



# Certified Tech Developer

The Ultimate Degree

## Infraestructura II

# Creamos un bucket S3

¡Les damos la bienvenida! En este espacio vamos a poner en práctica todo lo que aprendimos durante esta semana.

## Objetivo

Vamos a usar nuestra cuenta de AWS y utilizar un template de Ansible para crear un bucket S3 desde nuestra PC hacia nuestra cuenta de AWS, ¡automáticamente! Tu desafío va a ser llegar al final de este ejercicio.

## Instrucciones

Trabajando en infraestructura, hay ocasiones en que los desarrolladores nos piden poder guardar archivos estáticos dentro de la cuenta de la organización. Para este fin existen los buckets S3 en AWS.

Tenemos que tener en cuenta algunos requisitos previos antes de ejecutar un playbook de Ansible en nuestras computadoras. Recordemos que esta herramienta está escrita en Python, por lo que vamos a necesitar tener instalado Python 3 y, una vez hecho esto, instalar Boto3. Esta última es la librería de Python para automatizar tareas en AWS. ¡Nuestras mayores aliadas en este camino!



Podemos descargar Python 3 desde su web oficial, eligiendo la opción más adecuada para el sistema operativo que tengamos:

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

Una vez instalado Python, para instalar Boto3, vamos a usar pip —el sistema de administración de módulos de Python—. Esta herramienta está incluida con la instalación de Python.

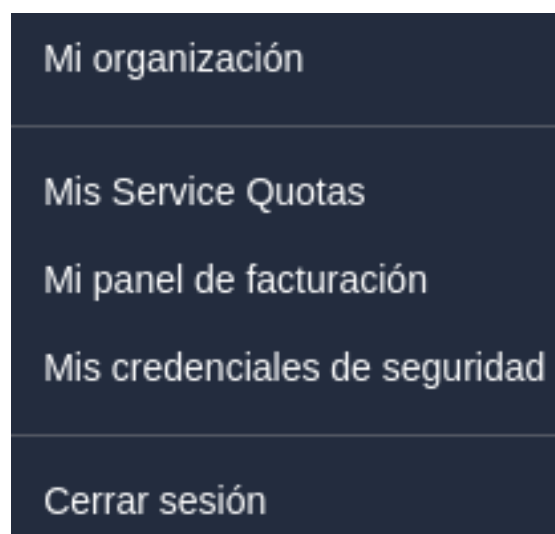
Debemos ejecutar este comando:

```
pip install boto3
```

Vamos a visualizar este mensaje al terminar:

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

Ahora solo nos queda configurar nuestra cuenta de AWS dentro de nuestras computadoras. De esta manera, no tendremos que ingresar ningún tipo de autenticación para ejecutar nuestros playbooks. Para acceder a nuestras credenciales, desde la cuenta de AWS, nos dirigimos a “Mis credenciales de seguridad”.





Guardemos las credenciales que generamos con la opción “Claves de acceso (ID de clave de acceso y clave 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](#)

Ahora sí, ¡vamos a nuestra computadora! Ejecutamos el comando **aws configure** e ingresamos los datos solicitados con nuestro ID de clave 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 instalación para esta guía (y que nos servirá para el resto de los ejercicios).

¡Instalemos Ansible! Como en el caso anterior, tenemos que elegir la opción correcta para nuestro sistema operativo:

[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)



¡Por fin llegó la hora de probar nuestro playbook! Creamos un archivo que se llame **main.yml**. El contenido de nuestro código es el siguiente:

```
---  
  
- hosts: localhost  
  
  tasks:  
  
  - name: Creamos un bucket S3  
  
    aws_s3:  
  
      bucket: my-digitalhouse-stored  
  
      mode: create
```


Vamos por partes. ¿Qué estamos escribiendo?

- Hosts: hace referencia a donde se va a ejecutar el playbook. En nuestro caso es nuestra PC, por lo que hacemos referencia a "localhost".
- Tasks: es una palabra reservada para indicar que comenzamos con tareas.
- Name: una de las bondades de Ansible es la posibilidad de documentar mientras escribimos código. En este caso, **name** siempre es una breve referencia a lo que estamos ejecutando.
- Módulo: ya tenemos todo listo para ejecutar nuestro módulo (¡estamos ejecutando código declarativo!). Ya tenemos casi todo resuelto con solo escribir el nombre del módulo: **aws\_s3**.
- Cada módulo tiene sus requerimientos y palabras reservadas. Para este en particular, la mejor práctica es consultar la documentación 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)


Consultando la misma, solo hay dos requisitos obligatorios:

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

¿Estamos cumpliendo estos requisitos? Al consultar nuestra cuenta de AWS, notaremos que no tenemos creado el bucket:

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

  Copiar ARN  Vaciar  Eliminar  Crear bucket

Nombre	Región de AWS	Acceso	Fe
No hay coincidencias			
No se encuentra ninguna coincidencia.			
 Borrar filtros			

Ya tenemos todo listo para ejecutar nuestro playbook. Vayamos a nuestra terminal y, simplemente, ejecutemos el 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
```

¿Qué pasó? Al ejecutarlo podemos ver cómo nos mostró cada paso que está descrito en nuestro código, coincidiendo los nombres y mostrando un resumen final de todo lo que se ejecutó. Hagamos zoom en ese resumen:

```
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
```

Los “ok” indican las conexiones exitosas. En nuestro caso fueron dos:

- La conexión de Ansible dentro de nuestro propio equipo y que esté bien instalado.
- La conexión de nuestro equipo hacia nuestra cuenta con AWS.



Todo lo que configuramos al principio estuvo perfecto, pero... ¿se creó el bucket?

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>

¡Excelente! Ya podemos automatizar todos los repositorios de archivos estáticos para nuestros equipos de trabajo.

¿Qué mejoras harías? Leyendo la documentación oficial, ¿podrías crear un bucket y subir archivos directamente? ¡Modificá el playbook para practicar aún más!