

mysql_native_password vs. caching_sha2_password

Nein, man kann den Passwort-Hash nicht umwandeln.

Das liegt an der Funktionsweise von kryptografischen Hashes: Sowohl `mysql_native_password` (SHA1-basiert) als auch `caching_sha2_password` (SHA-256-basiert) sind Einweg-Verschlüsselungen. Aus dem Hash auf dem alten Server lässt sich das ursprüngliche Klartext-Passwort mathematisch nicht wiederherstellen, um daraus den neuen SHA-256-Hash zu berechnen.

Du hast in dieser Situation jedoch **zwei praktische Wege**:

Option 1: Das alte Verfahren auf dem neuen Server aktivieren (Empfohlen, wenn Passwort unbekannt)

MySQL 8.0 unterstützt `mysql_native_password` weiterhin (in MySQL 8.0/8.4 ist es zwar deprecated oder als Plugin verfügbar, kann aber problemlos verwendet werden). Du kannst den User also **trotzdem mit seinem alten Hash anlegen**:

SQL

```
CREATE USER 'benutzername'@'localhost'
  IDENTIFIED WITH mysql_native_password AS '*2470C0C21F792A758062C1D2543813E071372E97';
```

(Ersetze den Stern-String mit dem Hash aus deinem alten `SHOW CREATE USER`-Befehl).

“ **Hinweis:** Falls MySQL auf dem neuen Server meckert (`Plugin 'mysql_native_password' is not loaded`), musst du das Plugin in der Server-Konfiguration (`my.cnf` / `mysqld.cnf`) unter `[mysqld]` aktivieren:

Ini, TOML

```
[mysqld]
mysql_native_password=ON
```

Danach den MySQL-Dienst neu starten (`systemctl restart mysql`).

Option 2: Neues Passwort mit `caching_sha2_password` vergeben

Wenn dir das Passwort im Klartext bekannt ist (oder du ein neues vergeben kannst), erstelle den User direkt mit dem neuen Standard-Verfahren:

SQL

```
CREATE USER 'benutzername'@'localhost'  
  IDENTIFIED WITH caching_sha2_password BY 'DeinNeuesOderAltesKlartextPasswort';
```

Fazit & Vorgehen

- **Kennst du das Passwort im Klartext?** → Nutze **Option 2** (sicherer und zukunftssicherer).
- **Kennst du das Passwort NICHT?** → Nutze **Option 1**, um den alten Hash direkt zu übernehmen.