

# Установка MySQL

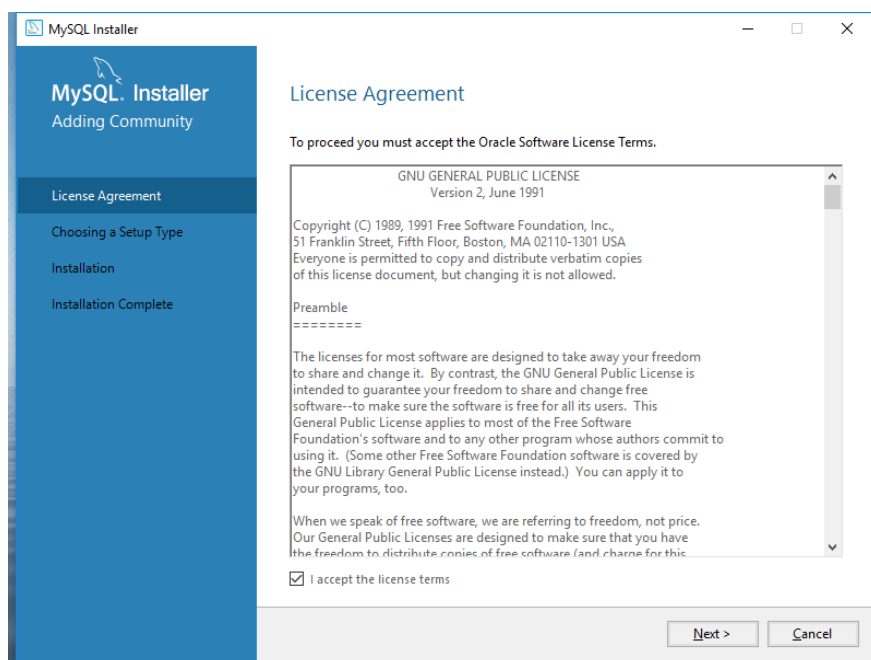
Сервер MySQL доступен для установки во всех современных операционных системах. Мы рассмотрим установку MySQL в Windows, Mac OS X и Linux (на примере дистрибутива Ubuntu).

## Установка в Windows

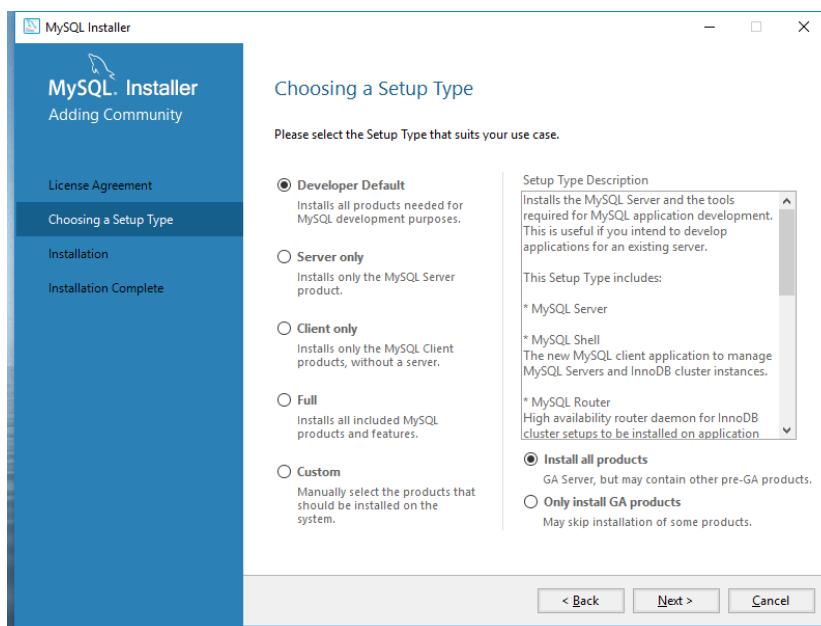
Для установки сервера MySQL необходимо загрузить дистрибутив с официальной страницы <https://dev.mysql.com/downloads/windows/installer/8.0.html>. На странице предлагается либо полный дистрибутив, либо загрузчик, который скачает дистрибутив во время установки. Выберите полный дистрибутив, если на момент установки компьютер не будет подключен к сети Интернет. В случае, если Интернет доступен можно выбирать любой из вариантов.

В случае если используется Windows версии ниже 10, MySQL 8 может отказаться работать. В этом случае можно прибегнуть к архиву дистрибутивов по ссылке <https://downloads.mysql.com/archives/community/>. На более старых версиях Windows, можно использовать версию MySQL 5.7.X.

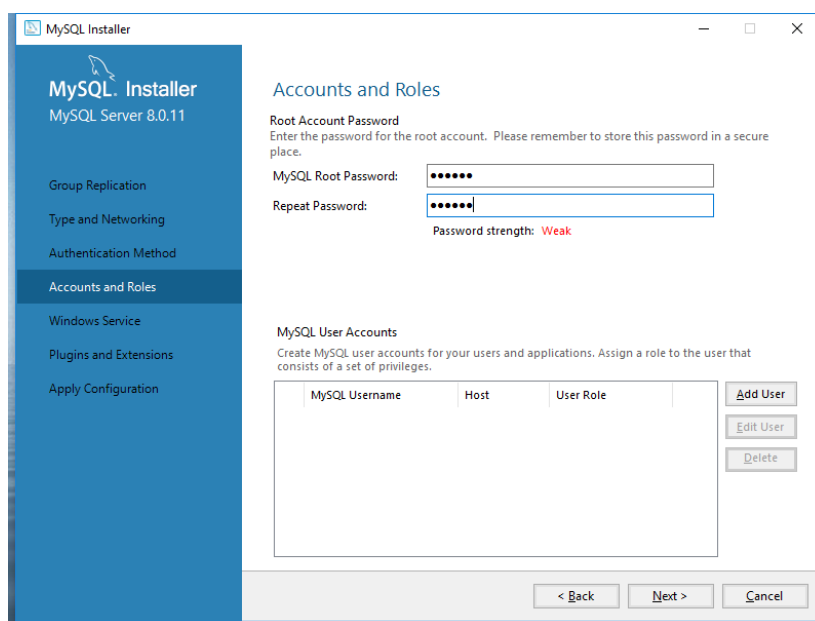
После загрузки дистрибутива можно приступить к установке, пройдя последовательно через серию диалоговых окон менеджера.



При первом знакомстве с MySQL предлагаемые настройки лучше оставить без изменений.

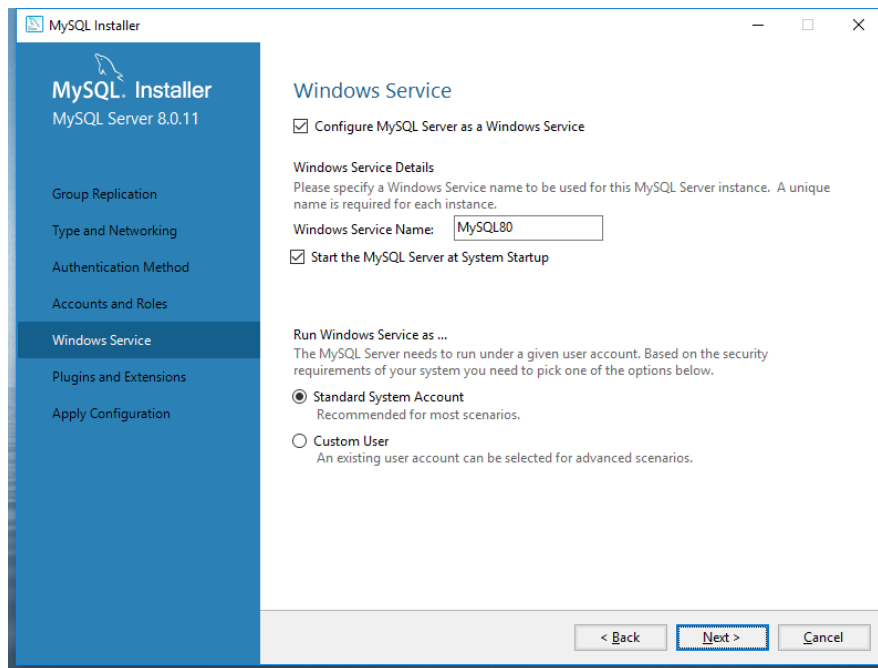


В одном из диалоговых окон будет предложено задать пароль для root-пользователя.

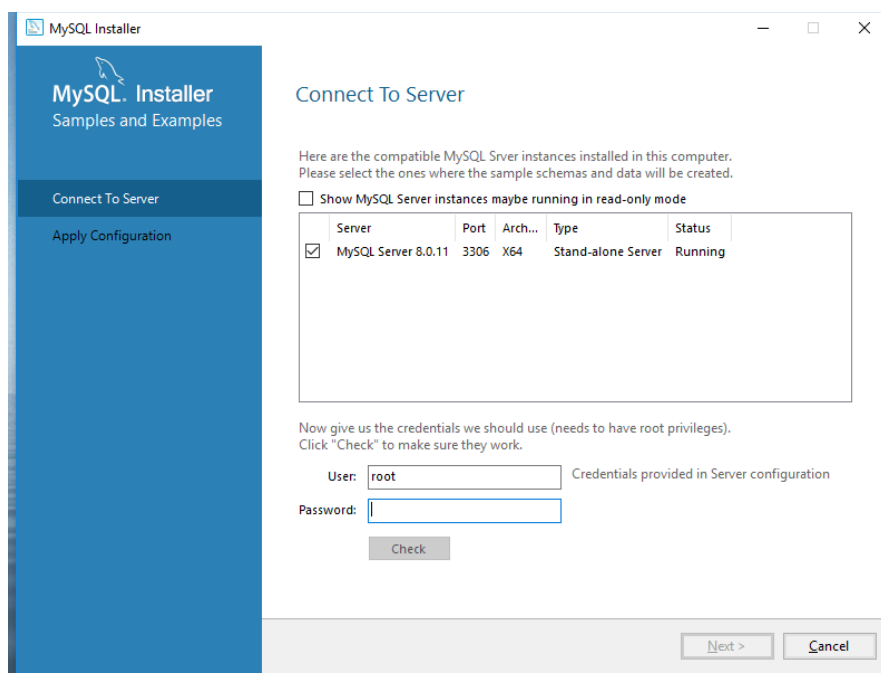


В этом же окне при необходимости можно добавить дополнительных mysql-пользователей. Впрочем, это можно сделать и позже, используя команды CREATE USER и GRANT.

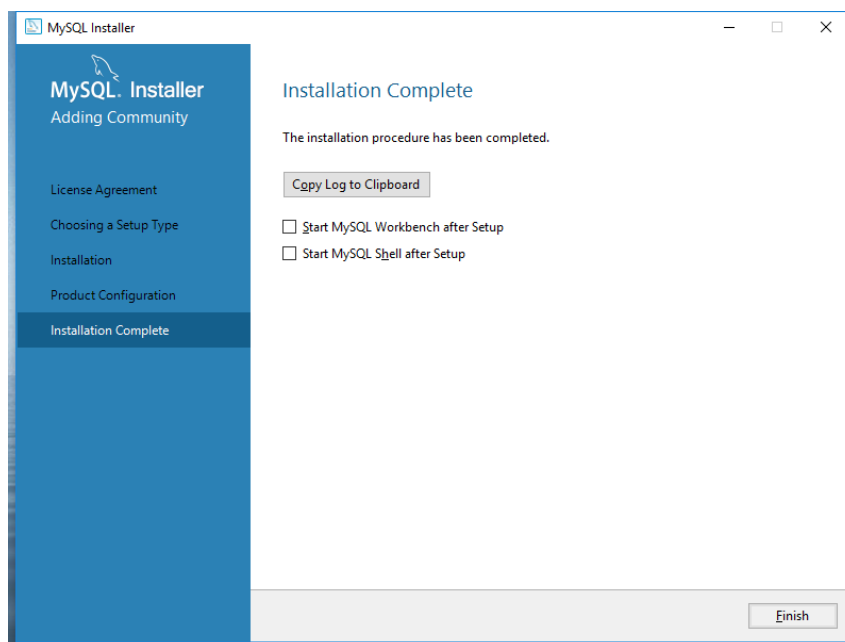
Чтобы вы могли запускать сервер MySQL при старте операционной системы, следует включить флажок "Configure MySQL Server as Windows Service". Это сообщит установщику о необходимости создать сервис, который будет запускать MySQL-сервер при входе в систему и штатно останавливать его работу при завершении сеанса.



После установки основных компонентов MySQL-сервера менеджер установки предложит проверить соединение с MySQL-сервером. Здесь потребуется ввести пароль root-пользователя, заданного на одном из предыдущих этапов.



По окончании установки менеджер предложит запустить MySQL Workbench и командную строку с консольным MySQL-клиентом. Можно снять соответствующие флажки и завершить работу менеджера установки.



## Установка в Mac OS X

Прежде чем устанавливать MySQL в Mac OS X, следует установить Command Line Tools for XCode из магазина AppStore. XCode — это интегрированная среда разработки приложений для Mac OS X и iOS. Полная загрузка XCode не обязательна, достаточно установить инструменты командной строки и компилятор. Убедиться в том, установлен ли XCode, можно при помощи команды:

```
xcode-select -p
/Applications/Xcode.app/Contents/Developer
```

Если вместо указанного выше пути выводится предложение установить Command Line Tools, следует установить этот пакет, выполнив команду:

```
xcode-select --install
```

Теперь можно приступать к установке PHP, для чего лучше всего воспользоваться менеджером пакетов Homebrew. На момент написания книги установить Homebrew можно было при помощи команды:

```
/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Впрочем, точную команду всегда можно выяснить на официальном сайте <http://brew.sh>. После установки в командной строке будет доступна команда **brew**, при помощи которой можно загружать, удалять и обновлять пакеты с программным обеспечением.

После того, как менеджер Homebrew установлен, приступаем к установке MySQL. Сделать это можно при помощи команды:

```
brew install mysql
```

Если необходимо, чтобы MySQL-сервер стартовал автоматически при загрузке операционной системы, в каталоге Library/LaunchAgents домашнего каталога необходимо создать символическую ссылку на plist-файл дистрибутива MySQL:

```
ln -sfv /usr/local/opt/mysql/*.plist ~/Library/LaunchAgents
```

## Установка в Linux (дистрибутив Ubuntu)

Перед тем как мы займемся установкой MySQL, нужно обновить индексы пакетного менеджера apt-get при помощи следующей команды:

```
sudo apt-get update
```

Это позволит обновить базу пакетов, доступных через apt-get. После этого рекомендуется выполнить обновление системы при помощи команды:

```
sudo apt-get upgrade
```

Для установки MySQL следует выполнить команду:

```
sudo apt-get install mysql-server
```

В ходе установки несколько раз будет предложено ввести пароль для пользователя root, под управлением учетной записи которого будет осуществляться администрирование сервера. Запомните введенный пароль.

Убедиться в том, что сервер запущен, можно при помощи утилиты ps. Результаты вывода утилиты можно отфильтровать по строке mysqld, чтобы было проще видеть результаты.

```
ps aux | grep mysqld
mysql      2036  0.0  9.5 1167692 195740 pts/0    Sl   17:31   0:00
/usr/sbin/mysqld --user=mysql --pid-file=/var/run/mysqld/mysqld1.pid
--socket=/var/run/mysqld/mysqld1.sock --port=3307 --basedir=/usr
--datadir=/var/lib/mysql1 --tmpdir=/tmp
root       2089  0.0  0.0   8984   360 pts/0    R+   17:49   0:00 grep
--color=auto mysqld
```

Если сервер оказался не запущенным, его можно запустить при помощи команды `service`, после которой указывается имя сервиса `mysql` и команда `start`

```
sudo service mysql start
```

Остановить сервер можно, передав команду `stop`

```
sudo service mysql stop
```

После установки MySQL нужно провести постинсталляционную настройку с целью повышения безопасности сервера. Для этого надо выполнить следующую команду:

```
sudo mysql_secure_installation
```

Сразу после выполнения команды и при необходимости ввода пароля для **sudo** будет затребован пароль для root-пользователя MySQL, который вводился при установке. Затем утилита в диалоговом режиме предложит настроить параметры безопасности. На вопросы, предлагаемые утилитой, следует отвечать либо N (нет), либо Y (да). Ниже приводится один из возможных вариантов ответов.

- N — Изменить пароль root? Отвечаем "нет", т. к. пароль был задан при установке MySQL.
- Y — Удалить анонимного пользователя без пароля? Анонимный пользователь — это пользователь без имени, в качестве пароля которого выступает пустая строка. У такого пользователя немного прав доступа, кроме доступа к базе данных `test`, однако на рабочем сервере лучше не оставлять возможностей для несанкционированного доступа, поэтому пользователя лучше удалить, ответив на запрос положительно.
- Y — Запретить удаленное обращение от имени пользователя root? У аккаунтов в MySQL существуют два режима: доступ с локального хоста и доступ через сеть. Пользователь `root` обладает безграничными полномочиями, поэтому лучше запретить сетевой доступ для этого аккаунта. Чтобы воспользоваться root-доступом, придется сначала попасть на сервер — лишний барьер безопасности не повредит. Поэтому отвечаем на этот вопрос положительно.
- Y — Удаление базы данных `test`? Это уникальная база данных, привилегии которой настроены таким образом, что к ней имеют доступ все `mysql`-пользователи. Во избежание недоразумений и проблем с безопасностью лучше сразу удалить эту базу данных.
- Y — Перегрузка привилегий? Отвечаем "да", чтобы внесенные изменения сразу вступили в силу.

## Запуск утилиты `mysqld_multi` в Linux (дистрибутив Ubuntu)

При запуске утилиты `mysqld_multi` в среде Linux следует учитывать ряд особенностей. Дело в том, что данная утилита не может разобрать конфигурационный файл, если в нем используются комментарии `#` и включения других конфигурационных файлов.

Поэтому при выполнении операций по настройке репликации (в уроке 7), когда требуется два одновременно запущенных MySQL-сервера лучше подготовить отдельный конфигурационный файл `mysqld_multi.cnf` и указывать путь к нему при помощи директивы `--defaults-file` сервера `mysqld`.

В дистрибутиве Ubuntu файл можно расположить в папке `/etc/mysql/mysql.conf.d/`. Ниже приводится его возможное содержимое.

```
[mysqld1]
user      = mysql
pid-file  = /var/run/mysqld/mysqld1.pid
socket    = /var/run/mysqld/mysqld1.sock
port      = 3307
basedir   = /usr
datadir   = /var/lib/mysql1
tmpdir    = /tmp

[mysqld2]
user      = mysql
pid-file  = /var/run/mysqld/mysqld1.pid
socket    = /var/run/mysqld/mysqld1.sock
port      = 3308
basedir   = /usr
datadir   = /var/lib/mysql1
tmpdir    = /tmp
```

Обратите внимание, что `pid` и `socket`-файл в секциях `[mysqld1]` и `[mysqld2]` должны отличаться. Серверам должны быть назначены отличающиеся друг от друга порты (в примере 3307 и 3308). Кроме того, у серверов должны быть независимые каталоги данных.

Прежде чем запускать утилиту `mysqld_multi` потребуется создать каталоги данных `/var/lib/mysql1` и `/var/lib/mysql2`. Для этого можно воспользоваться MySQL-сервером `mysqld` передав ему параметр `--initialize`, в параметре `--user` следует указать пользователя `mysql`, а в параметре `--datadir` указать путь до каталога данных. Команды следует выполнять при запущенном MySQL-сервере.

```
mysqld --initialize --user=mysql --datadir=/var/lib/mysql1
mysqld --initialize --user=mysql --datadir=/var/lib/mysql2
```

Команды создают новые каталоги данных по путям, указанным в параметре `--datadir`. Теперь можно остановить основной сервер MySQL

```
sudo service mysql stop
```

и запустить два сервера при помощи утилиты `mysqld_multi`, обязательно указав в параметре `--defaults-file` путь до подготовленного ранее конфигурационного файла (без комментариев и включений других конфигурационных файлов):

```
mysqld_multi --defaults-file=/etc/mysql/mysql.conf.d/mysqld_multi.cnf  
start
```

Остановить сервера можно передав утилите `mysqld_multi` команды `stop`

```
mysqld_multi --defaults-file=/etc/mysql/mysql.conf.d/mysqld_multi.cnf  
stop
```