Dokumentation LB3 Modul M141

Yassin Sobhy, Luca Greuter 06.05.2024



Inhalt

[Definition Infrastruktur 3](#_Toc165979728)

[Repo: https://github.com/LucaKubaGreuter/M141LB3 3](#_Toc165979729)

[Lokale Datenbank 3](#_Toc165979730)

[Erstellen der Datenbank (A1, A2) 3](#_Toc165979731)

[Umsetzung der User-Matrix (D1, C1) 3](#_Toc165979732)

[Importieren der CSV-Dateien (B1) 3](#_Toc165979733)

[Testprotokolle (C2) 4](#_Toc165979734)

[AWS-Datenbank 5](#_Toc165979735)

[Installation und Setup (A1) 5](#_Toc165979736)

[Migration (D1, B1, C1) 6](#_Toc165979737)

# Definition Infrastruktur

### Repo: <https://github.com/LucaKubaGreuter/M141LB3>

# Lokale Datenbank

### Erstellen der Datenbank (A1, A2)

Zuallererst haben wir auf unserem MySQL-Server Manuel die Datenbank «backpacker\_lb3» erstellt und dieses mit dem vorgegebenen ddl befüllt. Das war keine grosse Herausforderung da wir sowieso nur ein .sql Skript ausführen mussten.

### Umsetzung der User-Matrix (D1, C1)

Als nächstes haben wir die User-Matrix studiert und dann mit einem Skript die nötigen Benutzer nach Vorgabe erstellt. Wir haben einen «user» und ein «management» erstellt mit den angegebenen Berichtigungen.

Ein Bild, das Text, Screenshot, Software, Display enthält.

Automatisch generierte Beschreibung

Das war nicht so schwierig, sobald man wusste, wo, welche Berechtigungen hingehören. Doch es war schon eine Herausforderung, dass bei der Benutzer-Gruppe beim table tbl\_benutzer einzelne unter Berechtigungen erstellt werden mussten.

### Importieren der CSV-Dateien (B1)

Nun mussten wir die Datenbank noch mit Daten füllen, dazu bekamen wir für jeden table eine .csv Dateien, welche wir in den nötigen Table importieren mussten. Doch es funktionierte nicht beim ersten Mal, da anscheinend das CSV komisch formatiert war und wir das richtig im sql-skript angeben mussten.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Wenn man es wie im Bild importiert werden alle Daten korrekt in die tables importiert und sind genau wie in der CSV-Datei. Zudem haben wir die DB von unnötigem Index und Constraints bereinigt falls vorhanden. Zudem haben wir eine Optimierung durchgeführt.

Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

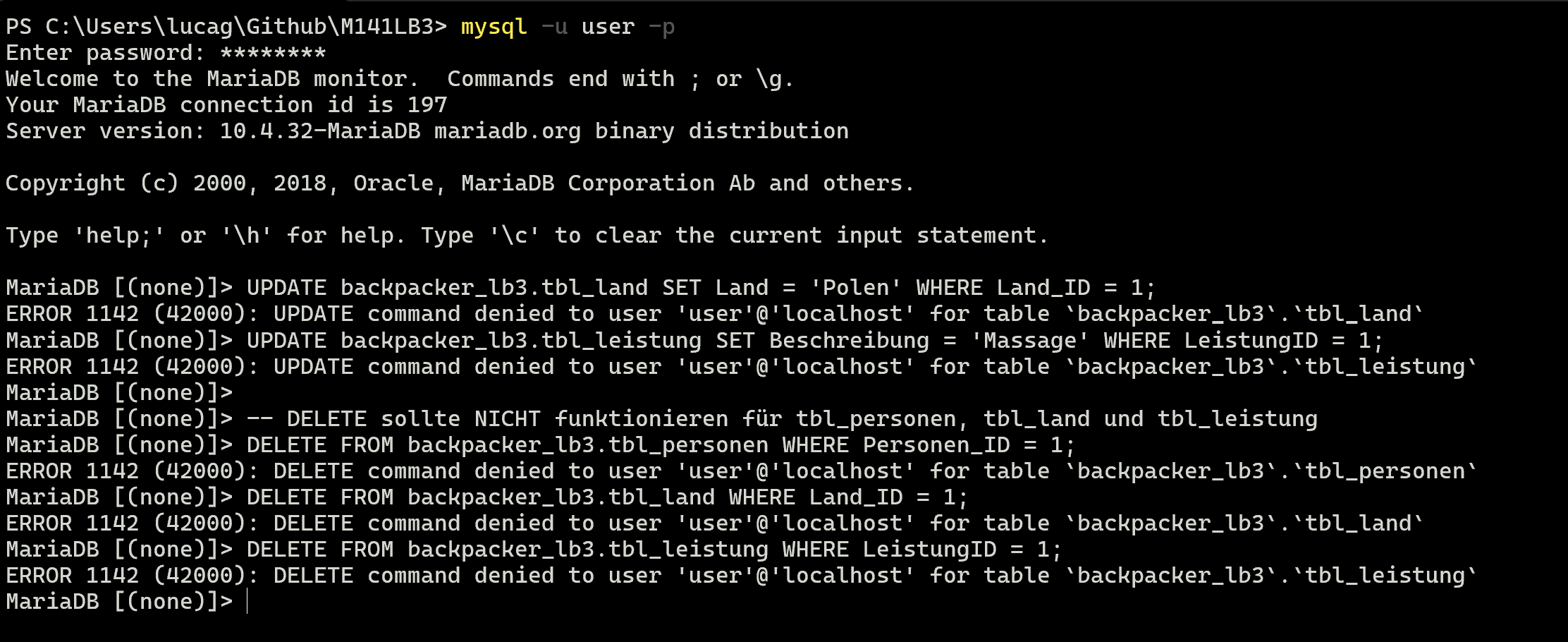
### Testprotokolle (C2)

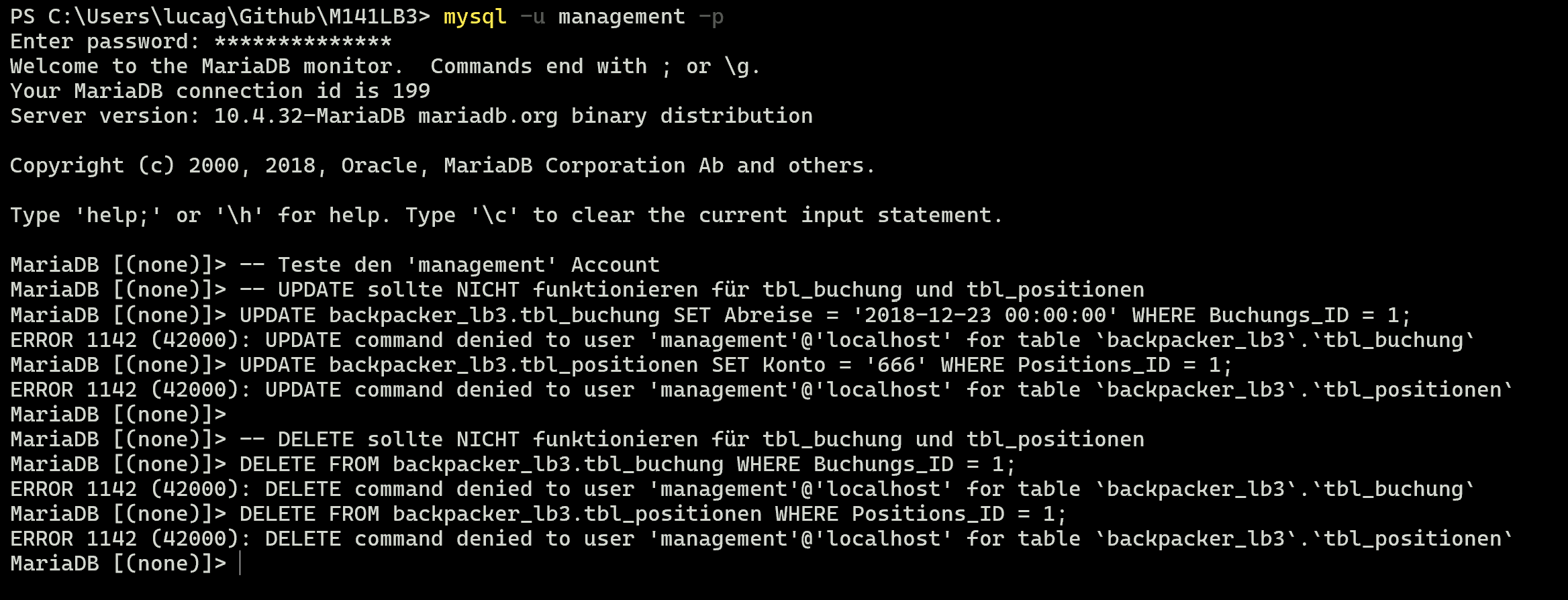
Um sicher zu gehen haben wir auch die Rollen der Benutzer getestet.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

In diesem Skript wird sichergestellt, dass die beiden Benutzer auch sicher nicht Sachen ändern können, welche sie nicht ändern sollten. Gleichzeitig wurde auch getestet, ob sich die Benutzer auf den MySQL-Server einloggen können.





Hier sieht man, dass man sich mit beide Benutzern auf den Server kommt und alles, was nicht funktionieren sollte, funktioniert auch nicht.

# AWS-Datenbank

### Installation und Setup (A1)

Wir haben eine AWS-Instanz erstellt mit Ubuntu, t2-medium da wir eh ausreichend Budget haben und wir haben ein Schlüsselpaar erstellt für die sichere SSH-Verbindung. Bei den eingehenden Regeln haben wir noch das port 3306 geöffnet, dass man von extern direkt auf den Server zugreifen kann.

Ein Bild, das Text, Screenshot, Software, Display enthält.

Automatisch generierte Beschreibung

Mit diesem init-file wird der mariadb richtig aufgesetzt und auch richtig konfiguriert für Verbindungen von extern.

### Migration (D1, B1, C1)

Bei der Migration haben wir ein Simples Skript geschrieben, mit welchem wir einen dump erstellen, diesen mit scp auf die Instanz zu kopieren.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Plus weitere Befehle, damit alles reibungslos lauft.

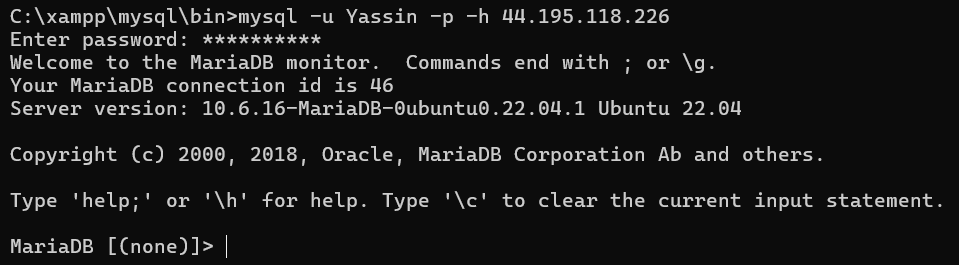
### Verbindung

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

User: Luca

PW: luca1234



User: Yassin

PW: yassin1234