

Interactuar con AWS usando Python

boto3

Definición

`boto3` es la Biblioteca de Amazon Web Services (AWS) para Python, que permite a los desarrolladores interactuar con los servicios de AWS mediante código Python. Proporciona una API sencilla y coherente para realizar acciones como la creación, administración y configuración de servicios y recursos en AWS.

Características

1. Clientes y Recursos:

- **Clientes:** Proporcionan acceso a las operaciones de bajo nivel disponibles en AWS. Permiten hacer llamadas directas a la API de AWS.
- **Recursos:** Proporcionan una abstracción de alto nivel sobre los clientes, facilitando la interacción con los servicios de AWS mediante una interfaz orientada a objetos.

2. Gestión de Credenciales:

- `boto3` puede manejar credenciales automáticamente si se configuran correctamente en tu entorno mediante herramientas como `aws configure` o mediante el uso de roles de IAM en instancias de EC2.

3. Sesiones:

- Una sesión es un contenedor que mantiene la configuración y las credenciales necesarias para interactuar con los servicios de AWS.

4. Operaciones Asíncronas:

- Soporte para operaciones asíncronas que permiten la ejecución de tareas en segundo plano.

5. Facilidad de Uso:

- `boto3` proporciona una interfaz intuitiva y fácil de usar para interactuar con los servicios de AWS, lo que facilita a los desarrolladores la integración de AWS en sus aplicaciones.

Servicios Compatibles

`boto3` es compatible con una amplia gama de servicios de AWS, incluidos pero no limitados a:

- Amazon EC2 (Elastic Compute Cloud)
- Amazon S3 (Simple Storage Service)
- Amazon RDS (Relational Database Service)
- AWS Lambda
- DynamoDB
- Amazon SNS (Simple Notification Service)
- Amazon SQS (Simple Queue Service)

Ventajas

- **Automatización:** Permite la automatización de tareas comunes en AWS.
- **Escalabilidad:** Facilita la creación y gestión de recursos de AWS a gran escala.
- **Integración:** Se integra fácilmente con aplicaciones basadas en Python.

Sesiones con boto3 y Python

¿Qué es una Sesión?

En el contexto de AWS y la biblioteca `boto3`, una sesión es una instancia de la clase `boto3.Session` que mantiene la configuración y las credenciales necesarias para interactuar con los servicios de AWS. Una sesión es un contenedor que guarda información como tus credenciales de acceso (Access Key ID y Secret Access Key), la región predeterminada, y otros parámetros de configuración que son necesarios para hacer llamadas a los servicios de AWS.

Características de una Sesión con boto3

1. Gestión de Credenciales:

- La sesión puede ser configurada con credenciales específicas o puede utilizar las credenciales configuradas globalmente en tu entorno (por ejemplo, mediante el comando `aws configure`).
- Esto permite reutilizar las credenciales de una manera segura y eficiente.

2. Configuración de la Región:

- Puedes especificar la región de AWS en la que deseas operar.
- La región predeterminada se puede configurar en la sesión para asegurarte de que todas las llamadas a los servicios de AWS se hagan en la región correcta.

3. Reutilización:

- Las sesiones permiten la reutilización de credenciales y configuración en múltiples servicios y clientes dentro de tu aplicación.

Beneficios de Usar Sesiones

- **Seguridad:** Almacenar credenciales fuera del código fuente ayuda a mantener las credenciales seguras.
- **Flexibilidad:** Puedes cambiar la configuración de la región o las credenciales sin necesidad de modificar el código de tus clientes y recursos.
- **Reutilización:** Puedes reutilizar la misma sesión para múltiples servicios, lo que simplifica la gestión de credenciales y configuración en tu aplicación.

Creación de Sesión en Python

En esta práctica, se aprenderá a crear y verificar una sesión de AWS usando `boto3`. Los pasos a seguir serán:

1. Crear un usuario IAM con la política de permisos necesaria (En este caso será de administrador) y obtener sus claves.
2. Configurar una sesión de boto3 usando las claves de acceso del usuario IAM.
3. Verificar que la sesión está activa.

```

In [ ]: import boto3
import os
from botocore.exceptions import NoCredentialsError, PartialCredentialsError, \
    EndpointConnectionError

session = boto3.Session(
    aws_access_key_id = os.getenv('AWS_ACCESS_KEY_ID'),
    aws_secret_access_key = os.getenv('AWS_SECRET_ACCESS_KEY'),
    region_name = 'us-east-1' # Puedes cambiar la región según tu preferencia
)

# Crear un cliente de EC2
ec2_client = session.client('ec2')

try:
    # Intentar hacer una llamada para describir las instancias EC2
    response = ec2_client.describe_instances()
    print("La sesión está activa y funcionando correctamente.")
    for reservation in response['Reservations']:
        for instance in reservation['Instances']:
            print(f'Instance ID: {instance["InstanceId"]}, State: {instance["State"]["Name"]}')
except NoCredentialsError:
    print("No se encontraron credenciales. Asegúrate de que las credenciales \
        están configuradas correctamente.")
except PartialCredentialsError:
    print("Las credenciales están incompletas. Por favor, verifica tu AWS \
        Access Key ID y Secret Access Key.")
except EndpointConnectionError:
    print("No se pudo conectar al punto final de AWS. Verifica tu conexión a \
        Internet y la región configurada.")
except Exception as e:
    print(f'Ocurrió un error: {e}')

# Otras validaciones
try:
    sts_client = session.client('sts')
    caller_identity = sts_client.get_caller_identity()
    print(f"Sesión activa para la cuenta: {caller_identity['Account']}, \
        usuario: {caller_identity['Arn']}")
except Exception as e:
    print(f"No se pudo obtener la identidad del llamador: {e}")

```

La sesión está activa y funcionando correctamente.

Sesión activa para la cuenta: 058264393125, usuario: arn:aws:iam::058264393125:root

En el contexto de boto3 y AWS, no existe un concepto formal de "cerrar sesión" como lo harías con una sesión de usuario en una aplicación web. Una vez que has configurado una sesión y realizado las operaciones necesarias, simplemente puedes dejar de usar la sesión. No es necesario liberar recursos manualmente, ya que boto3 gestiona esto automáticamente.