

22.15.0 Exkurs: MySQL-Datenbankserver – Installation, Konfiguration und Test

Wenn Sie MySQL-Datenbanken auf Ihrem System bearbeiten wollen, dann benötigen Sie notwendigerweise einen MySQL-Datenbankserver. MySQL ist ein bekanntes und verbreitetes Datenbanksystem für Client-Server-Architekturen.

Inhaltsverzeichnis

22.15.0 Exkurs: MySQL-Datenbankserver – Installation, Konfiguration und Test.....	1
22.15.0.1 Hinweise.....	1
22.15.0.2 Installation des MySQL-Datenbankservers.....	1
22.15.0.3 Passwort für den DB-Benutzer (Administrator) root festlegen.....	1
22.15.0.4 Absicherung der MySQL-Server-Installation.....	2
22.15.0.5 Dokumentation.....	3
22.15.0.6 Log-Dateien.....	3
22.15.0.7 Steuerung des MySQL-Datenbankservers.....	3
22.15.0.8 Fernzugriff auf den MySQL-Server.....	3
22.15.0.9 Übung.....	4
22.15.0.10 De-Installation des MySQL-Datenbankservers.....	5

22.15.0.1 Hinweise

- In der vorliegenden Beschreibung werden die Installation, Konfiguration und ein Vorschlag für einen ersten Test des MySQL-Datenbankservers für Linux Mint 20.2 beschrieben.
- Es wird die Version 8.0.30 des Datenbankservers MySQL eingesetzt.
- Auf die Dokumentation zum MySQL-Datenbankserver wird nur verwiesen.
- Für viele administrative Aufgaben und Tests wird in einer Konsole der MySQL-Client `mysql` eingesetzt.
- MariaDB wurde als "Drop-in"-Ersatz für MySQL entwickelt. Somit sind beide Softwarepakete funktional gleichwertig und austauschbar.

22.15.0.2 Installation des MySQL-Datenbankservers

So können Sie feststellen, welche MySQL-Pakete bereits auf Ihrem System installiert sind. So könnte eine Ausgabe aussehen:

```
hans@mint20:~$ dpkg --get-architecture | grep mysql
ii gmbas3-gb-db-mysql 3.17.3+git202207211120... amd64 MySQL driver for the Gmbas database
ii gmbas3-gb-mysql 3.17.3+git202207211120... amd64 Gmbas MySQL component
ii libmysqlclient21:amd64 8.0.30-0ubuntu0.20.04.2 amd64 MySQL database client library
ii mysql-common 5.8+1.0.5ubuntu2 all MySQL database common files, e.g. /etc/mysql/my.cnf
```

Der MySQL-Server kann über die Anwendungsverwaltung installiert werden oder Sie geben in einem Terminal folgende Zeilen ein:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install mysql-server
```

Gut zu wissen:

Während der Installation wird automatisch auch der MySQL-Konsolen-Client `mysql` mit installiert, obwohl dieser bei Linux Mint im Software-Manager als "nicht installiert" angezeigt wird.

Beachten Sie, dass der MySQL-Server zum Abschluss der Installation bereits gestartet wurde, was Sie mit dem folgenden Befehl in einer Konsole überprüfen können:

```
$ systemctl status mysql | grep 'Status'
Status: "Server is operational"
```

22.15.0.3 Passwort für den DB-Benutzer (Administrator) root festlegen

Während der Installation des MySQL-Datenbankservers (Version 8) wird automatisch ein Konto für den MySQL-Administrator `root@localhost` angelegt – aber kein MySQL-Administrator-Passwort erfragt.

Im folgenden Abschnitt wird dem Administrator-Konto ein Passwort hinzugefügt und die Authentifizierungsmethode geändert, damit sich ein Datenbank-Benutzer auch mit dem MySQL-Client in einer Konsole beim MySQL-Server anmelden darf.

Starten Sie in einer Konsole den MySQL-Client `mysql`. Geben Sie den folgenden Befehl ein und anschließend Ihr System-Passwort:

```
$ sudo mysql
```

Sie müssen jetzt für den MySQL-Administrator `root@localhost` das Passwort festlegen und das Authentifizierungs-Plugin von MySQL vom Standard-Plugin `auth_socket` (mindestens) auf das Plugin `mysql_native_password` umstellen:

```
$ mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '<root-password>';  
$ mysql> FLUSH PRIVILEGES; -- Berechtigungstabellen neu laden und Änderungen übernehmen
```

Das traditionelle Plugin `mysql_native_password` zur Authentifizierung ist zwar nicht sehr sicher, da es nur einen Hash des Passworts verwendet, ist dafür aber mit älteren DB-Treibern kompatibel. Alternativ wechseln Sie später auf ein anderes Plugin, die Sie unter dem Link: <https://dev.mysql.com/doc/refman/8.0/en/authentication-plugins.html> erkunden können.

Prüfen Sie nun, ob auf das Authentifizierungs-Plugin `mysql_native_password` umgestellt wurde:

```
$ mysql> SELECT user, host, plugin FROM mysql.user WHERE user='root';  
+-----+-----+-----+  
| user | host      | plugin                |  
+-----+-----+-----+  
| root | localhost | mysql_native_password |  
+-----+-----+-----+  
1 row in set (0,00 sec)  
  
mysql> quit  
Bye  
$
```

Sie können sich nun mit Ihrem DB-Konto (`root@localhost`, `<root-password>`) mit dem MySQL-Client-Programm `mysql` lokal am MySQL-Server anmelden, was mit `$ sudo mysql` nicht mehr möglich ist!

Die folgenden fünf Befehle zum Verbinden von MySQL-Client und MySQL-Server sind gleichwertig:

```
$ mysql -u root -p 'Standard-Host ist `localhost` und Standard-Port `3306`'  
$ mysql -h localhost --user=root root -p  
$ mysql -h localhost --port=3306 -u root -p  
$ mysql --host=<hostname> --port=3306 -u root -p  
$ mysql --host=<hostname> --port=3306 --user=root --password
```

So können Sie sich zum Beispiel die Version des installierten MySQL-Datenbankservers anzeigen lassen:

```
$ mysql -h localhost -u root -p  
mysql> SELECT VERSION();  
+-----+  
| VERSION() |  
+-----+  
| 8.0.30-0ubuntu0.20.04.2 |  
+-----+  
1 row in set (0,00 sec)  
mysql> quit  
Bye  
$
```

22.15.0.4 Absicherung der MySQL-Server-Installation

Zur Absicherung der MySQL-Server-Installation wird das installierte Sicherheitsskript aufgerufen:

```
$ sudo mysql_secure_installation
```

- Geben Sie das Root-Passwort ein, das Sie zuvor über das `ALTER_USER`-Kommando festgelegt hatten.
- Im ersten Schritt wird einmalig nachgefragt, ob Sie die 'VALIDATE PASSWORD COMPONENT'

verwenden wollen. Wenn Sie an dieser Stelle mit 'y' antworten – was empfohlen wird – dann wählen Sie direkt danach die gewünschte 'Password Policy' oder Passwort-Richtlinie aus und stellen Sie sich auf ein entsprechend starkes Passwort ein. In diesem Exkurs wird `1 = Medium` gewählt.

- Da die Passwort-Richtlinie auf `Medium` eingestellt wurde, muss ein gültiges DB-Passwort (für alle DB-Benutzer) daher dieser Richtlinie entsprechen:
 - Das Passwort muss mindestens 8 Zeichen lang sein.
 - Das Passwort hat mindestens 1 Kleinbuchstaben und mindestens 1 Großbuchstaben.
 - Die Anzahl der Ziffern ist mindestens 1.
 - Die Mindestanzahl an Sonderzeichen beträgt 1.
- Im Dialog wird die Stärke des verwendeten Passworts ermittelt und angezeigt.
- Bei einem Wert unter 100 beantworten Sie die Frage, ob Sie das Passwort ändern wollen mit `y` und geben ein starkes Passwort wie **adminPW9#** ein, das der Passwort-Richtlinie `Medium` entspricht. Anderenfalls können Sie das existierende starke Passwort mit `n` übernehmen.
- Ansonsten sollten Sie bei allen weiteren Nachfragen im Dialog 'y' eingeben, um sichere Standardeinstellungen des MySQL-Servers festzulegen.

22.15.0.5 Dokumentation

Eine umfangreiche Dokumentation zum MySQL-Datenbankserver finden Sie unter:

LINK1: <https://dev.mysql.com/doc/refman/8.0/en/>

22.15.0.6 Log-Dateien

Die letzten Einträge der *aktuellen* Fehler-Log-Datei des MySQL-Datenbankserver können Sie sich mit dem ersten Befehl in einer Konsole ansehen; den kompletten Inhalt mit dem zweiten Befehl:

```
$ tail -f /var/log/mysql/error.log
$ xed /var/log/mysql/error.log
```

22.15.0.7 Steuerung des MySQL-Datenbankservers

Achtung! Der MySQL-Datenbankserver wird bei jedem Systemstart *automatisch* gestartet (Standard). Das können Sie mit diesen beiden Befehlen ändern:

```
$ sudo systemctl disable mysql      -- MySQL aus der Autostart-Liste entfernen
$ sudo systemctl enable mysql      -- MySQL der Autostart-Liste wieder hinzufügen
```

Wenn der MySQL-Datenbankserver beim Systemstart nicht gestattet wird, dann können Sie ihn selbst starten.

Die folgenden Aufrufe in einem Terminal steuern den MySQL-Datenbankserver mit den in der Liste aufgeführten Parametern:

```
$ sudo systemctl parameter mysql {start|stop|restart|reload|force-reload|status}  oder
$ sudo service mysql parameter {start|stop|restart|reload|force-reload|status}
```

Beispiele:

```
$ sudo systemctl start mysql
$ sudo service mysql stop
$ systemctl status mysql      ' Aufruf ohne erhöhte Berechtigungen nur bei der Statusabfrage
```

22.15.0.8 Fernzugriff auf den MySQL-Server

Der MySQL-Server ist so (vor-)konfiguriert, dass sich der DB-Administrator nur lokal am MySQL-Server anmelden darf. Diese Einschränkung wird auch durch ein geändertes Binding nicht aufgehoben, denn es wird durch den Kontotyp – hier root@localhost – festgelegt!

Damit sich weitere DB-Benutzer von einem beliebigen Host zum MySQL-Server verbinden dürfen, muss

- (1) mit erhöhten Berechtigungen ein entsprechendes Binding in die (Haupt-)Konfigurationsdatei (ab Ubuntu 16) `/etc/mysql/mysql.conf.d/mysqld.cnf` eingetragen werden. Dafür muss bei der Variablen 'bind-address' die IP-Adresse 127.0.0.1 durch die IP-Adresse ersetzt werden, unter der der MySQL-Server erreichbar ist. Wenn die Zeile `bind-address = ::` lautet, dann akzeptiert der MySQL-Server TCP/IP-Verbindungen über alle IPv4- und IPv6-Schnittstellen des Server-Hosts. Wenn dagegen `bind-address = 192.168.100.12` angegeben wird, dann lauscht der MySQL-Server nur auf dieser IP-Adresse. Nachdem Sie die Bindungsadresse geändert haben, vergessen Sie nicht, Ihren MySQL-Server neu zu starten!
- (2) im DB-Konto-Namen eines DB-Benutzers die IP-Adresse(n) angegeben werden, von der er auf den MySQL-Server zugreifen darf. Hier einige Beispiele:

<code>user_name@localhost</code>	Zugriff nur von localhost,
<code>user_name@192.168.0.3</code>	Zugriff nur von 192.168.0.3,
<code>user_name@192.168.0.%</code>	Zugriff aus dem IP-Bereich 192.168.0.1 bis 192.168.0.254,
<code>user_name@%</code>	Zugriff von jedem Host.

So stellen Sie fest, an welchen IP-Adressen der MySQL-Server lauscht oder gebunden ist:

```
$ sudo netstat -tupan | grep mysql
[sudo] Passwort für hans:
Proto Recv-Q Send-Q Local Address           Foreign Address         State                   PID/Program name
-----
tcp        0      0 127.0.0.1:33060        0.0.0.0:*               LISTEN                  8678/mysqld
tcp        0      0 127.0.0.1:37926        127.0.0.1:3306         VERBUNDEN               9478/mysql
tcp6       0      0 :::3306                :::*                     LISTEN                  8678/mysqld
tcp6       0      0 127.0.0.1:3306        127.0.0.1:37926        VERBUNDEN               8678/mysqld
$
```

Diese oberen Ausgaben korrespondieren mit der Ausgabe des folgenden Befehls im MySQL-Client:

```
$ mysql -h localhost -u root -p
mysql> SHOW variables WHERE variable_name LIKE 'bind%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| bind_address  | *     |
+-----+-----+
1 row in set (0,00 sec)
mysql>
```

Detaillierte Informationen zum Binding finden Sie unter https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_bind_address.

22.15.0.9 Übung

Die Übung – ausgeführt in einer Konsole mit dem MySQL-Monitor ``psql`` (Client) – besteht darin, dass Sie wichtige Befehle noch einmal wiederholen und dabei

- zuerst den Status des MySQL-Datenbankservers ermitteln,
- darauf folgend den Server (auf dem Standard-Port 3306) starten, wenn das notwendig ist,
- als Nächstes die Konfiguration des MySQL-Servers neu einlesen,
- danach den MySQL-Client `mysql` starten und sich als DB-Benutzer ``root`` einloggen,
- anschließend herausfinden, welche Datenbanken existieren,
- dann die MySQL-Hilfe aufrufen und abfragen, wie eine neue Datenbank angelegt wird
- und abschließend den MySQL-Client mit dem Kommando `quit` oder `exit` oder `\q` beenden.

```
$ systemctl status mysql
$ sudo systemctl start mysql -- Option
$ sudo service mysql force-reload
$ mysql -h localhost --port=3306 -u root -p
mysql> SHOW DATABASES; -- Achten Sie hier auf das Semikolon
mysql> HELP -- Jede folgende Zeile einzeln abarbeiten
mysql> HELP CREATE DATABASE
mysql> quit
Bye
```

Wenn Sie eine bestimmte Hilfe aufrufen – wie 'HELP CREATE DATABASE' – dann wird Ihnen auch hinter 'URL:' ein passender Link wie <https://dev.mysql.com/doc/refman/8.0/en/create-database.html> angezeigt, den Sie im *Kontext-Menü* mit der rechten Maustaste sofort interaktiv öffnen können!

22.15.0.10 De-Installation des MySQL-Datenbankservers

Für den Fall einer De-Installation des Datenbank-Servers ist es immer eine gute Idee, wenn Sie von den bisher verwendeten Datenbanken eine komplette Sicherung von Struktur und Daten anlegen oder von ausgewählten Datenbanken:

```
$ mysqldump --comments --dump-date --no-tablespaces --user root --password --all-databases > all-databases.sql  
$ mysqldump --comments --dump-date --no-tablespaces --user root --password --databases DB1Name DB2Name > db1db2.sql
```

Wenn Sie sich von MySQL – hier in der Version 8 – trennen wollen, dann de-installieren Sie bitte nur den MySQL-Server und den MySQL-Client; entweder über die Synaptic-Paketverwaltung oder in einer Konsole mit folgendem Befehl:

```
$ sudo apt-get remove --purge mysql-client mysql-client-8.0 mysql-client-core-8.0 mysql-server mysql-ser-  
ver-8.0 mysql-server-core-8.0
```

Anschließend können Sie die folgenden Verzeichnisse löschen:

```
$ sudo rm -rf /var/lib/mysql* /var/log/mysql
```

Hinweis

```
└─ mysql  
   └─ conf.d  
      ├── mysql.cnf  
      └── mysqldump.cnf  
  └─ my.cnf -> /etc/alternatives/my.cnf  
  └─ my.cnf.fallback
```

Diese Ordner und Verzeichnisse existieren in der Regel schon, bevor MySQL installiert wurde und sollten daher nicht gelöscht werden!