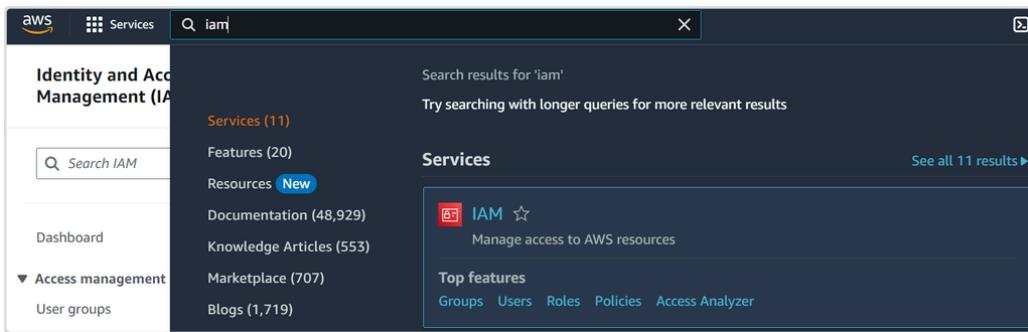


# EMAIL - AWS - Como gerar key e secret

Passo 1: Acesse o Console da AWS

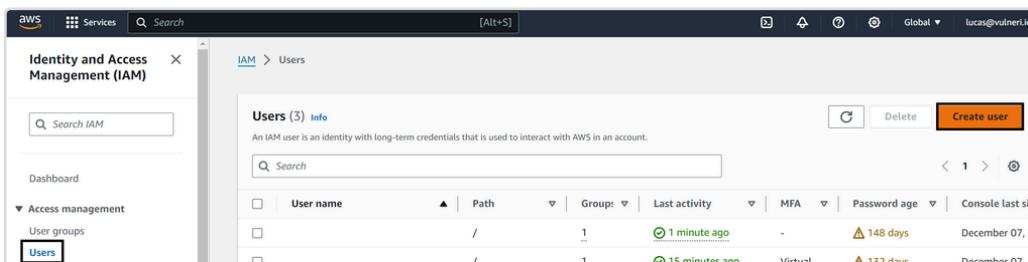
Acesse o Console da AWS em <https://aws.amazon.com/> e faça login na sua conta.

Passo 2: Navegue até o IAM (Identity and Access Management)



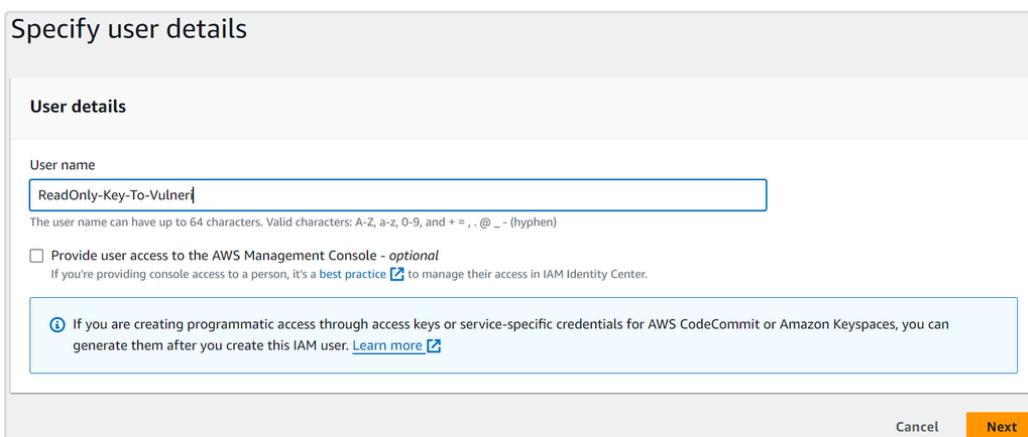
No Console da AWS, vá até o serviço IAM. Você pode encontrá-lo no menu de serviços ou pesquisar por "IAM".

Passo 3: Crie um Usuário



No painel de navegação do IAM, clique em "Usuários / Users" e, em seguida, clique em "Adicionar usuário / Create Users".

Escolha um nome para o usuário (ReadOnly-Key-To-Vulneri) e clique em próximo / Next



User details

User name

ReadOnly-Key-To-Vulneri

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)

Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

[Learn more](#)

Cancel **Next**

Passo 4: Atribua Permissões ao Usuário

No passo "Definir permissões", clique em "Anexar políticas existentes diretamente", em seguida, em Filter by Type, selecione "AWS Managed - job function".

**Set permissions**

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

**Permissions options**

- Add user to group
- Copy permissions
- Attach policies directly

**Permissions policies (2/1191)**

Choose one or more policies to attach to your new user.

Policy name	Type	Attached entities
AdministratorAccess	AWS managed - job function	3
Billing	AWS managed - job function	1
DatabaseAdministrator	AWS managed - job function	0
DataScientist	AWS managed - job function	0
NetworkAdministrator	AWS managed - job function	0
PowerUserAccess	AWS managed - job function	0
<b>ReadOnlyAccess</b>	AWS managed - job function	1
<b>SecurityAudit</b>	AWS managed - job function	1
SupportUser	AWS managed - job function	0
SystemAdministrator	AWS managed - job function	2
ViewOnlyAccess	AWS managed - job function	0

**Set permissions boundary - optional**

Cancel Previous **Next**

Procure e selecione as políticas "ReadOnlyAccess" e "SecurityAudit" e clique em Next.

Essa política fornece permissões de leitura padrão para muitos serviços da AWS.

## Passo 5: Revise e Crie o Usuário

**Review and create**

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

**User details**

User name ReadOnly-Key-To-Vulneri	Console password type None	Require password reset No
--------------------------------------	-------------------------------	------------------------------

**Permissions summary**

Name	Type	Used as
ReadOnlyAccess	AWS managed - job function	Permissions policy

**Tags - optional**

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous **Create user**

Revise as configurações e clique em "Criar usuário".

## Passo 6: Clique para visualizar os detalhes do usuário

Users (3) <a href="#">Info</a>								
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.								
<input type="checkbox"/>	User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Action
<input type="checkbox"/>	/	1	<a href="#">...</a>	54 minutes ago	-	149 days	December 07, 2023	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	/	1	<a href="#">...</a>	7 minutes ago	Virtual	133 days	December 08, 2023	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	<a href="#">ReadOnly-Key-To-Vulnerability</a>	/	0	-	-	-	-	<a href="#">Edit</a> <a href="#">Delete</a>

Passo 7: Clique em “Security Credentials”

Permissions   Groups   Tags   **Security credentials**   Access Advisor

**Console sign-in**

Console sign-in link [Copy](#)   Console password **Not enabled**

[Enable console access](#)

Passo 8: Procure pelo grupo Access keys e clique em “Create access key”

**Access keys (0)**

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

[Create access key](#)

Passo 9: Selecione o tipo “Other” e em seguida clique em next

## Access key best practices & alternatives Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

### Use case

**Command Line Interface (CLI)**

You plan to use this access key to enable the AWS CLI to access your AWS account.

**Local code**

You plan to use this access key to enable application code in a local development environment to access your AWS account.

**Application running on an AWS compute service**

You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

**Third-party service**

You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

**Application running outside AWS**

You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.

**Other**

Your use case is not listed here.



**It's okay to use an access key for this use case, but follow the best practices:**

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access keys when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

Cancel

Next

Passo 10: Coloque uma descrição ( Vulneri Access Key) para a key e em seguida clique em Create access Key

## Set description tag - optional Info

The description for this access key will be attached to this user as a tag and shown alongside the access key.

### Description tag value

Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

Vulneri Access Key

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: \_ . : / = + - @

Cancel

Previous

Create access key

Passo 11: Clique em “Download .csv file” e em seguida em “Done”

## Retrieve access keys Info

### Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

#### Access key

#### Secret access key

AKIA2

\*\*\*\*\* [Show](#)

### Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#)

**Done**

Passo 12 - Abra o arquivo ReadOnly-Key-To-Vulneri\_accessKeys.csv. Perceba que a Access Key compõe a primeira parte da linha 2 (até a vírgula) e a Secret Access Key começa depois da vírgula.

	A
1	Access key ID,Secret access key
2	AKIA2SDFSDKQ5OWETSDF4TE,N3UasdfawGFDRtqfwgvwe4qg4pYsnciAJmPbt

Passo 13 - Envie o arquivo ReadOnly-Key-To-Vulneri\_accessKeys.csv para o email **security@vulneri.io**