

A Quick Guide to Using the MySQL SLES Repository

Abstract

This is a quick guide for using the MySQL SLES repository to install and upgrade MySQL on SUSE Linux Enterprise Server (SLES).

For supported versions and architectures of SUSE and openSUSE, see the official [Supported Platforms by MySQL](#) list.

For legal information, see the [Legal Notices](#).

For help with using MySQL, please visit the [MySQL Forums](#), where you can discuss your issues with other MySQL users.

Document generated on: 2021-10-18 (revision: 71116)

Table of Contents

..... 1

- [Adding SLES SDK Dependencies](#)
- [Steps for a Fresh Installation of MySQL](#)
- [Installing Additional MySQL Products and Components](#)
- [Upgrading MySQL with the MySQL SLES Repository](#)
- [Replacing MySQL Installed by an RPM from Other Sources](#)
- [Installing MySQL NDB Cluster Using the SLES Repository](#)
- [Installing Additional MySQL NDB Cluster Products and Components](#)

SLES Dependencies

In order to resolve required MySQL dependencies, add the SLES SDK repository to your repository list and Zypper will handle the required dependencies. For example, MySQL 8 has a `pkgconfig(openssl)` dependency.

MySQL binary distributions are built using GCC 7, and the lowest supported GCC version is GCC 5.3 (previously 4.8.5).

SLES 12: Installing MySQL RPM packages on SLES 12 requires that the GCC Devel repository is enabled, for example:

```
$> cd /etc/zypp/repos.d/  
$> wget https://download.opensuse.org/repositories/devel:/gcc/SLE-12/devel:gcc.repo
```

Steps for a Fresh Installation of MySQL

Note

The following instructions assume that no versions of MySQL (whether distributed by Oracle or other parties) have already been installed on your system; if that is not the case, follow instead the instructions given in [Replacing MySQL Installed by an RPM from Other Sources](#).

Adding the MySQL SLES Repository

First, add the MySQL SLES repository to your system's repository list. Follow these steps:

- Go to the download page for MySQL SLES repository at <https://dev.mysql.com/downloads/repo/suse/>.
- Select and download the release package for your SLES version.
- Install the downloaded release package with the following command, replacing `package-name` with the name of the downloaded package:

```
$> sudo rpm -Uvh package-name.rpm
```

The commands for installing the packages for SLES 12 and 15 are shown here, where `#` indicates the release number within a version such as `15-1`:

```
# SLES 12  
# Enables MySQL 8.0 by default  
$> sudo rpm -Uvh mysql80-community-release-sles12-#.noarch.rpm
```

```
# SLES 15 / openSUSE 15
# Enables MySQL 8.0 by default
$> sudo rpm -Uvh mysql80-community-release-s115-#.noarch.rpm
```

Importing MySQL GnuPG Key

Import into the system the GnuPG key for MySQL products, which will be used for checking signatures of the downloaded packages from the MySQL SLES repository, with the following command:

```
$> sudo rpm --import /etc/RPM-GPG-KEY-mysql
```

Selecting a Release Series

Within the MySQL SLES repository, different release series of the MySQL Community Server are hosted in different subrepositories. For SLES 12 and SLES 15, the subrepository for the latest GA series (currently MySQL 8.0) is enabled by default, and the subrepositories for all other series are disabled. Use this command to see all of the subrepositories in the MySQL SLES repository, and to see which of them are enabled or disabled:

```
$> zypper repos | grep mysql.*community
```

To install the latest release from a specific series, before running the installation command, make sure that the subrepository for the series you want is enabled and the subrepositories for other series are disabled. For example, on SLES 12, to disable the subrepository for MySQL 8.0, which is enabled by default, use the following command:

```
$> sudo zypper modifyrepo -d mysql80-community
```

Then, enable the subrepository for the release series you want. For example, to enable the subrepository for MySQL 5.7:

```
$> sudo zypper modifyrepo -e mysql57-community
```

You should only enable the subrepository for one release series at any time. When subrepositories for more than one release series are enabled, the latest series will be used.

Verify that the correct subrepositories have been enabled by running the following command and checking its output:

```
$> zypper repos -E | grep mysql.*community
4 | mysql80-community | MySQL 5.7 Community Server
```

After that, use the following command to refresh the repository information for the enabled subrepository:

```
$> sudo zypper refresh
```

Installing MySQL with Zypper

Install MySQL by the following command:

```
$> sudo zypper install mysql-community-server
```

This installs the package for the MySQL server, as well as other required packages.

Starting the MySQL Server

Start the MySQL server with the following command:

```
$> systemctl start mysql
```

You can check the status of the MySQL server with the following command:

```
$> systemctl status mysql
```

If the operating system is systemd enabled, standard `systemctl` (or alternatively, `service` with the arguments reversed) commands such as `stop`, `start`, `status`, and `restart` should be used to manage the MySQL server service. The `mysql` service is enabled by default, and it starts at system reboot. See [Managing MySQL Server with systemd](#) for additional information.

MySQL Server Initialization (for MySQL 8.0 and later): When the server is started for the first time, the server is initialized, and the following happens (if the data directory of the server is empty when the initialization process begins):

- The SSL certificate and key files are generated in the data directory.
- The [validate_password plugin](#) is installed and enabled.
- A superuser account `'root'@'localhost'` is created. A password for the superuser is set and stored in the error log file. To reveal it, use the following command:

```
$> sudo grep 'temporary password' /var/log/mysql/mysqlld.log
```

Change the root password as soon as possible by logging in with the generated, temporary password and set a custom password for the superuser account:

```
$> mysql -uroot -p
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass4!';
```

Note

MySQL's [validate_password](#) plugin is installed by default. This will require that passwords contain at least one uppercase letter, one lowercase letter, one digit, and one special character, and that the total password length is at least 8 characters.

You can stop the MySQL Server with the following command:

```
$> sudo service mysql stop
```

Securing the MySQL Installation (for MySQL 5.6 only)

The program `mysql_secure_installation` allows you to perform important operations like setting the root password, removing anonymous users, and so on. Always run it to secure your MySQL installation:

```
$> mysql_secure_installation
```

It is important to remember the root password you set. See [mysql_secure_installation — Improve MySQL Installation Security](#) for details.

Do not run `mysql_secure_installation` after an installation of MySQL 5.7 or 8.0, as the function of the program has already been performed by the SLES repository installation.

Installing Additional MySQL Products and Components

You can install more components of MySQL. List subrepositories in the MySQL SLES repository with the following command:

```
$> zypper repos | grep mysql.*community
```

Use the following command to list the packages for the MySQL components available for a certain subrepository, changing `subrepo-name` to the name of the subrepository you are interested in :

```
$> zypper packages subrepo-name
```

Install any packages of your choice with the following command, replacing `package-name` with name of the package (you might need to enable first the subrepository for the package, using the same method for selecting a subrepository for a specific release series outlined in [Selecting a Release Series](#)):

```
$> sudo zypper install package-name
```

For example, to install the MySQL benchmark suite from the subrepository for the release series you have already enabled:

```
$> sudo zypper install mysql-community-bench
```

Upgrading MySQL with the MySQL SLES Repository

Note

- Before performing any update to MySQL, follow carefully the instructions in [Upgrading MySQL](#). Among other instructions discussed there, it is especially important to back up your database before the update.

Use the MySQL SLES repository to perform an in-place update (that is, replacing the old version of the server and then running the new version using the old data files) for your MySQL installation by following these steps (they assume you have installed MySQL with the MySQL SLES repository; if that is not the case, following the instructions in [Replacing MySQL Installed by an RPM from Other Sources](#) instead):

Selecting a Target Series

During an update operation, by default, the MySQL SLES repository updates MySQL to the latest version in the release series you have chosen during installation (see [Selecting a Release Series](#) for details), which means, for example, a 5.7.x installation will *not* be updated to a 8.0.x release

automatically. To update to another release series, you need to first disable the subrepository for the series that has been selected (by default, or by yourself) and enable the subrepository for your target series. To do that, follow the general instructions given in [Selecting a Release Series](#).

As a general rule, to upgrade from one release series to another, go to the next series rather than skipping a series.

Important

- For important information about upgrading from MySQL 5.5 to 5.6, see [Upgrading from MySQL 5.5 to 5.6](#).
- For important information about upgrading from MySQL 5.6 to 5.7, see [Upgrading from MySQL 5.6 to 5.7](#).
- For important information about upgrading from MySQL 5.7 to 8.0, see [Upgrading from MySQL 5.7 to 8.0](#).
- In-place downgrading of MySQL is not supported by the MySQL SLES repository. Follow the instructions in [Downgrading MySQL](#).

Upgrading MySQL

Upgrade MySQL and its components by the following command:

```
$> sudo zypper update mysql-community-server
```

Alternatively, you can update MySQL by telling Zypper to update everything on your system (this might take considerably more time):

```
$> sudo zypper update
```

Upgrading the Data

The MySQL Server always restarts after an update by Zypper. Prior to MySQL 8.0.16, run `mysql_upgrade` after the server restarts to check and possibly resolve any incompatibilities between the old data and the upgraded software. `mysql_upgrade` also performs other functions; see [mysql_upgrade — Check and Upgrade MySQL Tables](#) for details. As of MySQL 8.0.16, this step is not required, as the server performs all tasks previously handled by `mysql_upgrade`.

Note

After upgrading from MySQL 5.6 to 5.7, if you have problem connecting to the server as `root` (which will make it impossible to run `mysql_upgrade`), stop the server and then restart it with the `--skip-grant-tables` option before you run `mysql_upgrade`. See [Upgrading from MySQL 5.6 to 5.7](#) for details.

You can also update a specific component only. Use the following command to list all the installed packages from the MySQL SLES repository:

```
$> zypper packages -i | grep mysql-.*community
```

After identifying the package name of the component of your choice, update the package with the following command, replacing `package-name` with the name of the package:

```
$> sudo zypper update package-name
```

Replacing MySQL Installed by an RPM from Other Sources

RPMs for installing the MySQL Community Server and its components can be downloaded from MySQL either from the [MySQL Developer Zone](#), from the native software repository of SLES, or from the MySQL SLES repository. The RPMs from the those sources might be different, and they might install and configure MySQL in different ways.

If you have installed MySQL with RPMs from the MySQL Developer Zone or the native software repository of SLES and want to replace the installation using the RPM from the MySQL SLES repository, follow these steps:

1. Back up your database to avoid data loss. See [Backup and Recovery](#) on how to do that.
2. Stop your MySQL Server, if it is running. If the server is running as a service, you can stop it with the following command:

```
$> sudo service mysql stop
```

3. Follow the steps given for [Adding the MySQL SLES Repository](#).
4. Follow the steps given for [Selecting a Release Series](#).
5. Follow the steps given for [Installing MySQL with Zypper](#). You will be asked if you want to replace the old packages with the new ones; for example:

```
Problem: mysql-community-server-5.6.22-2.sles11.x86_64 requires mysql-community-client = 5.6.22-2.sles11,
but this requirement cannot be provided uninstalleable providers:
mysql-community-client-5.6.22-2.sles11.x86_64[mysql56-community]
Solution 1: replacement of mysql-client-5.5.31-0.7.10.x86_64 with mysql-community-client-5.6.22-2.sles11.x86_64
Solution 2: do not install mysql-community-server-5.6.22-2.sles11.x86_64
Solution 3: break mysql-community-server-5.6.22-2.sles11.x86_64 by ignoring some of its dependencies

Choose from above solutions by number or cancel [1/2/3/c] (c)
```

Choose the “replacement” option (“Solution 1” in the example) to finish your installation from the MySQL SLES repository.

6. If you have replaced your MySQL installation with a higher version from the MySQL SLES repository, follow the instructions given in [Upgrading the Data](#).

Installing MySQL NDB Cluster Using the SLES Repository

Notes

- The following instructions assume that neither the MySQL Server nor MySQL NDB Cluster has already been installed on your system; if that is not the case, remove the MySQL Server or MySQL NDB Cluster, including all its executables, libraries, configuration files, log files, and data directories, before you continue. However there is no need to remove the release package you might have used to enable the MySQL SLES repository on your system.
- The NDB Cluster Auto-Installer package has a dependency on the [python2-crypto](#) and [python-paramiko](#) packages. Zypper can take care of this dependency if the Python repository has been enabled on your system. You can do this with the following command:

```
$> sudo zypper addrepo \
http://download.opensuse.org/repositories/devel:languages:python/SLE_12_SP2/devel:languages:python
```

- The MySQL SLES repository supports installation of MySQL NDB Cluster only for release 7.5.6 and later, and only for SLES 12. For other methods of installing NDB Cluster, see [Installation of NDB Cluster on Linux](#), or [Installation of NDB Cluster on Linux](#), depending on the release series that you are using.

Adding the MySQL SLES Repository for MySQL NDB Cluster and Importing the GnuPG Key

Follow the steps in [Adding the MySQL SLES Repository](#) and [Importing MySQL GnuPG Key](#) to add the MySQL SLES repository to your system's repository list and import the GnuPG Key. If you have already performed those steps before, make sure you have the most up-to-date version of the release package by running the following command:

```
$> sudo zypper update mysql57-community-release
```

Selecting the MySQL NDB Cluster Subrepository

Within the MySQL SLES repository, the MySQL Community Server and MySQL NDB Cluster are hosted in different subrepositories. By default, the subrepository for the latest GA series of the MySQL Server is enabled and the subrepository for MySQL NDB Cluster is disabled. To install NDB Cluster, disable the subrepository for the MySQL Server and enable the subrepository for NDB Cluster. For example, disable the subrepository for MySQL 5.7, which is enabled by default, with the following command:

```
$> sudo zypper modifyrepo -d mysql57-community
```

Then, enable the subrepository for MySQL NDB Cluster:

```
$> sudo zypper modifyrepo -e mysql-cluster-7.5-community
```

Verify that the correct subrepositories have been enabled by running the following command and checking its output:

```
$> zypper repos -E | grep mysql.*community
10 | mysql-cluster-7.5-community | MySQL Cluster 7.5 Community | Yes | No
```

After that, use the following command to refresh the repository information for the enabled subrepository:

```
$> sudo zypper refresh
```

Installing MySQL NDB Cluster

For a minimal installation of MySQL NDB Cluster, follow these steps:

- Install the components for SQL nodes:

```
$> sudo zypper install mysql-cluster-community-server
```

After the installation is completed, start and initialize the SQL node by following the steps given in [Starting and Stopping the MySQL Server](#).

If you choose to initialize the data directory manually using the `mysqld --initialize` command (see [Initializing the Data Directory](#) for details), a `root` password is going to be generated and stored in the SQL node's error log; see [MySQL Server Initialization \[3\]](#) for how to find the password, and for a few things you need to know about it.

- Install the executables for management nodes:

```
$> sudo zypper install mysql-cluster-community-management-server
```

- Install the executables for data nodes:

```
$> sudo zypper install mysql-cluster-community-data-node
```

To install more NDB Cluster components, see [Installing Additional MySQL Products and Components](#).

Configuring and Starting MySQL NDB Cluster

See [Initial Configuration of NDB Cluster](#) on how to configure MySQL NDB Cluster and [Initial Startup of NDB Cluster](#) on how to start it for the first time.

Installing Additional MySQL NDB Cluster Products and Components

You can use Zypper to install individual components and additional products of MySQL NDB Cluster from the MySQL SLES repository. To do that, assuming you already have the MySQL SLES repository on your system's repository list (if not, follow Step 1 and 2 of [Installing MySQL NDB Cluster Using the SLES Repository](#)), follow the same steps given in [Installing Additional MySQL Products and Components](#).

Note

Known issue: Currently, not all components required for running the MySQL NDB Cluster test suite are installed automatically when you install the test suite package (`mysql-cluster-community-test`). Install the following packages with `zypper install` before you run the test suite:

- `mysql-cluster-community-auto-installer`
- `mysql-cluster-community-management-server`
- `mysql-cluster-community-data-node`
- `mysql-cluster-community-memcached`
- `mysql-cluster-community-java`
- `mysql-cluster-community-ndbclient-devel`

Legal Notices

Copyright © 1997, 2021, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms:

You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Oracle disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility/>.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab>.