



TUTORIAL

Comment installer MySQL sur Ubuntu 20.04

Ubuntu MySQL Databases Ubuntu 20.04

By [Mark Drake](#)

Published on May 13, 2020 21k

Français

Une version précédente de ce tutoriel a été rédigée par [Hazel Virdó](#).

Introduction

MySQL est un système de gestion de base de données open source, généralement installé dans le cadre de la populaire pile LAMP (Linux, Apache, MySQL, PHP/Python/Perl). Il met en œuvre le modèle relationnel et utilise le langage de requête structuré (mieux connu sous le nom de SQL – Structured Query Language) pour gérer ses données.

Ce tutoriel passe en revue la façon d'installer la version 8 de MySQL sur un serveur Ubuntu 20.04. En le terminant, vous disposerez d'une base de données relationnelle fonctionnelle que vous pourrez utiliser pour construire votre prochain site web ou votre prochaine application.

Conditions préalables

Pour suivre ce tutoriel, vous aurez besoin de :

- Un serveur Ubuntu 20.04 avec un utilisateur administratif non root et un pare-feu configuré avec UFW. Pour configurer cela, suivez notre [guide de configuration initiale du serveur pour Ubuntu 18.20](#).

Étape 1 - Installation de MySQL

Sur Ubuntu 20.04, vous pouvez installer MySQL en utilisant le référentiel de paquets APT. Au moment de la rédaction de ce document, la version de MySQL disponible dans le référentiel Ubuntu par défaut est la version 8.0.19.

Pour l'installer, mettez à jour l'index des paquets sur votre serveur si vous ne l'avez pas fait récemment :

```
$ sudo apt update
```

Ensuite, installez le paquet `mysql-server` :

```
$ sudo apt install mysql-server
```

Cela installera MySQL, mais ne vous demandera pas de définir un mot de passe ou de faire d'autres changements de configuration. Comme cela rend votre installation de MySQL non sécurisée, nous allons aborder ce point.

Étape 2 – Configuration de MySQL

Pour les nouvelles installations de MySQL, vous devrez exécuter le script de sécurité inclus dans le SGBD. Ce script modifie certaines des options par défaut les moins sûres pour des choses comme les connexions root distantes et les sample users.

Exécutez le script de sécurité avec `sudo` :

```
$ sudo mysql_secure_installation
```

Vous serez alors guidé à travers une série d'invites où vous pourrez apporter quelques modifications aux options de sécurité de votre installation MySQL. La première invite vous demandera si vous souhaitez configurer le plugin Validate Password, que vous pouvez utiliser pour tester la solidité de votre mot de passe MySQL.

Si vous choisissez de mettre en place le plugin Validate Password, le script vous demandera de choisir un niveau de validation du mot de passe. Le niveau le plus fort - que vous sélectionnez en entrant 2 - exigera que votre mot de passe comporte au moins huit caractères, dont un mélange de majuscules, de minuscules, de chiffres et de caractères spéciaux :

Output

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords and improve security. It checks the strength of password and allows the users to set only those passwords which are secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: Y

There are three levels of password validation policy:

LOW Length >= 8

MEDIUM Length >= 8, numeric, mixed case, and special characters

STRONG Length >= 8, numeric, mixed case, special characters and dictionary

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG:

2

Que vous choisissiez ou non de configurer le plugin Validate Password, l'invite suivante vous demandera de définir un mot de passe pour l'utilisateur **root** de MySQL. Entrez et confirmez le mot de passe sécurisé de votre choix :

Output

Please set the password for root here.

New password:

Re-enter new password:

Si vous avez utilisé le plugin Validate Password, vous recevrez des commentaires sur la force de votre nouveau mot de passe. Ensuite, le script vous demandera si vous voulez continuer avec le mot de passe que vous venez de saisir ou si vous voulez en saisir un nouveau. En supposant que vous êtes satisfait de la force du mot de passe que vous venez d'entrer, saisissez Y pour poursuivre le script :

Output

Estimated strength of the password: 100

```
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No)
```

À partir de là, vous pouvez appuyer sur **Y** puis sur **ENTER** pour accepter les valeurs par défaut pour toutes les questions suivantes. Cela supprimera les utilisateurs anonymes et la base de données de test, désactivera les connexions **root** à distance, et chargera ces nouvelles règles afin que MySQL respecte immédiatement les modifications que vous avez apportées.

Notez que même si vous avez défini un mot de passe pour l'utilisateur MySQL **root**, cet utilisateur n'est pas configuré pour s'authentifier avec un mot de passe lors de la connexion au shell MySQL. Si vous le souhaitez, vous pouvez ajuster ce paramètre en suivant l'étape 3.

Étape 3 - (Facultatif) Ajustement de l'authentification et des privilèges de l'utilisateur

Dans les systèmes Ubuntu fonctionnant sous MySQL 5.7 (et versions ultérieures), l'utilisateur MySQL **root** est configuré pour s'authentifier en utilisant le plugin `auth_socket` par défaut plutôt qu'avec un mot de passe. Cela permet d'améliorer la sécurité et la convivialité dans de nombreux cas, mais cela peut aussi compliquer les choses si vous devez autoriser un programme externe (par exemple, phpMyAdmin) à accéder à l'utilisateur.

Afin d'utiliser un mot de passe pour se connecter à MySQL en tant que **root**, vous devrez changer sa méthode d'authentification de `auth_socket` à un autre plugin, tel que `caching_sha2_password` ou `mysql_native_password`. Pour ce faire, ouvrez l'invite MySQL depuis votre terminal :

```
$ sudo mysql
```

Ensuite, vérifiez la méthode d'authentification utilisée par chacun de vos comptes utilisateurs MySQL à l'aide de la commande suivante :

```
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
```

Output

```
+-----+-----+
| user          | authentication_string |
+-----+-----+
| debian-sys-maint | $A$005$1S|M#3K #XslZ.xXUq.crEqTjMvhgOIX7B/zki5DeLA3JB9nh0KwENTwQ4 | ca
| mysql.infoschema | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED
| mysql.session    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED
| mysql.sys        | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED
| root            |                        |
+-----+-----+
5 rows in set (0.00 sec)
```

Dans cet exemple, vous pouvez voir que l'utilisateur **root** s'authentifie effectivement en utilisant le plugin `auth_socket`. Pour configurer le compte **root** afin qu'il s'authentifie avec un mot de passe, lancez une instruction `ALTER USER` pour modifier le plugin d'authentification utilisé et définir un nouveau mot de passe.

Veillez à remplacer `password` par un mot de passe fort de votre choix, et souvenez-vous que cette commande modifiera le mot de passe **root** que vous avez défini à l'étape 2 :

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY 'password';
```

Remarque : L'instruction `ALTER USER` précédente définit l'utilisateur MySQL **root** pour s'authentifier avec le plugin `caching_sha2_password`. [Selon la documentation officielle de MySQL](#), `caching_sha2_password` est le plugin d'authentification préféré de MySQL, car il fournit un cryptage de mot de passe plus sûr que l'ancien, mais encore largement utilisé, `mysql_native_password`.

Cependant, de nombreuses applications PHP – phpMyAdmin, par exemple – ne fonctionnent pas de manière fiable avec `caching_sha2_password`. Si vous prévoyez d'utiliser cette base de données avec une application PHP, vous pouvez définir **root** pour vous authentifier avec `mysql_native_password` à la place :

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';
```

Exécutez ensuite `FLUSH PRIVILEGES` qui ordonne au serveur de recharger les tableaux de subventions et de mettre vos nouvelles modifications en vigueur :

```
mysql> FLUSH PRIVILEGES;
```

Vérifiez à nouveau les méthodes d'authentification utilisées par chacun de vos utilisateurs pour confirmer que **root** ne s'authentifie plus à l'aide du plugin `auth_socket` :

```
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
```

Output

```
+-----+-----+
| user                | authentication_string |
+-----+-----+
| debian-sys-maint    | $A$005$lS|M#3K #XslZ.xXUq.crEqTjMvhgOIX7B/zki5DeLA3JB9nh0KwENTwQ4 | ca
| mysql.infoschema    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED
| mysql.session       | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED
| mysql.sys           | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED
| root                | *3636DACC8616D997782ADD0839F92C1571D6D78F
+-----+-----+
5 rows in set (0.00 sec)
```

Vous pouvez voir dans cet exemple que l'utilisateur MySQL **root** s'authentifie maintenant en utilisant le plugin `caching_sha2_password`. Une fois que vous avez confirmé cela sur votre propre serveur, vous pouvez quitter le shell MySQL :

```
mysql> exit
```

Certains peuvent aussi trouver qu'il est plus adapté à leur travail de se connecter à MySQL avec un utilisateur dédié. Pour créer un tel utilisateur, ouvrez à nouveau le shell MySQL :

```
$ sudo mysql
```

Remarque : si vous avez activé l'authentification par mot de passe pour **root**, comme décrit dans les paragraphes précédents, vous devrez utiliser une commande différente pour accéder au shell MySQL. Ce qui suit permettra d'exécuter votre client MySQL avec les privilèges d'utilisateur habituels, et vous n'obtiendrez les privilèges d'administrateur au sein de la base de données qu'en vous authentifiant :

```
$ mysql -u root -p
```

De là, créez un nouvel utilisateur et attribuez-lui un mot de passe fort :

```
mysql> CREATE USER 'sammy'@'localhost' IDENTIFIED BY 'password';
```

Ensuite, accordez à votre nouvel utilisateur les privilèges appropriés. Par exemple, vous pourriez accorder les privilèges d'utilisateur à tous les tableaux de la base de données, ainsi que le pouvoir d'ajouter, de modifier et de supprimer des privilèges d'utilisateur, avec cette commande :

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'sammy'@'localhost' WITH GRANT OPTION;
```

Notez qu'à ce stade, vous n'avez pas besoin d'exécuter à nouveau la commande `FLUSH PRIVILEGES`. Cette commande n'est nécessaire que lorsque vous modifiez les tableaux de subvention en utilisant des instructions comme `INSERT`, `UPDATE` ou `DELETE`. Comme vous avez créé un nouvel utilisateur, au lieu de modifier un utilisateur existant, le `FLUSH PRIVILEGES` est inutile ici.

Ensuite, quittez le shell MySQL :

```
mysql> exit
```

Enfin, testons l'installation MySQL.

Étape 4 – Test de MySQL

Quelle que soit la manière dont vous l'avez installé, MySQL aurait dû commencer à fonctionner automatiquement. Pour le tester, vérifiez son état.

```
$ systemctl status mysql.service
```

Vous verrez une sortie semblable à ce qui suit :

Output

```
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2020-04-21 12:56:48 UTC; 6min ago
 Main PID: 10382 (mysqld)
    Status: "Server is operational"
     Tasks: 39 (limit: 1137)
  Memory: 370.0M
    CGroup: /system.slice/mysql.service
            └─10382 /usr/sbin/mysqld
```

Si MySQL ne tourne pas, vous pouvez le démarrer avec `sudo systemctl start mysql`.

Pour une vérification supplémentaire, vous pouvez essayer de vous connecter à la base de données en utilisant l'outil `mysqladmin` qui est un client vous permettant d'exécuter des commandes administratives. Par exemple, cette commande indique de se connecter à MySQL en tant que **root** (`-u root`), de demander un mot de passe (`-p`) et de renvoyer la version.

```
$ sudo mysqladmin -p -u root version
```

Vous devriez voir une sortie qui ressemble à ça :

Output

```
mysqladmin Ver 8.0.19-0ubuntu5 for Linux on x86_64 ((Ubuntu))
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.
```

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

```
Server version      8.0.19-0ubuntu5
Protocol version    10
Connection          Localhost via UNIX socket

UNIX socket         /var/run/mysqld/mysqld.sock
Uptime:             10 min 44 sec
```

```
Threads: 2  Questions: 25  Slow queries: 0  Opens: 149  Flush tables: 3  Open tables: 69  C
```

Cela signifie que MySQL est opérationnel.

Conclusion

Vous avez maintenant installé une configuration MySQL de base sur votre serveur. Voici quelques exemples des prochaines étapes que vous pouvez entreprendre :

- [Configurer une pile LAMP](#)
- [S'entraîner à exécuter des requêtes avec SQL](#)

Qu'avez-vous pensé de cette traduction?



Was this helpful?

Yes

No



[Report an issue](#)

About the authors



Mark Drake

Technical Writer @ DigitalOcean

Still looking for an answer?



Ask a question



Search for more help

RELATED



Managed MySQL, Redis, Postgres on DigitalOcean
[Product](#)

Comment utiliser SSH pour se connecter à un serveur distant

 [Tutorial](#)

Comment créer un nouvel utilisateur et octroyer des autorisations dans MySQL

 [Tutorial](#)

Comments

0 Comments

Leave a comment...

Sign In to Comment



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



**GET OUR BIWEEKLY NEWSLETTER**

Sign up for Infrastructure as a
Newsletter.

**HUB FOR GOOD**

Working on improving health and
education, reducing inequality,
and spurring economic growth?
We'd like to help.

**BECOME A CONTRIBUTOR**

You get paid; we donate to tech
nonprofits.

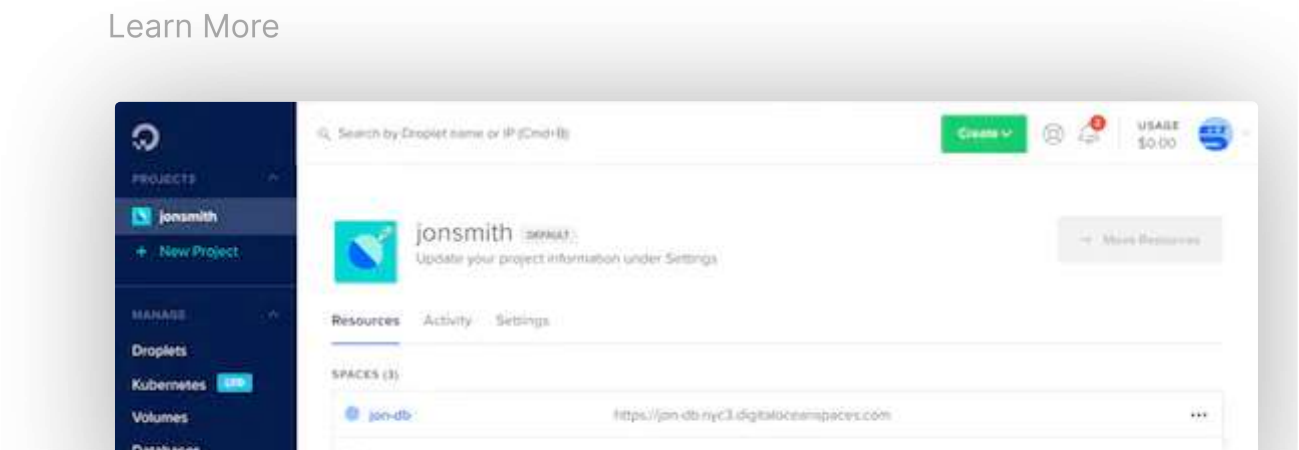
Featured on [Community](#) [Kubernetes Course](#) [Learn Python 3](#) [Machine Learning in Python](#)
[Getting started with Go](#) [Intro to Kubernetes](#)

DigitalOcean Products [Virtual Machines](#) [Managed Databases](#) [Managed Kubernetes](#) [Block Storage](#)
[Object Storage](#) [Marketplace](#) [VPC](#) [Load Balancers](#)

Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you’re running one virtual machine or ten thousand.

[Learn More](#)



© 2021 DigitalOcean, LLC. All rights reserved.

- Company
- About
- Leadership
- Blog
- Careers
- Partners
- Referral Program
- Press
- Legal
- Security & Trust Center

Products

- Pricing
- Products Overview
- Droplets
- Kubernetes
- Managed Databases
- Spaces
- Marketplace

Community

- Tutorials
- Q&A
- Tools and Integrations
- Tags
- Product Ideas
- Write for DigitalOcean
- Presentation Grants

Contact

- Get Support
- Trouble Signing In?
- Sales
- Report Abuse
- System Status

- Load Balancers

Block Storage

API Documentation

Documentation

Release Notes
- Hatch Startup Program

Shop Swag

Research Program

Open Source

Code of Conduct