04

AWS IAM Políticas de Grupo

Transcrição

Conseguimos dar acesso a um instrutor da Alura e criamos um grupo para quando quisermos gerar um novo usuário para outro instrutor, adicionando-o no grupo, o que dará acesso total ao nosso S3.

Isso funciona, mas antes vamos verificar como fazemos para que o usuário `romulo-henrique` , criado anteriormente, tenha acesso à nossa conta. No *dashboard* do IAM (https://console.aws.amazon.com/iam/home) encontraremos o link cujo usuário ou qualquer outro instrutor que estiver cadastrado irá usar para se logar, no campo "IAM users sign-in link". Então, enviaremos esta URL para eles poderem fazer o login.

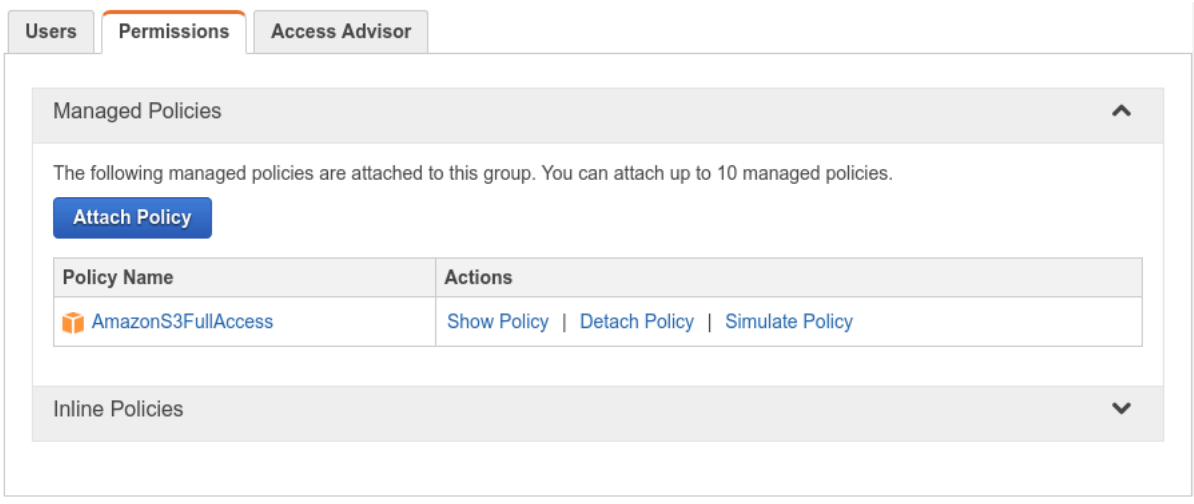
Mas a URL possui vários números, é pouco legível, e não conseguiremos gravar isso. Então a ideia é customizá-la, por meio do botão/link "Customize". Ao clicarmos nesse link, poderemos nomear a URL, chamando-a de "alura" por exemplo, e finalizaremos a alteração clicando em "Yes, Create". Assim, a URL ficará muito mais legível.

Supondo que somos o instrutor Rômulo, abriremos uma página anônima e acessaremos a URL de login. O campo "Account" já vem preenchido e só precisaremos das credenciais do Rômulo para fazermos o login. Feito isso, a partir do usuário, vamos tentar acessar o S3, cuja permissão foi dada. Se o analisarmos, veremos todos os *buckets* da conta, podendo-se inclusive fazer uploads em um *bucket* que ele nem deveria ter acesso. Agora, vamos começar a limitar esse acesso, já que estamos dando acesso total ao usuário em relação a todos os *buckets* da nossa conta.

Limitando o acesso

Retornaremos ao console do IAM e acessaremos a seção Groups (https://console.aws.amazon.com/iam/home#/groups). Vamos analisar o grupo **instrutores**, porque a política dada a um instrutor será replicada em todos eles, isto é, não faz sentido aplicarmos políticas apenas ao Rômulo; poderemos **expandir isso ao grupo**.

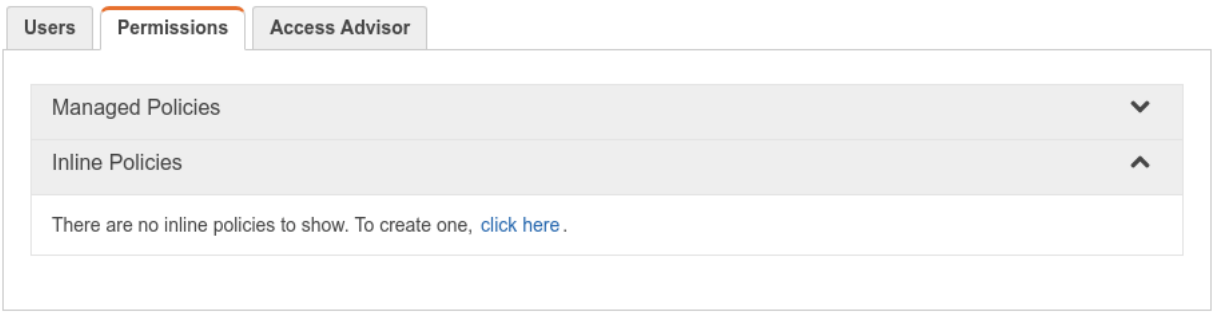
Veremos que já existe uma permissão cadastrada que demos ao criarmos o grupo ***AmazonS3FullAccess***, ou seja, o grupo tem acesso total ao S3, mas não queremos isso.



O que faremos então? Primeiro vamos remover essa política, clicando em "Detach Policy". Agora o Rômulo deixou de ter acesso ao S3, mas ainda queremos permitir seu acesso ao *bucket* `alura-s3` . Sendo assim, precisaremos de uma política mais específica para ele, já que aquelas com que estávamos trabalhando até então abrangiam termos gerais, com total acesso ao S3 ou somente acesso de leitura.

Criando uma política

Se queremos políticas mais estreitas, poderemos **criá-las**, indo a "Inline Policies". Vamos expandir esse campo e clicar em "click here":



Existem duas opções: uma para gerarmos uma política e outra para customizarmos uma já existente. Em "Policy Generator", clicaremos em "Select", e isto faz com que se abra a página de criação de políticas.

No campo "Effect" precisaremos dizer que tipo de política queremos, se de **permissão** (*Allow*) ou de **negação** (*Deny*). Queremos liberar ou controlar, negar o acesso? No nosso caso, queremos permitir, então optaremos por "Allow". No campo "AWS Service" precisaremos selecionar o serviço que iremos permitir, lembrando que ao trabalharmos com o IAM temos diversos serviços da Amazon, e escolheremos o **Amazon S3**.

No campo "Actions", informaremos as ações envolvidas na política, dentre as quais a primeira permissão será a de verificação dos *buckets* da nossa conta. Portanto, selecionaremos a ação "ListAllMyBuckets". Em seguida, no campo "Amazon Resource Name (ARN)", diremos onde esta ação será efetiva.

> Existe uma nomenclatura padrão da Amazon, chamada de ***Amazon Resource Name*** (ARN), que é basicamente um protocolo, um *namespace* usada por ela. Os *namespaces* relacionados ao S3 podem ser vistos aqui (http://docs.aws.amazon.com/pt_br/general/latest/gr/aws-arns-and-namespaces.html#arn-syntax-s3).

Como a nossa ideia é permitir que sejam listados todos os *buckets*, vamos procurar pelo ARN dos *buckets*, que é `arn:aws:s3:::nome_do_bucket` , mas se formos lidar com todos os *buckets* em vez de utilizarmos os seus nomes, usaremos o asterisco (`*`). Logo, vamos utilizar o ARN `arn:aws:s3:::*` .

Agora, poderemos clicar em "Add Statement" e em "Next Step", em que veremos uma prévia e perceberemos que uma política é criada como se fosse um arquivo JSON:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1505251989000",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ]
        }
    ]
}
```

Para entendermos o que foi gerado, a primeira chave é a ***Version***, relacionada à versão da API e políticas da Amazon, não a versão da política que nós criamos. Depois vem o campo "Sid", que é uma identificação cujo nome até poderíamos alterar caso quiséssemos, mas vamos manter o padrão. Em seguida é mostrado o efeito da política, basicamente o que preenchemos no cadastro.

Então, poderemos aplicar a política clicando em "Apply Policy", e verificaremos que o Rômulo conseguirá visualizar os *buckets*, mas ele não possui acesso total ao `alura-s3`. Logo, estará faltando liberar o acesso a este *bucket* para os instrutores.

Liberando o acesso ao *bucket*

Novamente será necessário criarmos outra política. Em "Inline Policies", clicaremos em "Create Group Policy", em "Policy Generator", e depois em "Select". Escolheremos novamente mais um efeito de permissão relacionada ao S3, cuja ação é liberar total acesso aos instrutores, de forma que eles poderão adicionar, atualizar, remover tudo aquilo que for relacionado ao *bucket* `alura-s3`. Para simplificar, vamos marcar a opção que permite todas as ações do S3, `All Actions (*)`. E o ARN relacionado a esta política será do *bucket* `alura-s3`, portanto `arn:aws:s3:::alura-s3`.

Resumindo, estaremos permitindo qualquer ação oriunda de um instrutor e relacionada ao *bucket* `alura-s3`. Então prosseguiremos com a criação da política e logo em seguida verificaremos novamente se o Rômulo conseguirá visualizar os *buckets* e acessar o `alura-s3` e veremos que sim. Mas ao tentarmos subir algum arquivo, ocorre um erro, ou seja, os instrutores ainda não possuem o privilégio para envio e atualização de objetos, pois a política que acabamos de criar existe a nível de *bucket*, e não aos objetos.

Liberando o acesso aos objetos

Precisaremos criar outra liberação, agora para os dados internos do nosso *bucket*. Poderemos editar a política recém criada adicionando-a ao *resource*, com que teremos acesso total ao *bucket* e também a tudo que está contido nele:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1505254613000",
            "Effect": "Allow",
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::alura-s3",
                "arn:aws:s3:::alura-s3/*"
            ]
        }
    ]
}
```

Essa ação total será tanto para o *bucket* em si quanto para os objetos contidos nele. Ao salvarmos a política, tentaremos acessá-lo novamente e fazer o upload de um arquivo no *bucket* `alura-s3` como um instrutor, e o upload é realizado com sucesso. Isto significa que foi necessário colocar esses dois critérios. É um detalhe simples, mas que pode passar despercebido.

Se algum instrutor tentar entrar em outro *bucket*, o acesso será negado, exatamente como queríamos: os instrutores têm total acesso ao `alura-s3`, mas não a qualquer outro *bucket*.

Assim, conseguimos definir regras e políticas de acesso a determinados grupos, no nosso caso, de **instrutores**. E isso será bastante útil em se tratando do uso do S3.