Linux中的mysql原生指令，以及相关备份操作

一、实验介绍

1.内容描述

本实验主要介绍了在Ubuntu操作系统中，mysql的原生指令，以及相关备份操作。系统中mysql的root初始密码为空。

2.实验目的

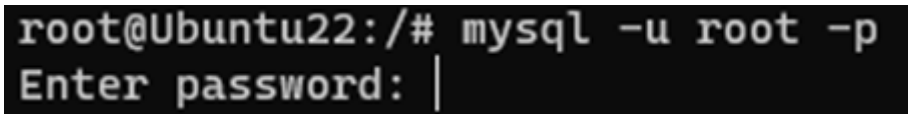
- 掌握mysql的基础原生指令
- 掌握mysql轻量备份
- 掌握mysql主从同步

二、基础原生指令

1. **mysql**：用于启动MySQL命令行客户端，可以连接到MySQL服务器并执行SQL语句。

   ```
   mysql -u username -p
   ```

   其中，`username`是MySQL用户名。执行此命令后，系统会提示输入密码。

2. **mysqladmin**：用于执行管理任务，如创建数据库、删除数据库、管理用户等。

   - 检查MySQL服务器状态：

   ```
   mysqladmin -u username -p status
   ```

   该命令将提示您输入MySQL用户名和密码，然后显示MySQL服务器的当前状态，包括运行时间、连接数、线程信息等。

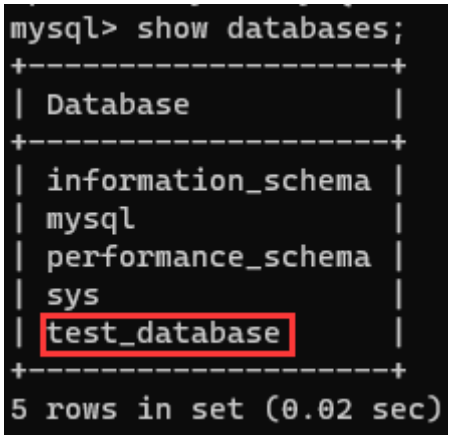
   ```
   root@Ubuntu22:/# mysqladmin -u root -p status
   Enter password:
   Uptime: 11355  Threads: 2  Questions: 7  Slow queries: 0  Opens: 119  Flush tables: 3
     Open tables: 38  Queries per second avg: 0.000
   ```

   - 创建新的数据库：

   ```
   mysqladmin -u username -p create database_name
   ```

   这个命令用于创建一个新的数据库。将`username`替换为具有适当权限的MySQL用户名，`database_name`是要创建的数据库名。

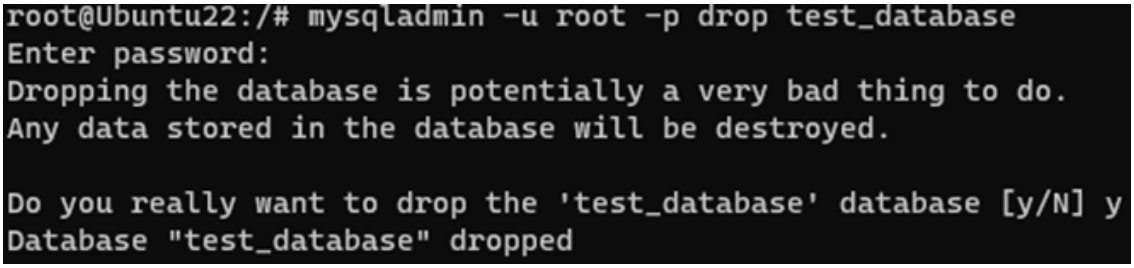
   ```
   root@Ubuntu22:/# mysqladmin -u root -p create test_database
   Enter password:
   ```



- 删除数据库：

```
mysqladmin -u username -p drop database_name
```

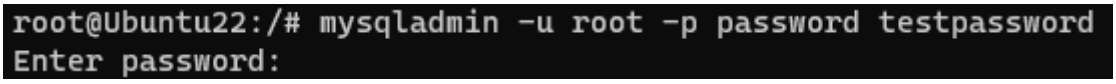
这个命令用于删除一个数据库。将`username`替换为具有适当权限的MySQL用户名，`database_name`是要删除的数据库名。



- 修改MySQL用户密码：

```
mysqladmin -u username -p password new_password
```

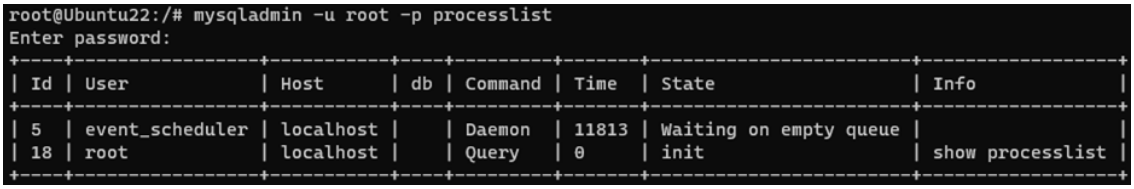
这个命令用于更改MySQL用户的密码。将`username`替换为要修改密码的MySQL用户名，`new_password`是新的密码。



- 查看MySQL服务器进程列表：

```
mysqladmin -u username -p processlist
```

这个命令用于显示当前正在运行的MySQL服务器进程列表。将`username`替换为具有适当权限的MySQL用户名。



3. **mysqldump**：MySQL提供的备份工具，用于导出数据库或表的内容。

- 备份整个数据库：

```
mysqldump -u username -p database_name > backup.sql
```

其中，`username`是MySQL用户名，`database_name`是要备份的数据库名，`backup.sql`是备份文件名。

```
root@Ubuntu22:/# mysqldump -u root -p mysql > backup.sql
Enter password:
```

backup.sql

- 备份单个表：

```
mysqldump -u username -p database_name table_name > backup.sql
```

`table_name`是要备份的表名。

```
root@Ubuntu22:/# mysqldump -u root -p mysql user > mysql_user_backup.sql
Enter password:
```

mysql_user_backup.sql

- 备份时包括创建数据库和表结构：

```
mysqldump -u username -p --no-data database_name > backup.sql
```

```
root@Ubuntu22:/# mysqldump -u root -p --no-data mysql > mysql_nodata_backup.sql
Enter password:
```

mysql_nodata_backup.sql

4. **mysqlbinlog**：MySQL提供的二进制日志解析工具，用于解析和查看二进制日志文件的内容。

- 进入mysql数据文件目录下/var/lib/mysql，可看到日志文件

```
root@Ubuntu22:/# cd /var/lib/mysql
root@Ubuntu22:/var/lib/mysql# ls
 auto.cnf         binlog.000008   client-cert.pem     '#innodb_redo'         server-key.pem
 binlog.000001    binlog.000009   client-key.pem      '#innodb_temp'         sys
 binlog.000002    binlog.000010   debian-5.7.flag      mysql                 test_database
 binlog.000003    binlog.000011  '#ib_16384_0.dblwr'   mysql.ibd             test_database2
 binlog.000004    binlog.000012  '#ib_16384_1.dblwr'   performance_schema    Ubuntu22.pid
 binlog.000005    binlog.index    ib_buffer_pool       private_key.pem       undo_001
 binlog.000006    ca-key.pem      ibdata1              public_key.pem        undo_002
 binlog.000007    ca.pem          ibtmp1               server-cert.pem
```

- 解析并显示二进制日志内容：

```
mysqlbinlog mysql-bin.xxxxxx
```

`mysql-bin.xxxxxx`是二进制日志文件名。

```
root@Ubuntu22:/var/lib/mysql# mysqlbinlog binlog.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 /*!*/;
# at 4
#230619 15:09:01 server id 1  end_log_pos 126 CRC32 0x0e54d572  Start: binlog v 4, server v 8.0
.33-0ubuntu0.22.04.2 created 230619 15:09:01 at startup
ROLLBACK/*!*/;
BINLOG '
Df+PZA8BAAAAegAAAH4AAAAAAAQAOC4wLjMzLTB1YnVudHUwLjIyLjA0LjIAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAN/49kEwANAAgAAAAABAAEAAAAYgAEGggAAAAICAgCAAAACgoKKioAEjQA
CigAAXLVVA4=
'/*!*/;
# at 126
#230619 15:09:01 server id 1  end_log_pos 157 CRC32 0x18f021b7  Previous-GTIDs
# [empty]
# at 157
#230619 15:09:02 server id 1  end_log_pos 180 CRC32 0x862c7098  Stop
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*/;
```

- 将二进制日志内容导出到文件：

```
mysqlbinlog mysql-bin.xxxxxx > output.txt
```

```
root@Ubuntu22:/var/lib/mysql# mysqlbinlog binlog.000001 > output.txt
root@Ubuntu22:/var/lib/mysql# cat output.txt
# 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 /*!*/;
# at 4
#230619 15:09:01 server id 1  end_log_pos 126 CRC32 0x0e54d572  Start: binlog v 4, server v 8.0
.33-0ubuntu0.22.04.2 created 230619 15:09:01 at startup
ROLLBACK/*!*/;
BINLOG '
Df+PZA8BAAAAegAAAH4AAAAAAAQAOC4wLjMzLTB1YnVudHUwLjIyLjA0LjIAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAN/49kEwANAAgAAAAABAAEAAAAYgAEGggAAAAICAgCAAAACgoKKioAEjQA
CigAAXLVVA4=
'/*!*/;
# at 126
#230619 15:09:01 server id 1  end_log_pos 157 CRC32 0x18f021b7  Previous-GTIDs
# [empty]
# at 157
#230619 15:09:02 server id 1  end_log_pos 180 CRC32 0x862c7098  Stop
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*/;
```

- 从指定位置开始解析并显示二进制日志内容：

```
mysqlbinlog --start-position=xxx mysql-bin.xxxxxx
```

`xxx`是要开始解析的位置。

```
root@Ubuntu22:/var/lib/mysql# mysqlbinlog --start-position='230619' binlog.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 /*!*/;
# at 157
#230619 15:09:01 server id 1  end_log_pos 126 CRC32 0x0e54d572  Start: binlog v 4, server v 8.0
.33-0ubuntu0.22.04.2 created 230619 15:09:01 at startup
ROLLBACK/*!*/;
BINLOG '
Df+PZA8BAAAAegAAAH4AAAAAAAQA0C4wLjMzLTB1YnVudHUwLjIyLjA0LjIAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAN/49kEwANAAgAAAAABAAEAAAAYgAEGggAAAAICAgCAAAACgoKKioAEjQA
CigAAXLVVA4=
'/*!*/;
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*/;
```

三、轻量备份

MySQL的轻量备份是指只备份数据库的变更部分，而不是完整备份整个数据库。这种备份方法可以减少备份的时间和存储空间，因为它只关注最新的变更内容。要使用原生指令实现MySQL的轻量备份，可以结合使用`mysqldump`和`mysqlbinlog`命令。

以下是使用原生指令实现轻量备份的步骤：

1. **创建完整备份**：首先，使用`mysqldump`命令创建数据库的完整备份。

   ```
   mysqldump -u username -p --single-transaction --flush-logs --source-data=2
   database_name > backup.sql
   ```

   其中，`username`是MySQL用户名，`database_name`是要备份的数据库名，`backup.sql`是备份文件名。

   该命令使用了`--single-transaction`参数来确保备份是一致性的，并使用`--flush-logs`参数来刷新二进制日志文件。`--source-data=2`参数会在备份文件中添加二进制日志文件的位置信息。

   ```
   root@Ubuntu22:/# mysqldump -u root -p --single-transaction --flush-logs --source-data=2 mysql >
   backup.sql
   Enter password:
   ```

2. **获取二进制日志文件名和位置**：备份完成后，查看备份文件中的二进制日志文件名和位置信息。打开备份文件，搜索类似于下面的内容：

   ```
   CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.xxxxxx', MASTER_LOG_POS=xxx;
   ```

   记下`mysql-bin.xxxxxx`和`xxx`的值，它们表示二进制日志文件的名称和位置。

   ```
   -- CHANGE MASTER TO MASTER_LOG_FILE='binlog.000014', MASTER_LOG_POS=157;
   ```

3. **应用差量备份**：执行差量备份时，将只备份自上次完整备份后的数据库变更部分。
   - 首先，将数据库切换到只读模式，以确保备份期间不会发生数据修改。

     ```
     mysql -u username -p -e "FLUSH TABLES WITH READ LOCK;"
     ```

     ```
     root@Ubuntu22:/# mysql -u root -p -e "FLUSH TABLES WITH READ LOCK;"
     Enter password:
     ```

   - 接下来，使用`mysqlbinlog`命令应用差量备份。

```
mysqlbinlog --no-defaults --start-position=xxx --stop-position=yyy
mysql-bin.xxxxxx | mysql -u username -p
```

将上一步中获取的二进制日志文件名和位置值替换到命令中的`mysql-bin.xxxxxx`、`xxx`和`yyy`。

这个命令会解析并应用自上次完整备份后的二进制日志内容，并将其还原到数据库中。

```
root@Ubuntu22:/var/lib/mysql# mysqlbinlog --no-defaults --start-position='230619' --stop-positi
on='230701' binlog.000014 | mysql -u root -p
Enter password:
```

- 最后，解锁数据库，允许进行数据修改。

```
mysql -u username -p -e "UNLOCK TABLES;"
```

```
root@Ubuntu22:/var/lib/mysql# mysql -u root -p -e "UNLOCK TABLES;"
Enter password:
```

四、主从同步

MySQL的主从同步是一种数据库复制技术，它允许将一个MySQL服务器（主服务器）的数据自动复制到一个或多个其他MySQL服务器（从服务器）。主从同步提供了数据冗余、负载均衡和故障恢复等优势。

要使用原生指令实现MySQL的主从同步，可以按照以下步骤进行设置：

1. **配置主服务器**：
   - 编辑主服务器的配置文件（通常是`my.cnf`），添加以下配置：

```
[mysqld]
server-id=1
log-bin=mysql-bin
binlog-do-db=database_name
```

其中，`server-id`是唯一标识主服务器的ID，`log-bin`启用二进制日志功能，`binlog-do-db`指定要复制的数据库名。

```
root@Ubuntu22:/# cd /etc/mysql/mysql.conf.d/
root@Ubuntu22:/etc/mysql/mysql.conf.d# vim mysqld.cnf
```

在最下方添加：

```
server-id=1
log-bin=mysql-bin
binlog-do-db=test_database2
```

创建数据库test_database2：

```
root@Ubuntu22:/etc/mysql/mysql.conf.d# mysqladmin -u root -p create test_database2
Enter password:
```

   - 重启主服务器以应用配置更改。

```
root@Ubuntu22:/etc/mysql/mysql.conf.d# service mysql restart
```

2. **配置从服务器**：
   - 编辑从服务器的配置文件，添加以下配置：

```
[mysqld]
server-id=2
relay-log=mysql-relay
log-slave-updates=1
```

`server-id`是唯一标识从服务器的ID，`relay-log`定义从服务器的中继日志文件，`log-slave-updates`启用从服务器记录自身的二进制日志。

```
root@Ubuntu22:/# cd /etc/mysql/mysql.conf.d/
root@Ubuntu22:/etc/mysql/mysql.conf.d# vim mysqld.cnf
```

在最下方添加：

```
server-id=2
relay-log=mysql-relay
log-slave-updates=1
```

- 重启从服务器以应用配置更改。

```
root@Ubuntu22:/etc/mysql/mysql.conf.d# service mysql restart
```

3. **连接主从服务器**：
   - 使用`SHOW MASTER STATUS;`命令在主服务器上查找二进制文件和位置：

```
mysql> SHOW MASTER STATUS;
+------------------+----------+---------------+------------------+-------------------+
| File             | Position | Binlog_Do_DB  | Binlog_Ignore_DB | Executed_Gtid_Set |
+------------------+----------+---------------+------------------+-------------------+
| mysql-bin.000001 |      157 | test_database2 |                  |                   |
+------------------+----------+---------------+------------------+-------------------+
1 row in set (0.00 sec)
```

   - 在从服务器上使用`CHANGE MASTER TO`语句连接到主服务器，执行以下命令：

```
mysql -u username -p
CHANGE MASTER TO
MASTER_HOST='master_host',
MASTER_USER='replication_user',
MASTER_PASSWORD='replication_password',
MASTER_PORT=3306,
MASTER_LOG_FILE='mysql-bin.xxxxxx',
MASTER_LOG_POS=xxx;
```

其中，`master_host`是主服务器的主机名或IP地址，`replication_user`和`replication_password`是具有复制权限的MySQL用户的用户名和密码。`mysql-bin.xxxxxx`和`xxx`是主服务器上的二进制日志文件名和位置。

```
mysql> CHANGE MASTER TO
    -> MASTER_HOST='172.20.0.1',
    -> MASTER_USER='root',
    -> MASTER_PASSWORD='',
    -> MASTER_PORT=3306,
    -> MASTER_LOG_FILE='mysql-bin.000001',
    -> MASTER_LOG_POS=157;
Query OK, 0 rows affected, 9 warnings (0.01 sec)
```

   - 启动从服务器的复制进程：

```
START SLAVE;
```

```
mysql> START SLAVE;
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

- 检查从服务器的状态，确保复制进程已启动：

```
SHOW SLAVE STATUS\G
```

```
mysql> START SLAVE;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> SHOW SLAVE STATUS\G
*************************** 1. row ***************************
               Slave_IO_State: Connecting to source
                  Master_Host: 172.20.0.1
                  Master_User: root
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000001
          Read_Master_Log_Pos: 157
               Relay_Log_File: mysql-relay.000001
                Relay_Log_Pos: 4
        Relay_Master_Log_File: mysql-bin.000001
             Slave_IO_Running: Connecting
            Slave_SQL_Running: Yes
```