

Laboratório - AWS Security Token Service (STS)

Resumo

Observação: A interface do Console de Gerenciamento da AWS pode sofrer pequenas alterações visuais ao longo do tempo, mas os conceitos e a localização geral dos serviços permanecem consistentes. As instruções neste resumo seguem a estrutura geral das funcionalidades.

Este laboratório prático demonstra o uso de credenciais temporárias na AWS via AWS Security Token Service (STS). Você aprenderá a criar e assumir uma role IAM com permissões restritas, executar scripts Python no CloudShell para gerar credenciais temporárias e validar acessos permitidos/negados.

Objetivo do laboratório

- Criar uma role IAM temporária com `AmazonS3FullAccess`.
- Alternar para a role no console e via CLI.
- Gerar credenciais temporárias via STS.
- Configurar e testar credenciais na AWS CLI.
- Simular expiração de credenciais.
- Compreender o impacto da política de confiança da role.
- Excluir todos os recursos criados.

Cenário:

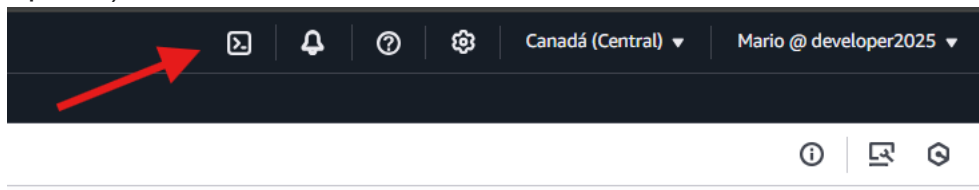
Você atuará como um usuário IAM com permissão para criar e assumir roles temporárias, navegando por etapas práticas no console e no CloudShell.

Pré-requisitos

- Acesso ao Console de Gerenciamento da AWS.
- Usuário IAM (não o root) com permissões para:
 - Criar e gerenciar roles no IAM.
 - Usar o AWS CloudShell (incluindo instalar pacotes como Python e Boto3).
 - Assumir roles.
- Familiaridade básica com a linha de comando (Linux) e o AWS CLI.

Passo 1: Configuração do Ambiente

1. **Abrir CloudShell:** No Console AWS, clique no ícone do CloudShell (canto superior).



2. **Atualizar e Instalar Python:**

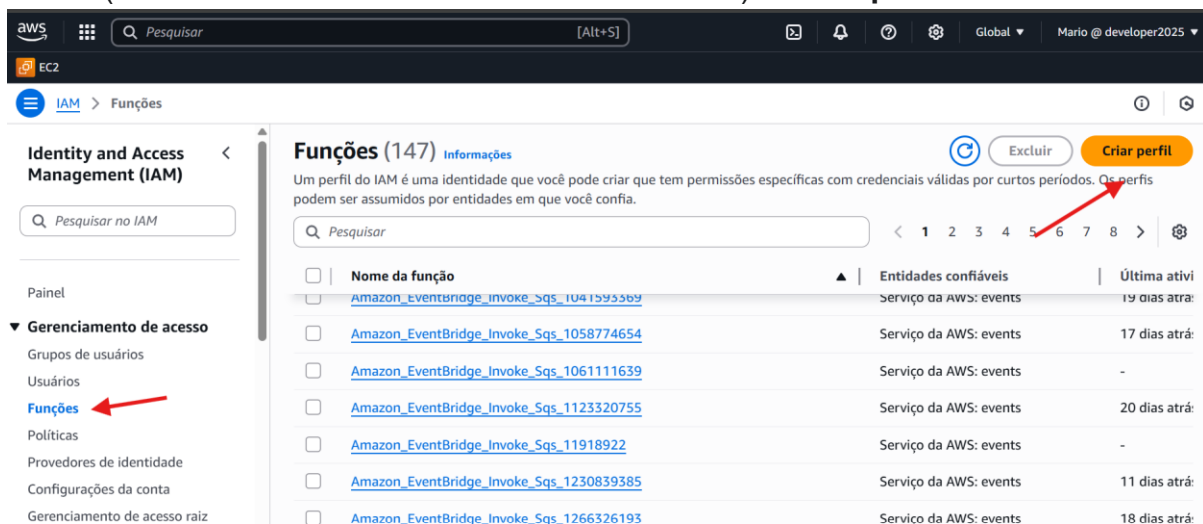
```
sudo yum update -y
sudo yum install -y python3
python3 --version
```

3. **Instalar Boto3:**

```
pip3 install boto3 --upgrade
python3 -c "import boto3; print(boto3.__version__)"
```

Passo 2: Criando uma Role Temporária

1. **Navegar para IAM Roles:** Acesse o console AWS, vá para **IAM > Funções** (no menu lateral "Gerenciamento de acesso") > **Criar perfil**.



2. **Configurar Política de Confiança:**

- a. Em "Tipo de entidade confiável", selecione "Política de confiança personalizada".

Tipo de entidade confiável

- ☐ Serviço da AWS
Permitir que serviços da AWS, como o EC2, Lambda ou outros executem ações nessa conta.
- ☐ Conta da AWS
Permitir que entidades em outras contas da AWS pertencentes a você ou a terceiros executem ações nessa conta.
- ☐ Identidade Web
Permite que os usuários federados pelo provedor de identidade da Web externo especificado assumam essa função para executar ações nessa conta.
- ☐ Federação SAML 2.0
Permitir que os usuários federados com o SAML 2.0 de um diretório corporativo executem ações nessa conta.
- ☒ Política de confiança personalizada
Crie uma política de confiança personalizada para permitir que outras pessoas executem ações nessa conta.

b. Role a página, clique em "Adicionar uma entidade principal".

Política de confiança personalizada

Crie uma política de confiança personalizada para permitir que outras pessoas executem ações nessa conta.

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "Statement1",  
6       "Effect": "Allow",  
7       "Principal": {},  
8       "Action": "sts:AssumeRole"  
9     }  
10  ]  
11 }
```

c. Em "Tipo de entidade principal", selecione "Usuários IAM".

d. Em "ARN", cole o ARN do seu usuário IAM (encontre-o em **IAM > Usuários > Seu Usuário**).

IAM > Usuários > Mario

Identity and Access Management (IAM)

Pesquisar no IAM

Painel

Gerenciamento de acesso

Grupos de usuários

Usuários

Mario Informações

Resumo

ARN

arn:aws:iam::804364734712:user/Mario

Criado

March 20, 2025, 09:06 (UTC-03:00)

Acesso ao console

Habilitado com MFA

Último login no console

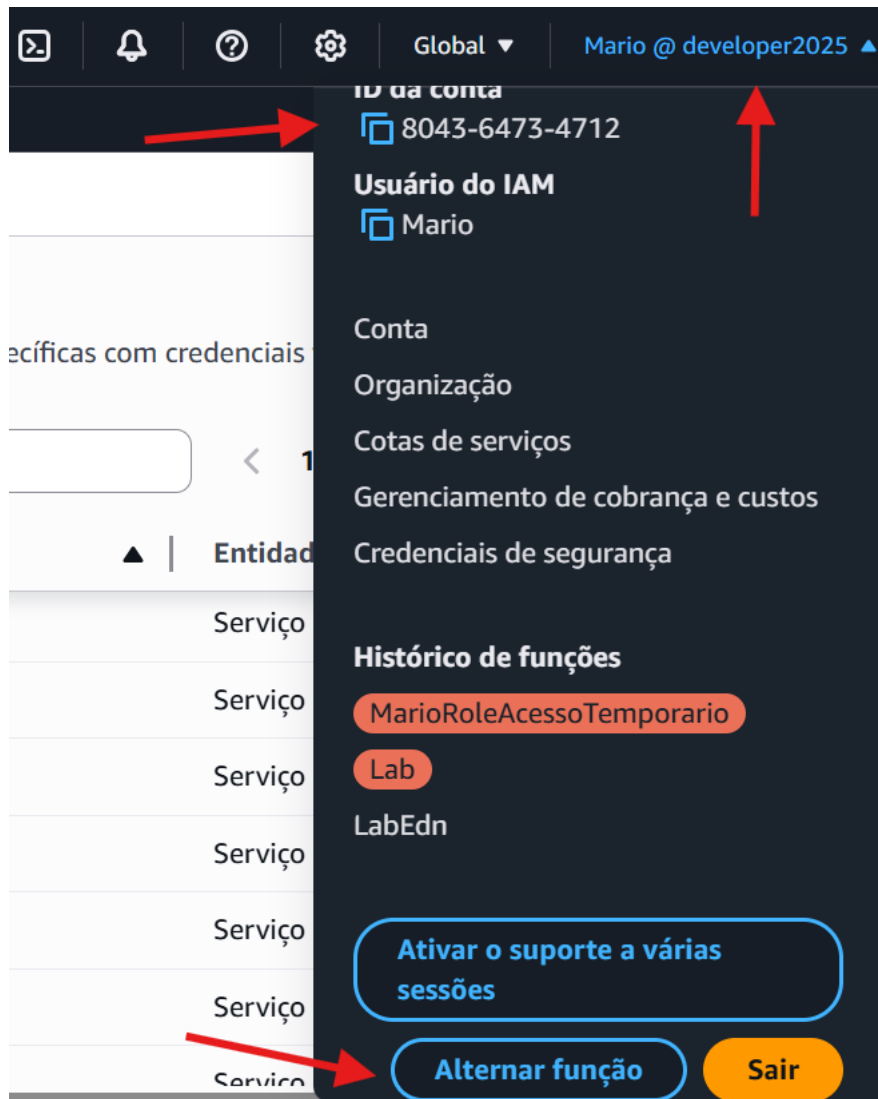
Hoje

e. Clique em "Adicionar entidade principal" e depois em "Próximo".

Passo 3: Alternando para a Role Temporária

1. Alternar no Console:

- Duplique a aba do seu navegador.
- Na aba duplicada, clique no ícone do nome da sua conta (canto superior direito).
- Copie o "ID da conta".
- Clique em "Alternar função".



- Preencha os campos:
 - ID da conta:** Cole o ID da sua conta.
 - Nome do perfil do IAM:** SeuNomeRole
 - Nome de exibição opcional:** SeuNomeAcessoTemporario
 - Cor da tela opcional:** Escolha uma cor.
- Clique em "Alternar perfil".

Switch Role

Switching roles enables you to manage resources across Amazon Web Services accounts using a single user. When you switch roles, you temporarily take on the permissions assigned to the new role. When you exit the role, you give up those permissions and get your original permissions back. [Learn more](#)

Account ID
The 12-digit account number or the alias of the account in which the role exists.

804364734712

IAM role name
The name of the role that you want to assume which can be found at the end of the role's ARN. For example, provide the **TestRole** role name from the following role ARN: `arn:aws:iam::123456789012:role/TestRole`.

MarioSilvaRole

Display name - optional
This name will appear in the console navigation bar when active. Choose a name to help identify the permission set assigned to the role.

MarioRoleAcessoTemporario

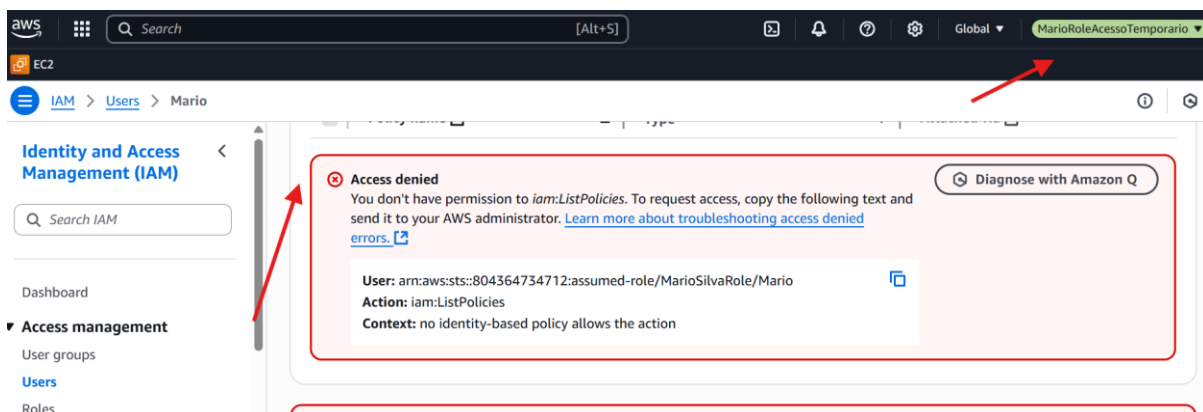
Display color - optional
The selected color displays in the console navigation when this role is active

Green

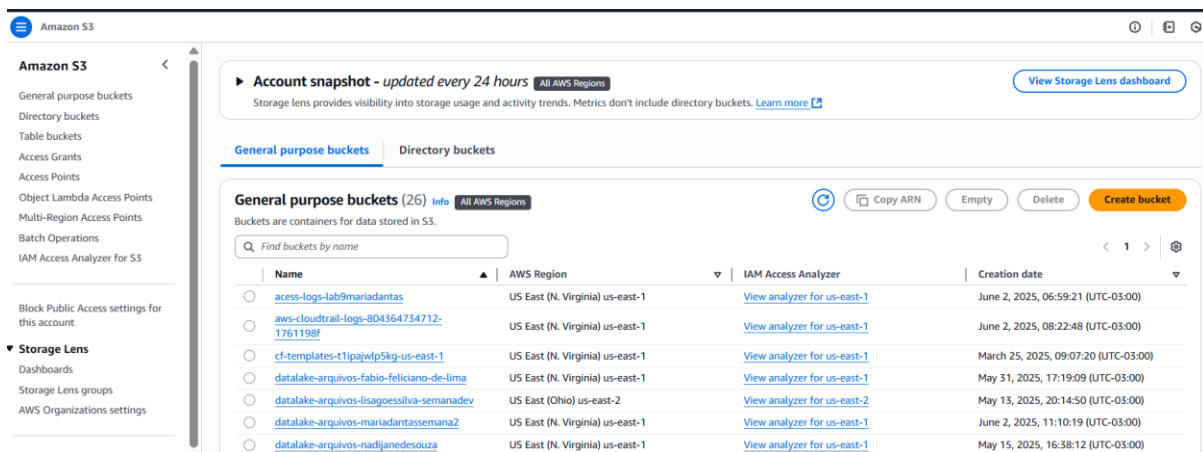
Cancel

Switch Role

- g. **Validação:** Observe que o console mudará de cor e você terá acesso ao S3, mas acesso negado a outros serviços como Lambda.

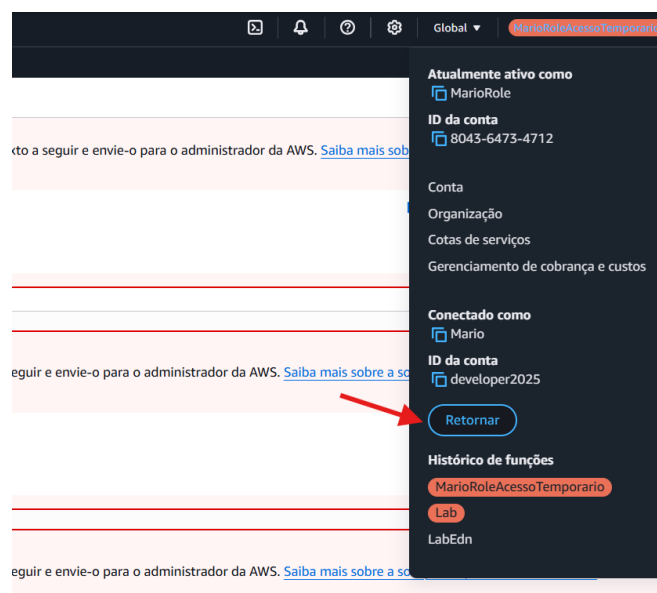


2. **Validar Acesso S3:** Na barra de busca, digite S3 e valide o acesso.



3. Como sair da role e voltar para sua conta original no console da AWS

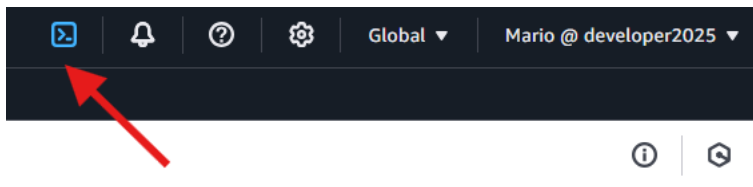
- No canto superior direito do console da AWS, clique no ícone com o seu nome de usuário (ou no nome da role que está em uso).
- No menu suspenso, clique em “Retornar” (ou “Switch back”, se o console estiver em inglês) para voltar à sua conta original.



4. Verificar Identidade no CloudShell:

- Localize o ícone do terminal, ele está à esquerda do ícone do sino (notificações)
- Clique nesse ícone para abrir o AWS CloudShell, o terminal interativo integrado ao console e execute:

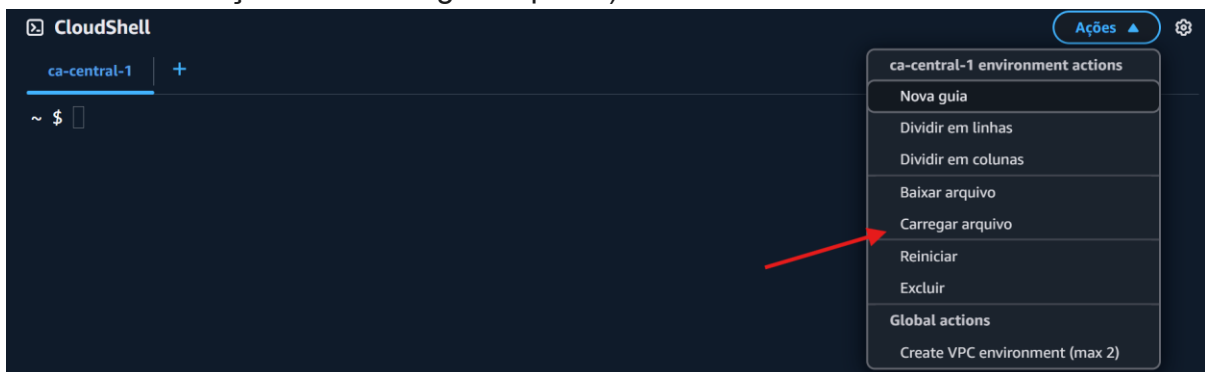
```
aws sts get-caller-identity
```



```
CloudShell
ca-central-1 +
~ $ aws sts get-caller-identity
{
  "UserId": "AIDA3WR672D4JJLLQF4DD",
  "Account": "804364734712",
  "Arn": "arn:aws:iam::804364734712:user/Mario"
}
~ $
```

5. Assumir uma role via AWS CLI:

- Acesse o CloudShell
- Faça o download do arquivo em python [credenciais temporarias.py](#).
- Depois faça o upload do arquivo em python para o CloudShell (botão "Ações" > "Carregar arquivo").



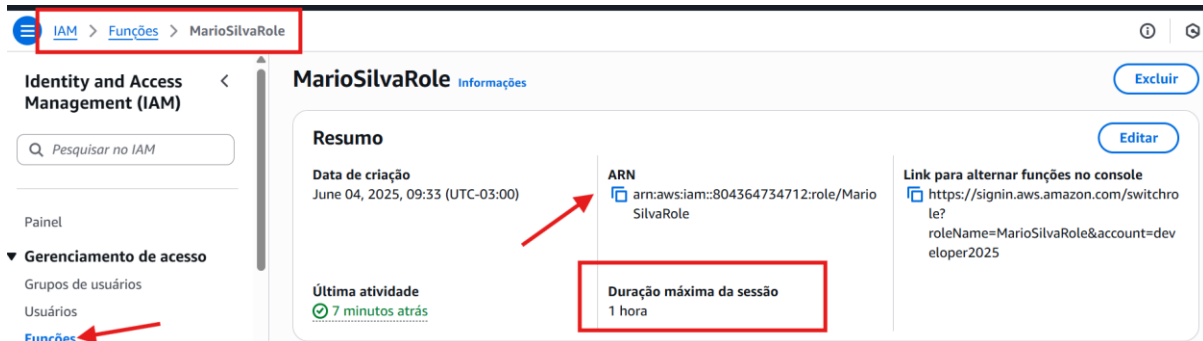
- Verifique o upload com: `ls`

```
CloudShell
ca-central-1 +
~ $ ls
credenciais_temporarias.py
~ $
```


Passo 4: Executar o Script

1. Obter ARN da Role:

- No console, vá para **IAM > Funções**.
- Pesquise e clique em `SeuNomeRole`.
- Copie o **ARN da função**.



No campo “**Duração máxima da sessão**” indica por quanto tempo uma sessão/login permanece ativa antes de expirar. Por padrão, esse tempo é de **1 hora**. Caso queira aumentar esse período, basta clicar em **Editar**.

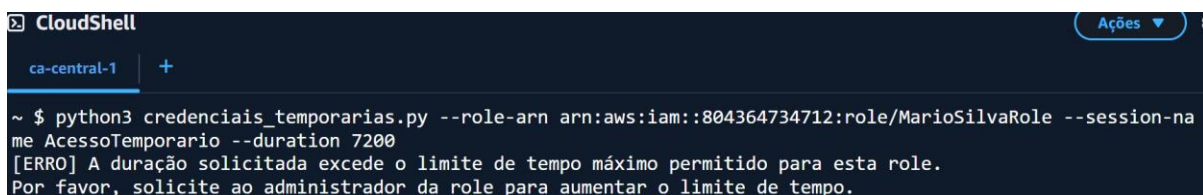
Neste laboratório, a duração da sessão permanece no valor padrão de 1 hora.

2. Testar Duração (Erro Esperado):

- Execute o comando abaixo, substituindo `SEU_ROLE_ARN` pelo ARN copiado e `SeuNomeRole` pelo nome da sua role.

```
python3 credenciais_temporarias.py --role-arn
arn:aws:iam::SEU_ROLE_ARN:role/SeuNomeRole --session-name AcessoTemporario --
duration 7200
```

- Observe o erro, pois a duração padrão máxima é 3600 segundos igual a (1 hora).



3. Executar com Duração Válida:

- Altere `--duration` para 3600 e execute novamente:

```
python3 credenciais_temporarias.py --role-arn  
arn:aws:iam::SEU_ROLE_ARN:role/SeuNomeRole --session-name AcessoTemporario --  
duration 3600
```



```
CloudShell  
ca-central-1 +  
~ $ python3 credenciais_temporarias.py --role-arn arn:aws:iam::804364734712:role/MarioSilvaRole --session-na  
me AcessoTemporario --duration 3600  
AWS Access Key ID: ASIA3WR672D4FKXLWHRL  
AWS Secret Access Key: EIUKx85Qvch4EKG254zAJRiRXYG1AxkThk4o6HNU  
AWS Session Token: FwoGZXIvYXZlEGcaDn0KEiaJEEMlWj+0CK4AQcxRPOziag7tnk7tI6FgSCRS GPOL6VtKIXQN HK8125mq3V0t6f9x  
UCVb3HIwDI6Yq1Pm8Vsb nHW4UlcCs1StJ4cNCVbvHbnf9ZYaSaKDi9OuoIxx3MEPUE7UhXzSoiCHUGEpP7WTVKI9IjA10EpY0gUc jxWvIAN3  
m4B3GNQt71Aq/iw9XPzEk+LHjrPcVHMo12k4eJfJmxdlwU1FQ2QgWb0i0Y4tCYD8SG4+Ry3PqYvAU/67mG/Qsoy4mBwgYyLXCuPTExUZV9m  
5+do6LGwXokotYZjBZJz/Qf0jmt47987xnVT+w0xViGM9it3g==
```

- b. **Copie as informações:** AWS Access Key ID, AWS Secret Access Key, AWS Session Token exibidas no terminal.

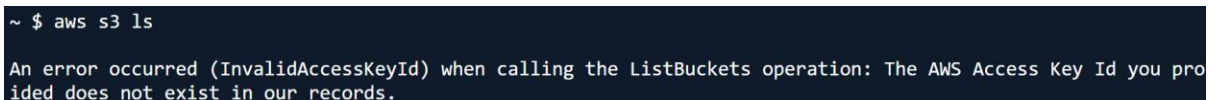
Passo 5: Configurar no AWS CLI

1. **Configurar Credenciais Temporárias:** No CloudShell, digite `aws configure` e preencha:
 - a. AWS Access Key ID [None]: (Cole seu AWS Access Key ID)
 - b. AWS Secret Access Key [None]: (Cole seu AWS Secret Access Key)
 - c. Default region name [None]: `us-east-1`
 - d. Default output format [None]: `json`

2. Testar Acesso S3 (Erro Esperado):

```
aws s3 ls
```

- a. Observe o erro, pois o Session Token ainda não foi configurado.



```
~ $ aws s3 ls  
An error occurred (InvalidAccessKeyId) when calling the ListBuckets operation: The AWS Access Key Id you pro  
vided does not exist in our records.
```

Passo 6: Adicionar o Session Token

1. Editar Arquivo de Credenciais:

```
sudo nano ~/.aws/credentials
```

2. Adicionar Session Token:

- a. Vá para a última linha e adicione:

```
aws_session_token = SEU_SESSION_TOKEN
```

```
ca-central-1 +
GNU nano 8.3 /home/cloudshell-user/.aws/credential
[default]
aws_access_key_id = ASIA3WR672D4FKXLWHRL
aws_secret_access_key = EIUKx85Qvch4EKG254zAJRiRXYG1AxkThk4o6HNU
aws_session_token = FwoGZXIvYXdzEGcaDGn0KEiaJEEMlWj+0CK4AQcxRPoziag7tnk7tI
↑
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

(Substitua `SEU_SESSION_TOKEN` pelo token que você copiou).

- b. Pressione `CTRL+O` e `Enter` para salvar o arquivo e depois `CTRL+X` para sair.

3. Verificar Conteúdo:

```
cat ~/.aws/credentials
```

Passo 7: Testar Permissões

1. Testar Acesso S3 (Permitido):

```
aws s3 ls
```

- a. Deve listar seus buckets S3.

```
~ $ aws s3 ls
2025-06-02 10:09:51 acess-logs-lab9mariadantas
2025-06-02 11:22:50 aws-cloudtrail-logs-804364734712-1761198f
2025-03-25 12:07:21 cf-templates-t1ipajwlp5kg-us-east-1
2025-05-31 20:40:32 datalake-arquivos-fabio-feliciano-de-lima
2025-05-14 00:50:11 datalake-arquivos-lisagoessilva-semanadev
2025-06-02 14:21:52 datalake-arquivos-mariadantassemana2
2025-05-15 20:49:29 datalake-arquivos-nadijanedesouza
2025-05-20 08:53:21 datalake-arquivos-robertmello
2025-04-24 23:29:56 do-not-delete-ssm-diagnosis-804364734712-us-east-1-1h2sh
2025-06-01 23:15:43 elasticbeanstalk-us-east-1-804364734712
2025-05-04 22:22:52 fabrod-bucket-lab3
```

2. Verificar Identidade Atual:

```
aws sts get-caller-identity
```

```
~ $ aws sts get-caller-identity
{
  "UserId": "ARO3WR672D4NFEGSTN4R:AcessoTemporario",
  "Account": "804364734712",
  "Arn": "arn:aws:sts::804364734712:assumed-role/MarioSilvaRole/AcessoTemporario"
}
~ $
```

3. Testar Acesso Lambda (Negado):

```
aws lambda list-functions
```

- a. Deve retornar "acesso negado".

```
~ $ aws lambda list-functions

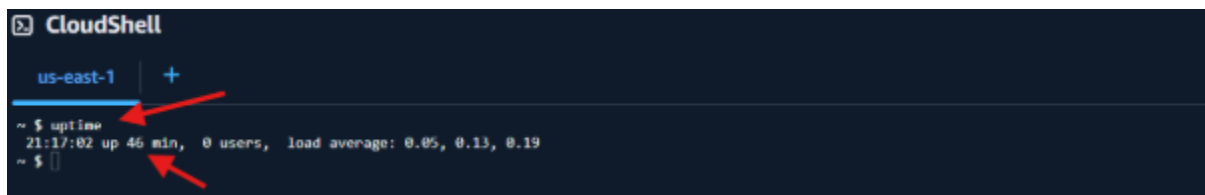
An error occurred (AccessDeniedException) when calling the ListFunctions operation: User: arn:aws:sts::804364734712:assumed-role/MarioSilvaRole/AcessoTemporario is not authorized to perform: lambda:ListFunctions on resource: * because no identity-based policy allows the lambda:ListFunctions action
~ $
```

Passo 8: Simular Expiração

1. Verificar Tempo de Sessão:

```
uptime
```

- a. Observe o tempo de atividade do CloudShell.



```
CloudShell
us-east-1 +
~ $ uptime
21:17:02 up 46 min, 0 users, load average: 0.05, 0.11, 0.19
~ $
```

2. **Testar Acesso Após Expiração:** Após a duração definida (`--duration 3600` = 1 hora), tente novamente:

```
aws s3 ls
```

- a. Deve retornar um erro de credenciais expiradas.

A screenshot of the AWS CloudShell interface. At the top, it says 'CloudShell' and 'us-east-1'. Below that, the command prompt shows '~ \$ aws s3 ls'. A red arrow points to this command. Below the command, a red rectangular box highlights an error message: 'An error occurred (ExpiredToken) when calling the ListBuckets operation: The provided token has expired.' The prompt '~ \$' is visible below the error message.

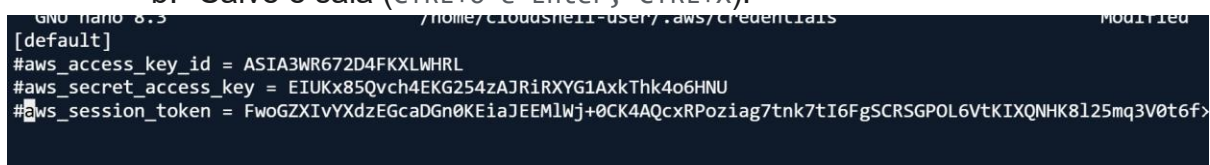
```
CloudShell
us-east-1
~ $ aws s3 ls
An error occurred (ExpiredToken) when calling the ListBuckets operation: The provided token has expired.
~ $
```

Passo 9: Restaurar Credenciais Originais

1. **Editar** ~/.aws/credentials:

```
sudo nano ~/.aws/credentials
```

- a. Comente (#) ou remova as linhas: aws_access_key_id, aws_secret_access_key, aws_session_token.
- b. Salve e saia (CTRL+O e Enter, CTRL+X).

A screenshot of the nano text editor. The top line shows 'GNU nano 2.9.3 /home/cloudshell-user/.aws/credentials'. Below that, the content of the file is shown: '#[default]', '#aws_access_key_id = ASIA3WR672D4FKXLWHRL', '#aws_secret_access_key = EIUKx85Qvch4EK6254zAJRiRXYG1AxkThk4o6HNU', and '#aws_session_token = FwoGZXIvYXZlEGcaDgn0KEiaJEEMlwj+0CK4AQcxRPoziag7tnk7tI6FgSCRS5GPOL6VtKIXQNHK8125mq3V0t6f>'.

```
GNU nano 2.9.3 /home/cloudshell-user/.aws/credentials
[default]
#aws_access_key_id = ASIA3WR672D4FKXLWHRL
#aws_secret_access_key = EIUKx85Qvch4EK6254zAJRiRXYG1AxkThk4o6HNU
#aws_session_token = FwoGZXIvYXZlEGcaDgn0KEiaJEEMlwj+0CK4AQcxRPoziag7tnk7tI6FgSCRS5GPOL6VtKIXQNHK8125mq3V0t6f>
```

2. **Editar** ~/.aws/config:

```
sudo nano ~/.aws/config
```

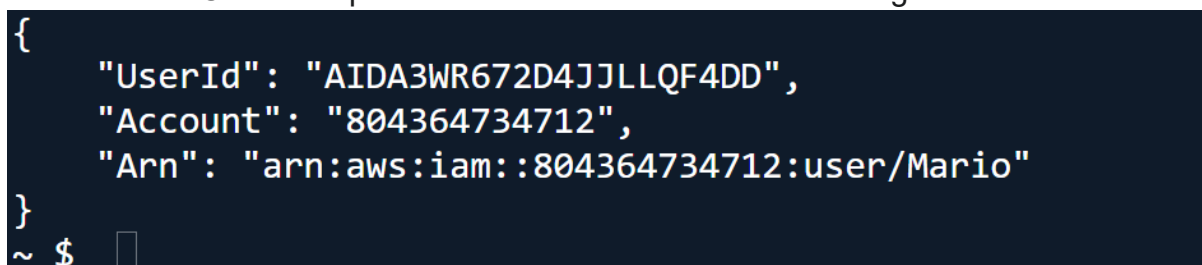
- a. Comente (#) ou remova as linhas: region = us-east-1, output = json.
- b. Salve e saia (CTRL+O e Enter, CTRL+X).

Passo 10: Confirmar Identidade Atual

1. **Verificar Identidade:**

```
aws sts get-caller-identity
```

- a. Confirme que você voltou ao seu usuário IAM original.

A screenshot of a terminal window showing the output of the 'aws sts get-caller-identity' command. The output is a JSON object with three fields: 'UserId', 'Account', and 'Arn'.

```
{
  "UserId": "AIDA3WR672D4JJLLQF4DD",
  "Account": "804364734712",
  "Arn": "arn:aws:iam::804364734712:user/Mario"
}
~ $
```

Passo 11: Acessar a Role no Console – Teste de Segurança - Política de Confiança

1. **Navegar para IAM Roles:** No console, vá para **IAM > Funções**.
2. **Selecionar Role:** Clique em `SeuNomeRole`.

Passo 12: Editar Política de Confiança

1. **Editar Política:** Vá para a aba "Relações de Confiança" > "Editar política de confiança".

MarioSilvaRole [Informações](#) [Excluir](#)

Resumo [Editar](#)

Data de criação
June 04, 2025, 09:33 (UTC-03:00)

ARN
[arn:aws:iam::804364734712:role/MarioSilvaRole](#)

Link para alternar funções no console
<https://signin.aws.amazon.com/switchrole?roleName=MarioSilvaRole&account=developer2025>

Última atividade
✔ 39 minutos atrás

Duração máxima da sessão
1 hora

Permissões **Relações de confiança** Etiquetas Último acesso Revogar sessões

Entidades confiáveis [Editar política de confiança](#)

Entidades que podem assumir essa função em condições especificadas.

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "Statement1",
6       "Effect": "Allow",
7       "Principal": {
8         "AWS": "arn:aws:iam::804364734712:user/teste"
9       },
10      "Action": "sts:AssumeRole"
11    }
12  ]
13 }
```

2. **Modificar JSON:**
 - a. Apague o JSON existente.
 - b. Adicione uma nova instrução (clique em "Adicionar nova instrução").
 - c. Clique para adicionar uma entidade principal e selecione "IAM Roles".
 - d. Informe o ARN da role de um colega (ou outra role que não seja a sua).
 - e. Salve as alterações.

Editar política de confiança

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "Statement1",
6       "Effect": "Allow",
7       "Principal": {
8         "AWS": "arn:aws:iam::804364734712:user/teste"
9       },
10      "Action": "sts:AssumeRole"
11    }
12  ]
13 }
```

Passo 13: Reexecutar o Script

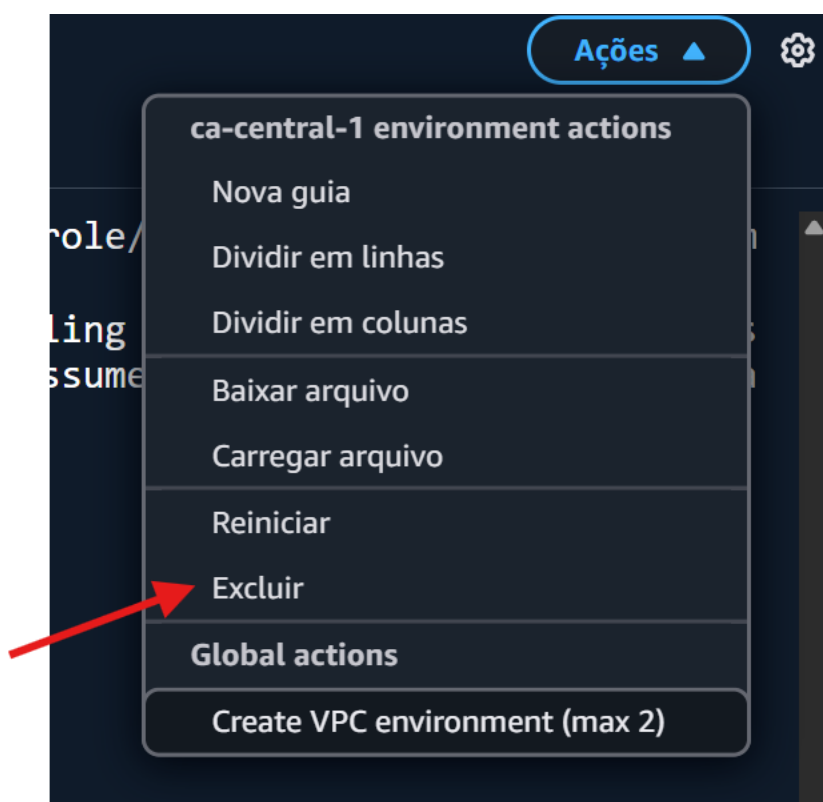
1. **Executar Script Novamente:** No CloudShell, execute o script `credenciais_temporarias.py` novamente com os mesmos parâmetros do Passo 4.
 - a. **Resultado:** Você receberá um erro de "acesso negado" (ou similar).

```
~ $ python3 credenciais_temporarias.py --role-arn arn:aws:iam::804364734712:role/MarioSilvaRole --session-name AcessoTemporario --duration 3600
[ERRO] Falha ao obter credenciais: An error occurred (AccessDenied) when calling the AssumeRole operation: User: arn:aws:iam::804364734712:user/Mario is not authorized to perform: sts:AssumeRole on resource: arn:aws:iam::804364734712:role/MarioSilvaRole
~ $
```

- b. **Explicação:** A política de confiança da role foi modificada, e seu usuário não está mais autorizado a assumir essa role.

Passo 14: Excluindo Recursos

1. **Excluir CloudShell:** No CloudShell, clique em "Ações" > "Excluir". Digite delete e clique em "Excluir".



2. **Excluir Role IAM:**
 - a. No console, vá para **IAM > Funções**.
 - b. Pesquise por `SeuNomeRole`, marque a caixa ao lado e clique em "Excluir".
 - c. Digite o nome da role para confirmar e clique em "Excluir".

Identity and Access Management (IAM)

Painel

Gerenciamento de acesso

- Grupos de usuários
- Usuários
- Funções**

Funções (1/147) Informações

Um perfil do IAM é uma identidade que você pode criar que tem permissões específicas com credenciais válidas por curtos períodos. Os perfis podem ser assumidos por entidades em que você confia.

Q mario 1 correspondência

<input checked="" type="checkbox"/>	Nome da função	Entidades confiáveis	Última ativa
<input checked="" type="checkbox"/>	MarioSilvaRole	Conta: 804364734712	44 minutos

Roles Anywhere Informações

Gerenciar

Parabéns, você concluiu o laboratório e removeu todos os recursos!