios do S3, sobre essa divisão entre *buckets* e **objetos**, veremos outra opção do *bucket*.

Vamos acessar a nossa conta no S3, o *bucket* criado anteriormente (que no caso foi **alura-s3**) e as propriedades deste balde na aba "Properties". Percebemos que ele apresenta as mesmas opções disponibilizadas na criação do *bucket*, como a habilitação do **versionamento** e o **logging**, e também a opção de criar um **host** para uma página web estática.

Isso significa que o *bucket* pode atuar como um servidor simples, sem nenhuma tecnologia *back-end* ou linguagem de programação, sendo simplesmente uma página estática, para o caso de precisarmos criar um guia para algum usuário nosso, algo nesse sentido.

Utilizando o *bucket* como host

Para entendermos melhor, habilitaremos essa opção em "Static website hosting", clicando em "Disabled". São mostradas três opções, sendo a última a que atualmente está ativa, ou seja, é a opção de desabilitação do hospedamento do nosso site. Então, marcaremos a primeira opção para habilitá-la. Nos será perguntado qual será o arquivo de *index*, o HTML principal do nosso *bucket*.

Faremos essa configuração mesmo não tendo este arquivo, preenchendo esse campo com "index.html". Além disso, há uma página de erro, se por exemplo alguém fizer uma requisição para um recurso, uma URL não existente. Nesse caso, vamos dar o nome de "erro.html".

Ainda temos a opção de criar regras de redirecionamento e para onde esse redirecionamento irá ocorrer, mas não habilitaremos isso. Então, clicamos em **Save** e pronto, o nosso *bucket* já serve como uma página estática.

Acessando a página

Quando clicamos novamente na opção "Static website hosting", na parte de cima aparece um *endpoint*, vamos copiá-lo e acessá-lo no nosso navegador para ver o que ele nos mostra. Veremos que ele está mostrando um erro **403**, um erro gerado pelo lado do cliente, sendo que na verdade ele não causou erro nenhum, pois ainda não temos nada configurado, e não temos sequer acesso à página, sinal de que está faltando alguma configuração no *bucket*.

Configuramos a página principal e a de erro, mas não temos os respectivos arquivos. Então, vamos fazer o upload desses dois arquivos.

Caso queira, você pode baixar o `index.html` [aqui \(https://s3.amazonaws.com/caelum-online-public/617-aws-s3/02/arquivos/index.html\)](https://s3.amazonaws.com/caelum-online-public/617-aws-s3/02/arquivos/index.html), e o `erro.html` pode ser baixado [aqui \(https://s3.amazonaws.com/caelum-online-public/617-aws-s3/02/arquivos/erro.html\)](https://s3.amazonaws.com/caelum-online-public/617-aws-s3/02/arquivos/erro.html).

O arquivo principal é o mesmo utilizado nos vídeos anteriores. E no arquivo de erro temos uma página bem simples com uma mensagem de erro. Faremos o upload dos arquivos para nosso *bucket*, exatamente como fizemos anteriormente.

Com os arquivos no *bucket*, poderemos acessar novamente o *endpoint* para verificar se conseguimos usar a página, mas o erro 403 continua aparecendo. Para verificar o que está acontecendo, acessaremos o `index.html` de dentro do *bucket* e perceberemos que ele ainda está com a opção "Make public" habilitada, ou seja, ele ainda não é público, que é o padrão para todos os objetos enviados para o *bucket*.

Tornando os objetos públicos

Então, se precisamos que outras pessoas acessem esse arquivo, precisamos torná-lo público. Feito isto, verificamos novamente o *endpoint*, e a página que queríamos carregar aparecerá corretamente - é um serviço simples, mas que já ajuda.

Vamos ver se a página de erros está funcionando? Na URL, vamos adicionar um `/e`, que sabemos que o *bucket* não possui. Continuamos com o erro 403; não era para a página de erro que tínhamos visto ser apresentada? Ocorreu o mesmo que tinha acontecido com o objeto `index.html`, isto é, a página `erro.html` ainda não está pública.

Portanto, faremos o mesmo, marcando o objeto como público e verificando se quando fazemos uma requisição para algo inexistente a página de erro é exibida. Novamente acessaremos a URL adicionando um `/e` e, como queríamos, a mensagem de erro é mostrada.

Resumindo, sempre que precisamos usar um objeto que outros usuários precisarão acessar, temos que torná-lo público, marcando a opção "Make public". É uma dica, talvez não usemos isso sempre, mas é mais um serviço que a Amazon nos oferece com o S3, uma página estática.

Claro que talvez usemos coisas mais robustas em uma aplicação web, como aplicação PHP, Python, Ruby, Java, aí teremos outros serviços da Amazon, como o S2. Existe um curso na Alura sobre isso: [Amazon EC2: Faça um deploy da sua webapp com alta disponibilidade e escalabilidade \(https://cursos.alura.com.br/course/introducao-ao-cloud-do-ec2-no-aws\)](https://cursos.alura.com.br/course/introducao-ao-cloud-do-ec2-no-aws).