



# Certified Tech Developer

The Ultimate Degree

## Infraestrutura II

# Criando um bucket S3

Damos as boas-vindas! Neste espaço, vamos colocar em prática todo o aprendizado durante esta semana.

## Objetivo

Vamos fazer uso da nossa conta da AWS, e utilizar um template da Ansible para criar um bucket do nosso computador até a nossa conta da AWS automaticamente! Seu desafio será chegar até o final do exercício.

## Instruções

Trabalhando em infraestrutura, há ocasiões em que desenvolvedores nos pedem para salvarmos arquivos estáticos dentro da conta da organização. Para isso, existem os buckets S3 no AWS.

Antes de executarmos um playbook da Ansible nos nossos computadores, deveremos levar em conta alguns requerimentos prévios. Lembrando que esta ferramenta está escrita em Python, de modo que iremos precisar ter o Python 3 instalado e, uma vez feito isso, deveremos instalar Boto3, que é uma biblioteca de Python para automatizar tarefas no AWS. Nossas maiores aliadas neste caminho!



Podemos baixar o Python 3 a partir da sua web oficial, escolhendo a alternativa mais adequada para nosso sistema operacional:

<https://www.python.org/downloads/>

Assim que o Python estiver instalado, iremos usar pip - o sistema de gerenciamento de pacotes de Python - para instalar o Boto3. Esta ferramenta é incluída na instalação de Python.

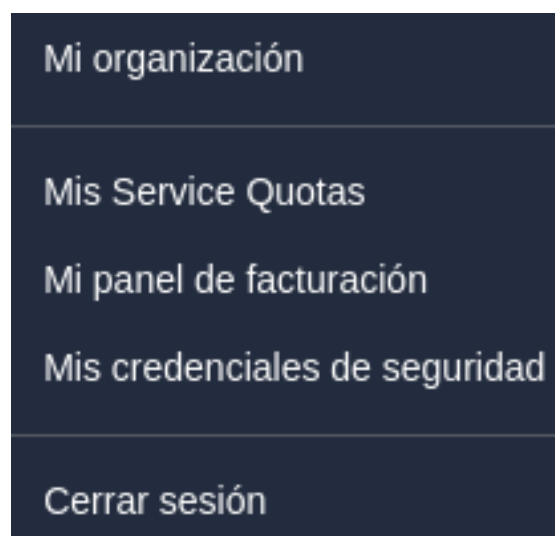
Deveremos executar este comando:

```
pip install boto3
```

Ao finalizar, visualizamos esta mensagem:

```
Installing collected packages: boto3  
Successfully installed boto3-1.18.5
```

Agora, só resta configurar nossa conta de AWS dentro dos nossos computadores. Assim, para executar nossos playbooks, não será preciso inserirmos nenhum tipo de autenticação. Para acessar nossas credenciais, vamos, a partir da conta de AWS, para "Minhas credenciais de segurança".





Salvamos as credenciais criadas com a opção “Chaves de acceso (ID de chave de acceso e chave de acceso secreta)”.

▼ Claves de acceso (ID de clave de acceso y clave de acceso secreta)

Use access keys to make programmatic calls to AWS from the AWS CLI, 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.

Para su protección, no comparta nunca las claves secretas. Como práctica recomendada, sugerimos un cambio frecuente de las claves. **La clave secreta solo se puede ver o descargar durante el proceso de creación. Cree una nueva clave de acceso si ha perdido la ya existente.** [Más información](#)

Creado	ID de clave de acceso	Último uso	Última región utilizada	Último servicio utilizado	Estado	Acciones
jul. 14º 2021	AKIA6AINKP3IHOPBZHWY	2021-07-14 15:43 UTC-0300	us-west-1	cloudformation	Activo	<a href="#">Desactivar</a>   <a href="#">Eliminar</a>

[Crear una clave de acceso](#)

Agora sim, vamos para nosso computador! Executamos o comando **aws configure** e inserimos os dados requisitados com nosso ID de chave de acceso.

```
[enuel@enuel ansible_aws]$ aws configure
AWS Access Key ID [*****DCZA]:
AWS Secret Access Key [*****MB4y]:
Default region name [us-west-1]:
Default output format [json]:
```

Última instalação para este guia (e que será de utilidade para os demais exercícios).

Vamos instalar a Ansible! Como no caso anterior, deveremos escolher opção correta para nosso sistema operacional:

[https://docs.ansible.com/ansible/latest/installation\\_guide/intro\\_installation.html#installing-ansible-on-specific-operating-systems](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#installing-ansible-on-specific-operating-systems)



Finalmente, chegou a hora de testar nosso playboock! Vamos criar um arquivo chamado **main.yml**. O conteúdo do nosso código é o seguinte:

```
---  
- hosts: localhost  
  
tasks:  
- name: Criamos um bucket S3  
  
  aws_s3:  
  
    bucket: my-digitalhouse-stored  
  
    mode: create
```

Vamos por partes. O que estamos escrevendo?


Hosts: faz referência a onde o playbook será executado. Neste caso, será no nosso computador, portanto fazemos referência a "localhost".

- Tasks: é uma palavra reservada para indicar que estamos começando com tarefas.
- Name: uma das vantagens da Ansible é a possibilidade de documentar enquanto escrevemos código. Neste caso, **name** é sempre uma breve referência ao que estamos executando.
- Módulo: já temos tudo pronto para executarmos nosso módulo (estamos executando código declarativo!). Basta escrever o nome do módulo: **aws\_s3**, e já teremos quase tudo resolvido.
- Cada módulo tem seus requerimentos e palavras reservadas. Para este em específico, a melhor prática será verificar a documentação oficial:









[https://docs.ansible.com/ansible/latest/collections/amazon/aws/aws\\_s3\\_module.html](https://docs.ansible.com/ansible/latest/collections/amazon/aws/aws_s3_module.html)



Segundo ela, existem apenas dois requerimentos obrigatórios:

- **bucket**  
string / required
- **mode**   
string / required

Estamos atendendo aos requisitos? Verificando nossa conta da AWS, iremos notar que o bucket não foi criado:

Los buckets son contenedores de datos almacenados en S3. [Más información](#) 


  Copiar ARN  Vaciar  Eliminar  Crear bucket

 digitalhouse 

Nombre	Región de AWS	Acceso	Fe
--------	---------------	--------	----

No hay coincidencias

No se encuentra ninguna coincidencia.

 Borrar filtros

Já temos tudo pronto para executar nosso playbook. Vamos para nosso terminal e basta executar o comando: **ansible-playbook main.yml**



```
! main.yml x
! main.yml > {} 0 > [ ] tasks > {} 0 > name
1 ---
2 - hosts: localhost
3   tasks:
4     - name: Creamos un bucket S3
5       aws_s3:
6         bucket: my-digitalhouse-stored
7         mode: create
8
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
[enuel@enuel ansible_aws]$ ansible-playbook main.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [localhost] *****

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

TASK [Creamos un bucket S3] *****
changed: [localhost]

PLAY RECAP *****
localhost                : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

O que aconteceu? Ao executá-lo, podemos ver como ele exibe cada passo descrito no nosso código, com os nomes coincidentes e mostrando um resumo final de tudo o que foi executado. Vamos fazer zoom nesse resumo:

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

TASK [Creamos un bucket S3] *****
changed: [localhost]

PLAY RECAP *****
localhost                : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Os “ok” indicam as conexões bem-sucedidas. No nosso caso, foram duas:

- A conexão da Ansible dentro do nosso próprio computador, e que esteja bem instalada.
- A conexão da nossa equipe com nossa conta com AWS.



Tudo o configurado no início foi perfeito, mas... o bucket foi criado?

Los buckets son contenedores de datos almacenados en S3. [Más Información](#)

1 coincidencia < 1 >

	Nombre ▲	Región de AWS ▼	Acceso ▼
<input type="radio"/>	<a href="#">my-digitalhouse-stored</a>	EE. UU. Oeste (Norte de California) us-west-1	<u>Los objetos pueden ser públicos</u>

Ótimo! Já podemos automatizar todos os repositórios de arquivos estáticos para nossas equipes de trabalho.

Quais melhorias você faria? Segundo a documentação oficial, seria possível você criar um bucket e carregar os arquivos diretamente? Modifique o playbook para praticar ainda mais!