

 05

## Obtendo os arquivos salvos no Bucket

### Transcrição

Conseguimos enviar as informações do *backup* para o bucket da Amazon. Agora, precisamos fazer o sentido inverso, que é pegar essas informações do bucket e restabelecer no banco de dados, caso esse banco venha apresentar problemas nas tabelas de *produtos* e *usuarios*.

O primeiro passo é criar um diretório para armazenar as informações que virão do bucket dentro da `/home`.

```
$ cd  
$ mkdir restore_mutillidae_amazon
```

Uma vez criado esse diretório, podemos voltar para o diretório de Scripts e começar a criar um novo script que vai pegar de volta as informações do bucket da Amazon.

```
$ cd Scripts/  
$ nano restore_amazon.sh
```

Como sabemos, a primeira linha do script é o **interpretador**.

```
#!/bin/bash
```

Agora, colocaremos o caminho completo do diretório `restore_mutillidae_amazon`, salvando-o em uma **constante**, para facilitar a manipulação:

```
#!/bin/bash  
  
CAMILHO_RESTORE=/home/rafael/restore_mutillidae_amazon
```

Vamos salvar a informação do bucket no diretório que acabamos de criar. Para isso, precisamos sincronizar a **origem** (*bucket*) no **destino** (*diretório local*). Utilizaremos o comando a seguir:

```
#!/bin/bash  
  
CAMILHO_RESTORE=/home/rafael/restore_mutillidae_amazon  
aws s3 sync s3://curso-shell-script/$(date + %F) $CAMILHO_RESTORE
```

Quando essa linha for executada, todo o conteúdo que estiver dentro dessa pasta com a data atual, virá para o diretório `restore_mutillidae_amazon`. Uma vez que o conteúdo tiver lá no diretório, precisamos verificar o que o usuário irá passar como parâmetro para nós. O usuário ficará encarregado de dizer qual tabela que ele quer restabelecer no banco da *Mutillidae*.

Para chegar até esse diretório, colocamos:

```
#!/bin/bash

CAMINHO_RESTORE=/home/rafael/restore_mutillidae_amazon
aws s3 sync s3://curso-shell-script/$(date + %F) $CAMINHO_RESTORE

cd $CAMINHO_RESTORE
```

Agora, o usuário deverá especificar, ou a tabela `produtos` ou a tabela `usuarios`. Entretanto, como já vimos, ele pode passar um nome qualquer lá, sendo necessário fazer a validação. Verificaremos se o parâmetro passado de fato vai existir aqui no diretório `restore_mutillidae_amazon`.

```
#!/bin/bash

CAMINHO_RESTORE=/home/rafael/restore_mutillidae_amazon
aws s3 sync s3://curso-shell-script/$(date + %F) $CAMINHO_RESTORE

cd $CAMINHO_RESTORE
if [ -f $1.sql ]
then
    mysql -u root mutillidae < $1.sql
```

Então, se o arquivo( `-f` ) passado como parâmetro( `$1.sql` ) existir, acessaremos o banco `mysql` com o usuário `root`, dentro do banco `mutillidae` e realizar o `restore` da tabela passada pelo usuário( `$1.sql` ).

Caso não seja um arquivo válido e ele não exista dentro do diretório, mostraremos a mensagem de que o arquivo procurado não existe:

```
cd $CAMINHO_RESTORE
if [ -f $1.sql ]
then
    mysql -u root mutillidae < $1.sql
else
    echo "O arquivo procurado não existe no diretório"
fi
```

Vamos fazer mais uma melhoria? Então, uma vez que o comando do `restore` for realizado, vamos verificar se de fato, foi feito com sucesso, ou se ele teve algum problema nesse processo de restabelecer as informações. Como podemos validar isso? Pelo **status de saída!!**

```
#!/bin/bash

CAMINHO_RESTORE=/home/rafael/restore_mutillidae_amazon
aws s3 sync s3://curso-shell-script/$(date + %F) $CAMINHO_RESTORE

cd $CAMINHO_RESTORE
if [ -f $1.sql ]
then
    mysql -u root mutillidae < $1.sql
    if [ $? -eq 0 ]
    then
        echo "O restore foi realizado com sucesso"
```

```
fi
else
    echo "O arquivo procurado não existe no diretório"
fi
```

Legal! Aparentemente o nosso script está pronto! Vamos salvá-lo e sair. Acessaremos o banco da *Mutillidae*, e vamos **deletar** a tabela *produtos*.

```
$ sudo mysql -u root
> use mutillidae
> delete from produtos;
```

Realizaremos uma rápida consulta só para ver se realmente os dados foram apagados:

```
> select * from produtos;
```

Então conseguimos perder as informações da tabela *produtos*! E felizmente, temos as informações salvas na Amazon. Então vamos pegar as informações de volta. Só que, para pegar essas informações, tínhamos feito a configuração de acesso pelo usuário root. Com isso, o script de fazer o restore deve ser executado pelo usuário root através do comando `$ sudo -i`.

Agora, vamos colocar o comando de acesso ao diretório e executar o script:

```
# cd /home/rafael/Scripts
# bash restore_amazon.sh produtos
```

A mensagem que foi nos retornada é `O restore foi realizado com sucesso!` Bom, se tudo deu certo, devemos ter dentro do diretório o conteúdo *produtos.sql* e também o *usuarios.sql*! E realmente, esses arquivos estão aqui! Foi feito um download desses arquivos que estavam no bucket da Amazon. Só falta verificarmos no banco da *Mutillidae*, e ver se de fato, as informações foram restabelecidas.

```
# mysql -u root
> use mutillidae
> select * from produtos;
```

E a informação foi restabelecida no banco de dados novamente! Então, tanto a tabela *produtos* quanto a tabela *usuarios* está tudo ok! Nós conseguimos pegar o conteúdo do bucket da Amazon, criamos um diretório local para salvar essas informações, e pegamos a tabela *produtos* e fizemos o *restore* e restabelecemos o conteúdo dela no banco *mutillidae*. Tarefa concluída com sucesso!