

Ubuntu 安装 Mysql 数据库

首先更新apt-get工具，执行命令如下：

```
apt-get upgrade
```

```
(base) root@4U:~# apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  bridge-utils cuda-cccl-11-8 cuda-command-line-tools-11-8 cuda-compiler-11-8 cuda-cudart-11-8 cuda-cudart-dev-11-8 cuda-cuobjdump-11-8 cuda-cu
cuda-driver-dev-11-8 cuda-gdb-11-8 cuda-libraries-11-8 cuda-libraries-dev-11-8 cuda-memcheck-11-8 cuda-nsight-11-8 cuda-nsight-compute-11-8 c
cuda-nvprof-11-8 cuda-nvprune-11-8 cuda-nvrtc-11-8 cuda-nvrtc-dev-11-8 cuda-nvtx-11-8 cuda-nvvp-11-8 cuda-profiler-api-11-8 cuda-sanitizer-11-8
cuda-toolkit-11-config-common cuda-tools-11-8 cuda-visual-tools-11-8 docker-scan-plugin gds-tools-11-8 gds-tools-12-3 genders libcublas-11-8
libcuffile-dev-11-8 libcurand-11-8 libcurand-dev-11-8 libcusolver-11-8 libcusolver-dev-11-8 libcuspars-11-8 libcuspars-dev-11-8 libgenders0
libnjpeg-dev-11-8 linux-generic-hwe-22.04 linux-headers-generic-hwe-22.04 linux-image-generic-hwe-22.04 nsight-compute-2022.3.0 nsight-compu
Use 'apt autoremove' to remove them.
The following packages have been kept back:
  libreoffice-base-core libreoffice-calc libreoffice-common libreoffice-core libreoffice-draw libreoffice-gnome libreoffice-gtk3 libreoffice-he
  libreoffice-writer libuno-cppuhelpergcc3-3 linux-generic-hwe-20.04 linux-headers-generic-hwe-20.04 linux-image-generic-hwe-20.04 python3-uno
The following packages will be upgraded:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin google-chrome-stable intel-microcode nodejs thunderbird
12 upgraded, 0 newly installed, 0 to remove and 22 not upgraded.
Need to get 238 MB of archives.
After this operation, 9,594 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates/main amd64 ubuntu-pro-client-l10n amd64 32.3-20.04 [18.5 kB]
Get:2 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates/main amd64 ubuntu-pro-client amd64 32.3-20.04 [208 kB]
Get:3 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates/main amd64 ubuntu-advantage-tools all 32.3-20.04 [11.0 kB]
Get:4 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates/main amd64 thunderbird-locale-en-us all 1:115.11.0+build2-0ubuntu0.20.04.1 [9,26
Get:5 http://mirrors.tuna.tsinghua.edu.cn/ubuntu focal-updates/main amd64 intel-microcode amd64 3.20240514.0ubuntu0.20.04.1 [6,894 kB]
Get:6 https://dl.google.com/linux/chrome/deb stable/main amd64 google-chrome-stable amd64 125.0.6422.141-1 [108 MB]
Get:7 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.32-1 [30.0 MB]
Get:8 https://deb.nodesource.com/node_20.x nodistro/main amd64 nodejs amd64 20.14.0-1nodesource [31.6 MB]
Get:9 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:26.1.3-1-ubuntu.22.04~jammy [14.6 MB]
Get:10 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce amd64 5:26.1.3-1-ubuntu.22.04~jammy [25.3 MB]
Get:11 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-rootless-extras amd64 5:26.1.3-1-ubuntu.22.04~jammy [9,319 kB]
Get:12 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-compose-plugin amd64 2.27.0-1-ubuntu.22.04~jammy [12.5 MB]
```

安装Mysql，执行如下命令：

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

```
(base) root@4U:~# apt-get install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:

The following packages have unmet dependencies:
  libevent-pthreads-2.1-7 : Depends: libevent-core-2.1-7 (= 2.1.11-stable-1) but 2.1.12-stable-1build3 is to be installed
E: Unable to correct problems, you have held broken packages.
(base) root@4U:~# sudo apt-get -f install
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  bridge-utils cuda-cccl-11-8 cuda-command-line-tools-11-8 cuda-compiler-11-8 cuda-cudart-11-8 cuda-cudart-dev-11-8 cuda-cu
cuda-driver-dev-11-8 cuda-gdb-11-8 cuda-libraries-11-8 cuda-libraries-dev-11-8 cuda-memcheck-11-8 cuda-nsight-11-8 cuda-r
cuda-nvprof-11-8 cuda-nvprune-11-8 cuda-nvrtc-11-8 cuda-nvrtc-dev-11-8 cuda-nvtx-11-8 cuda-nvvp-11-8 cuda-profiler-api-11-8
cuda-toolkit-11-config-common cuda-tools-11-8 cuda-visual-tools-11-8 docker-scan-plugin gds-tools-11-8 gds-tools-12-3 gen
libcuffile-dev-11-8 libcurand-11-8 libcurand-dev-11-8 libcusolver-11-8 libcusolver-dev-11-8 libcuspars-11-8 libcuspars-dev-11-8
libnjpeg-dev-11-8 linux-generic-hwe-22.04 linux-headers-generic-hwe-22.04 linux-image-generic-hwe-22.04 nsight-compute-2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 22 not upgraded.
(base) root@4U:~# sudo apt-get install libevent-core-2.1-7=2.1.11-stable-1
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  bridge-utils cuda-cccl-11-8 cuda-command-line-tools-11-8 cuda-compiler-11-8 cuda-cudart-11-8 cuda-cudart-dev-11-8 cuda-cu
cuda-driver-dev-11-8 cuda-gdb-11-8 cuda-libraries-11-8 cuda-libraries-dev-11-8 cuda-memcheck-11-8 cuda-nsight-11-8 cuda-r
cuda-nvprof-11-8 cuda-nvprune-11-8 cuda-nvrtc-11-8 cuda-nvrtc-dev-11-8 cuda-nvtx-11-8 cuda-nvvp-11-8 cuda-profiler-api-11-8
cuda-toolkit-11-config-common cuda-tools-11-8 cuda-visual-tools-11-8 docker-scan-plugin gds-tools-11-8 gds-tools-12-3 gen
libcuffile-dev-11-8 libcurand-11-8 libcurand-dev-11-8 libcusolver-11-8 libcusolver-dev-11-8 libcuspars-11-8 libcuspars-dev-11-8
libnjpeg-dev-11-8 linux-generic-hwe-22.04 linux-headers-generic-hwe-22.04 linux-image-generic-hwe-22.04 nsight-compute-2
Use 'sudo apt autoremove' to remove them.
The following packages will be DOWNGRADED:
```

开启Mysql 服务，执行命令如下：

```
service mysql start
```

并确认是否成功开启mysql,执行命令如下：

```
service mysql status
```

```
(base) root@4U:~# service mysql start
(base) root@4U:~# service mysql status
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-06-04 10:45:33 CST; 3min 9s ago
     Process: 3820578 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 3820586 (mysqld)
      Status: "Server is operational"
        Tasks: 37 (limit: 231379)
       Memory: 352.7M
          CPU: 2.525s
      CGroup: /system.slice/mysql.service
              └─3820586 /usr/sbin/mysqld

6月 04 10:45:32 4U systemd[1]: Starting MySQL Community Server...
6月 04 10:45:33 4U systemd[1]: Started MySQL Community Server.
(base) root@4U:~# █
```

确认是否启动成功，在LISTEN状态下，启动成功：

```
netstat -tap | grep mysql
```

```
(base) root@4U:~# netstat -tap | grep mysql
tcp        0      0 0 localhost:mysql      0.0.0.0:*           LISTEN      3820586/mysqld
tcp        0      0 0 localhost:33060      0.0.0.0:*           LISTEN      3820586/mysqld
(base) root@4U:~# █
```

在最新使用的 MySQL 版本中，默认使用 auth_socket 插件来进行身份验证，这意味着 root 用户通过操作系统的用户身份进行认证，而不是使用密码。这种配置在许多 Linux 系统上是默认的，特别是在安装 MySQL 时不要求设置密码的情况下。如果我们希望使用传统的密码验证方式来登录 MySQL，需要更改 root 用户的认证方式。以下是如何将 root 用户从 auth_socket 插件更改为使用密码认证的步骤：

1. 登录 MySQL:

首先，使用以下命令登录到 MySQL：

```
sudo mysql
```

```
(base) root@4U:~# sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.36-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

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

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

2. 更改认证插件和设置密码:

在 MySQL 命令行中, 使用以下命令来更改 `root` 用户的认证插件并设置一个新密码:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '新密码';
FLUSH PRIVILEGES;
```

把 `'新密码'` 替换为你想要设置的密码。

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Sn0wflake'&e('';
Query OK, 0 rows affected (0.02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

mysql>
```

3. 退出并测试登录:

更改完成后, 退出 MySQL:

```
exit;
```

然后尝试使用新密码重新登录:

```
sudo mysql -u root -p
```

系统将提示你输入密码, 此时应输入你刚才设置的密码。

```
(base) root@4U:~# sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.36-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

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

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

MySQL 配置文件中的 bind-address 参数限制了可以接受连接的 IP 地址。需要确认它是否设置为允许从你的客户端 IP 访问。查看 /etc/mysql/mysql.conf.d/mysqld.cnf 文件中的 bind-address:

```
(base) root@4U:~# vim /etc/mysql/mysql.conf.d/mysqld.cnf
(base) root@4U:~#
```

如果设置为 127.0.0.1 (只允许本地连接), 需要改为 0.0.0.0 (允许任何 IP 连接) 或具体的外部 IP 地址, 然后重启 MySQL 服务:

```
[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket              = /var/run/mysqld/mysqld.sock
# port                = 3306
# datadir             = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.
# tmpdir              = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address         = 0.0.0.0
mysqlx-bind-address  = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size      = 16M
# max_allowed_packet = 64M
# thread_stack        = 256K
# thread_cache_size   = -1
```

执行重启命令:

```
sudo systemctl restart mysql
```

再次登录MySQL，确保 MySQL 用户的主机设置允许从你的客户端 IP 地址连接。可以在 MySQL 中运行以下 SQL 命令来检查：

```
(base) root@4U:~# vim /etc/mysql/mysql.conf.d/mysqld.cnf
(base) root@4U:~# vim /etc/mysql/mysql.conf.d/mysqld.cnf
(base) root@4U:~# sudo systemctl restart mysql

(base) root@4U:~#
```

接下来，我们需要使用 root 用户或具有相应权限的用户登录到 MySQL。可以通过以下命令登录：

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

登录后，需要选择 mysql 数据库，因为用户信息存储在这个数据库中：

```
USE mysql;
```

执行以下 SQL 命令来查看所有用户及其主机：

```
SELECT user, host FROM user;
```

```
mysql> USE mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT user, host FROM user;
+-----+-----+
| user          | host          |
+-----+-----+
| debian-sys-maint | localhost    |
| mysql.infoschema | localhost    |
| mysql.session   | localhost    |
| mysql.sys       | localhost    |
| root           | localhost    |
+-----+-----+
5 rows in set (0.00 sec)
```

决定修改现有用户（如 root 用户），可以更改用户的 host 值，以允许从任意 IP 地址连接。执行如下命令：

```
UPDATE user SET host = '%' WHERE user = 'root' AND host = 'localhost';  
FLUSH PRIVILEGES
```

这里将 root 用户的 host 从 localhost 改为 %，表示从任何 IP 地址都允许连接。

```
mysql> UPDATE user SET host = '%' WHERE user = 'root' AND host = 'localhost';  
Query OK, 1 row affected (0.01 sec)  
Rows matched: 1  Changed: 1  Warnings: 0  
  
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> █
```

Windows下载Navicat, 地址: https://www.navicat.com/en/download/direct-download?product=navicat170_premium_en_x64.exe&location=1



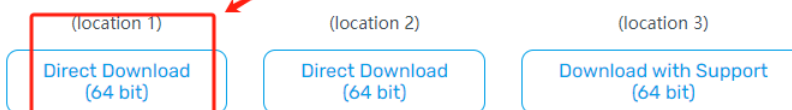
Navicat
Premium

Download Trial

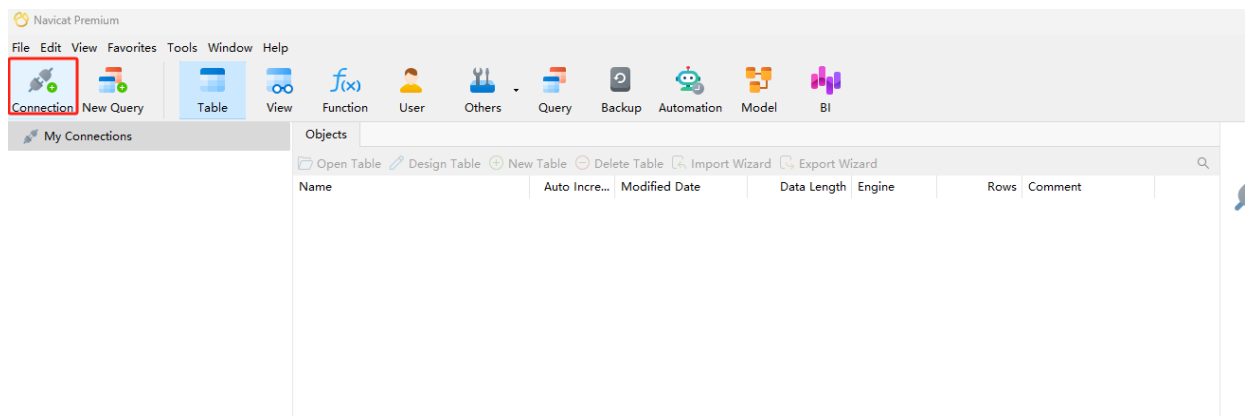
We offer a 14-day fully functional FREE trial of Navicat.

Windows

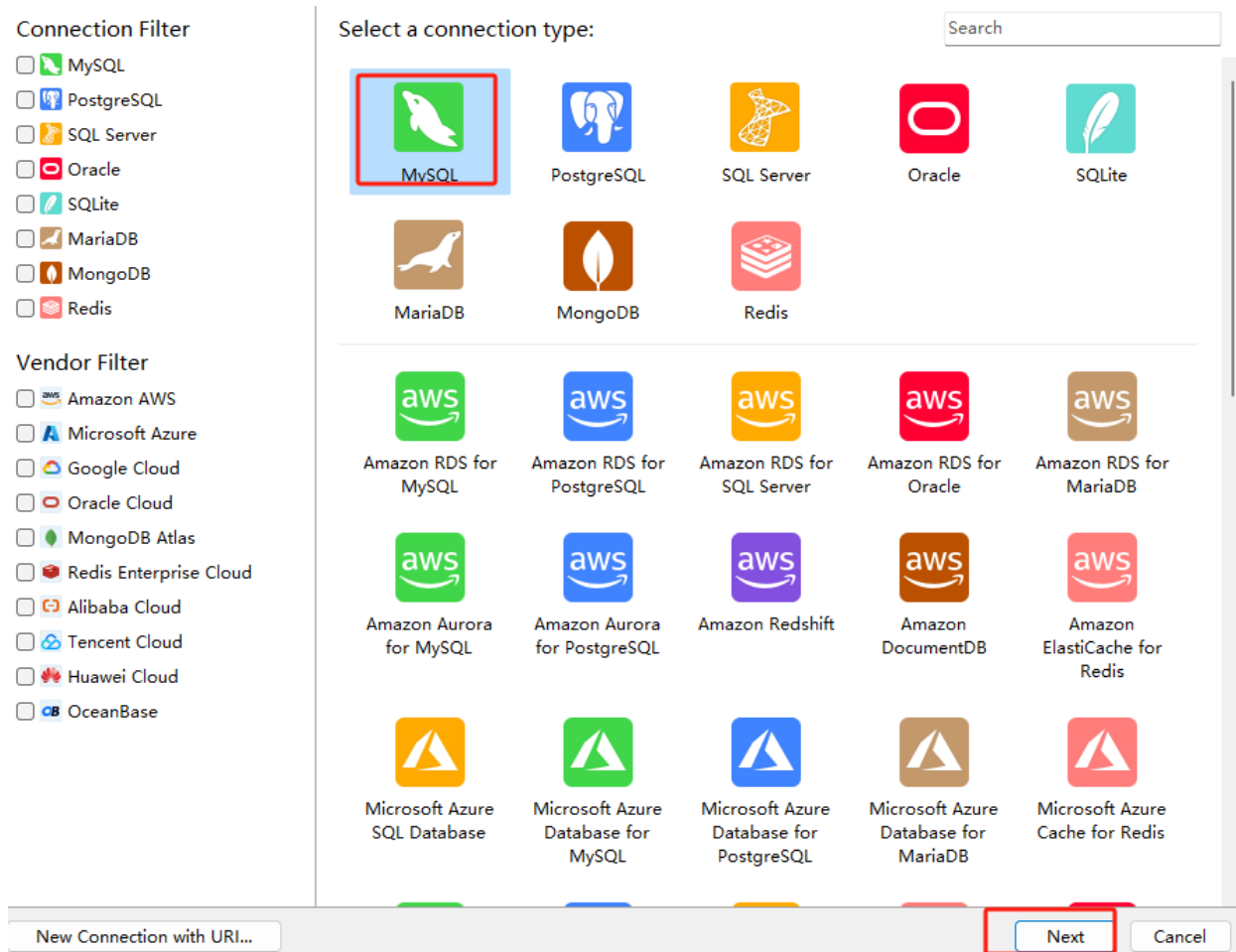
Navicat Premium 17



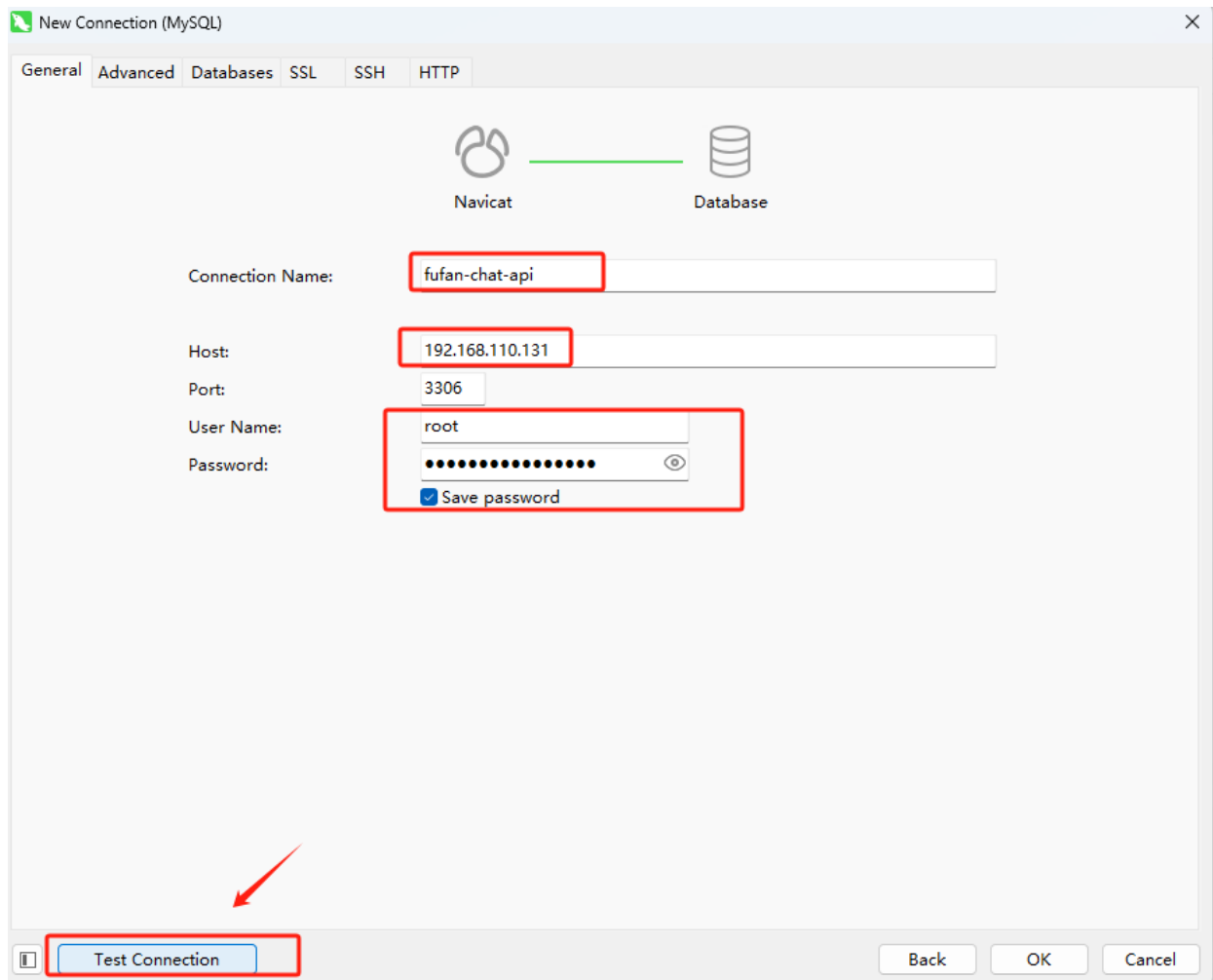
选择Connection:



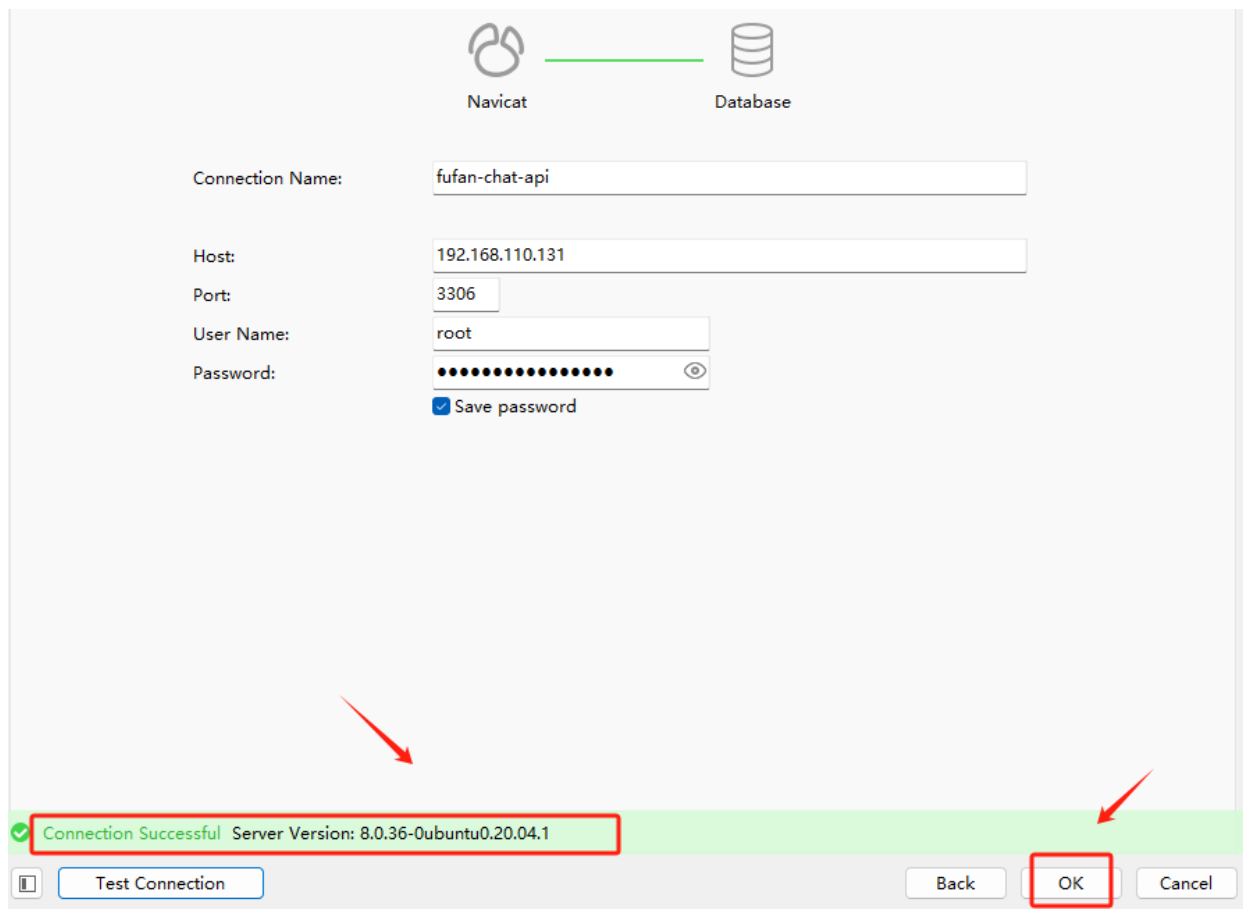
新建一个Mysql连接:



输入远程服务器的IP, Mysql的用户名和密码, 执行连通性测试。



如何能够正常连接，会提示 Connection Successful 字样。



接下来就可以正常在Navicate工具中进行Mysql操作了。

