Training - Grundlagen MySQL Admininstration (deutsch)

Agenda

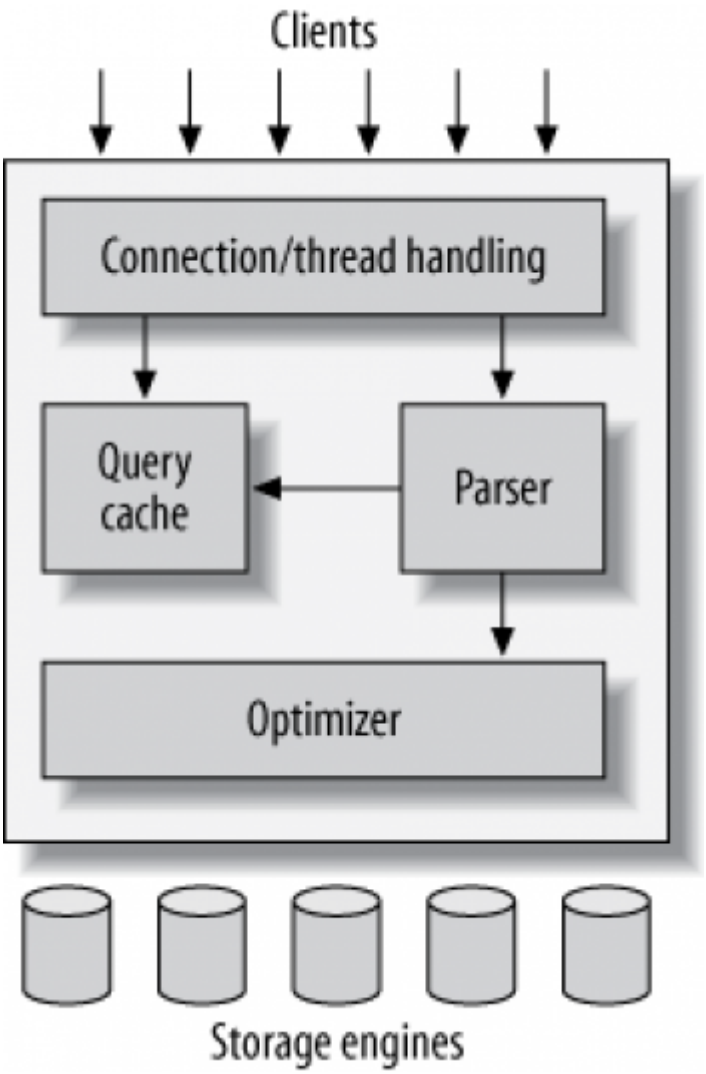
1. Architektur von MySQL
    - Verarbeitungsschritte Server (Schritte)
    - InnoDB Struktur
    - Storage Engines
    - Unterschiede MySQL 5.7 -> 8
2. Installation
    - Installation (Ubuntu)
    - Start/Status/Stop/Enable von MySQL
    - Lauscht mysql nach draussen ?
3. Konfiguration
    - Konfiguration anpassen und neu starten
4. Administration
    - Globale und Session Variablen (Server System Variables)
    - Error-log
    - Slow Query Log
5. Backup
    - Backup mit mysqldump - best practices
    - Backups PIT (Point-In-Time recovery)
    - Backup mit xtrabackup
    - Backup mit xtrabackup mit Verschlüsselung
    - Backup und Wiederherstellen in neuer Datenbank
    - mysqldump mit asynchroner Verschlüsselung
    - mydumper und myloader
6. Sicherheit
    - Absichern von Server/Client mit ssl
    - Verschlüsselte Backups mit xtrabackup
    - Prüfen ob socket verwendet, bei lokalem System
    - mysql_secure_installation - validate plugin aktivieren
    - general log deaktivieren
    - plugin vs. components
    - User Passwort-Länge und Passwort-Ablauf
    - Trigger ohne Super-Rechte anlegen
7. Sicherheit (CIS)
    - 1.1 Place Databases on Non-System Partitions
    - 1.2 Use dedicated Least Privileged Account
8. Tools
    - Testdatenbank Sakila installieren

9. Authentifizierung / User-Management
    - Für User altes Password-Verfahren mysql_native_password verwenden in MySQL 8
    - Rollen
10. Upgrade
    - Upgrade von MySQL 5.7 -> 8
11. Windows
    - Welchen Benutzer für den Service verwenden?
12. Documentation
    - Server System Variables - Reference
    - MySQL Performance Dokument - en

Architektur von MySQL

Verarbeitungsschritte Server (Schritte)

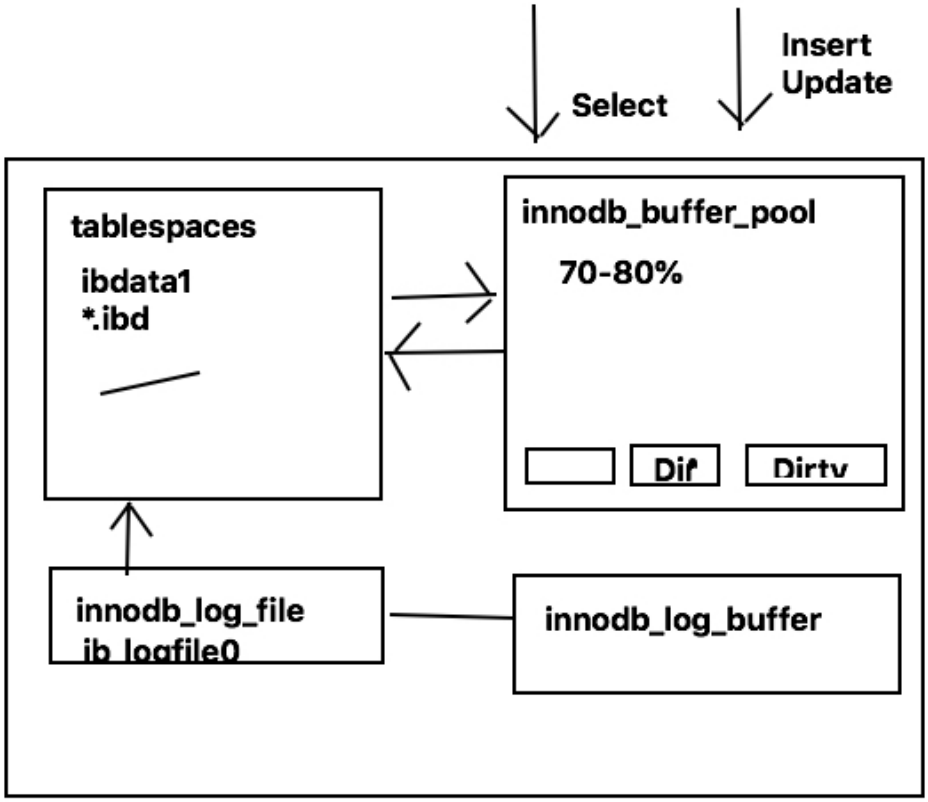
Overview



Changes in MySQL 8

- Query Cache gibt's nicht mehr.

InnoDB Struktur



Storage Engines

Warum ?

```
Du triffst die Auswahl:
Wie sollen Deine Daten gespeichert werden
```

Wie unterscheiden sich die Storage Engines ?

- In der Performance, Features und anderen Charakteristiken, die Du brauchst

Was machen Sie ?

- Sie sind zuständig für: Speichern und Lesen aller in MySQL Daten
- Jede Storage Engine hat:
  - Vor- und Nachteile
- Der Server kommuniziert mit den Storage Engines über die storage engine API
  - Unterschiede kann ich durch das Interface nicht sehen.
  - Die api enthält mehrere Dutzend low-level Funktionen z.B. “Beginne eine Transkation”, “Hole die Zeilen, die diesen Primärschlüssel hat”

Storage Engine machen folgendes NICHT ....

- Storage Engines parsen kein SQL
- Storage Engines kommunizieren nicht miteinander.

Welches sind die Wichtigsten ?

- MyISAM
- InnoDB (Default)
- Memory
- CSV
- Blackhole (/dev/null)
- Archive
- Federated

In Detail: MyISAM - Storage Engine

```
Feautures/Vorteile/Nachteile

Tabellen-Locks auf Tabellenebene.
Kein automatisches Data-Recovery
Daten z.B. bei Stromausfall können verloren (bis zu 8 Sekunden)
Kein Transaktionen
Indizes werden im Arbeitsspeicher vorgehalten

Vorteil:
Kompakte Datenspeicherung
Table Scans sind sehr schnell
```

In Detail: InnoDB - Storage Engine

```
Features

Unterstützt hot backups (wg. Transaktionen)
Transaktionen werden unterstüzt
Foreign Keys werden unterstützt
row-level locking
multi-versioning

indexes referenzieren die Daten über Primärschlüssel
indexes can quickly get huge in size
→ if size of primary index is not small

Sehr effektives Handling von Daten im Arbeitsspeicher
```

Unterschiede MySQL 5.7 -> 8

In Version 8 schnellere Feature-Wechsel

```
Von minor zu minor version, sehr viele neue Features

8.0.23 -> 8.0.24

Das war vorher eher stabiler in der Form ganz wenigen bis gar keinen neuen Features
```

Wegfall von *.frm - Datein von MySQL 5.7 -> 8

Set persist (neu in Version 8)

```
Während der Laufzeit server system variablen persistent setzen
```

mysql ssl verbindung

```
--ssl geht nicht mehr in MySQL 8
stattdessen:
--ssl-mode=REQUIRED
```

Komponenten / Components

```
## Alternative zu den Plugins

Components/Komponenten sind neu in MySQL 8.
```

Installation

Installation (Ubuntu)

Walkthrough

```
apt update
apt install mysql-server
```

Secure installation

```
mysql_secure_installation
```

Start/Status/Stop/Enable von MySQL

start/stop/status

```
## als root - Benutzer
systemctl status mysql
systemctl stop mysql
systemctl start mysql
```

Aktivieren/Deaktivieren (enable/disable)

```
## Automatischen Starten nach dem Booten (enable)
systemctl enable mysql

## is dienst aktiviert
systemctl is-enabled mysql
```

```
## deaktivieren
systemctl disable mysql
systemctl is-enabled mysql

## systemctl status -> Zeile disabled/enabled
```

Lauscht mysql nach draussen ?

Wie finde ich das raus ?

```
lsof -i | grep mysql
## localhost means it does NOT listen to the outside now
## mysqld  5208           mysql   19u  IPv4  56942      0t0  TCP localhost:mysql
(LISTEN)
```

Konfiguration

Konfiguration anpassen und neu starten

Administration

Globale und Session Variablen (Server System Variables)

```
mysql> show session variables like 'PERFORMANCE%schema';
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| performance_schema | ON    |
+--------------------+-------+
1 row in set (0.00 sec)

mysql> select @@performance_schema;
+----------------------+
| @@performance_schema |
+----------------------+
|                    1 |
+----------------------+
1 row in set (0.00 sec)

mysql> select @@SESSION.performance_schema;
ERROR 1238 (HY000): Variable 'performance_schema' is a GLOBAL variable
mysql> select @@performance_schema;
+----------------------+
| @@performance_schema |
+----------------------+
|                    1 |
+----------------------+
1 row in set (0.00 sec)

mysql> select @@GLOBAL.long_query_time;
```

```
+--------------------------+
| @@GLOBAL.long_query_time |
+--------------------------+
|                10.000000 |
+--------------------------+
1 row in set (0.00 sec)

mysql> select @@SESSION.long_query_time;
+---------------------------+
| @@SESSION.long_query_time |
+---------------------------+
|                 10.000000 |
+---------------------------+
1 row in set (0.00 sec)

mysql> set SESSION long_query_time=0.000001
    -> ;
Query OK, 0 rows affected (0.00 sec)

mysql> select @@SESSION.long_query_time;
+---------------------------+
| @@SESSION.long_query_time |
+---------------------------+
|                  0.000001 |
+---------------------------+
1 row in set (0.00 sec)

mysql> select @@GLOBAL.long_query_time;
+--------------------------+
| @@GLOBAL.long_query_time |
+--------------------------+
|                10.000000 |
+--------------------------+
1 row in set (0.00 sec)

mysql>
```

Error-log

```
show variables like 'log_error';
+---------------+---------------------------+
| Variable_name | Value                     |
+---------------+---------------------------+
| log_error     | /var/log/mysql/error.log |
+---------------+---------------------------+
1 row in set (0.00 sec)
```

Slow Query Log

Walkthrough

```
mysql> show variables like '%slow%';
+------------------------------+-------------------------------------+
| Variable_name                | Value                               |
+------------------------------+-------------------------------------+
| log_slow_admin_statements    | OFF                                 |
| log_slow_extra               | OFF                                 |
| log_slow_replica_statements | OFF                                 |
| log_slow_slave_statements    | OFF                                 |
| slow_launch_time             | 2                                   |
| slow_query_log               | OFF                                 |
| slow_query_log_file          | /var/lib/mysql-data/mysql2-slow.log |
+------------------------------+-------------------------------------+
7 rows in set (0.01 sec)

mysql> show variables like '%long%';
+-----------------------------------------------------------+-----------+
| Variable_name                                             | Value     |
+-----------------------------------------------------------+-----------+
| long_query_time                                           | 10.000000 |
| performance_schema_events_stages_history_long_size        | 10000     |
| performance_schema_events_statements_history_long_size    | 10000     |
| performance_schema_events_transactions_history_long_size | 10000     |
| performance_schema_events_waits_history_long_size         | 10000     |
+-----------------------------------------------------------+-----------+
5 rows in set (0.00 sec)

mysql> set slow_query_log = on;
ERROR 1229 (HY000): Variable 'slow_query_log' is a GLOBAL variable and should be set
with SET GLOBAL
mysql> set global slow_query_log = on;
Query OK, 0 rows affected (0.00 sec)

mysql> set global long_query_time = 0.000001;
Query OK, 0 rows affected (0.00 sec)

mysql> show session variables like 'long_query_time';
+-----------------+-----------+
| Variable_name   | Value     |
+-----------------+-----------+
| long_query_time | 10.000000 |
+-----------------+-----------+
1 row in set (0.00 sec)

mysql> show global variables like 'long_query_time';
+-----------------+----------+
| Variable_name   | Value    |
+-----------------+----------+
| long_query_time | 0.000001 |
+-----------------+----------+
1 row in set (0.01 sec)

mysql> quit
```

```
Bye
root@mysql2:/etc/mysql/mysql.conf.d# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.27-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show session variables like 'long_query_time';
+-----------------+----------+
| Variable_name   | Value    |
+-----------------+----------+
| long_query_time | 0.000001 |
+-----------------+----------+
1 row in set (0.00 sec)
```

Backup

Backup mit mysqldump - best practices

Useful options for PIT

```
## —quick not needed, because included in —opt which is enabled by default

## on local systems using socket, there are no huge benefits concerning --compress
## when you dump over the network use it for sure
mysqldump --all-databases --single-transaction --gtid --master-data=2 --routines --
events --flush-logs --compress > /usr/src/all-databases.sql;
```

With PIT_Recovery you can use --delete-master-logs

- All logs before flushing will be deleted

```
mysqldump --all-databases --single-transaction --gtid --master-data=2 --routines --
events --flush-logs --compress --delete-master-logs > /usr/src/all-databases.sql;
```

Version with zipping

```
mysqldump —-all-databases —-single-transaction —-gtid —-master-data=2 —-routines
--events —-flush-logs --compress | gzip > /usr/src/all-databases.sql.gz
```

Performance Test mysqldump (1.7 Million rows in contributions)

```
date; mysqldump --all-databases --single-transaction --gtid --master-data=2 --routines
--events --flush-logs --compress > /usr/src/all-databases.sql; date
```

```
Mi 20. Jan 09:40:44 CET 2021
Mi 20. Jan 09:41:55 CET 2021
```

Seperated sql-structure files and data-txt files including master-data for a specific database

```
 # backups needs to be writeable for mysql
 mkdir /backups
 chmod 777 /backups
 chown mysql:mysql /backups
 mysqldump --tab=/backups contributions
 mysqldump --tab=/backups --master-data=2 contributions
 mysqldump --tab=/backups --master-data=2 contributions > /backups/master-data.tx
```

Backups PIT (Point-In-Time recovery)

Meta - Weg

```
## Es ist wichtig, diese Reihenfolge
1) Spielen ein dump aus:

## --delete-source-logs nur wenn keine master-slave-Replikation
mysqldump --events --routines --flush-logs --source-data=2 --delete-source-logs --all-
databases > /usr/src/all-databases

2) Machen wir Änderungen in der sakral

use sakila
insert into actor (last_name,first_name) values (‚Hans‘,’Metzger’);
insert into actor (last_name,first_name) values (‚Hansi‘,’Metzgerei’);
delete from actor where id > 200;

——

3) Recovery vor Delete
1. Rausfinden wann der Fehler aufgetreten ist.
cd /var/lib/mysql
mysqlbinlog -vv bin-log.000005

2. Einschränken des binlogs und in recovery.sql ausspielen
## bei mehreren binlogs im Zeitraum bitte alle angeben
mysqlbinlog -vv --start-position=156 --stop-position=462 bin-log.000005 >
/usr/src/recovery.sql


3. Vollständigen Dump einspielen
mysql < /usr/src/all-databases.sql

4. recovery.sql einspielen
mysql < /src/src/recovery.sql
```

```
5. Überprüfen, ob die beiden Datensätze wieder da sind .
mysql> use sakila; select * from actor;
```

Backup mit xtrabackup

Installation

```
wget https://repo.percona.com/apt/percona-release_latest.$(lsb_release -sc)_all.deb
## installationsquelle werden in /etc/apt/sources.list.d geschrieben
dpkg -i percona-release_latest.focal_all.deb
## neue installationsquellen einlesen
apt update
apt search xtrabackup
apt install percona-xtrabackup-80
```

Walkthrough

```
## server version differs to xtrabackup
## xtrabackup 8.0.26
## mysql-server 8.0.27

xtrabackup --backup --target-dir /usr/src/backups/ --no-server-version-check
xtrabackup --prepare --target-dir /usr/src/backups/ --no-server-version-check

## altes datenverzeichnis aus dem Weg räumen
cd /var/lib
systemctl stop mysql
mv mysql mysql.bkup

##
xtrabackup --copy-back --target-dir /usr/src/backups/ --datadir=/var/lib/mysql --no-
server-version-check
## adjust permission
chown -R mysql:mysql mysql
chmod -R g=,o= mysql
systemctl start mysql
```

Walkthrough

```
## server version differs to xtrabackup
## xtrabackup 8.0.26
## mysql-server 8.0.27

## important to check if installed
## and eventually install
apt search qpress
apt install qpress

xtrabackup --backup --compress --target-dir /usr/src/backups/ --no-server-version-
check
xtrabackup --decompress --target-dir /usr/src/backups/
```

```
xtrabackup --prepare --target-dir /usr/src/backups/ --no-server-version-check

## altes datenverzeichnis aus dem Weg räumen
cd /var/lib
systemctl stop mysql
mv mysql mysql.bkup

##
xtrabackup --copy-back --target-dir /usr/src/backups/ --datadir=/var/lib/mysql --no-
server-version-check
## adjust permission
chown -R mysql:mysql mysql
chmod -R g=,o= mysql
systemctl start mysql
```

Ref:

- https://www.percona.com/blog/2020/10/23/mysql-new-releases-and-percona-xtrabackup-incompatibilities/

Backup mit xtrabackup mit Verschlüsselung

Walkthrough

```
## use output -> this key as encrypt-key
openssl rand -base64 24
xtrabackup --backup --target-dir=/usr/src/backups-encrypted --encrypt=AES256 --
encrypt-key="yIzl4skb1/Nn/t8g3cuEzpjGoYQQzo9l" --no-server-version-check
xtrabackup --decrypt=AES256 --encrypt-key="yIzl4skb1/Nn/t8g3cuEzpjGoYQQzo9l" --target-
dir=/usr/src/backups-encrypted
xtrabackup --prepare --target-dir=/usr/src/backups-encrypted

##
systemctl stop mysql
cd /var/lib
mv mysql mysql.bkup4
## datadir needs to in config of /etc/mysql/ - folders (in one config with category
[mysqld]
xtrabackup --copy-back --target-dir=/usr/src/backups-encrypted  --no-server-version-
check
cd /var/lib/
chown -R mysql:mysql mysql
chmod -R g=,o= mysql
systemctl start mysql
```

Refs:

- https://www.percona.com/doc/percona-xtrabackup/2.4/backup_scenarios/encrypted_backup.html
- https://www.percona.com/doc/percona-xtrabackup/LATEST/security/pxb-apparmor.html

Backup und Wiederherstellen in neuer Datenbank

```
## using --databases sakila  instead does not work here
mysqldump --events --routines sakila > /usr/src/sakila.sql
```

```
cd /usr/src

## Version 1
## echo "create schema sakilatraining" | mysql

## Version 2 - works also on Windows
mysql -e 'create schema sakilatraining'
mysql sakilatraining < sakila.sql
```

mysqldump mit asynchroner Verschlüsselung

```
## Asynchrones Schlüsselpaar erstellen
openssl req -x509 -nodes -newkey rsa:2048 -keyout mysqldump-key.priv.pem -out
mysqldump-key.pub.pem

## Öffentlichen Schlüssel Verwenden zum Verschlüsseln
mysqldump --routines --events --triggers --all-databases | openssl smime -encrypt -
binary -text -aes256 -out database.sql.enc -outform DER mysqldump-key.pub.pem

## Entschlüsseln
openssl smime -decrypt -in database.sql.enc -binary -inform DEM -inkey mysqldump-
key.priv.pem -out mysql-backup.sql
mysql < mysql-backup.sql
```

mydumper und myloader

- https://github.com/maxbube/mydumper

Sicherheit

Absichern von Server/Client mit ssl

Teil 1: 1-Weg-Sicherheit (nur auf Server validiert)

```
Bei MySQL 8 werden Zertfikate in der Regel bereits erstellt.
Ob ssl funktioniert können wir mit

mysql>show variables like '%HAVE_SSL%';

## Es funktioniert bereits, allerdings mit den automatisch
## erstellten Zertifikaten
```

Herausfinden, ob SSL verwendet wird

```
## auf client auf dem mysql-server
mysql
mysql>status
SSL:            Not in use
Connection:        Localhost via UNIX socket
```

Bitte das nicht verwenden, weil man damit nicht den Common Name setzen kann

```
sudo mysql_ssl_rsa_setup --uid=mysql
```

CA (Certificate Authority) und Server-Key erstellen

```
## On Server - create ca and certificates
mkdir -p /etc/mysql/ssl
cd /etc/mysql/ssl

## create ca.
openssl genrsa 4096 > ca-key.pem

## create ca-certificate
## Common Name: MariaDB CA
openssl req -new -x509 -nodes -days 365 -key ca-key.pem -out ca-cert.pem

## create server-cert
## Common Name: MariaDB Server
## Password: --- leave empty ----
openssl req -newkey rsa:2048 -days 365 -nodes -keyout server-key.pem -out server-
req.pem

## Next process the rsa - key
openssl rsa -in server-key.pem -out server-key.pem

## Now sign the key
openssl x509 -req -in server-req.pem -days 365 -CA ca-cert.pem -CAkey ca-key.pem -
set_serial 01 -out server-cert.pem
```

Zertifikate validieren

```
openssl verify -CAfile ca-cert.pem server-cert.pem
```

Configure Server

```
## create file
## /etc/mysql/mysql.cnf.d/mysqld.cnf
[mysqld]
ssl-ca=/etc/mysql/ssl/ca-cert.pem
ssl-cert=/etc/mysql/ssl/server-cert.pem
ssl-key=/etc/mysql/ssl/server-key.pem
### Set up TLS version here. For example TLS version 1.2 and 1.3 ##
tls_version = TLSv1.2,TLSv1.3

## Set ownership
chown -vR mysql:mysql /etc/mysql/ssl/
```

Restart and check for errors

```
systemctl restart mysql
journalctl -u mysql
```

Externen user auf server einrichten

```
## Einloggen in mysql-client als root

mysql> create user ext@'%' identified by 'P@ssw0rd';
Query OK, 0 rows affected (0.01 sec)

mysql> grant all on sakila.* to ext@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> alter user ext@'%' REQUIRE SSL;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from user where user = 'ext' \G

ssl_type: ANY
ssl_cipher: 0x
x509_issuer: 0x
x509_subject: 0x
```

Test on Client (1. Versuch)

```
## Er verbindet sich per SSL
## Zertifikatsüberprüfung findet nur auf SERVER statt
mysql -uext -p -h<ip-des-servers>
mysql> status
mysql> exit
## Wir probieren es ohne SSL
mysql -uext -p -h<ip-des-servers> --ssl-mode=DISABLED
## Trotz richtigem Passwort
Enter password:
ERROR 1045 (28000): Access denied for user 'ext'@'139.59.215.179' (using password:
YES)
```

Client verpflichten ein eigenes Zertifikat zu haben

```
## auf server als root
mysql>ALTER USER ext@'%' REQUIRE X509
```

On Client - fails because of missing client certificate

```
mysql -uext -p -h159.223.23.99
Enter password:
ERROR 1045 (28000): Access denied for user 'ext'@'139.59.215.179' (using password:
YES)
```

Teil 2: 2-Weg-Sicherheit (auf Server und Client validiert)

Client - Zertifikate auf Server erstellen

- Wir verwenden die gleiche CA wie beim Server

```
## auf dem Server
cd /etc/mysql/ssl
## Bitte Common-Name: MariaDB Client
openssl req -newkey rsa:2048 -days 365 -nodes -keyout client-key.pem -out client-
req.pem

## process RSA - Key
openssl rsa -in client-key.pem -out client-key.pem

## sign certficate with CA
openssl x509 -req -in client-req.pem -days 365 -CA ca-cert.pem -CAkey ca-key.pem -
set_serial 01 -out client-cert.pem
```

Client - Zertifikate validieren

```
openssl verify -CAfile ca-cert.pem client-cert.pem
```

Zertifikate für Client zusammenpacken

```
mkdir cl-certs; cp -a client* cl-certs; cp -a ca-cert.pem cl-certs ; tar cvfz cl-
certs.tar.gz cl-certs
```

Zertifikate auf Client transferieren

```
scp cl-certs.tar.gz 11trainingdo@<ip-des-clients>:/tmp
```

Zertifikate einrichten

```
## auf client
mv /tmp/cl-certs.tar.gz /etc/mysql/
cd /etc/mysql; tar xzvf cl-certs.tar.gz

cd /etc/mysql/cl-certs
ls -la


cd /etc/mysql/conf.d
vi mysql.cnf
[mysql]
ssl-ca=/etc/mysql/cl-certs/ca-cert.pem
ssl-cert=/etc/mysql/cl-certs/client-cert.pem
ssl-key=/etc/mysql/cl-certs/client-key.pem
```

Zertifikate testen

```
## Auf Server überprüfen dass X509 für user eingestellt ist
select user,ssl_type from mysql.user where user='ext'

## Auf Client zum server connecten
## Sollte die Verbindung nicht klappen stimmt auf dem
## Client etwas mit der Einrichtung nicht
```

```
mysql -uext -p -h<ip-des-mysql-servers>
mysql> status
```

Ref

- https://dev.mysql.com/doc/refman/8.0/en/alter-user.html```

Verschlüsselte Backups mit xtrabackup

Walkthrough

```
## use output -> this key as encrypt-key
openssl rand -base64 24
xtrabackup --backup --target-dir=/usr/src/backups-encrypted --encrypt=AES256 --
encrypt-key="yIzl4skb1/Nn/t8g3cuEzpjGoYQQzo9l" --no-server-version-check
xtrabackup --decrypt=AES256 --encrypt-key="yIzl4skb1/Nn/t8g3cuEzpjGoYQQzo9l" --target-
dir=/usr/src/backups-encrypted
xtrabackup --prepare --target-dir=/usr/src/backups-encrypted

##
systemctl stop mysql
cd /var/lib
mv mysql mysql.bkup4
## datadir needs to in config of /etc/mysql/ - folders (in one config with category
[mysqld]
xtrabackup --copy-back --target-dir=/usr/src/backups-encrypted  --no-server-version-
check
cd /var/lib/
chown -R mysql:mysql mysql
chmod -R g=,o= mysql
systemctl start mysql
```

Refs:

- https://www.percona.com/doc/percona-xtrabackup/2.4/backup_scenarios/encrypted_backup.html
- https://www.percona.com/doc/percona-xtrabackup/LATEST/security/pxb-apparmor.html

Prüfen ob socket verwendet, bei lokalem System

Voraussetzung

```
Linux - System
Applikation und Datenbank-Server sind auf gleichen Virtuellen bzw. Physischen Server
```

Testfolge

```
lsof -i
localhost:mysql
```

mysql_secure_installation - validate plugin aktivieren

Sicherstellen, dass die Komponente Validate als Passwort-Mechanismus aktiviert wird

```
mysql_secure_installation

und keine root-benutzer von extern erlauben

mysql>
select * from mysql.user where user='root' and host != 'localhost'
    -> ;
Empty set (0.00 sec)
```

general log deaktivieren

Warum ?

- Wird sehr schnell, sehr groß
- Schlecht für die Performance

Überprüfung ?

```
select @@general_log
## sollte auf 0 stehen
show variables like 'general_log'
OFF
```

plugin vs. components

Components

- Abgeschlossene Einheiten
- MySQL-Server ist ein Komponente
- Eine weitere Komponenten kann geschrieben werden.
- Diese kommunziert über einen Service mit der anderen Komponenten

Plugins

- Server, stellt eine API / bzw. verschiedene bereit
- Auf dieses greift das Plugin dann zu
- Alles innerhalb der Komponenten MySQL-Server
- Oftmals schlecht implementiert
    - Eigentlich respektiv auf bestimmte apis
    - In der Realität, scope ist oft auf alle api
- Plugin kann alles auslesen

Vorteile von Komponentens

- Keine Endung mehr beim Laden notwendig

User Passwort-Länge und Passwort-Ablauf

Passwort-Ablauf pro User setzten

```
## Passwort läuft nach 60 Tagen ab und muss neu gesetzt werden
ALTER USER training@localhost PASSWORD EXPIRE INTERVAL 60 day;
```

- https://dev.mysql.com/doc/refman/8.0/en/password-management.html

Läßt sich eine Passwort - Länge pro User festlegen ?

```
Nein.
Nur auf Server-Ebene für alle Benutzer möglich (über Validation Komponente)
```

- https://dev.mysql.com/doc/refman/8.0/en/validate-password.html

Trigger ohne Super-Rechte anlegen

Warum ist es so ?

- Trigger können nicht auf DETERMINISTIC gesetzt werden.
- Wenn ein Trigger nicht-deterministisch ist, kann es zu Problemen kommen
- In diesem Fall kann es beim BINLOG_FORMAT=STATEMENT zu Problemen beim Slave kommen

Test auf MySQL 8.0.27 BINLOG_FORMAT = ROW

```
## User training@localhost mit folgenden Rechten
show grants;
+-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-------------------------------------------------------------+
| Grants for training@localhost
|
+-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-------------------------------------------------------------+
| GRANT USAGE ON *.* TO `training`@`localhost`
|
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, REFERENCES, INDEX, ALTER, CREATE
TEMPORARY TABLES, LOCK TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER
ROUTINE, EVENT, TRIGGER ON `sakila`.* TO `training`@`localhost` |
+-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-------------------------------------------------------------+
2 rows in set (0.01 sec)

## Test - Case
use sakila;

## aus bug-report
mysql> CREATE TABLE t1 ( a int );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TRIGGER g1 BEFORE INSERT ON t1 FOR EACH ROW SET new.a=new.a+1;
ERROR 1419 (HY000): You do not have the SUPER privilege and binary logging is enabled
(you *might* want to use the less safe log_bin_trust_function_creators variable)
```

(Dirty-)Fix

```
## Either set it in the config or as SUPER-privileges user:
/etc/mysql/mysql.conf.d/mysqld.cnf
log-bin-trust-function-creators = 1
systemctl restart mysql

## now login as unprivileged (NON SUPERUSER PERMS) and try again
## on localhost
mysql -utraining -p
use sakila
mysql> CREATE TABLE if not exists t1 ( a int );
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> DROP TRIGGER IF EXISTS g1;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> CREATE TRIGGER g1 BEFORE INSERT ON t1 FOR EACH ROW SET new.a=new.a+1;
Query OK, 0 rows affected (0.00 sec)
```

Refs:

- https://bugs.mysql.com/bug.php?id=39489

Sicherheit (CIS)

1.1 Place Databases on Non-System Partitions

Überprüfen, wo das datadir liegt

```
mysql> select @@datadir;
+----------------------+
| @@datadir            |
+----------------------+
| /var/lib/mysql-data/ |
+----------------------+
1 row in set (0.00 sec)

mysql> show variables like 'datadir';
+---------------+----------------------+
| Variable_name | Value                |
+---------------+----------------------+
| datadir       | /var/lib/mysql-data/ |
+---------------+----------------------+
1 row in set (0.01 sec)
```

Walkthrough

```
## /etc/apparmod.d/
vi usr.sbin.mysqld
## ---> change these lines
## Allow data dir access
##  /var/lib/mysql/ r,
##  /var/lib/mysql/** rwk,
```

```
  /var/lib/mysql-data/ r,
  /var/lib/mysql-data/** rwk,
### <-----

systemctl stop mysql
systemctl restart apparmor
systemctl status apparmor
aa-status

## Change config of mysql
## datadir
cd /etc/mysql/mysql.conf.d/
vi mysqld.cnf
## change datadir to /var/lib/mysql-data # on seperate partition
datadir=/var/lib/mysql-data

cd /var/lib
cp -a mysql mysql-data
systemctl restart mysql

## Bei Erfolg ist das Datadir jetzt geändert
mysql>
show variables like 'datadir';
```

Debuggen bei Problemen

```
journalctl -u mysql -e
/var/log/mysql/error_log
```

systemctl stop mysql

1.2 Use dedicated Least Privileged Account

Check

```
## simple
ps aux | grep mysql

## sophistiscated
ps aux | head -n 1 && ps aux | grep mysql | grep -v grep
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
mysql      18341  0.5 42.4 1842172 426204 ?      Ssl  14:22   0:01 /usr/sbin/mysqld
```

Tools

Testdatenbank Sakila installieren

```
cd /usr/src
wget https://downloads.mysql.com/docs/sakila-db.tar.gz
```

```
tar xvf sakila-db.tar.gz
cd sakila-db/
ls -la
mysql < sakila-schema.sql
mysql < sakila-data.sql
```

Authentifizierung / User-Management

Für User altes Password-Verfahren mysql_native_password verwenden in MySQL 8

```
create user scanner@localhost identified with mysql_native_password by 'Passw0rd';
```

Rollen

Konzept

- Rolen

Walkthrough

```
## Rolle anlegen
mysql> CREATE ROLE sakiladb
## Berechtigungen der Rolle zuordnen / alle Berechtigungen für sakila
mysql> GRANT ALL ON sakila.db TO sakiladb
## Nutzer anlegen
mysql> CREATE USER roleuser@localhost identified by 'P@ssw0rd';

## Die Rolle dem Nutzer zugeordnet
mysql> GRANT sakiladb TO rolesuser@localhost

## Die Standardrolle festlegen, wenn er sich einloggt
SET DEFAULT ROLE ALL TO roleuser@localhost
```

Abgekürzte From

```
create user roleuser2@localhost identified by 'P@ssw0rd' DEFAULT ROLE sakiladb;
```

Ref

- https://www.mysqltutorial.org/mysql-roles/

Upgrade

Upgrade von MySQL 5.7 -> 8

Walkthrough (Teil 1)

```
## Download repo - apt install package on server
cd /usr/src
wget https://dev.mysql.com/downloads/repo/apt/
-> mysql-apt.....
```

```
dpkg -i  mysql-apt-config_0.8.20-1_all.deb
## all settings can be like, then select OK

## Repostände lokal updaten
apt update

apt install mysql-shell

mysqlsh>
JS \c root@localhost

### Probleme mit socket evtl durch unterschiedliche Paketherkünfte (MySQL 5.7 -> MySQL
8)
MySQL 5.7. kam von Ubuntu und verwendete einen anderen Socket
MySQL 8 verwendet den socket /var/lib/mysql/mysql.sock
```

Anpassen des Socket in Server und Client - Teil 2

```
cd /etc/mysql/mysql.conf.d/mysqld.cnf
## socket geändert in
[mysqld]
socket = /var/lib/mysql/mysql.sock

## Server neu starten und prüfen ob sockt da ist im Verzeichnis
systemctl restart mysql
/var/lib/mysql/

## socket geändert für client - config
cd /etc/mysql/conf.d/mysql.cnf

## mysql.conf
[mysql]
socket = /var/lib/mysql/mysql.sock

## Achtung auch für mysqldump setzten z.B. in Datei
## client.cnf - neu erstellen
[client]
socket = /var/lib/mysql/mysql.sock
```

Test Script für MySQL Migration aufgerufen

```
mysqlsh
## Vorne connection, hinten Options
JS> util.checkForServerUpgrade('root@localhost',{"configPath":"/etc/mysql/my.cnf"})

## Dann Ausgabe sieht ungefähr so aus.
The MySQL server at localhost:33060, version 5.7.31-log - MySQL Community

Server (GPL), will now be checked for compatibility issues for upgrade to MySQL

8.0.21...
```

```
1) Usage of old temporal type

  No issues found

......
```

Ausgabe überprüft, was muss evtl geändert, berücksichtigt werden.

```
z.B. AUTO_CREATE_NO_USER - unkritisch, da in MySQL 8 per default der Fall ist.
(Es wird bei grants keine user angelegt, wenn diese nicht existieren)
```

Sicherung der Datenbank VOR !! Update

```
mysqldump --all-databases --routines --events > /usr/src/all-database.sql
## Wenn 0 ausgabe, dann ist das Script erfolgreich durchgelaufen
echo $?
## Zur Sicherheit noch letzte in Dump anschauen
## Hier muss Dump completed stehen
```

Server stoppen und deinstallieren

```
systemctl stop mysql
apt remove mysql-server-5.7
## alte Abhängigkeiten, die nicht mehr benötigt werden, werden gelöscht
apt autoremove
```

Neuen Server installieren und Fehler bereinigen

```
## mysql-server ist in der Regel die neueste, zur Sicherheit nochmal checken
## mit apt search mysql-server
apt install mysql-server

## mysqld.conf behalten !!

## Wenn server nicht starten Fehler bereinigen
## Analysieren
/var/log/mysql/error.log

## mysqld.conf entsprechend anpassen

## danach start probieren, so lange bis es geht !! (fehlerbereinigung -> starten ->
fehlerber....)
systemctl start mysqld

##### Upgrade erfolgt beim Starten in Place sowohl in Installationspackage als auch
tar.gz
```

Refs:

- https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-utilities-upgrade.html

Windows

Welchen Benutzer für den Service verwenden?

LocalSystem is nicht gut !!

```
Zu viele Rechte
```

Optimal wäre: LocalService

```
## geht nur auf, wenn applikation und DB-Server auf gleichem Host
mysqld --install --local-service

##
mysqld --remove "ServiceNamen"
```

2. Alternative: NetworkService

```
Frage: Nimmt der installer diesen beider Installatio
```

Refs:

- https://www.netikus.net/documents/MySQLServerInstallation/index.html?moresecurity.htm

Documentation

Server System Variables - Reference

- https://dev.mysql.com/doc/refman/8.0/en/server-system-variable-reference.html

MySQL Performance Dokument - en

- https://schulung.t3isp.de/documents/pdfs/mysql/mysql-performance.pdf