

# AUTOMAÇÃO DE CRIAÇÃO DE GRUPOS, POLÍTICAS E USUÁRIOS NA AWS USANDO O ANSIBLE

Conceitos e tecnologias usadas:

- Ansible;
- Módulo Ansible community.aws collection;
- AWS CLI;
- Console AWS;
- Planilhas CSV com dados de origem;
- Playbook em linguagem Yaml;
- Código em linguagem Python;

Passo a passo:

**1º Passo:** Instalação da AWS CLI na máquina que será usada para enviar os comandos do Ansible. As opções para todos os sistemas operacionais podem ser encontradas em:

<https://aws.amazon.com/pt/cli/>

**2º Passo:** Criação de usuário na console da AWS com acesso do tipo “Programmatic Access”, e permission policy do tipo AdministratorAccess.

A Access key ID e a secret access key desse usuário devem ser armazenadas em local seguro para serem usadas no próximo passo.

**3º Passo:** Configuração da AWS CLI na máquina em que foi instalada. Na linha de comando digitar:

**\$ AWS configure**

Em seguida será pedido:

- AWS Access Key ID: (do usuário que foi criado no passo anterior)
- AWS Secret Access Key: (do usuário que foi criado no passo anterior)
- Default Region Name: (região da AWS que será usada)
- Default Output Format: json

```
daniel@Notebook-danielgil:~$ aws configure
AWS Access Key ID [*****2TVU]:
AWS Secret Access Key [*****RjSZ]:
Default region name [us-east-1]:
Default output format [json]:
```

**4º Passo:** Instalação do Ansible via linha de comando.

Todas as instruções estão em:

[https://docs.ansible.com/ansible/latest/installation\\_guide/intro\\_installation.html](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html)

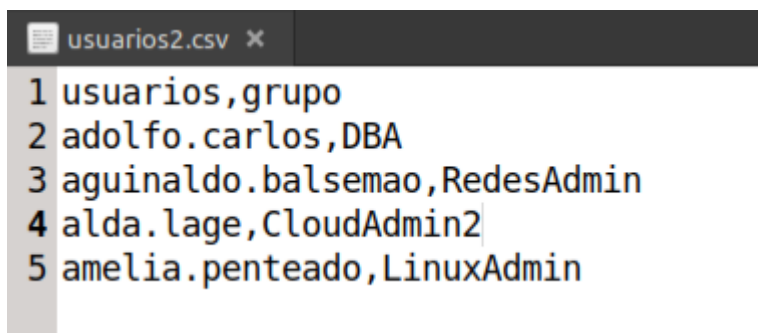
**5º Passo:** Instalação do módulo do Ansible community.aws collection via linha de comando:

**\$ ansible-galaxy collection install amazon.aws**

**6º passo:** Armazenar em um único diretório 5 arquivos:

- Um arquivo CSV com 2 colunas, uma com nomes dos usuários e outra com o nome do grupo que esses usuários pertencerão, o nome dado para o arquivo foi usuarios2.csv e o número de registros pode ser maior ou menor;

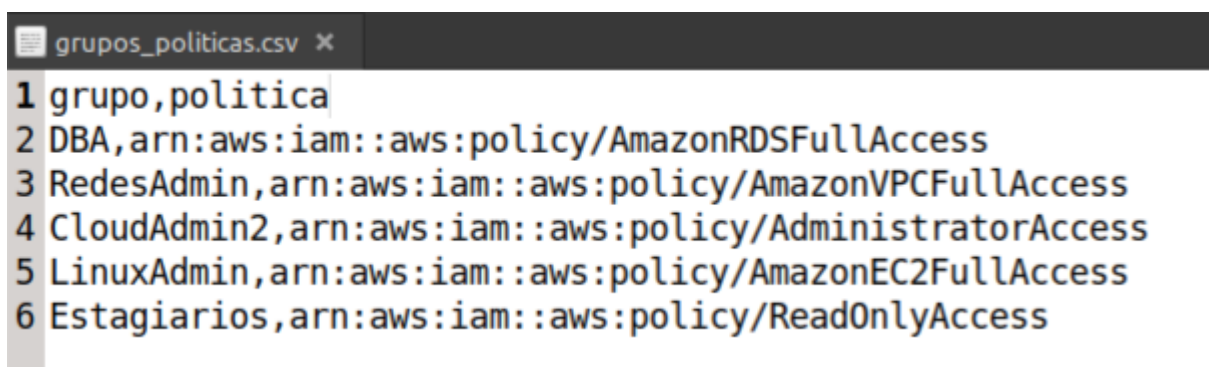
Exemplo:



```
usuarios2.csv x
1 usuarios,grupo
2 adolfo.carlos,DBA
3 aguinaldo.balsemao,RedesAdmin
4 alda.lage,CloudAdmin2
5 amelia.penteado,LinuxAdmin
```

- Um arquivo CSV com 2 colunas, uma com os nomes dos grupos e outra com a ARN da política que será atachada a esse grupo, o nome dado para o arquivo foi grupos\_politicas.csv e o número de registros pode ser maior ou menor;

Exemplo:



```
grupos_politicas.csv x
1 grupo,politica
2 DBA,arn:aws:iam::aws:policy/AmazonRDSFullAccess
3 RedesAdmin,arn:aws:iam::aws:policy/AmazonVPCFullAccess
4 CloudAdmin2,arn:aws:iam::aws:policy/AdministratorAccess
5 LinuxAdmin,arn:aws:iam::aws:policy/AmazonEC2FullAccess
6 Estagiarios,arn:aws:iam::aws:policy/ReadOnlyAccess
```

- Um arquivo de código em Python que será executado pelo Ansible e que fará a leitura do arquivo usuarios2.csv, gerará uma senha aleatória dentro dos padrões da AWS para cada um e armazenará em um novo arquivo CSV com o nome de usuarios\_com\_senha.csv

Código Python:

```

aws_iam.py
21 import csv
22 from random import randint, choice, shuffle
23
24
25 def gera_senha():
26     letras = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u',
27             'v', 'w', 'x', 'y', 'z']
28     caracteres = ['!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '_', '+', '-', '=', '[', ']', '{', '}', '|']
29     senha = []
30
31     for n in range(2): # Gera 4 letras, 2 minúsculas e 2 maiúsculas
32         senha.append(choice(letras))
33         senha.append(choice(letras).upper())
34
35     for n in range(4): # Gera 4 números entre 0 e 9
36         senha.append(str(randint(0, 9)))
37
38     senha.append(choice(caracteres)) # Gera 1 caractere especial
39     shuffle(senha) # Embaralha a lista aleatoriamente
40     senha = ''.join(senha) # Unifica os elementos da lista em uma única string
41     return senha
42
43
44 usuario = []
45 grupo = []
46
47
48 with open('usuarios2.csv') as original: # Abre o arquivo original para leitura
49     leitor = csv.DictReader(original)
50     for row in leitor:
51         usuario.append(row['usuarios']) # Armazena todos os usuários em uma lista
52         grupo.append(row['grupo']) # Armazena todos os grupos em uma lista
53
54
55 with open('usuarios_com_senha.csv', 'w') as destino: # Cria o novo arquivo
56     fieldnames = ['usuarios', 'grupo', 'senha']
57     writer = csv.DictWriter(destino, fieldnames=fieldnames)
58     writer.writeheader()
59     for n in range(len(usuario)):
60         # Escreve linhas no arquivo com cada usuário e respectivos grupos e senhas:
61         writer.writerow({'usuarios': usuario[n], 'grupo': grupo[n], 'senha': gera_senha()})
62

```

Exemplo que resulta da execução do código Python (este novo CSV será gerado automaticamente quando o playbook for executado):

```

usuarios_com_senha.csv x
1 usuarios,grupo,senha
2 adolfo.carlos,DBA,9QR4rq8_9
3 aguinaldo.balsemao,RedesAdmin,2HL$58gs9
4 alda.lage,CloudAdmin2,p9@3PQ18s
5 amelia.penteado,LinuxAdmin,)2x0L2C9m

```

- Um arquivo para armazenamento da ARN da política EnforceMFA que é uma política customizada para obrigar os usuários a habilitar o MFA para acessar os recursos da AWS. Foi optado armazenar este dado separadamente porque a ARN desta política tem o número da ID da conta da AWS, deixando separado é possível encriptá-lo. O nome dado a este arquivo foi id.yml;

Exemplo:

```
id.yml x
1 ---
2 nome_enforce: arn:aws:iam::[REDACTED]:policy/EnforceMFAPolicy
```

ID da Conta AWS

- O arquivo do playbook em formato Yaml que executará todas as tasks:

```
! importa_usuarios.yml > ...
1 ---
2 |
3 # Importação de usuários para AWS
4
5 - name: Importa usuarios
6
7   hosts: localhost
8   connection: local
9   vars_files: id.yml
10
11
12   tasks:
13
14     # Leitura do arquivo csv de grupos
15
16     - name: ler arquivo csv de grupos
17       community.general.read_csv:
18         path: grupos_politicas.csv
19         key: grupo
20         register: grupo
21
22
23     # Cria grupos e adiciona as politicas especificas
24
25     - name: cria grupos e adiciona politicas
26       community.aws.iam_group:
27         name: "{{ item.value.grupo }}"
28         managed_policies:
29           - "{{ item.value.politica }}"
30           - "{{ nome_enforce }}"
31         state: present
32
33     loop: "{{ grupo.dict|dict2items }}"
34
35     # Executa localmente o script python para criação de arquivos com senhas
36     # fortes e aleatórias
37
38     - name: gera csv com senhas
39       script: aws_iam.py
40       args:
41         executable: python3
42
43
44     # Leitura do arquivo csv criado anteriormente
45
46     - name: ler arquivo csv de usuarios
47       community.general.read_csv:
48         path: usuarios_com_senha.csv
49         key: usuarios
50         register: usuarios
51
52
53     # Criação de usuários na AWS baseado no novo arquivo csv
54
55     - name: cria usuario na AWS
56       community.aws.iam:
57         iam_type: user
58         name: "{{ item.value.usuarios }}"
59         state: present
60         password: "{{ item.value.senha }}"
61         groups: "{{ item.value.grupo }}"
62         access_key_state: create
63         loop: "{{ usuarios.dict|dict2items }}"
64
65 ...
66
```

- **Passo 7:** Criptografar o arquivo que contém a ARN da política EnforceMFA digitando o comando:

**\$ ansible-vault encrypt id.yml**

Em seguida será pedido que crie uma senha para esse arquivo encriptado:

```
daniel@Notebook-danielgil:~/Área de Trabalho/Cloud Computing/GitHub/aws-user-creation-ansible-cp$ ansible-vault encrypt id.yml
[WARNING]: log file at /var/log/ansible.log is not writeable and we cannot create it, aborting

New Vault password:
Confirm New Vault password:
Encryption successful
```

- **Passo Final:** Executar a playbook com o comando:

**\$ ansible-playbook --ask-vault-pass importa\_usuarios.yml**

Com o parâmetro `--ask-vault-pass` o Ansible pede a senha do arquivo criptografado antes de executar a playbook.

```
daniel@Notebook-danielgil:~/Área de Trabalho/Cloud Computing/GitHub/aws-user-creation-ansible-cp$ ansible-playbook --ask-vault-pass importa_usuarios.yml
[WARNING]: log file at /var/log/ansible.log is not writeable and we cannot create it, aborting

Vault password:
```

Por fim, depois de digitar a senha ele executa a playbook:

```

daniel@Notebook-danielgil:~/Área de Trabalho/Cloud Computing/GitHub/aws-user-creation-ansible-cp$ ansible-playbook --ask-vault-pass importa_usuarios.yml
[WARNING]: log file at /var/log/ansible.log is not writeable and we cannot create it, aborting

Vault password:
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [Importa usuarios] *****

TASK [Gathering Facts] *****
ok: [localhost]

TASK [ler arquivo csv de grupos] *****
ok: [localhost]

TASK [cria grupos e adiciona politicas] *****
changed: [localhost] => (item={'key': 'DBA', 'value': {'grupo': 'DBA', 'politica': 'arn:aws:iam::aws:policy/AmazonRDSFullAccess'}})
changed: [localhost] => (item={'key': 'RedesAdmin', 'value': {'grupo': 'RedesAdmin', 'politica': 'arn:aws:iam::aws:policy/AmazonVPCFullAccess'}})
changed: [localhost] => (item={'key': 'CloudAdmin2', 'value': {'grupo': 'CloudAdmin2', 'politica': 'arn:aws:iam::aws:policy/AdministratorAccess'}})
changed: [localhost] => (item={'key': 'LinuxAdmin', 'value': {'grupo': 'LinuxAdmin', 'politica': 'arn:aws:iam::aws:policy/AmazonEC2FullAccess'}})
changed: [localhost] => (item={'key': 'Estagiarios', 'value': {'grupo': 'Estagiarios', 'politica': 'arn:aws:iam::aws:policy/ReadOnlyAccess'}})

TASK [gera csv com senhas] *****
changed: [localhost]

TASK [ler arquivo csv de usuarios] *****
ok: [localhost]

TASK [cria usuario na AWS] *****
changed: [localhost] => (item={'key': 'adolfo.carlos', 'value': {'usuarios': 'adolfo.carlos', 'grupo': 'DBA', 'senha': '9QR4rq8_9'}})
changed: [localhost] => (item={'key': 'aguinaldo.balsemao', 'value': {'usuarios': 'aguinaldo.balsemao', 'grupo': 'RedesAdmin', 'senha': '2HL$58gs9'}})
changed: [localhost] => (item={'key': 'alda.lage', 'value': {'usuarios': 'alda.lage', 'grupo': 'CloudAdmin2', 'senha': 'p9@3PQ18s'}})
changed: [localhost] => (item={'key': 'amelia.penteado', 'value': {'usuarios': 'amelia.penteado', 'grupo': 'LinuxAdmin', 'senha': '2x0L2C9m'}})

PLAY RECAP *****
localhost : ok=6 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

```

Todos os arquivos se encontram no repositório:

<https://github.com/danielcgil83/aws-user-creation-ansible>

Daniel Gil