Установка и настройка MySQL в Ubuntu 22.04

Установка MySQL

Обновим сведения о пакетах:

```
user@ubuntu–server:~$ sudo apt update_
```

Установка MySQL сервера и клиента:

```
user@ubuntu-server:~$ sudo apt install mysql-server mysql-client
```

Если пароль root не запросил установщик, устанавливаем его самостоятельно (и запоминаем):

```
user@ubuntu-server:~$ sudo mysqladmin password -u root -p
Enter password:
New password:
Confirm new password:
```

Далее нужно выполнить **mysql_secure_installation**, для того чтобы защитить ваш MySQL сервер:

```
user@ubuntu-server:~$ sudo mysql secure installation
```

Здесь можно ответить "n":

```
user@ubuntu-server:~$ sudo mysql_secure_installation

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: n
```

На все остальные вопросы нужно отвечать утвердительно:

- Remove anonymous users удалить аноноимных пользователей
- **Disallow root login remotely** отключить удалённые подключения для root пользователя, это касается только удалённых подключений, подключения с localhost удалёнными не считаются.
- Remove test database and access to it удалить тестовую базу данных.
- Reload privilege tables now перезагрузить таблицу привелегий.

```
All done!
user@ubuntu-server:~$ [
```

Проверим статус сервера:

создадим БД и пользователя с полными правами на эту базу:

```
user@ubuntu-server:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.39-Oubuntu0.22.04.1 (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> CREATE DATABASE test_db;
Query OK, 1 row affected (0.01 sec)

mysql> CREATE USER 'testuser'@'%' IDENTIFIED BY '1234';
Query OK, 0 rows affected (0.04 sec)

mysql> GRANT ALL PRIVILEGES ON test_db.* TO 'testuser'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

Настроим MySQL сервер для приёма удалённых подключений:

```
GNU nano 6.2
                        /etc/mysql/mysql.conf.d/mysqld.cnf *
 * Basic Settings
               = mysql
# pid-file
              = /var/run/mysqld/mysqld.pid
              = /var/run/mysqld/mysqld.sock
 socket
               = 3306
 port
              = /var/lib/mysql
 datadir
# 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.
                    = 0.0.0.0
bind-address
mysqlx-bind-address
```

И перезапустим сервер:

```
user@ubuntu-server:~$ sudo systemctl restart mysql
user@ubuntu-server:~$ [
```

Откроем порт 3306 на сервере:

```
user@ubuntu-server:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
user@ubuntu-server:~$ sudo ufw allow 3306/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
user@ubuntu-server:~$ sudo ufw status
Status: active
То
                          Action
                                      From
3306/tcp
                          ALLOW
                                     Anywhere
3306/tcp (v6)
                          ALLOW
                                     Anywhere (v6)
user@ubuntu-server:~$
```

Убедимся, что MySQL слушает порт 3306:

Проверка правильности установки

Унаем IP сервера:

```
user@ubuntu-server:~$ ip a
1: lo: <LOOPBACK, UP, LOWER UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
      valid lft forever preferred lft forever
   inet6 ::1/128 scope host
      valid lft forever preferred lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc fq codel state UP gr
oup default qlen 1000
   link/ether 08:00:27:9a:5a:89 brd ff:ff:ff:ff:ff
  inet 192.168.1.67/24 me ric 100 brd 192.168.1.255 scope global dynamic enp0s
3
      valid 1ft 85672sec preferred 1ft 85672sec
    inet6 fe80::a00:27ff:fe9a:5a89/64 scope link
      valid lft forever preferred lft forever
user@ubuntu-server:~$
```

Подключаемся с удаленной машины:

```
C:\Users\user>mysql -h 192.168.1.67 -u testuser -p
Enter password: ****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.39-0ubuntu0.22.04.1 (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.

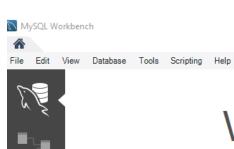
mysql>
```

Настройка сервера завершена.

Резервное копирование.

Создадим тестовую базу данных с несколькими таблицами и заполним их.

B MySQL WorkBanch создадим новое соединение:



Welcome to MySQL V

MySQL Workbench is the official graphical user interface (GUI) tool for Normal create and browse your database schemas, work with database object design and run SQL queries to work with stored data. You can also migrat database vendors to your MySQL database

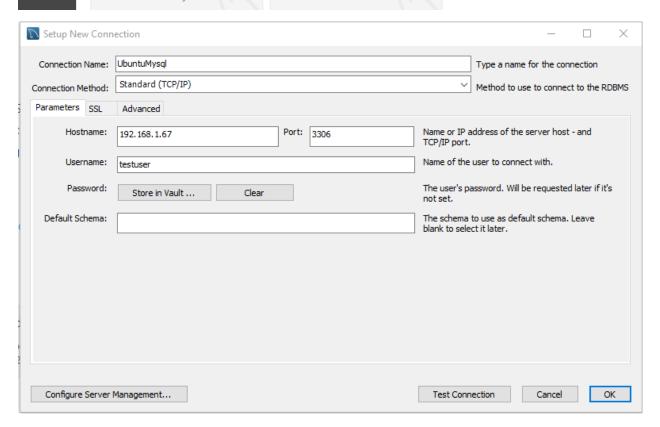
Browse Documentation >

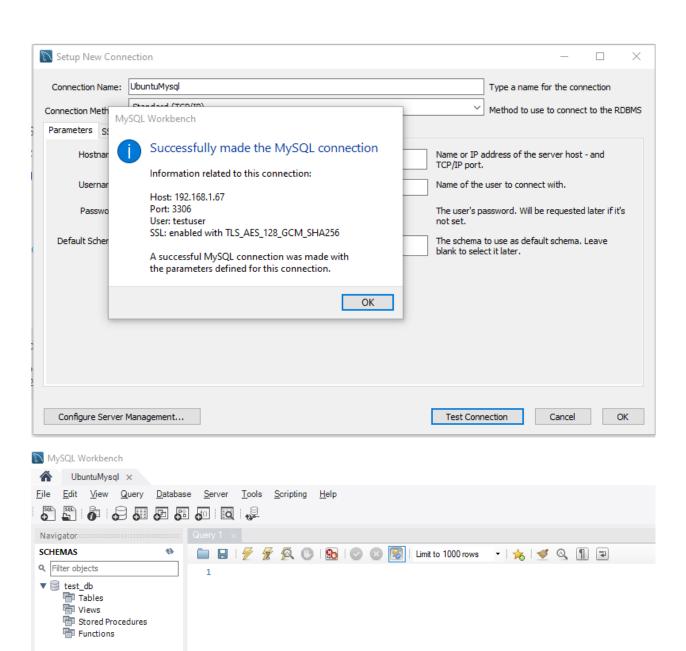
Read the Blog >



Local instance MySQL84

studentdb



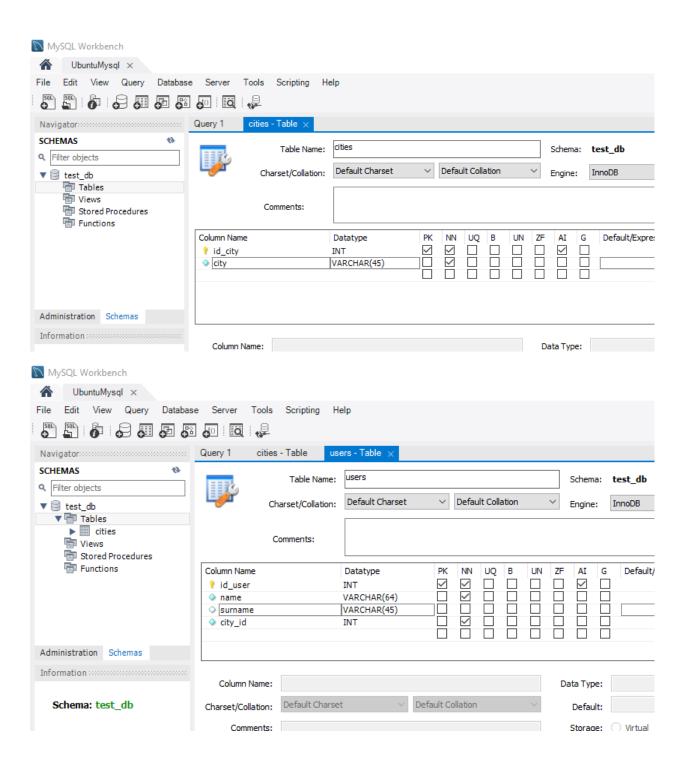


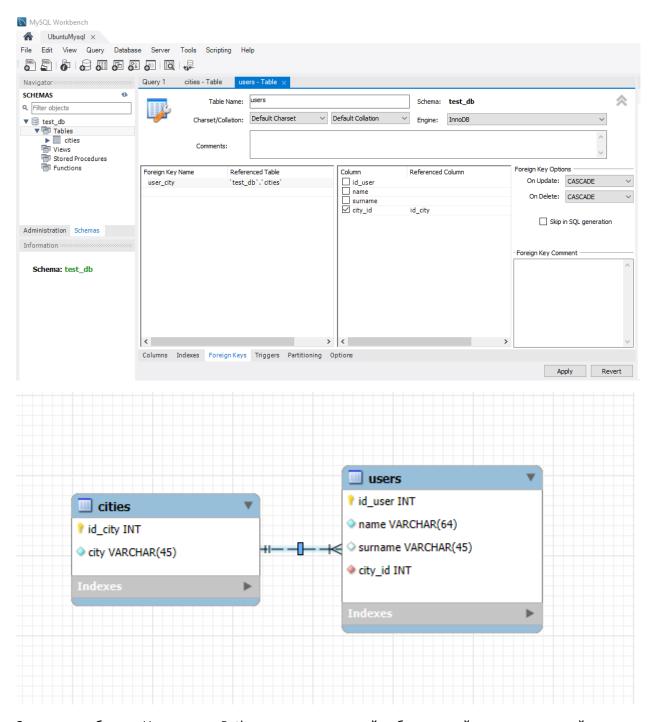
Создадим две таблицы:

Administration Schemas

No object selected

Information :





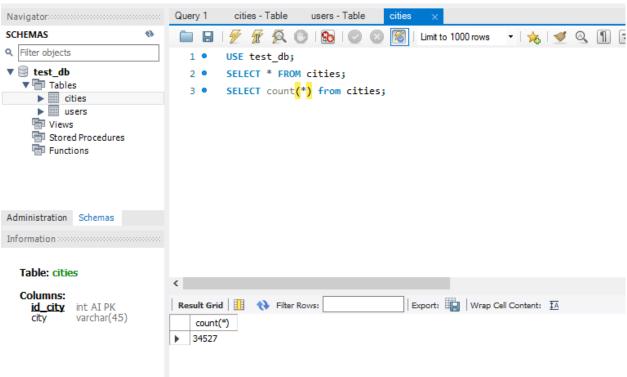
Заполним таблицы. Используем Python скрипт с прошлой лабораторной, изменив настройки подключения:

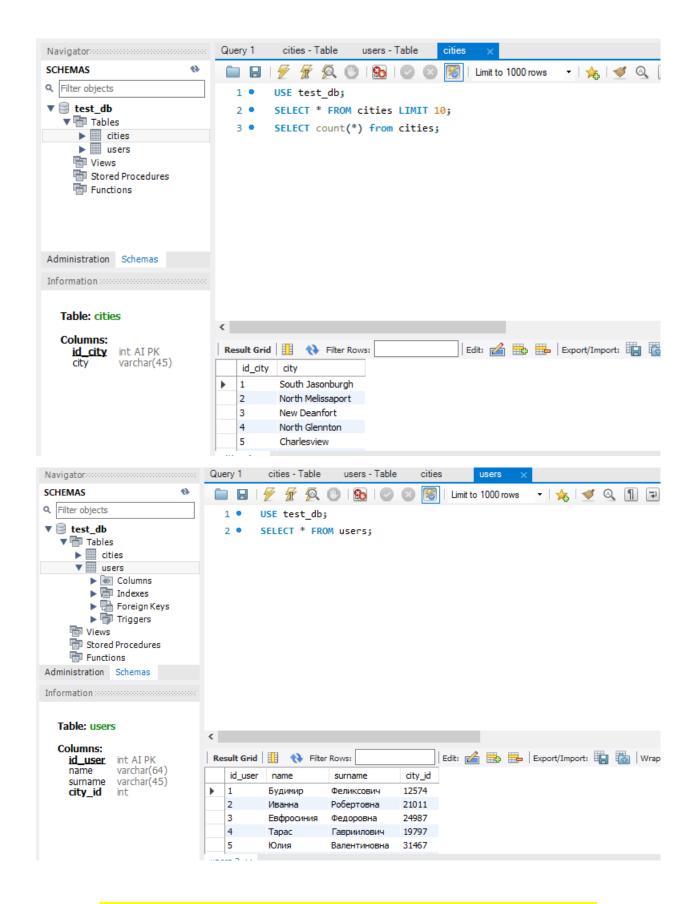
```
import pymysql
from faker import Faker

connection = pymysql.connect(
host='192.168.1.67',
port=3306,
user='testuser',
password='1234',
database='test_db',
cursorclass=pymysql.cursors.DictCursor

11
)

# Генерация списка городов
faker = Faker('en_GB')
```





Резервное копирование базы данных с использованием утилиты mysqldump.

Создание резервной копии:

```
user@ubuntu-server:~$ mysqldump -u testuser -p test_db > testdb_backup.sql
Enter password:
mysqldump: Error: 'Access denied; you need (at least one of) the PROCESS privilege(s) for this
operation' when trying to dump tablespaces
user@ubuntu-server:~$ [
```

Попытка неудачная, юзеру не хватает привелегии "PROCESS", исправим:

```
user@ubuntu-server:~$ mysql -u root -p
Enter password:
ERROR 1698 (28000): Access denied for user 'root'@'localhost'
user@ubuntu-server:~$ [
```

Опять неудача, root не может подключаться к БД удаленно, мы сами запретили. Пробуем непосредственно с сервера:

```
user@ubuntu-server:~$ sudo mysql -u root -p
[sudo] password for user:
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 27
Server version: 8.0.39-Oubuntu0.22.04.1 (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> GRANT PROCESS ON *.* TO 'testuser'@'%';
Query OK, O rows affected (0.02 sec)
mysql> FLUSH PRIVILEGES;
Query OK, O rows affected (0.01 sec)
mysql>
```

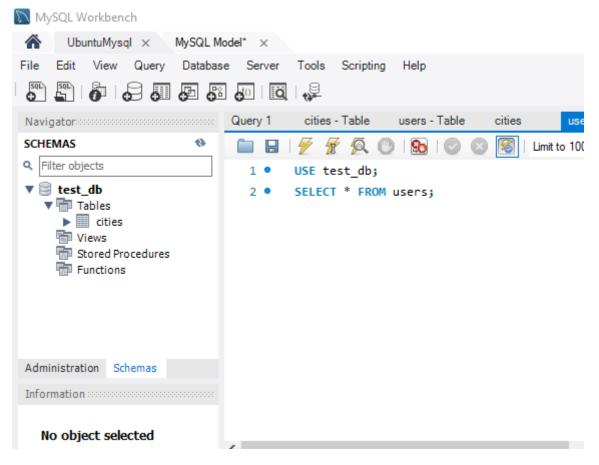
Теперь повторим попытку создания резервной копии:

```
user@ubuntu-server:~$ mysqldump -u testuser -p test_db > testdb_backup.sql
Enter password:
user@ubuntu-server:~$ [
```

Если мы не указывали путь, файл резервной копии создается в текущем каталоге, проверим:

```
user@ubuntu-server:~$ 1s -1
total 5648
-rw-rw-r-- 1 user user 5780519 Sep 28 10:17 testdb_backup.sql
user@ubuntu-server:~$ []
```

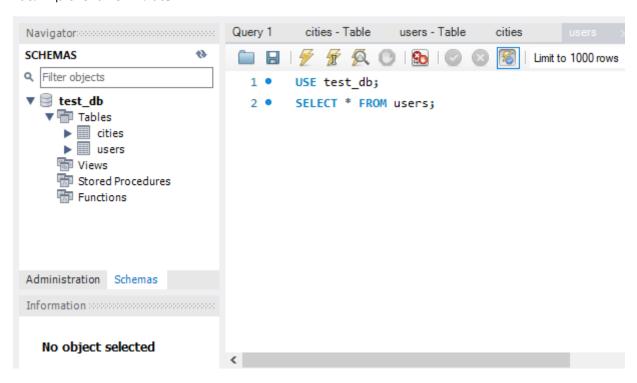
Файл создан. Теперь дропнем одну из таблиц, например users:



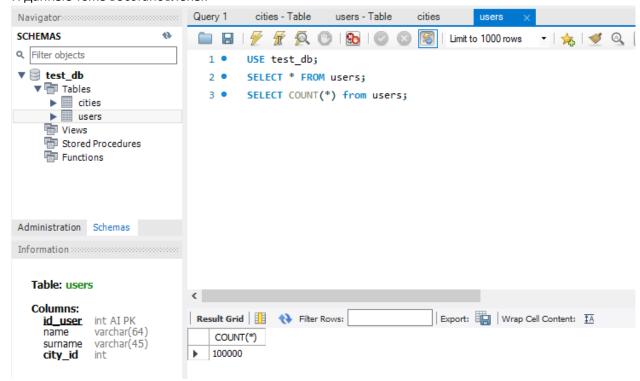
И попробуем восстановить:

```
user@ubuntu-server:~$ mysql -u testuser -p test_db < testdb_backup.sql
Enter password:
user@ubuntu-server:~$ [
```

Таблица снова появилась:



И данные тоже восстановлены:



Настройка автоматического резервного копирования.

Создадим bash-скрипт для резервного копирования. Создадим файл:

echo "Резервная копия \$DATABASE создана в \$BACKUP_FILE.gz"

```
user@ubuntu-server:~$ sudo nano /usr/local/bin/backup mysql.sh
[sudo] password for user:
И вставим код:
#!/bin/bash
# Настройки
USER="root"
                  # Имя пользователя MySQL
PASSWORD="your_password" # Пароль MySQL пользователя
DATABASE="my database"
                         # Имя базы данных
BACKUP DIR="/var/backups/mysql" # Каталог для хранения резервных копий
DATE=$(date +%F-%H-%M-%S) # Текущая дата для уникального имени файла
BACKUP_FILE="$BACKUP_DIR/$DATABASE-$DATE.sql"
# Убедитесь, что директория для резервных копий существует
mkdir -p "$BACKUP DIR"
# Создание резервной копии
mysqldump -u $USER -p$PASSWORD $DATABASE > $BACKUP_FILE
# Сжатие резервной копии для экономии места
gzip $BACKUP_FILE
# Удаление резервных копий старше 7 дней (опционально)
find $BACKUP_DIR -type f -name "*.sql.gz" -mtime +7 -exec rm {} \;
```

```
GNU nano 6.2
                              /usr/local/bin/backup mysql.sh *
 !/bin/bash
# Настройки
USER="root"
                     # Имя пользователя MySQL, тут можно и root
PASSWORD="1111"
                    # Пароль MySQL пользователя
PASSWORD IIII # Naposib Nyogb Nov
BACKUP FILE="$BACKUP DIR/$DATABASE-$DATE.sql"
# Убедитесь, что директория для резервных копий существует
mkdir -p "$BACKUP DIR"
# Создание резервной копии
mysqldump -u $USER -p$PASSWORD $DATABASE > $BACKUP FILE
# Сжатие резервной копии для экономии места
gzip $BACKUP FILE
Удаление резервных копий старше 7 дней (опционально)
find $BACKUP DIR -type f -name "*.sql.gz" -mtime +7 -exec rm {} \;
echo "Резервная копия $DATABASE создана в $BACKUP FILE.gz"
```

Сделаем скрипт исполняемым:

```
user@ubuntu-server:~$ sudo chmod +x /usr/local/bin/backup_mysql.sh
user@ubuntu-server:~$ |
```

Теперь нужно настроить задачу в планировщике **cron**, чтобы резервное копирование выполнялось автоматически, например, каждый день в определённое время. Откроем файл crontab:

```
user@ubuntu-server:~$ sudo crontab -e
```

И добавим в конец файла:

0 2 * * * /usr/local/bin/backup_mysql.sh

```
GNU nano 6.2
                                   /tmp/crontab.mJSIX0/crontab *
 Each task to run has to be defined through a single line
 indicating with different fields when the task will be run
 and what command to run for the task
 To define the time you can provide concrete values for
 minute (m), hour (h), day of month (dom), month (mon),
 and day of week (dow) or use '*' in these fields (for 'any').
 Notice that tasks will be started based on the cron's system
 daemon's notion of time and timezones.
# Output of the crontab jobs (including errors) is sent through
email to the user the crontab file belongs to (unless redirected).
For example, you can run a backup of all your user accounts
 at 5 a.m every week with:
 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
For more information see the manual pages of crontab(5) and cron(8)
# m h dom mon dow command
 2 * * * /usr/local/bin/backup mysql.sh
```

0 2 * * * — означает, что скрипт будет выполняться каждый день в 2:00.

Убедимся, что задача добавлена в cron:

```
user@ubuntu-server:~$ sudo crontab -1
# Edit this file to introduce tasks to be run by cron.
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
 and what command to run for the task
# To define the time you can provide concrete values for
 minute (m), hour (h), day of month (dom), month (mon),
 and day of week (dow) or use '*' in these fields (for 'any').
 Notice that tasks will be started based on the cron's system
 daemon's notion of time and timezones.
 Output of the crontab jobs (including errors) is sent through
 email to the user the crontab file belongs to (unless redirected).
# For example, you can run a backup of all your user accounts
 at 5 a.m every week with:
 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
# For more information see the manual pages of crontab(5) and cron(8)
#mh dom mon dow command
0 2 * * * /usr/local/bin/backup mysql.sh
user@ubuntu-server:~$
```

Для теста запустим скрипт вручную:

```
user@ubuntu-server:~$ sudo /usr/local/bin/backup_mysql.sh
mysqldump: [Warning] Using a password on the command line interface can be insecure.
Pезервная копия test_db создана в /var/backups/mysql/test_db-2024-09-28-10-42-53.sql.gz
user@ubuntu-server:~$ []

user@ubuntu-server:~$ ls -1 /var/backups/mysql
total 1460
-rw-r--r-- l root root 1493822 Sep 28 10:42 test_db-2024-09-28-10-42-53.sql.gz
user@ubuntu-server:~$ []
```

Включение бинарного логирования в MySQL

Редактируем конфигурационный файл MySQL:

```
user@ubuntu-server:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
[sudo] password for user: [
```

В секции [mysqld] нужно добавить или изменить следующие строки:

```
[mysqld]
```

max binlog size = 100M

log-bin = /var/log/mysql/mysql-bin.log # Включает бинарное логирование и задает путь к файлам лога

Максимальный размер файла бинарного лога

```
GNU nano 6.2
                         /etc/mysql/mysql.conf.d/mysqld.cnf *
 Run program with --help to get a list of available options and with
  --print-defaults to see which it would actually understand and use.
 For explanations see
 http://dev.mysql.com/doc/mysql/en/server-system-variables.html
# Here is entries for some specific programs
The following values assume you have at least 32M ram
[mysqld]
  * Basic Settings
log-bin = /var/log/mysql/mysql-bin.log # Включает бинарное логирование и задае
server-id = 1
                                         # Уникальный идентификатор сервера (ну
binlog-format = ROW
                                         # Формат логирования (ROW, STATEMENT,
expire_logs_days = 7
                                         # Указывает, сколько дней хранить бина
max binlog size = 100M
                                         # Максимальный размер файла бинарного
```

Теперь нужно перезапустить MySQL:

```
user@ubuntu-server:~$ sudo systemctl restart mysql
user@ubuntu-server:~$ |
```

И проверить, что бинарное логирование включено:

```
user@ubuntu-server:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.39-0ubuntu0.22.04.1 (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> SHOW VARIABLES LIKE 'log_bin';
   -----+
| Variable_name | Value |
| log_bin | ON |
l row in set (0.03 sec)
mysql>
```

Просмотр активных бинарных логов:

Просмотр содержимого бинарного лога:

```
user@ubuntu-server:~$ sudo mysqlbinlog /var/log/mysql/mysql-bin.000001
# The proper term is pseudo_replica_mode, but we use this compatibility alias
# to make the statement usable on server versions 8.0.24 and older.
/*!50530 SET @@SESSION.PSEUDO SLAVE MODE=1*/;
/*!50003 SET @OLD COMPLETION TYPE=@@COMPLETION TYPE,COMPLETION TYPE=0*/;
DELIMITER /*!*/;
#240928 11:17:35 server id 1 end_log_pos 126 CRC32 0xf5458c85 Start: binlog v
4, server v 8.0.39-0ubuntu0.22.04.1 created 240928 11:17:35 at startup
# Warning: this binlog is either in use or was not closed properly.
ROLLBACK/*!*/;
BINLOG '
z+X3Zg8BAAAAegAAAH4AAAABAAQAOC4wLjM5LTB1YnVudHUwLjIyLjA0LjEAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAADP5fdmEwANAAqAAAAABAAEAAAAYqAEGqqAAAAICAqCAAAAACqoKKioAEjQA
CiaAAYWMRfU=
1/*!*/;
# at 126
#240928 11:17:35 server id 1 end log pos 157 CRC32 0xb5687cl3 Previous-GTIDs
SET @@SESSION.GTID NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
DELIMITER ;
# End of log file
/*!50003 SET COMPLETION TYPE=@OLD COMPLETION TYPE*/;
/*!50530 SET @@SESSION.PSEUDO SLAVE MODE=0*/;
user@ubuntu-server:~$
```

Теперь MySQL сервер настроен на ведение бинарных логов, которые могут быть использованы для восстановления данных, репликации и аудита изменений.