



UNIVERSIDAD IBEROAMERICANA DEL ECUADOR
LABORATORIOS PRACTICOS

UNIVERSIDAD IBEROAMERICANA DEL ECUADOR

FACULTAD:

Facultad de Comunicación y TICS

CARRERA

Software

Laboratorio de:

Programación en la Nube

Práctica No.: 7

Tema:

Laboratorio S3 y EC2

Realizado por:

Barros Cristhian, Chango Jordán, Naula Alex



UNIVERSIDAD IBEROAMERICANA DEL ECUADOR
LABORATORIOS PRACTICOS

Objetivos:

Objetivo General

Analizar la integración entre Amazon S3 y Amazon EC2 destacando su utilidad y eficacia en el almacenamiento, procesamiento y gestión de datos en la nube.

Objetivos específicos

- Creación de un bucket en Amazon S3 y almacenamiento de datos
- Configuración de una estructura en Amazon S3 para la modificación de archivos y bucket mediante un script de Python
- Instalar aws cli en una estructura de EC2

Marco teórico:

Amazon S3 (Simple Storage Service):

Amazon S3 es un servicio de almacenamiento en la nube altamente escalable y duradero ofrecido por Amazon Web Services (AWS). Permite almacenar y recuperar cualquier cantidad de datos desde cualquier lugar en la web. Algunos conceptos importantes relacionados con Amazon S3 son:

- **Buckets:** Los buckets son contenedores para almacenar objetos en Amazon S3. Cada bucket tiene un nombre único global en la región de AWS en la que se crea y es la unidad básica de almacenamiento en S3.
- **Objetos:** Los objetos son los datos que se almacenan en Amazon S3. Cada objeto consiste en los datos en sí, metadatos y una clave única. Los metadatos son pares clave-valor que describen atributos del objeto, como la fecha de creación, el tipo de contenido, el propietario, etc.
- **Regiones y Zonas de Disponibilidad:** Amazon S3 se replica automáticamente en múltiples ubicaciones dentro de una región de AWS para garantizar la alta disponibilidad y durabilidad de los datos. Estas ubicaciones se denominan "Zonas de Disponibilidad".



UNIVERSIDAD IBEROAMERICANA DEL ECUADOR
LABORATORIOS PRACTICOS

Integración de Metadatos en la Estructura del Sistema:

La integración de metadatos en la estructura del sistema en Amazon S3 permite proporcionar información adicional sobre los objetos almacenados y mejorar la organización y gestión de los datos en el bucket. Algunos aspectos clave de cómo se logra esta integración son:

- **Asignación de Metadatos:** Los metadatos se pueden asignar a los objetos almacenados en un bucket de Amazon S3. Los metadatos son pares clave-valor que se agregan al objeto y describen detalles específicos sobre ese objeto.
- **Información Adicional:** Los metadatos proporcionan información adicional sobre los objetos, como detalles sobre el contenido del objeto, la fuente de los datos, el propósito del objeto o cualquier otra información relevante que pueda ser útil para su gestión y uso.
- **Acceso a los Metadatos:** Los metadatos se pueden acceder y modificar utilizando la API de Amazon S3 o la consola de administración de AWS. También es posible configurar políticas de acceso y permisos para controlar quién puede ver o modificar los metadatos.
- **Uso en Scripts y Aplicaciones:** La presencia de metadatos permite que scripts y aplicaciones interactúen con los objetos de manera más inteligente y eficiente. Por ejemplo, un script podría consultar los metadatos para determinar la fecha de creación de un objeto y realizar acciones basadas en esa información.
- **Automatización de Tareas:** Los metadatos también pueden utilizarse para automatizar tareas en función de su contenido. Se pueden configurar eventos de Amazon S3 para activar acciones específicas cuando se modifiquen los metadatos o cuando se agregue un nuevo objeto al bucket.



UNIVERSIDAD IBEROAMERICANA DEL ECUADOR
LABORATORIOS PRACTICOS

Boto3

Es una biblioteca de Python desarrollada por AWS que permite interactuar y gestionar servicios en la nube de Amazon Web Services (AWS) de manera programática. Esta poderosa biblioteca facilita la automatización de tareas comunes, como el acceso, creación, modificación y eliminación de recursos en AWS, mediante el uso de API (Application Programming Interface) de servicios como Amazon S3, Amazon EC2, DynamoDB, entre otros. Boto3 proporciona una interfaz de alto nivel y fácil de usar para desarrolladores, lo que les permite integrar funcionalidades de AWS en sus aplicaciones Python de manera rápida y eficiente, lo que resulta especialmente útil para implementar soluciones escalables y flexibles en la nube de AWS.

cliente.list_buckets()

En este contexto, "cliente" se refiere a un objeto de la clase "boto3.client()" que representa la conexión con el servicio de Amazon S3. El método "list_buckets()" se utiliza para obtener una lista de todos los buckets disponibles en la cuenta de AWS asociada al cliente.

Aplicación:

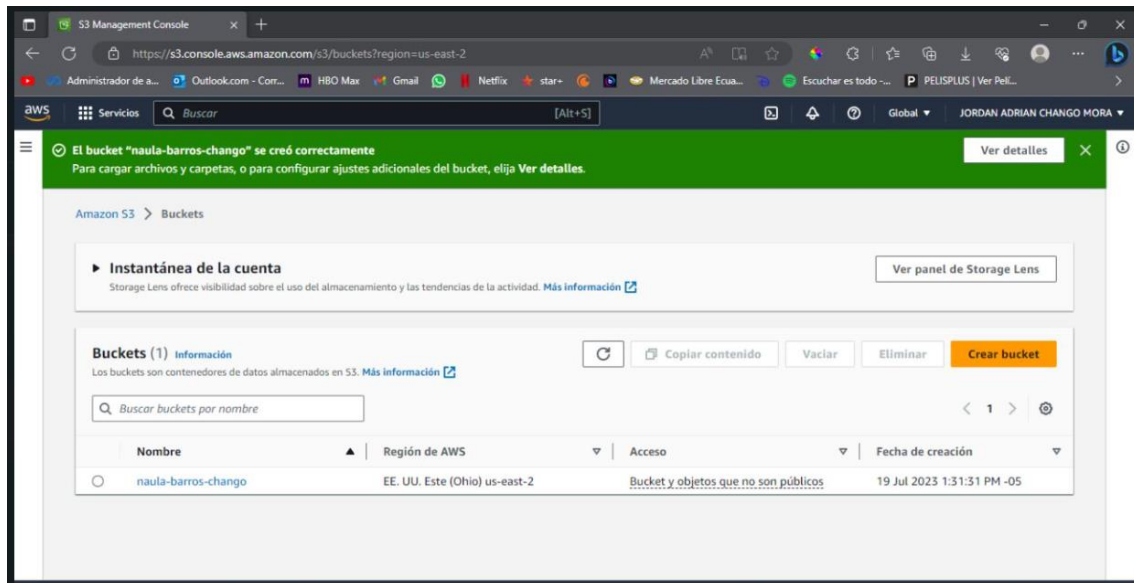
Mediante "cliente.list_buckets()", los desarrolladores pueden obtener una lista de los nombres de los buckets existentes en su cuenta de AWS, lo que resulta útil para realizar tareas de administración, monitoreo y gestión de recursos en Amazon S3. Esta información puede utilizarse para validar la existencia de determinados buckets, listarlos para que el usuario los seleccione, o realizar operaciones específicas en función de los nombres de los buckets obtenidos.



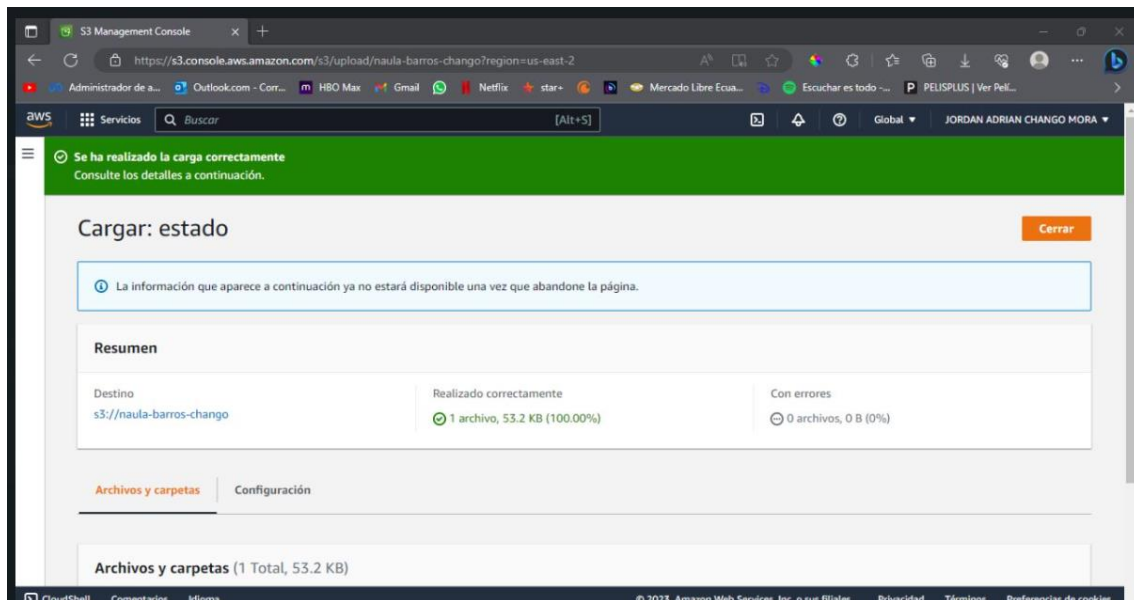
UNIVERSIDAD IBEROAMERICANA DEL ECUADOR LABORATORIOS PRACTICOS

Desarrollo de la práctica:

Creación del Bucket



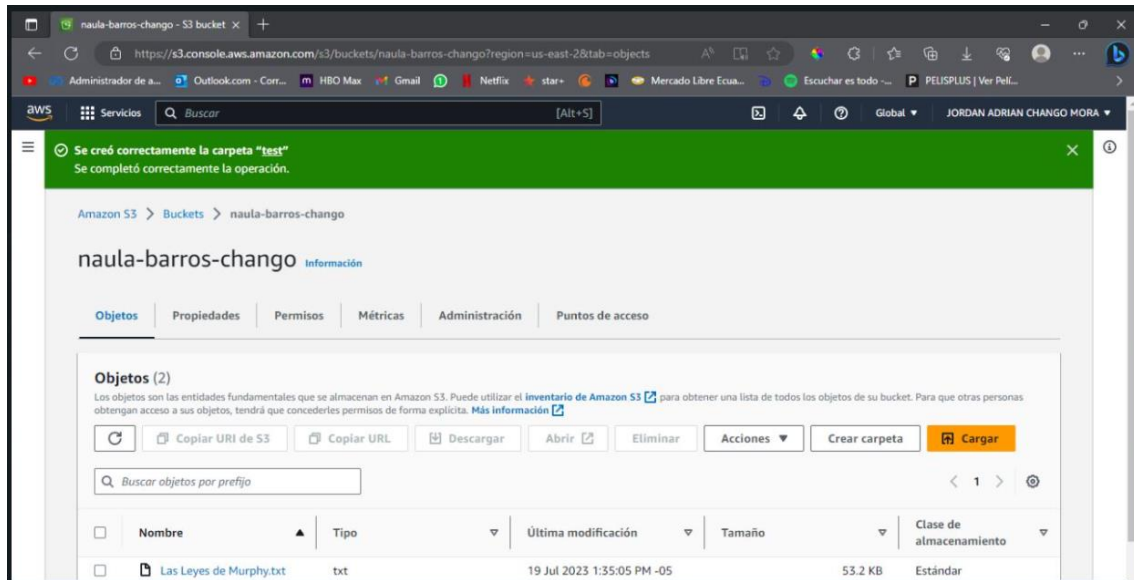
Cargando Archivos



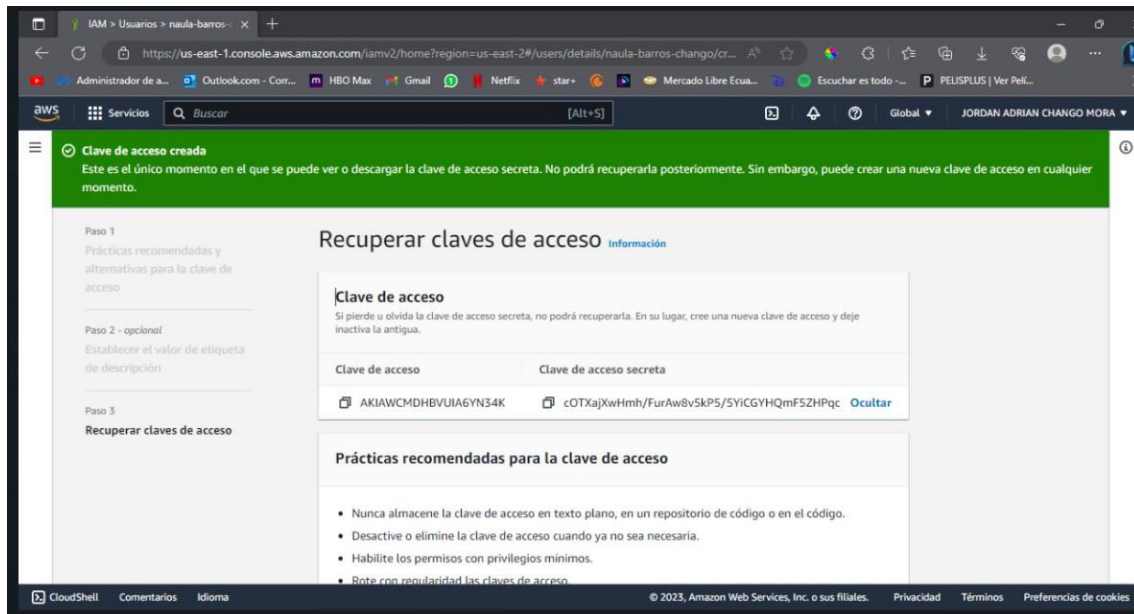
Creando Carpeta tests



UNIVERSIDAD IBEROAMERICANA DEL ECUADOR LABORATORIOS PRACTICOS



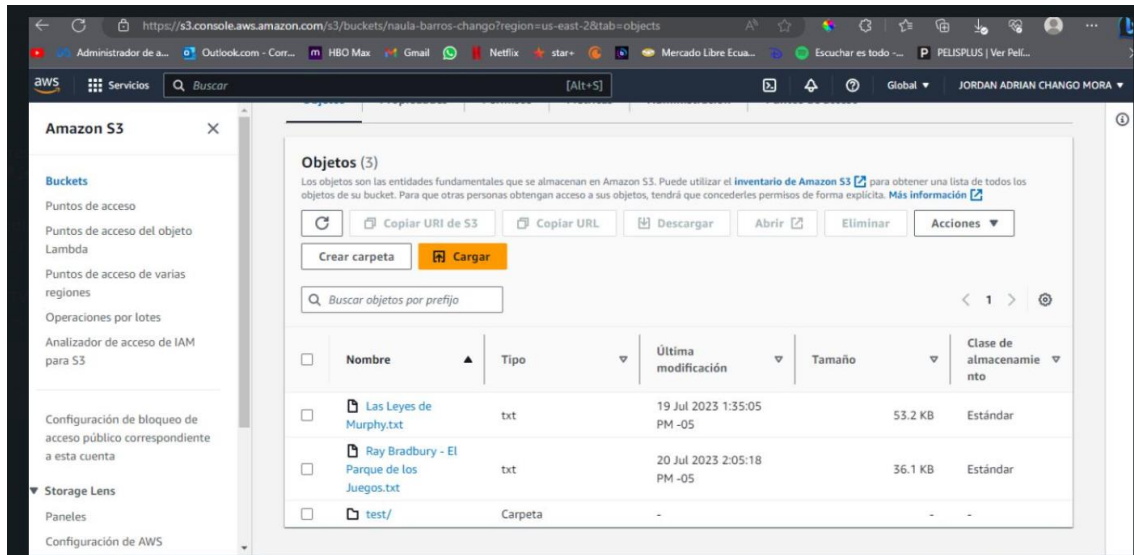
Creando Clave de Acceso (Credenciales de Seguridad)



Estructura de S3



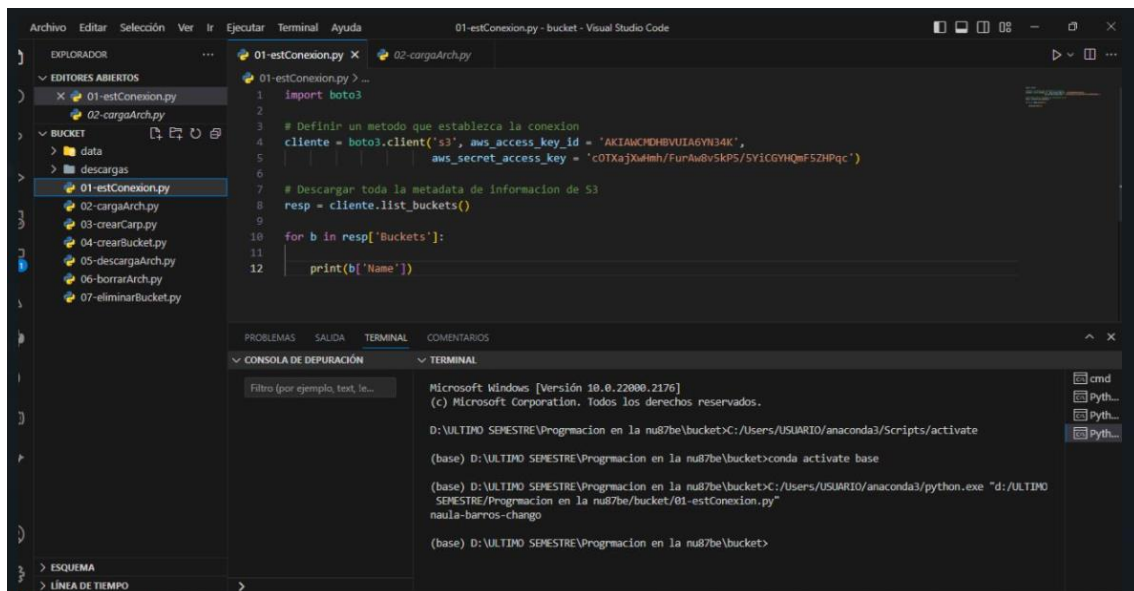
UNIVERSIDAD IBEROAMERICANA DEL ECUADOR LABORATORIOS PRACTICOS



Instalación de boto3

```
(base) D:\ULTIMO SEMESTRE\Programacion en la nu87be\bucket> pip install boto3
```

Estableciendo conexión



Cargando Archivo



UNIVERSIDAD IBEROAMERICANA DEL ECUADOR LABORATORIOS PRACTICOS

```
02-cargaArch.py > ...
4 # Definir un metodo que establezca la conexion
5 cliente = boto3.client('s3', aws_access_key_id = 'AKIAWCMDBVUIA6YN34K',
6                               aws_secret_access_key = 'cOTXajXwHmh/FurAw8v5kP5/5YiCGVHQmF5ZHPqc')
7
8 # Variables de acceso al sistema
9
10 bucket = 'naula-barros-chango'
11 path_act = os.getcwd()
12 arch = 'Ray Bradbury - El Parque de los Juegos.txt'
13 path_arch = os.path.join(path_act, 'data', arch)
14 print(path_arch)
15
```

TERMINAL

```
base
(base) D:\ULTIMO SEMESTRE\Progrmmacion en la nu87be\bucket>C:
/Users/USUARIO/anaconda3/python.exe "d:/ULTIMO SEMESTRE/Prog
mmacion en la nu87be\bucket/01-estConexion.py"
naula-barros-chango

(base) D:\ULTIMO SEMESTRE\Progrmmacion en la nu87be\bucket>C:
/Users/USUARIO/anaconda3/python.exe "d:/ULTIMO SEMESTRE/Prog
mmacion en la nu87be\bucket/02-cargaArch.py"
D:\ULTIMO SEMESTRE\Progrmmacion en la nu87be\bucket\data\Ray
Bradbury - El Parque de los Juegos.txt

(base) D:\ULTIMO SEMESTRE\Progrmmacion en la nu87be\bucket>
```

Creación de una carpeta

```
03-crearCarp.py > ...
6 # se crean dos carpetas dentro del bucket definido
7 bucket = 'naula-barros-chango'
8 subcarp1 = 'ejemplo/'
9 subcarp2 = 'otroEjemplo/'
10
11 cliente.put_object(Bucket = bucket,
12                   Key = subcarp1)
13 cliente.put_object(Bucket = bucket,
14                   Key = subcarp2)
15
16 # Mostrar las objetos almacenados en el bucket en funcion de su ultima fecha de modificacion
17 objt_int = cliente.list_objects(Bucket = bucket)
18
19 for a in objt_int['Contents']:
20
```

TERMINAL

```
(base) D:\ULTIMO SEMESTRE\Progrmmacion en la nu87be\bucket>conda activate base

(base) D:\ULTIMO SEMESTRE\Progrmmacion en la nu87be\bucket>C:/Users/USUARIO/anaconda3/python.exe "d:/ULTIMO
SEMESTRE/Progrmmacion en la nu87be\bucket/01-estConexion.py"
naula-barros-chango

(base) D:\ULTIMO SEMESTRE\Progrmmacion en la nu87be\bucket>C:/Users/USUARIO/anaconda3/python.exe "d:/ULTIMO
SEMESTRE/Progrmmacion en la nu87be\bucket/03-crearCarp.py"
Las Leyes de Murphy.txt 2023-07-19 18:35:05+00:00
Ray Bradbury - El Parque de los Juegos.txt 2023-07-20 19:05:18+00:00
ejemplo/ 2023-07-23 12:41:45+00:00
otroEjemplo/ 2023-07-23 12:41:45+00:00
test/ 2023-07-19 18:35:30+00:00

(base) D:\ULTIMO SEMESTRE\Progrmmacion en la nu87be\bucket>
```

Creando un nuevo bucket



UNIVERSIDAD IBEROAMERICANA DEL ECUADOR LABORATORIOS PRACTICOS

```
1 import boto3
2
3 # Definir un metodo que establezca la conexion
4 cliente = boto3.client('s3', aws_access_key_id = 'AKIAIHCHEBVUIA6YN34K', aws_secret_access_key = 'cOTXaJxHmh/FurAv')
5
6 # Crear nuevo bucket dentro de mi S3, la definicion de las variables
7
8 bucket = 'naula-barros-chango'
9
10 cliente.create_bucket(Bucket = bucket)
11
12 # Verificar la creacion del nuevo Bucket
13
14 resp = cliente.list_buckets()
15
16 for b in resp['Buckets']:
17     print(b['Name'])
```

Ray Bradbury - El Parque de los Juegos.txt 2023-07-20 19:05:10+00:00
ejemplo/ 2023-07-23 12:41:45+00:00
otroEjemplo/ 2023-07-23 12:41:45+00:00
test/ 2023-07-19 18:35:30+00:00

(base) D:\ULTIMO SEMESTRE\Programacion en la nu87be\bucket>C:\Users\USUARIO\anaconda3\python.exe "d:\ULTIMO SEMESTRE\Programacion en la nu87be\bucket\04-crearBucket.py"

creación de bucket

Descargando Archivos

```
1 import boto3
2 import os
3
4 # Definir un metodo que establezca la conexion
5 cliente = boto3.client('s3', aws_access_key_id = 'AKIAIHCHEBVUIA6YN34K', aws_secret_access_key = 'cOTXaJxHmh/FurAv')
6
7 # Definir las variables de descarga
8
9 bucket = 'naula-barros-chango'
10 path_act = os.getcwd()
11 arch = 'Ray Bradbury - El Parque de los Juegos.txt'
12 path_arch = os.path.join(path_act, 'descargas', arch)
13
14 # Definir la funcion de descarga
15
16 cliente.download_file(Bucket = bucket,
17                       Key = arch,
18                       Filename = path_arch)
19
20 down_dir = os.path.join(path_act, 'descargas')
21
22 for root, dirs, files in os.walk(down_dir):
```

t this request was sent to.

(base) D:\ULTIMO SEMESTRE\Programacion en la nu87be\bucket>C:\Users\USUARIO\anaconda3\python.exe "d:\ULTIMO SEMESTRE\Programacion en la nu87be\bucket\05-descargaArch.py"

Ray Bradbury - El Parque de los Juegos.txt

(base) D:\ULTIMO SEMESTRE\Programacion en la nu87be\bucket>

descarga de archivo

Borrando Archivos



UNIVERSIDAD IBEROAMERICANA DEL ECUADOR LABORATORIOS PRACTICOS

```
13 cliente.delete_object(bucket = bucket,
14                           Key = archivo)
15
16 objetosBorr = cliente.list_objects(bucket = bucket)
17
18 # Imprimir todos los objetos que estan en el bucket
19
20 print(f"Lista de objetos almacenados en el (bucket)")
21
22 for a in objetosBorr['Contents']:
23     print(a['Key'])
24
25 print("\nLista de todos los archivos borrados: ")
26
27 for a in objetosBorr['Contents']:
28     if 'ejemplo/' in a['Key'] and a['Key'] != 'ejemplo/':
29         print(f"El archivo que se va a borrar es(a['Key'])")
30         borrar(a['Key'])
31
32
33
34
```

boarrar archivos

Eliminando Bucket

```
1 import boto3
2
3 # Definir un metodo que establezca la conexion
4 cliente = boto3.client('s3', aws_access_key_id = 'AKIAWCHDHBVUIAGYH34K', aws_secret_access_key = 'c0TXajXwHmh/FurAw
5
6 bucket = 'naula-barros-chango'
7
8 lista = cliente.list_buckets()
9
10 for n in lista['Buckets']:
11     print(n['Name'])
12
13 cliente.delete_bucket(bucket = bucket)
14
15 print('\n')
16 lista = cliente.list_buckets()
17
18 for n in lista['Buckets']:
19     print(n['Name'])
20
21
```

eliminar bucket

Instalación de aws cli en mi estructura de EC2



UNIVERSIDAD IBEROAMERICANA DEL ECUADOR LABORATORIOS PRACTICOS

```
18.190.101.4 (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2. 18.190.101.4 (ubuntu)
Selecting previously unselected package libxrender1:amd64.
Preparing to unpack .../66-libxrender1_1:0.9.10-1build4_amd64.deb ...
Unpacking libxrender1:amd64 (1:0.9.10-1build4) ...
Selecting previously unselected package libcairo2:amd64.
Preparing to unpack .../67-libcairo2_1.16.0-5ubuntu2_amd64.deb ...
Unpacking libcairo2:amd64 (1.16.0-5ubuntu2) ...
Selecting previously unselected package libdatrie1:amd64.
Preparing to unpack .../68-libdatrie1_0.2.13-2_amd64.deb ...
Unpacking libdatrie1:amd64 (0.2.13-2) ...
Selecting previously unselected package libdjvulibre-text.
Preparing to unpack .../69-libdjvulibre-text_3.5.28-2build2_all.deb ...
Unpacking libdjvulibre-text (3.5.28-2build2) ...
Selecting previously unselected package libdjvulibre2:amd64.
Preparing to unpack .../70-libdjvulibre2_3.5.28-2build2_amd64.deb ...
Unpacking libdjvulibre2:amd64 (3.5.28-2build2) ...
Selecting previously unselected package libgraphite2-3:amd64.
Preparing to unpack .../71-libgraphite2-3_1.3.14-1build2_amd64.deb ...
Unpacking libgraphite2-3:amd64 (1.3.14-1build2) ...
Selecting previously unselected package libharfbuzz0b:amd64.
Preparing to unpack .../72-libharfbuzz0b_2.7.4-1ubuntu3.1_amd64.deb ...
Unpacking libharfbuzz0b:amd64 (2.7.4-1ubuntu3.1) ...
Selecting previously unselected package liblmbase25:amd64.
Preparing to unpack .../73-liblmbase25_2.5.7-2_amd64.deb ...
Unpacking liblmbase25:amd64 (2.5.7-2) ...
Selecting previously unselected package libimagequant0:amd64.
Preparing to unpack .../74-libimagequant0_2.17.0-1_amd64.deb ...
Unpacking libimagequant0:amd64 (2.17.0-1) ...
Selecting previously unselected package libjxr0:amd64.
Preparing to unpack .../75-libjxr0_1.2-g1t20170615-f752187-5_amd64.deb ...
Unpacking libjxr0:amd64 (1.2-g1t20170615-f752187-5) ...
Selecting previously unselected package libjxr-tools.
Preparing to unpack .../76-libjxr-tools_1.2-g1t20170615-f752187-5_amd64.deb ...
Unpacking libjxr-tools (1.2-g1t20170615-f752187-5) ...
Selecting previously unselected package libopenxr25:amd64.
Preparing to unpack .../77-libopenxr25_2.5.7-1_amd64.deb ...
Unpacking libopenxr25:amd64 (2.5.7-1) ...
```

Verificando la Versión

```
18.190.101.4 (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2. 18.190.101.4 (ubuntu)
update-alternatives: using /usr/bin/montage-im6.q16 to provide /usr/bin/montage (montage) in auto mode
update-alternatives: using /usr/bin/montage-im6.q16 to provide /usr/bin/montage-im6 (montage-im6) in auto mode
update-alternatives: using /usr/bin/mogrify-im6.q16 to provide /usr/bin/mogrify (mogrify) in auto mode
update-alternatives: using /usr/bin/mogrify-im6.q16 to provide /usr/bin/mogrify-im6 (mogrify-im6) in auto mode
Setting up imagemagick (8:6.9.11-60+dfsg-1.3ubuntu0.22.04.3) ...
Processing triggers for install-info (6.8-4build1) ...
Processing triggers for mailcap (3.70+nm1ubuntu1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for shared-mime-info (2.1-2) ...
Processing triggers for sgml-base (1.30) ...
Setting up docutils-common (0.17.1+dfsg-2) ...
Processing triggers for sgml-base (1.30) ...
Setting up python3-docutils (0.17.1+dfsg-2) ...
Setting up awscli (1.22.34-1) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Restarting services...
/etc/needrestart/restart.d/systemd-manager
systemctl restart chrony.service multipathd.service mysql.service packagekit.service polkit.service rsyslog.service ssh.service systemd-journald.service systemd-networkd.service systemd-resolved.service systemd-udev.service systemd-udevd.service
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-15-75:~$ aws --version
aws-cli/1.22.34 Python/3.10.6 Linux/5.19.0-1025-aws botocore/1.23.34
ubuntu@ip-172-31-15-75:~$
```

Ingreso de credenciales (clave valor)



UNIVERSIDAD IBEROAMERICANA DEL ECUADOR LABORATORIOS PRACTICOS

```
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for shared-mime-info (2.1-2) ...
Processing triggers for sgml-base (1.30) ...
Setting up docutils-common (0.17.1+dfsg-2) ...
Processing triggers for sgml-base (1.30) ...
Setting up python3-docutils (0.17.1+dfsg-2) ...
Setting up awscli (1.22.34-1) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Restarting services...
/etc/needrestart/restart.d/systemd-manager
systemctl restart chrony.service multipathd.service mysql.service packagekit.service polkit.service rsyslog.service ssh.service systemd-
journal.service systemd-networkd.service systemd-resolved.service systemd-udev.service
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-15-75:~$ aws --version
aws-cli/1.22.34 Python/3.10.6 Linux/5.19.0-1025-aws botocore/1.23.34
ubuntu@ip-172-31-15-75:~$ aws s3 ls
Unable to locate credentials. You can configure credentials by running "aws configure".
ubuntu@ip-172-31-15-75:~$ aws configure
AWS Access Key ID [None]: AKIAWCHQBUIA6YN34K
AWS Secret Access Key [None]: c0TXaJxWmhH/FurAw8v5kP5/SYlCGYHqmf5ZHPqc
Default region name [None]:
Default output format [None]:
ubuntu@ip-172-31-15-75:~$
```

Conexión de EC2 a S3 con el bucket naula-barros-chango

```
Processing triggers for shared-mime-info (2.1-2) ...
Processing triggers for sgml-base (1.30) ...
Setting up docutils-common (0.17.1+dfsg-2) ...
Processing triggers for sgml-base (1.30) ...
Setting up python3-docutils (0.17.1+dfsg-2) ...
Setting up awscli (1.22.34-1) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Restarting services...
/etc/needrestart/restart.d/systemd-manager
systemctl restart chrony.service multipathd.service mysql.service packagekit.service polkit.service rsyslog.service ssh.service systemd-
journal.service systemd-networkd.service systemd-resolved.service systemd-udev.service
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-15-75:~$ aws --version
aws-cli/1.22.34 Python/3.10.6 Linux/5.19.0-1025-aws botocore/1.23.34
ubuntu@ip-172-31-15-75:~$ aws s3 ls
ubuntu@ip-172-31-15-75:~$ aws configure
AWS Access Key ID [None]: AKIAWCHQBUIA6YN34K
AWS Secret Access Key [None]: c0TXaJxWmhH/FurAw8v5kP5/SYlCGYHqmf5ZHPqc
Default region name [None]:
Default output format [None]:
ubuntu@ip-172-31-15-75:~$ aws s3 ls
2023-07-19 18:31:31 naula-barros-chango
ubuntu@ip-172-31-15-75:~$
```

Análisis de resultados:

- Creación exitosa de un bucket en Amazon S3 con un nombre único y adecuado.
- Almacenamiento exitoso de diversos tipos de datos (archivos) en el bucket creado.



**UNIVERSIDAD IBEROAMERICANA DEL ECUADOR
LABORATORIOS PRACTICOS**

- Organización correcta de los objetos dentro del bucket para facilitar su acceso y recuperación.
- Implementación de una estructura en Amazon S3 que permitió la modificación de archivos y metadatos mediante un script de Python.
- Definición y asignación de metadatos a los objetos almacenados en el bucket para facilitar la interacción inteligente con los datos desde el script.
- Instalación exitosa de AWS CLI en una instancia de Amazon EC2.
- Configuración de las credenciales de AWS en la instancia EC2 para permitir la interacción con los servicios de AWS desde la línea de comandos.

Conclusiones y recomendaciones:

Conclusiones

- La integración de Amazon S3 y Amazon EC2 permite una gestión efectiva de datos en la nube, facilitando el almacenamiento y procesamiento escalable de recursos en AWS.
- La configuración adecuada de metadatos en Amazon S3 mejora la organización y automatización de tareas, optimizando el acceso y manipulación de datos en el bucket.
- La instalación de AWS CLI en instancias EC2 simplifica la administración de servicios de AWS desde la línea de comandos, mejorando la eficiencia operativa en la nube.

Recomendaciones

- Se recomienda seguir buenas prácticas de seguridad y acceso, como la implementación de políticas de control de acceso, para proteger los datos almacenados en Amazon S3.
- Es conveniente documentar y etiquetar adecuadamente los objetos y buckets en Amazon S3, para facilitar su identificación y seguimiento en entornos con múltiples recursos.



UNIVERSIDAD IBEROAMERICANA DEL ECUADOR
LABORATORIOS PRACTICOS

- Se sugiere explorar más funcionalidades de AWS CLI y Boto3 para aprovechar al máximo el potencial de la gestión programática y automatizada en la nube de AWS.

Bibliografía:

Descripción general de los buckets - Amazon Simple Storage Service. (n.d.).

https://docs.aws.amazon.com/es_es/AmazonS3/latest/userguide/UsingBucket.html

Jhonny. (2019, May 21). *AWS SDK para Python (Boto3)* - ▷ Cursos de

Programación de 0 a Experto © Garantizados. ▷ Cursos De

Programación De 0 a Experto © Garantizados.

[https://unipython.com/aws-sdk-para-python-](https://unipython.com/aws-sdk-para-python-boto3/#:~:text=%C2%BFQu%C3%A9%20es%20Boto3%3F,manejar%20y%20configurar%20dichos%20servicios)

[boto3/#:~:text=%C2%BFQu%C3%A9%20es%20Boto3%3F,manejar%20y%20configurar%20dichos%20servicios.](https://unipython.com/aws-sdk-para-python-boto3/#:~:text=%C2%BFQu%C3%A9%20es%20Boto3%3F,manejar%20y%20configurar%20dichos%20servicios)

What is Amazon EC2? - Amazon Elastic Compute Cloud. (n.d.).

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>