

AWS IAM

Transcrição

Como parte do serviço do S3, geramos chaves para usarmos a nossa conta com o acesso *root*, e vimos que isso é extremamente perigoso, uma vez que existe o risco de perdê-las e, como consequência disso todos poderão controlar os serviços da nossa conta.

Nosso objetivo agora é acessarmos o *bucket* do S3 de forma a conseguirmos prover serviços para nossos instrutores, pra o envio de arquivos de imagens ao fim de todos os dias. Queremos começar a limitar o acesso a determinados instrutores, para que não tenhamos que enviar o acesso *root* do nosso serviço a eles.

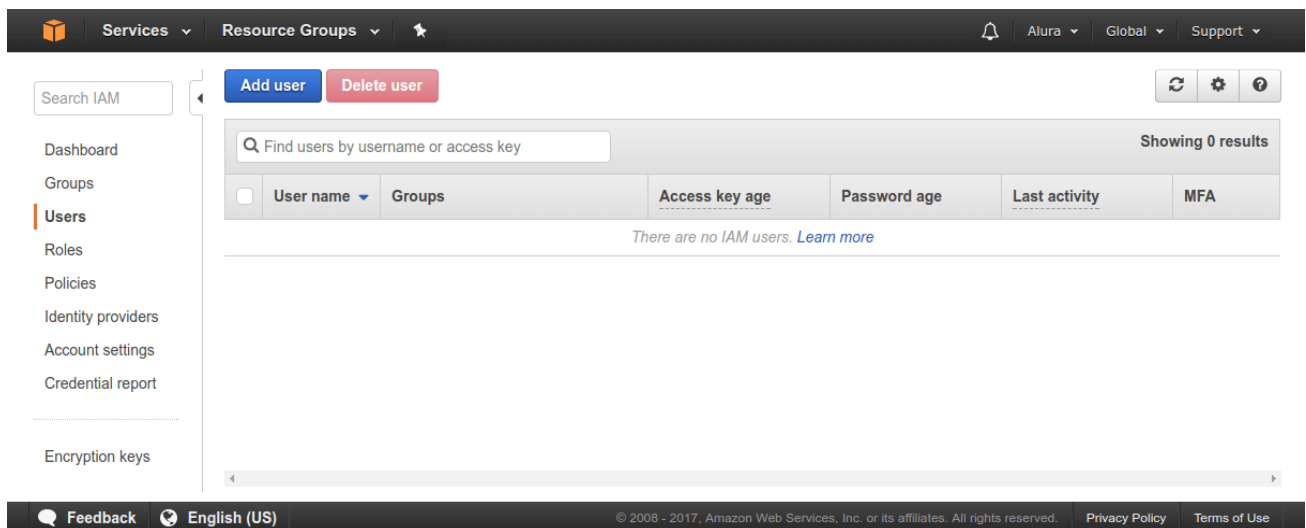
AWS IAM

O AWS possui um serviço pronto para uso, o **IAM** (*Identity and Access Management*). Quando geramos as credenciais anteriormente, vimos um alerta ser exibido solicitando termos cuidado, e que o melhor e mais seguro seria usar o serviço IAM. Então, vamos analisá-lo melhor.

Utilizaremos o IAM para gerarmos usuários para cada instrutor da Alura.

Criando um usuário

Na [página do IAM \(https://console.aws.amazon.com/iam/home\)](https://console.aws.amazon.com/iam/home), no menu à esquerda, há a seção "Users":



Vamos criar um usuário para o instrutor Rômulo Henrique para que ele possa acessar a nossa conta. Clicaremos no botão "Add User" e seremos redirecionados a uma página com formulário para preenchimento dos dados de cadastro do usuário:

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

- Access type* ☐ **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- ☐ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required


[Cancel](#)

[Next: Permissions](#)


Colocaremos o nome do usuário e depois teremos que selecionar quais tipos de acesso do AWS esse usuário terá. Existem basicamente dois tipos, o **programático**, que é quando queremos que o usuário só trabalhe na nossa conta através do CLI ou SDK, e o **console web**, se quisermos que o usuário trabalhe na nossa conta através do navegador. Como ainda não estamos trabalhando de uma forma programática, marcaremos somente a segunda opção.

Depois, é solicitado que criemos uma senha qualquer para o usuário, e deixaremos o *checkbox* "Require password reset" selecionado para exigir que o usuário altere a senha assim que ele acessar o sistema. Em seguida, clicaremos em "Next: Permissions" para ir à segunda etapa, em que daremos permissões para o usuário.


Set permissions for romulo-henrique



Add user to group



Copy permissions from existing user



Attach existing policies directly

i **Get started with groups**
You haven't created any groups yet. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. Get started by creating a group. [Learn more](#)

[Create group](#)

[Cancel](#)

[Previous](#)

[Next: Review](#)

Nessa etapa, é possível darmos alguma permissão específica ao usuário, inclusive através de um grupo. Como no nosso caso teremos vários instrutores e todos terão a mesma permissão de acesso ao *bucket*, o que faremos é criar o grupo "instrutores", clicando no botão "Create group":

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name

Create policy

Refresh

Filter: Policy type

Search

Showing 276 results

	Policy name	Type	Attachments	Description
<input type="checkbox"/>	AdministratorAccess	Job function	0	Provides full access to AWS services and resources.
<input type="checkbox"/>	AmazonAPIGatewayAdministra...	AWS managed	0	Provides full access to create/edit/delete APIs in Amazon API Gateway vi...
<input type="checkbox"/>	AmazonAPIGatewayInvokeFull...	AWS managed	0	Provides full access to invoke APIs in Amazon API Gateway.
<input type="checkbox"/>	AmazonAPIGatewayPushToCl...	AWS managed	0	Allows API Gateway to push logs to user's account.
<input type="checkbox"/>	AmazonAppStreamFullAccess	AWS managed	0	Provides full access to Amazon AppStream via the AWS Management C...
<input type="checkbox"/>	AmazonAppStreamReadOnlyA...	AWS managed	0	Provides read only access to Amazon AppStream via the AWS Managem...

Cancel

Create group

Durante a criação do grupo, na parte inferior da tela, há as políticas de permissão que o grupo poderá ter. Tais permissões não se restringem ao S3, e o IAM abrange todo o serviço do AWS. No nosso caso, queremos trabalhar com políticas relacionadas ao S3, então para filtrarmos, pesquisaremos por "S3", e aparecerão as políticas relacionadas apenas a ele.

Por ora, queremos dar acesso total ao nosso serviço do S3 a todos os instrutores, e depois vamos restringindo isso, então selecionaremos a política "AmazonS3FullAccess" e clicaremos em "Create group". Em seguida, vamos para a próxima etapa, clicando em "Next: Review", onde veremos um *review* das configurações feitas, e assim poderemos finalizar a criação do usuário clicando em "Create user".

Ao fazermos isso, aparecerá um alerta que nos mostra qual link devemos passar ao usuário para que ele consiga fazer o login de acesso ao S3 da Alura. Então, acessaremos esse link em outro navegador ou janela anônima, simulando o acesso desse link pelo usuário. Ele será direcionado a uma tela relacionada à nossa conta do AWS.

Preencheremos o nome do usuário e a senha que criamos anteriormente, partindo do princípio de que já enviamos esses dados para o instrutor. Quando o login é realizado, o AWS exibe o cadastro de uma nova senha, como definimos nas configurações. Com a nova senha cadastrada, o instrutor já poderá acessar seu painel.

É importante notarmos que em nenhum momento exigimos que o Rômulo criasse uma conta do AWS, pois ele será um usuário da nossa conta, pré-existente, via o serviço do IAM.

Verificando o acesso do usuário ao S3, e ao acessá-lo, todos os *buckets* da nossa conta pessoal devem ser mostrados, como esperado. O Rômulo, então, tem total acesso aos *buckets*, inclusive ao `alura-s3`. Caso haja outros baldes na nossa conta, o Rômulo também terá acesso, mas não queremos permitir que ele faça esse tipo de acesso... Isso é assunto para os próximos vídeos!

<https://cursos.alura.com.br/course/aws-s3-manipule-e-armazene-na-nuvem/task/28138>

3/3