

# Gerenciando acesso à AWS com o IAM

"Identity and Access Management"



# Definição

O IAM, é um serviço de configuração de usuários, grupos, políticas e funções de acesso a ações e recursos de uma conta na AWS. Ele é o motor de partida para garantir a funcionalidade em todas as atividades executadas em uma conta AWS.

Sua correta configuração, seguindo as boas práticas sugeridas pela AWS, garante um ambiente bem mais seguro para trabalhar.

A lógica é definir usuários ou grupos que terão acesso à conta da AWS, mediante políticas e/ou funções de permissionamento.

Ou seja, determinar quais ações poderão ser executadas em determinados recursos da AWS.

# Composição

- Usuários (**Users**)
- Grupos (**Groups**)
- Políticas (**Policies**)
- Funções (**Roles**)

# Usuários

Entidades individuais de acesso à conta AWS.

Ao efetuar o login com o usuário ele assume as políticas de permissionamento que estiverem atribuídas à ele.

IAM > Users

## Users (22) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

dev 2 matches

User name	Last activity	MFA	Password age	Active key age	Access key last use
<a href="#">tf_devops_admin-user</a>	4 days ago	Virtual...	-	35 days	4 days ago
<a href="#">tf_devops_jr-user</a>	4 hours ago	Virtual...	-	36 days	4 hours ago

IAM > Users > tf\_devops\_admin-user

## tf\_devops\_admin-user Info

Delete

### Summary

ARN: arn:aws:iam::[redacted]:user/tf\_devops\_admin-user  
Console access: Disabled  
Access key 1: Active (Used 4 days ago, 35 days old)  
Access key 2: Create access key

Created: December 30, 2025, 15:34 (UTC-03:00)  
Last console sign-in: -

### Permissions

Groups: Tags (4) Security credentials Last Accessed

### Console sign-in

Console sign-in link: https://futuraplataforma.signin.aws.amazon.com/console  
Console password: Not enabled  
Enable console access

### Multi-factor authentication (MFA) (1)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more

Type	Identifier	Certifications	Created on
Virtual	arn:aws:iam::[redacted]:mfa/tf-devops-admin	Not Applicable	Tue Dec 30 2025

# Grupos

Entidades de agrupamento de usuários utilizadas para a configuração de permissões que serão compartilhadas entre todos os membros.  
Ou seja: Todos os usuários vinculados à ela possuirão as mesmas permissões.

The screenshot shows the AWS IAM console for the 'devops' user group. The breadcrumb navigation is 'IAM > User groups > devops'. The 'Summary' section includes:

- User group name: devops
- Creation time: December 17, 2025, 09:07 (UTC-03:00)
- ARN: arn:aws:iam::[redacted]:group/devops

Below the summary, there are tabs for 'Users (2)', 'Permissions', and 'Access Advisor'. The 'Users (2)' tab is active, showing a search bar and a list of users:

User name
<a href="#">leonardo</a>
<a href="#">rogerio</a>

The screenshot shows the 'Permissions policies (21)' section for the 'devops' user group. It includes a search bar, a 'Filter by Type' dropdown set to 'All types', and a table of attached policies:

Policy name	Type	Attached entities
<a href="#">AcessarSomenteUSEast2Policy</a>	Customer managed	1
<a href="#">AmazonEC2ReadOnlyAccess</a>	AWS managed	1
<a href="#">AmazonRDSReadOnlyAccess</a>	AWS managed	1
<a href="#">AmazonRoute53FullAccess</a>	AWS managed	1
<a href="#">AmazonSNSReadOnlyAccess</a>	AWS managed	1

# Políticas

Regras de permissionamento ou recusa de acesso a recursos dentro da conta AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2ReadAndStartStop",
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:RebootInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

## Composição:

Sid → Nome do controle que será efetuado

Effect → Informa se a política é de permissão (Allow) ou negação (DENY)

Action → Quais ações de um determinado recurso serão cobertas pela configuração

Resource → Identificação do recurso que será afetado

# Políticas (Exemplos)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAllIfNoMFA",
      "Effect": "Deny",
      "NotAction": [
        "iam:CreateVirtualMFADevice",
        "iam:EnableMFADevice",
        "iam:ListVirtualMFADevices",
        "iam:ListMFADevices",
        "iam:GetUser",
        "iam:ChangePassword",
        "iam:GetAccountPasswordPolicy"
      ],
      "Resource": "*",
      "Condition": {
        "BoolIfExists": {
          "aws:MultiFactorAuthPresent": "false"
        }
      }
    }
  ]
}
```

Nega qualquer ação que não esteja entre as especificadas, **enquanto usuário não estiver logado com a autenticação multi-fator.**

Ele é obrigado a configurar a autenticação para poder ter acesso às demais permissões que possua.

# Políticas (Exemplos)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyCriticalActions",
      "Effect": "Deny",
      "Action": [
        "organizations:*",
        "account:*",
        "billing:*",
        "ec2:Delete*",
        "ecs:Delete*",
        "rds:Delete*"
      ],
      "Resource": "*"
    }
  ]
}
```

Nega o acesso à recursos como organizations, accounts e billing (recursos administrativos da conta) e não permite ações de exclusão nos recursos EC2, ECS e RDS.

# Funções

Configuração de papéis com as políticas de acesso que serão autorizadas para usuários ou serviços que as assumirem.

Podem ser atribuídas diretamente a recursos da AWS ou a usuários (através de uma ação chamada assume role) dando permissão para acessar recursos/serviços que estejam liberados em suas políticas.

The screenshot shows the AWS IAM console page for the role `tf_devops_jr-role`. The breadcrumb navigation is `IAM > Roles > tf_devops_jr-role`. The role's description is "Acesso nível junior para recursos no terraform".

**Summary**

- Creation date:** December 29, 2025, 10:52 (UTC-03:00)
- ARN:** `arn:aws:iam::[redacted]:role/tf_devops_jr-role`
- Link to switch roles in console:** `https://[redacted].console.aws.amazon.com/iam/home?#/roles/tf_devops_jr-role&account=futuraplatforma`
- Last activity:** 4 hours ago
- Maximum session duration:** 12 hours

**Permissions policies (8)**

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type
<input type="checkbox"/> <a href="#">AWSBackupFullAccess</a>	AWS managed
<input type="checkbox"/> <a href="#">Devops-Jr-EFS-Policy</a>	Customer inline
<input type="checkbox"/> <a href="#">Devops-Jr-Restriction-Policy</a>	Customer inline

The screenshot shows the "Trusted entities" section of the `tf_devops_jr-role` page. The breadcrumb navigation is `IAM > Roles > tf_devops_jr-role`. The role's description is "Acesso nível junior para recursos no terraform".

**Trusted entities**

Entities that can assume this role under specified conditions.

```
1- [{"
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Principal": {
7-         "AWS": "arn:aws:iam::[redacted]:user/tf_devops_jr-user"
8-       },
9-       "Action": "sts:AssumeRole",
10-      "Condition": {
11-        "IpAddress": {
12-          "aws:SourceIp": [
13-            "[redacted]/32",
14-            "[redacted]/32"
15-          ]
16-        }
17-      }
18-    }
19-   ]
20- }
```

# Assumindo Funções (Assume Role)

Assume Role, não é nada mais do que assumir uma função previamente configurada com determinadas permissões de acesso.

Antes de mais nada, precisa existir um usuário IAM configurado, sem políticas atribuídas e com credencial de acesso criada para ter acesso a um access key e um secret access key que posteriormente permitirá que obtenhamos as permissões de acesso para nossa máquina. Também é possível tornar o acesso ainda mais seguro habilitando o acesso multi-fator ao seu usuário (MFA).

Para este processo nós Usamos as funções com as políticas de acesso criadas no IAM para que usuários ou serviços possam assumam estas configurações.

# Assumindo Funções (Assume Role)

Após efetuadas as configurações de usuário e funções, precisamos executar comandos do AWS CLI (Interface de linhas de comando da AWS) para aplicar essas funções para nosso usuário e assim termos acesso aos recursos liberados por ela.

O primeiro passo é configurar as credenciais do usuário que poderá assumir a função.

Para isso podemos usar alguns recursos como:

- Setar a credencial no arquivo de configurações da aws usando o AWS Configure
- Setar manualmente a variável de ambientes do sistema operacional

**Como exemplo usaremos o powershell para a demonstração**

# Assumindo Funções (Assume Role) - Demonstração

Setaremos manualmente as credenciais e em seguida verificaremos se o usuário está configurado com o comando **aws sts get-caller-identity**.

```
$env:AWS_ACCESS_KEY_ID      = 'AKIAHFH6FGH213GFFG'  
$env:AWS_SECRET_ACCESS_KEY = 'FGFGDF564FDG52FD54FGG'  
$env:AWS_DEFAULT_REGION    = 'us-east-1'
```

```
C:\> aws sts get-caller-identity  
  
{  
  "UserId": "AKIAHFH6FGH213GFFG",  
  "Account": "123456789012",  
  "Arn": "arn:aws:iam::123456789012:user/seu-usuario"  
}
```

# Assumindo Funções (Assume Role) - Demonstração

Feito isso, agora vamos executar o comando que vai assumir para nosso usuário as permissões configuradas para a nossa função.

```
c:\> aws sts assume-role --role-arn
arn:aws:iam::123456789012:role/s3-role --role-session-name
sessao-s3 --duration-seconds 3600 --output json

{
  "Credentials": {
    "AccessKeyId": "ASIA5FEXAMPLE123456",
    "SecretAccessKey":
"wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "SessionToken":
"IQoJb3JpZ2luX2VjEjR////////wEaCXVzLWVhc3QtMSJHMEUCIQDzExample
SessionTokenPart1ExampleSessionTokenPart2ExampleSessionTokenPart
3",
    "Expiration": "2026-01-05T15:42:31Z"
  },
  "AssumedRoleUser": {
    "AssumedRoleId": "AROEXAMPLEID:sessao-s3",
    "Arn": "arn:aws:sts::123456789012:assumed-role/s3-
role/sessao-s3"
  }
}
```

- ARN da nossa Função AWS;
- Nome para a sessão;
- Tempo em segundos que permanecerá ativa (neste caso 3600s = 1h).

# Assumindo Funções (Assume Role) - Demonstração

Para assumir a função limpamos as credenciais da nossa sessão e informamos as novas credenciais retornadas pelo comando **aws sts assume-role**.

```
Remove-Item Env:AWS_ACCESS_KEY_ID -ErrorAction SilentlyContinue
Remove-Item Env:AWS_SECRET_ACCESS_KEY -ErrorAction
SilentlyContinue
Remove-Item Env:AWS_SESSION_TOKEN -ErrorAction SilentlyContinue
```

```
$env:AWS_ACCESS_KEY_ID      = 'ASIA5FEXAMPLE123456'
$env:AWS_SECRET_ACCESS_KEY =
'wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY'
$env:AWS_SESSION_TOKEN     =
'IQoJb3JpZ2luX2VjEJr////////wEaCXVzLWVhc3QtMSJHMEUCIQDzExample
SessionTokenPart1ExampleSessionTokenPart2ExampleSessionTokenPart
3'
$env:AWS_CREDENTIAL_EXPIRATION = '2026-01-05T15:42:31Z'
```

# Assumindo Funções (Assume Role) - Demonstração

Após isso, executando novamente comando **aws sts get-caller-identity**, não serão mais retornadas informações sobre o usuário, mas sim sobre a função assumida.

```
C:\> aws sts get-caller-identity

{
  "UserId": "AROA7BFF9FGHFG2GX1XD5D:sessao-s3",
  "Account": "123456789012",
  "Arn": "arn:aws:sts::123456789012:assumed-role/s3-
role/sessao-s3"
}
```

# Assumindo Funções (Assume Role) - Demonstração

A partir deste momento é possível acessar todos os recursos que estiverem configurados para a função .

```
c:\> aws s3 ls
2023-04-11 11:24:11 funcionarios
2023-04-11 14:01:23 fotos
2024-12-30 17:31:37 pagamentos
```

# Assumindo Funções (Assume Role) - Observações

Após vencer o período de expiração ou caso a sessão do Powershell seja fechada e aberta uma nova, não será mais possível acessar os recursos AWS, a menos que refaça todo o processo de autenticação.

Isso parece ruim, mas na verdade é MUITO BOM, pois garante segurança ao acesso na conta AWS. Pois você abre uma sessão, se autentica, faz o que precisa ser feito e ao final o acesso é extinto, prevenindo que terceiros tenham acesso indevido.

Neste exemplo, de forma didática executamos os comandos e informamos valores manualmente. Mas profissionalmente usaríamos scripts para automatizar este processo.

# Importante

Qualquer ação ou recurso que não esteja discriminado nas políticas de um usuário, grupo ou função é automaticamente negado pela AWS.

Fica a cargo do administrador da conta definir os acessos e ações para cada recurso.



