

DAT151

Database and System Administration



Spring 2018

Assignment 0

*No deadline, but has to be done before obligatory assignment 1.
No report to hand in this time.*

Last changed, 08.01.2018 (BK)

Installing Linux

Before installation:

Read lecture notes about installation before starting this task.

Choose a machine in the lab and use the same machine for the entire course.

It is possible to use your own PC for this course, but some of the later tasks require access to two computers in the same local network. One task requires access to a tape station.

Install CentOS 7 in the lab. You can choose other Linux distributions on your own computer if you like, but the lecturers are best able to help you if you chose CentOS.

Don't use all the free space on the disk for your installation – 40-50 GB is plenty! (The disk might be used for other things as well...)

All the machines in the lab needs dual boot, running both MS Windows and Linux. Don't delete any of the MS Windows installations during your Linux installation.

Installation:

Some important points (read all before you start installing):

- The machines in the lab supports EFI-boot, but we will use Legacy boot to avoid problems with EFI and dual boot.
 - **Press F9 during booting => Blue boot menu**
 - **Choose SATA3 if CD, or your USB disk**
- Partition the disk manually to make sure that you get the right structure on the disk. Select standard partitions. Do not use LVM!
 - INSTALLATION DESTINATION
 - “I will configure partitioning”
 - “New mount points...” --> “Standard partition”

- You should at least make the following partitions:
 - / - root system
 - /home - home folders (user files)
 - swap - swap partition, 1-2 times physical memory
 - At least 5 empty partitions, size 2GB in each partition – to be used in the database part of the course.
- Do not delete any Windows partitions.
- Choose the “Standard desktop” installation, select extra packages if you like.
- Install KDE or Gnome desktops (or both). Choose your favourite! If you have no favourite, use Gnome.
- In the lab, you should all use the root password given during the lectures.
- Select a Norwegian keyboard in the lab.
- Use static IP addresses in the lab. Use the IP addresses printed on the machines.
 - NETWORK & HOSTNAME
 - Enable Ethernet
 - choose 'Configure'
 - select 'General'
 - enable 'Automatically connect to this network when available'
 - Configure IPv4
- Other IP addresses to be used in the lab:
 - **Name server:** 10.0.0.20
 - **Mask:** 255.255.254.0
 - **Gateway:** 10.0.0.1

After installation, you can update all programs in the system. The easiest way to do this is by using [yum](#). Use man pages for yum (`man yum`) to find out how to do it.

Updating the boot loader, GRUB:

To be able to boot the Windows partition on your computer, you need to add a Windows entry to the GRUB configuration. CentOS 7 uses GRUB 2, i.e. configuration in `/etc/grub.d/`.

Follow the description given in the lecture notes. If you are using a system running GRUB Legacy, you find a description of this in the book.

Installing a DBMS

We will install MariaDB 5.5 that is shipped with CentOS 7. MariaDB is a community-developed fork of the MySQL database source code. For all practical purposes, MariaDB is a binary drop in replacement of the same MySQL version. Major Linux distros are switching from MySQL to MariaDB. Also major Web sites like Wikipedia and Google are doing the switch. The DAT151 lab exercises should work equally well on MySQL and MariaDB. MariaDB can be installed with *yum* from the CentOS repositories. MySQL can be downloaded from MySQL.

What should we install? A MariaDB-server will need as a minimum the packages **mariadb** and **mariadb-server**.

The CentOS repository also provide several connectors, but we will probably not need any of them in DAT151. For JDBC we can use **mysql-connector-java**. Also **mysql-connector-odbc** can be useful. The package **libdbi-dbd-mysql** contains a C connector.

For administration purposes, the GUI tool MySQL Workbench can be useful. The tool can be downloaded from MySQL. Observe that MySQL Workbench does not support Aria-specific features directly, but we can circumvent this by writing custom SQL code. (Aria is a MariaDB replacement of the MySQL engine MyISAM.)

The installation has to be done as root. Use *yum* and install the packages **mariadb** and **mariadb-server**.

Data base configuration

After the installation is finished, the MariaDB server should be started and we must configure CentOS to start MariaDB at boot.

As **root**, check if MariaDB is running and set to be started at boot:

```
[user@computer]$ systemctl status mariadb
```

From the output, the line starting with "Loaded" tells us whether the MariaDB server will be started at boot. The line starting with "Active" tells us if the server is currently running.

We enable the MariaDB server at boot by using:

```
[user@computer]$ systemctl enable mariadb
```

If the server is not running, we start it with:

```
[user@computer]$ systemctl start mariadb
```

A fresh MariaDB install will have root accounts accessible from the outside without password. The next tasks therefore is to protect our server:

- Remove all DBMS root accounts except **root@localhost**.
- Select a password for the DBMS user **root@localhost**.
- Remove all anonymous DBMS user accounts.
- Remove the database *test* which everyone can access.
- Remove the grants which allow everyone access to databases with names starting with *test_*.

All the above tasks can be achieved by issuing commands to the DBMS as the DBMS user **root**, but an easier solution is to use the script *mysql_secure_installation* which is found in the RPM package **mariadb-server**. The script produces an error message which can safely be ignored:

```
/bin/mysql_secure_installation: line 379: find_mysql_client: command not found
```

Check also the MariaDB documentation (<https://mariadb.com/kb/en/mariadb/documentation/>). The *Getting Started* documents explain how to install, set up and use MariaDB.

Next, let us create a database **privBase**, add a normal DBMS user and give the new DBMS user access to **privBase**. We connect to the MariaDB server as root from a terminal by:

```
[user@computer]$ mysql -u root -p
```

Giving a DBMS user access to a database is done with the **GRANT** statement. You will need the following SQL commands:

- **CREATE DATABASE** – See the MariaDB Documentation > SQL Commands > Data Definition > CREATE > CREATE DATABASE
- **CREATE USER** – See the MariaDB Documentation > SQL Commands Account Management > SQL Commands > CREATE USER
- **GRANT** – See the MariaDB Documentation > SQL Commands > Account Management > SQL Commands > GRANT

Read about the **my.cnf** configuration file in the in the MariaDB Documentation:

- **my.cnf** - See the MariaDB Documentation > Getting Started > Getting, Installing, and Upgrading MariaDB > Configuring MariaDB with my.cnf
- options to **my.cnf** – See the MariaDB Documentation > Getting Started Getting, Installing, and Upgrading MariaDB > Starting and Stopping MariaDB > mysqld Options

Use the documentation and find the purpose of the following options:

```
[client]
character-set-client=utf8

[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
log-bin=mysql-bin
log-bin-trust-function-creators
log=mysql.log

character-set-server=utf8
collation-server=utf8_danish_ci

default-storage-engine=MyISAM

max_connections=200

sql-mode = NO_AUTO_CREATE_USER

[mysqld_safe]
log-error=/var/log/mysql-error.log
pid-file=/var/run/mysqld/mysqld.pid
```

We will do more configuration later, but the above configuration is enough for now. Do not yet use any other table type then MyISAM. Some table type configurations can be specified only as long as there are no tables of this type.

More configuration

The computers in the network lab (that is E425) should have access to the MariaDB server. Check the use of the command **firewall-cmd** and open access through the firewall to the MariaDB server from all computers in the lab. Then test that you can access your MariaDB server from other lab computer.

The last task can be a little bit complicated if you have not worked with firewalld before. firewalld is a dynamic firewall system and used by default on CentOS 7.

firewalld divide the network into zones. Different zones can have different firewall settings. Examples of predefined zones are **public** for a computer connected directly to the internet, and **home** and **work** for computers behind a firewall.

We can ask firewalld to use a specific zone for some source addresses (IP numbers) or for specific network cards. If no zone is chosen, firewalld will use a default zone.

The system comes with a list of predefined zones that we can adapt for our use, but we can also create our own.

List all zones by:

```
[user@computer]$ firewall-cmd --list-all-zones
```

Get the default zone by:

```
[user@computer]$ firewall-cmd --get-default-zone
```

Set the default zone by:

```
[user@computer]$ firewall-cmd --set-default-zone=<Zone>
```

Ask FirewallD to remember the default zone after a reboot:

```
[user@computer]$ firewall-cmd --permanent \
--set-default-zone=<Zone>
```

There is no outside access to the lab, so we can use the predefined zone **dmz**. Configure firewalld to use **dmz**, either as the default zone, or for all computers in the lab, and then check that the zone got activated. We list active zones by:

```
[user@computer]$ firewall-cmd --get-active-zones
```

If the zone is not active, restart the network service, or restart the computer

Now, let us add the MariaDB service to the **dmz** zone for the current session, i.e. we are opening the network for connections to MariaDB:

```
[user@computer]$ firewall-cmd --zone=dmz --add-service mysql
```

We must also ask firewalld to remember our decision after a reboot:

```
[user@computer]$ firewall-cmd --permanent --zone=dmz \
--add-service mysql
```

Some final remarks

When logged in to MariaDB, find the active user by issuing the following command:

```
MariaDB [privBase]> SELECT USER();
```

Find current privileges by:

```
MariaDB [privBase]> SHOW GRANTS;
```