

06

AWS CLI Objects

Transcrição

Agora que já conhecemos os comandos relacionados à criação, remoção, listagem de *buckets* do S3, vamos trabalhar com comandos que serão úteis caso queiramos automatizar o processo de envio de imagens para o balde do S3 da Alura. A ideia agora é conhecer os comandos que precisamos para a automação.

No terminal, para relembrarmos, executaremos o comando `aws s3 ls` para ver quais *buckets* temos na nossa conta.

Agora, queremos ver o que tem dentro do *bucket* `alura-s3`. Para isso usaremos `aws s3 ls s3://alura-s3`, reforçando que no mundo do AWS representamos o *bucket* através da `S3Uri`. Veremos que no momento há 3 arquivos nesse *bucket*.

Se queremos manipular os objetos neste *bucket*, precisaremos conhecer comandos que nos permitam fazê-lo.

Fazendo o download de um objeto

O primeiro passo é copiar o arquivo `bucket.png` para a nossa máquina local. Para facilitar, pode-se acessar o diretório da nossa máquina em que colocaremos o arquivo do S3. E para copiar o arquivo do *bucket* utilizamos o comando `cp`, que serve tanto para fazer downloads como uploads de arquivos. No nosso caso, queremos fazer o download:

```
aws s3 cp s3://alura-s3/bucket.png
```

Por fim, informaremos onde queremos guardar o arquivo que estamos copiando, e representaremos o nosso diretório atual através do ponto, `.`:

```
aws s3 cp s3://alura-s3/bucket.png .
```

Perceberemos pela saída deste comando que o arquivo foi baixado, e ao verificarmos o diretório da nossa máquina, veremos que o arquivo está lá. Então, para realizarmos um download, usaremos o comando `cp`, que nos ajudará neste sentido.

Fazendo o upload de um objeto

O mesmo serve para o upload, mas antes criaremos o arquivo `teste.txt`, com um conteúdo simples, só para testarmos, mesmo. Arquivo criado, voltaremos ao terminal para enviar esse arquivo para o nosso *bucket*.

Para o upload, devemos informar a fonte, o arquivo que será enviado ao *bucket* e o seu destino, ou seja, a `S3Uri` do *bucket* para o qual esse objeto será enviado:

```
aws s3 cp teste.txt s3://alura-s3
```

Isso fará com que se mostre a saída do arquivo, que foi enviado ao nosso *bucket*. A nível de verificação, executaremos `aws s3 ls s3://alura-s3`, com que confirmaremos que o arquivo se encontra no nosso *bucket*. Então, estamos bem próximos da automatização do processo de upload de arquivos, por meio do comando `cp`, bastando saber quais os arquivos que precisaremos subir.

Removendo e renomeando objetos

Dentre os comandos básicos para lidarmos com objetos, existe o comando de remoção. Por exemplo, para deletar o objeto que acabamos de subir, utilizaremos o comando `rm`:

```
aws s3 rm s3://alura-s3/teste.txt
```

Da mesma forma que podemos enviar e remover um objeto, é possível movê-lo ou até mesmo renomeá-lo. Fazemos isso com o comando `mv`. Faremos o upload do arquivo `teste.txt` novamente para dentro do *bucket*, e vamos renomeá-lo para `testando.txt`:

```
aws s3 mv s3://alura-s3/teste.txt s3://alura-s3/testando.txt
```

Esse comando serve tanto para mover um objeto de um *bucket* para outro, quanto para renomeá-lo. Em nosso caso, moveremos o objeto para o mesmo *bucket*, mas com um nome diferente, acontecendo assim o processo de renomeação.

No entanto, é bem raro precisarmos desses comandos, já que normalmente há uma área de trabalho em que queremos manter um **sincronismo** com o que existe no *bucket*. Logo, não queremos nem precisamos tratar arquivos individualmente. Seria muito mais interessante se conseguíssemos sincronizar um *bucket* do S3 com um diretório local da nossa máquina. Como poderemos fazer isto?

Sincronizando o *bucket* com um ambiente local

Vamos trabalhar em um diretório vazio, portanto apagaremos os arquivos do diretório atual ou criaremos um novo, em que entraremos.

Queremos baixar todo o conteúdo do *bucket* para esse diretório e, para isso, no AWS CLI existe o comando do S3, `sync`, de *synchronized*. No nosso caso, queremos sincronizar **o quê e aonde?** Lembrando que para qualquer dúvida sobre o comando, temos sempre a ajuda do AWS, bastando adicionar `help` ao comando, por exemplo:

```
aws s3 sync help
```

Em se tratando do `sync`, ele nos permite sincronizar arquivos da nossa máquina no *bucket*, ou vice versa. Vamos visualizar isso com mais clareza sincronizando os arquivos localizados no *bucket* `alura-s3` com os do diretório `alura-s3` da nossa máquina:

```
aws s3 sync s3://alura-s3 alura-s3
```

Aqui, o diretório `alura-s3` é criado, e o download dos objetos que estavam contidos no nosso *bucket* é feito. Se executarmos o comando `ls` no terminal, notaremos o diretório `alura-s3` e, ao entrarmos nele e executarmos novamente um `ls`, poderemos ver os arquivos que acabamos de baixar.

Como o próprio nome diz, isso serve para mantermos uma sincronia, para que tudo que fizermos localmente seja refletido no *bucket*, e vice versa.

Sincronizando um ambiente local com o *bucket*

Para simularmos isso, na pasta **alura-s3** criaremos um novo arquivo chamado `testando2.txt`. Obviamente isso ainda não está refletido no *bucket*, como podemos ver ao listarmos seus itens:

```
aws s3 ls s3://alura-s3
```

Então, na pasta **alura-s3** sincronizaremos o que temos localmente com o *bucket*, executando novamente o `sync`:

```
aws s3 sync . s3://alura-s3
```

Isso faz com que os arquivos novos da nossa máquina local, que ainda não se encontravam no *bucket*, sejam transferidos, e agora façam parte do mesmo. Isto é, fizemos um upload de forma automática, sem precisarmos ficar executando o comando `cp`.

Sincronizando remoções de objetos

Só precisaremos ficar atentos com alguns detalhes ao trabalharmos com sincronismo, pois estaremos com os mesmos arquivos localmente e no *bucket*. Na nossa máquina local, deletaremos o arquivo `testando.txt`. Porém, ele continua no *bucket*, porque o sincronismo não é realizado de forma automática, e temos que usar o comando `sync`.

Para que essa mudança seja refletida no *bucket*, bastaria executar o mesmo comando usado anteriormente, certo? Mas executamos o comando e... nada acontece! Ou seja, removemos o arquivo da nossa máquina, mas essa mudança não foi refletida no *bucket*.

Quando fazemos uma remoção e queremos que isso fique devidamente sincronizado, precisamos indicar isso incluindo a opção de `delete` ao comando:

```
aws s3 sync . s3://alura-s3 --delete
```

Ao executarmos este comando, perceberemos na saída que o arquivo foi deletado.

O sincronismo funciona muito bem quando precisamos criar algo e mantê-lo atualizado. Significa que se temos uma versão nova no *bucket*, poderemos trazê-la com um `sync`, ou no caso de qualquer mudança de criação ou atualização. O sincronismo funciona sem colocarmos nenhuma *flag*, que só usaremos quando precisamos atualizar com alguma remoção, no caso, `--delete`.

Começamos a criar insumos para conseguirmos automatizar o processo de envio das imagens. Mas para realmente fazermos isso, iremos estudar novos assuntos mais à frente. Até lá!