AskPython

Corregir el error "El complemento de autenticación 'caching_sha2_password' no es compatible"

Por Gurpreet Kaur / 27 de noviembre de 2023

Conectarse a bases de datos MySQL desde Python a menudo resulta en el frustrante error: "El complemento de autenticación 'caching_sha2_password' no es compatible". Esto ocurre debido a una falta de coincidencia entre el complemento de autenticación utilizado por el servidor MySQL y el complemento compatible con la biblioteca de conectores MySQL en Python.

En esta guía, comprenderemos la causa raíz de este error, exploraremos soluciones para solucionarlo y veremos ejemplos del mundo real de cómo establecer conexiones de bases de datos desde aplicaciones de Python.

00:27

Lea también: ***Cómo corregir el error "La función no está definida" en Python***

¿Por qué no es compatible el complemento de autenticación 'caching_sha2_password'

Para comprender por qué se produce este error, vamos a crear una tabla de ejemplo para que podamos usarla como guía en nuestra explicación.

00:27

```
CREATE TABLE users (
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(50)
);
```

```
INSERT INTO users (name) VALUES
  ('John'),
  ('Sarah'),
  ('Peter');
```

Al intentar conectarse a esta base de datos desde Python utilizando la biblioteca mysql-connector, el siguiente código da como resultado un error:

```
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="myuser",
  password="mypass"
)
```

Error:

```
mysql.connector.errors.NotSupportedError: Authentication plugin 'caching
```

Este error se produce porque MySQL 8.0 y superior utilizan un complemento de autenticación de contraseña más seguro llamado para nuevas cuentas de usuario.`caching_sha2_password`

Sin embargo, la ampliamente utilizada biblioteca de Python mysql-connector aún no es compatible con este nuevo complemento. Solo es compatible con el complemento más antiguo utilizado en versiones anteriores de MySQL.`mysql_native_password`

00:27

Por lo tanto, hay una discrepancia entre el complemento de autenticación esperado en el servidor MySQL y el admitido en mysql-connector, lo que hace que las conexiones fallen.

Lea también: Firebase ImportError: No se pudo importar la biblioteca de Cloud Firestore

Cómo arreglar el error del complemento de autenticación 'caching_sha2_password'

Hay algunas formas de resolver este problema:

1. Cambiar el complemento de autenticación de usuario en MySQL

Podemos cambiar el plugin utilizado para la cuenta de usuario de MySQL a:`mysql_native_password`

00:27

```
ALTER USER 'myuser'@'localhost' IDENTIFIED WITH mysql_native_password BY
```

Ahora MySQL y python mysql-connector coinciden en términos de complemento:

```
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="myuser",
  password="mypass"
)

print(mydb) # <mysql.connector.connection_cext.CMySQLConnection object>
```

Sin embargo, este método compromete la seguridad, ya que el complemento mysql_native_password está desactualizado y es menos seguro que caching_sha2_password.

00:27

2. Utilice la biblioteca mysql-connector-python

La biblioteca mysql-connector-python es compatible con el nuevo complemento de autenticación.`caching_sha2_password`

Primero desinstale el paquete mysql-connector:

```
pip uninstall mysql-connector
```

A continuación, instale mysql-connector-python:

```
pip install mysql-connector-python
```

Ahora el código de conexión funciona con el plugin más nuevo:

```
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="myuser",
  password="mypass"
)
```

00:27

```
print(mydb) # <mysql.connector.connection_cext.CMySQLConnection object>
```

Este es el enfoque recomendado, ya que mantiene la seguridad intacta.

3. Especificar el complemento de autenticación en la llamada Connect

Podemos especificar explícitamente el complemento al conectarnos desde mysql-connector:`mysql_native_password`

```
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="myuser",
  password="mypass",
  auth_plugin='mysql_native_password'
)
```

Sin embargo, este método no se recomienda ya que el complemento de contraseña nativa tiene problemas de seguridad.

00:27

Comparación de soluciones

Solución	Seguridad	Pasos adicionales	Obras
Cambiar la autenticación de usuario de MySQL	Comprometido	Consulta ALTER USER	Sí
Usar mysql-connector-python	Intacto	Instalar biblioteca	Sí
Especificar el complemento de autenticación en el código	Comprometido	Ninguno	Sí

Real-World Example: Connecting Flask App to MySQL

Let's see examples of resolving this error when connecting Flask and Django apps to MySQL:

Error:

```
from flaskext.mysql import MySQL

mysql = MySQL()

app = Flask(__name__)
# MySQL configurations
app.config['MYSQL_DATABASE_USER'] = 'myuser'
```

00:27

```
app.config['MYSQL_DATABASE_PASSWORD'] = 'mypass'
app.config['MYSQL_DATABASE_DB'] = 'mydb'
app.config['MYSQL_DATABASE_HOST'] = 'localhost'
mysql.init_app(app)

conn = mysql.connect() # ERROR !
```

Solution:

Simply change to use the mysql-connector-python library instead of flaskext.mysql. This connector supports the new authentication plugin:

```
import mysql.connector

mysql = MySQLconnector()

app = Flask(__name__)
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'myuser'
app.config['MYSQL_PASSWORD'] = 'mypass'
app.config['MYSQL_DB'] = 'mydb'

mysql.init_app(app)
```

00:27

```
conn = mysql.connect() # Works!
```

Summary

Some key points to remember:

- The error occurs due to a mismatch between the MySQL plugin and Python mysql connector library
- Use mysql-connector-python library to resolve it without downgrading security
- Changing the plugin in MySQL also fixes it but compromises the security
- Can specify auth plugin in connect call but has vulnerabilities.

The “Authentication plugin ‘caching_sha2_password’ is not supported” error occurs due to a mismatch between the MySQL authentication plugin and the one supported by the Python MySQL connector library.

We explored various solutions like changing the plugin in MySQL, using the mysql-connector-python library or explicitly specifying the auth plugin in connect call.

Using the mysql-connector-python library is the recommended approach as it works without downgrading MySQL security.

I hope this guide gave you a comprehensive understanding of why this error occurs and actionable solutions to fix it for establishing connections from Python to MySQL databases.

00:27

Reference: ~~Reddit.~~

← Previous Post

Next Post →

Buscar...

Publicaciones recientes

¿Qué lista de criterios tienen los diseñadores y desarrolladores para garantizar el buen funcionamiento de un sitio?

¿Quieres convertirte en programador informático? Aquí hay 15 cursos que vale la pena analizar

Cómo construir una billetera criptográfica usando Python: una guía paso a paso

Los mejores métodos para obtener texto de las imágenes en 2024

Aprovechar Python para proyectos avanzados de desarrollo web

7 cosas que quizás no sabías que puedes hacer con Python

Cómo mejorar el rendimiento de la red de Kubernetes con eBPF

Creación de una solución personalizada de supervisión de existencias mediante Web Scraping

Python vs. otros lenguajes web: ¿cuál es el más popular?

Configurar la política de seguridad de contenido con Flask Talisman

00:27

Sitios favoritos

Tutoriales de GoLang

Ayuda de VM

Tutoriales de Linux

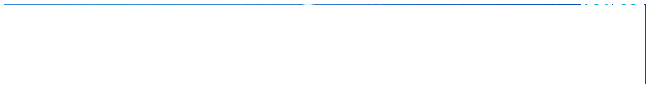
Tutoriales de MySQL

CodeForGeek

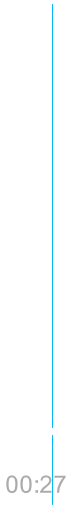
Mkyong

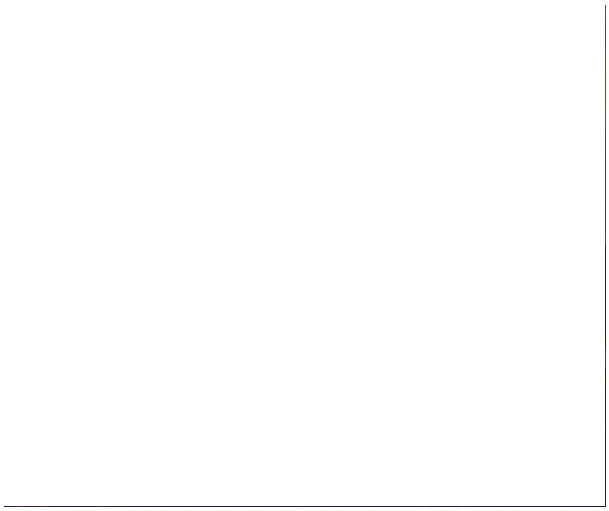
00:27

00:27



00:27


00:27



00:27

00:27

00:27