Odne Rindheim

Assignment 8

Task 1:

I saved the backups on a different partition on the lab computer.

Created partition for backup

Mounted at /run/media/odnerindheim/Backups

Device: /dev/sda3

Created a backup script:

Sudo nano /usr/local/bin/mariadb_backup.sh

```
# Configuration

BACKUP_DIR="/run/media/odnerindheim/Backups/mariadb"

DATE=$(date +"%Y%m%d_%H%M%S")


# Ensure the backup directory exists

mkdir -p "$BACKUP_DIR"


# Retrieve a list of databases, excluding certain system databases

databases=$(mysql -e "SHOW DATABASES;" | grep -Ev
"Database|information_schema|performance_schema")


# Backup each database individually

for db in $databases; do

  echo "Backing up $db"

  mysqldump --databases "$db" --master-data=2 --single-transaction >
"$BACKUP_DIR/${db}_${DATE}.sql"

done


# Flush binary logs

mysqladmin flush-logs


# Find and copy binary logs

binlogs=$(find /var/lib/mysql/ -name "mysql-bin.*")

cp $binlogs "$BACKUP_DIR"


# Compress all backups and binary logs into one tar file

tar -czf "$BACKUP_DIR/backup_${DATE}.tar.gz" -C "$BACKUP_DIR" .
```

Odne Rindheim

```
# Remove individual sql files and binary logs after creating the tar file
find "$BACKUP_DIR" -type f -not -name "backup_${DATE}.tar.gz" -delete
```

made the script executable

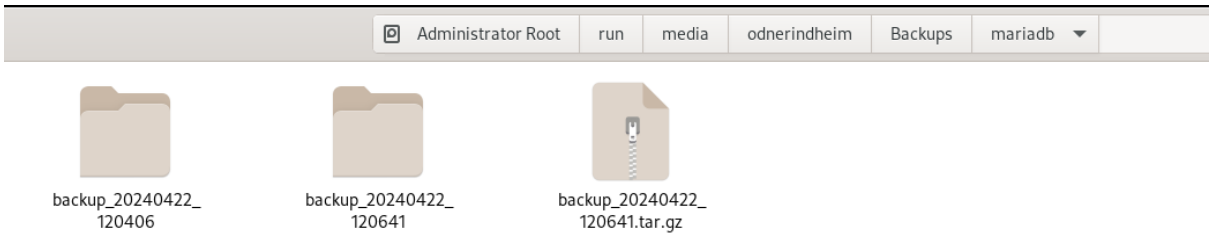
sudo chmod +x /usr/local/bin/mariadb_backup.sh

Tested the backup script:

Sudo /usr/local/bin/mariadb_backup.sh

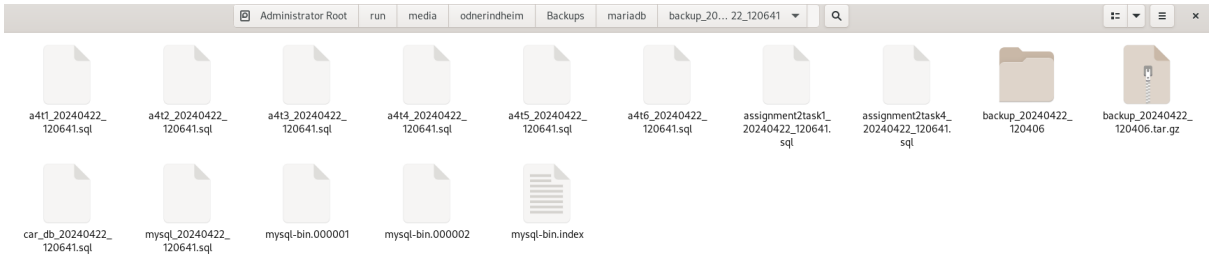
```
[odnerindheim@localhost ~]$ sudo /usr/local/bin/mariadb_backup.sh
Backing up a4t1
Backing up a4t2
Backing up a4t3
Backing up a4t4
Backing up a4t5
Backing up a4t6
Backing up assignment2task1
Backing up assignment2task4
Backing up car_db
Backing up mysql
tar: .: file changed as we read it
[odnerindheim@localhost ~]$
```

Checking the files:



Saved the backup tar, I had already done a backup a couple of minutes ago, and unzipped the newly created backup, that's why there are two folders.

Checked the folder:



Created a scheduled cron job to backup at 2am every day:

Sudo crontab -e

Odne Rindheim

inserted 2 0 * * * /usr/local/bin/mariadb_backup.sh

esc, :wq to write the cron

Task 2:

Took backup of the car_db

Mysql -u root –databases car_db > initial_backup.sql

Signed in with mysql:

Mysql -u odnerindheim -p

Selected car_db from assignemnt 7

Checked tables and count in passing:

```
MariaDB [car_db]> show tables;
+-----------------------+
| Tables_in_car_db      |
+-----------------------+
| Car                   |
| Car_fee_details       |
| Car_with_fees         |
| Car_with_subscription |
| Fee                   |
| Passing               |
| Passing_denorm        |
| Passing_with_owner    |
| Subscription          |
| TaxClass              |
| Tollstation           |
| car_passing_aux       |
+-----------------------+
12 rows in set (0.000 sec)

MariaDB [car_db]> select count(*) from Passing;
+----------+
| count(*) |
+----------+
|  4154196 |
+----------+
1 row in set (0.465 sec)

MariaDB [car_db]>
```

Odne Rindheim

Created a python script to add 1000 rows in Passing table:

```
import mysql.connector

from mysql.connector import Error

import random

from datetime import datetime, timedelta


def generate_timestamp(start_date, end_date):

    # Generates a random datetime between two dates

    delta = end_date - start_date

    random_days = random.randrange(delta.days)

    random_seconds = random.randrange(86400)

    return start_date + timedelta(days=random_days, seconds=random_seconds)


def main():

    try:

        # Establish connection to MariaDB

        conn = mysql.connector.connect(

            host='your_host',

            database='car_db',

            user='your_username',

            password='your_password'

        )

        if conn.is_connected():

            cursor = conn.cursor()


            # Fetch existing registration numbers from CAR table

            cursor.execute("SELECT regno FROM CAR")

            regnos = [row[0] for row in cursor.fetchall()]  # List of available regnos


            # Check if there are enough regnos

            if len(regnos) == 0:

                raise Exception("No registration numbers found in the CAR table. Please add
some first.")


            # Prepare the SQL insert statement

            insert_stmt = "INSERT INTO Passing (timestamp, regno, tollstation_id) VALUES (%s,
%s, %s)"


            # Generate and insert 1000 rows

            start_date = datetime(2023, 1, 1)
```

```
            end_date = datetime(2023, 12, 31)
            for _ in range(1000):
                timestamp = generate_timestamp(start_date, end_date)
                regno = random.choice(regnos)  # Choose a random regno from the list of
existing ones
                tollstation_id = random.randint(1, 29)  # Assuming 29 tollstations are
numbered 1 to 29
                data = (timestamp, regno, tollstation_id)
                cursor.execute(insert_stmt, data)

            # Commit the transactions
            conn.commit()
            print("1000 rows inserted successfully.")

    except Error as e:
        print("Error while connecting to MySQL", e)
    except Exception as e:
        print(e)
    finally:
        if conn.is_connected():
            cursor.close()
            conn.close()
            print("MySQL connection is closed")

if __name__ == "__main__":
    main()
```

Run the script:

Python insert_passing.py:

1000 rows inserted successfully.

Check the count of passing to verify:

Select count(*) from Passing;

Odne Rindheim

```
MariaDB [car_db]> select count(*) from Passing;
+----------+
| count(*) |
+----------+
|  4155196 |
+----------+
1 row in set (0.462 sec)
```

Flushed logs:

FLSUH LOGS;

Dropped table:

DROP TABLE car_db.Passing;

Created passing table again:

And inserted 100 rows using the python script from before, but changed to 100 instead of 1000.

Odne Rindheim

```
MariaDB [car_db]> FLUSH LOGS;
Query OK, 0 rows affected (0.016 sec)

MariaDB [car_db]> DROP TABLE car_db.Passing;
Query OK, 0 rows affected (0.029 sec)

MariaDB [car_db]> CREATE TABLE Passing(
    -> timestamp DATETIME,
    -> regno VARCHAR(15),
    -> tollstation INT,
    -> PRIMARY KEY(timestamp, regno, tollstation),
    -> FOREIGN KEY(regno) REFERENCES Car(regno),
    -> FOREIGN KEY(tollstation) REFERENCES Tollstatio
    -> ;
ERROR 1064 (42000): You have an error in your SQL syn
MariaDB [car_db]> CREATE TABLE Passing (
    ->     timestamp DATETIME,
    ->     regno VARCHAR(15),
    ->     tollstation INT,
    ->     PRIMARY KEY (timestamp, regno, tollstation
    ->     FOREIGN KEY (regno) REFERENCES Car(regno),
    ->     FOREIGN KEY (tollstation) REFERENCES Tolls
    -> );
Query OK, 0 rows affected (0.020 sec)

MariaDB [car_db]> select count(*) from Passing;
+----------+
| count(*) |
+----------+
|      100 |
+----------+
1 row in set (0.000 sec)

MariaDB [car_db]>
```

Performing point-in-time recovery:

Restore from backup:

Mysql -u root < initial_backup.sql

Replace the binary logs:

Odne Rindheim

Mysqlbinlog –stop-position=166170 /var/lib/mysql.binlog.00004 | mysql -u root

I used command sudo mysqlbinlog /var/lib/mysql/mysql-bin.00004 | less, to find the position.

Verified the table:

SELECT COUNT(*) FROM Passing;

```
MariaDB [car_db]> select count(*) from Passing;
+----------+
| count(*) |
+----------+
|  4155296 |
+----------+
1 row in set (0.494 sec)
```

Task 3:

This task was done using 2 Virtual Machines running AlmaLinux9

Installed mariadb and mariadb-server on both machines:

Sudo dnf install mariadb mariadb-server -y

I didn't see the point of creating a normal user for just this task, so everything in dbms is done as root as i didn't have any users set up at the time.

Entered the dbms, created database, and a table:

```
Sudo mysql

CREATE DATABASE ReplicationDB

CREATE TABLE users (

   id INT AUTO_INCREMENT PRIMARY KEY,

   username VARCHAR(50),

   email VARCHAR(100),

   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP

);
```

Created a python script to fill it with 100 rows of data:

```
import mysql.connector

from faker import Faker

import random


# Create a Faker instance

fake = Faker()
```

Odne Rindheim

```
# Database connection parameters
config = {
    'user': 'your_username',
    'password': 'your_password',
    'host': '127.0.0.1',
    'database': 'ReplicationDB',
    'raise_on_warnings': True
}

def insert_random_users(n):
    """Insert n random users into the database."""
    try:
        conn = mysql.connector.connect(**config)
        cursor = conn.cursor()

        for _ in range(n):
            username = fake.user_name()
            email = fake.email()
            # Generate INSERT statement
            insert_query = 'INSERT INTO users (username, email) VALUES (%s, %s)'
            cursor.execute(insert_query, (username, email))

        conn.commit()
        print(f"{n} users added successfully.")

    except mysql.connector.Error as err:
        print("Something went wrong: {}".format(err))
    finally:
        cursor.close()
        conn.close()

# Insert 100 random users
insert_random_users(100)
```

Ran the script:

Python3 script.py

Odne Rindheim

Checked if it was successful:

```
MariaDB [ReplicationDB]> SELECT * from users
    -> ;
+-----+-------------------+-------------------------------+---------------------+
| id  | username          | email                         | created_at          |
+-----+-------------------+-------------------------------+---------------------+
|   1 | kellyclark        | norriskaitlyn@example.org     | 2024-04-21 21:07:20 |
|   2 | dgonzalez         | davissharon@example.net       | 2024-04-21 21:07:20 |
|   3 | alyssarobinson    | philip39@example.net          | 2024-04-21 21:07:20 |
|   4 | michaelsmith      | erica21@example.org           | 2024-04-21 21:07:20 |
```

Edited my.cnf on both machines:

Client:

```
[mysqld]
Server-id=2
Log_bin=client-bin
Binlog_do_db=ReplicationDB
```

Server:

```
[mysqld]
Server-id=1
Log_bin=server-.bind
Binlog_do_db=ReplicationDB
```

Created replication user on both machines:

On server 192.168.0.214

```
CREATE USER 'replicator'@'192.168.0.173' IDENTIFIED BY;
GRANT REPLICATION SLAVE ON *.* TO 'replicator'@'192.168.0.173';
FLUSH PRIVILEGES;
```

On client 192.168.0.173

Odne Rindheim

```
CREATE USER 'replicator'@'192.168.0.214' IDENTIFIED BY;

GRANT REPLICATION SLAVE ON *.* TO 'replicator'@'192.168.0.214';

FLUSH PRIVILEGES;
```

Configured each server to connect to the other as a slave

On server 192.168.0.214

```
CHANGE MASTER TO

MASTER_HOST='192.168.0.173',

MASTER_USER='replicator',

MASTER_PASSWORD='password',

MASTER_LOG_FILE='client-bin.000001',

MASTER_LOG_POS=107;

START SLAVE;
```

On Client 192.168.0.173

```
CHANGE MASTER TO

MASTER_HOST='192.168.0.214',

MASTER_USER='replicator',

MASTER_PASSWORD='password',

MASTER_LOG_FILE='server-bin.000001',

MASTER_LOG_POS=107;

START SLAVE;
```

Verify replication status

SHOW SLAVE STATUS \G

Client:

```
MariaDB [(none)]> SHOW SLAVE STATUS \G

*************************** 1. row ***************************

            Slave_IO_State: Waiting for master to send event

               Master_Host: 192.168.0.214

               Master_User: replicator
```

Odne Rindheim

```
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: server-bin.000001
Read_Master_Log_Pos: 329
Relay_Log_File: mariadb-relay-bin.000002
Relay_Log_Pos: 629
Relay_Master_Log_File: server-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

Server:

```
MariaDB [(none)]> SHOW SLAVE STATUS\G
*************************** 1. row ***************************
Slave_IO_State: Waiting for master to send event
Master_Host: 192.168.0.173
Master_User: replicator
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: client-bin.000001
Read_Master_Log_Pos: 329
Relay_Log_File: mariadb-relay-bin.000002
Relay_Log_Pos: 629
Relay_Master_Log_File: client-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

Try dropping table on server, and check if its also dropped on client:

Server:

```
Drop table users;
```

Odne Rindheim

Select * from users;

MariaDB [ReplicationDB]> SELECT * from users;

ERROR 1146 (42S02): Table 'ReplicationDB.users' doesn't exist

Client:

MariaDB [ReplicationDB]> SELECT * from users;

ERROR 1146 (42S02): Table 'ReplicationDB.users' doesn't exist

Create the table on server, and check if its also created on client:

Server:

```
MariaDB [ReplicationDB]> CREATE TABLE users (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     username VARCHAR(50),
    ->     email VARCHAR(100),
    ->     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    -> );
Query OK, 0 rows affected (0,002 sec)
```

Client:

MariaDB [ReplicationDB]> SELECT * from users;

Empty set (0.000 sec)

Running the python script on client, and checking if the users also appears on server:

Client:

Python3 script.py (Adds 100 users)

```
MariaDB [ReplicationDB]> SELECT * from users;
+-----+------------------+------------------------------+---------------------+
| id  | username         | email                        | created_at          |
+-----+------------------+------------------------------+---------------------+
|   1 | mcdonaldcrystal  | michaelbrooks@example.org    | 2024-04-21 21:43:11 |
```

Odne Rindheim

```
|   2 | sara16          | tferguson@example.com         | 2024-04-21 21:43:11 |
|   3 | ohouston        | katherine96@example.com       | 2024-04-21 21:43:11 |
|   4 | yvettewilson    | gonzalezjillian@example.net   | 2024-04-21 21:43:11 |
|   5 | padillarobert   | sherylcline@example.org       | 2024-04-21 21:43:11 |
|   6 | acampbell       | julie83@example.org           | 2024-04-21 21:43:11 |
|   7 | sarah01         | cruzelaine@example.net        | 2024-04-21 21:43:11 |
|   8 | tatedaniel      | weekscarol@example.org        | 2024-04-21 21:43:11 |
|   9 | andrew84        | patrickstevenson@example.net  | 2024-04-21 21:43:11 |
|  10 | stephanieduncan | emily04@example.net           | 2024-04-21 21:43:11 |
```

Server:

```
MariaDB [ReplicationDB]> select * from users;
+-----+------------------+------------------------------+---------------------+
| id  | username         | email                        | created_at          |
+-----+------------------+------------------------------+---------------------+
|   1 | mcdonaldcrystal  | michaelbrooks@example.org    | 2024-04-21 21:43:11 |
|   2 | sara16          | tferguson@example.com         | 2024-04-21 21:43:11 |
|   3 | ohouston        | katherine96@example.com       | 2024-04-21 21:43:11 |
|   4 | yvettewilson    | gonzalezjillian@example.net   | 2024-04-21 21:43:11 |
|   5 | padillarobert   | sherylcline@example.org       | 2024-04-21 21:43:11 |
|   6 | acampbell       | julie83@example.org           | 2024-04-21 21:43:11 |
|   7 | sarah01         | cruzelaine@example.net        | 2024-04-21 21:43:11 |
|   8 | tatedaniel      | weekscarol@example.org        | 2024-04-21 21:43:11 |
|   9 | andrew84        | patrickstevenson@example.net  | 2024-04-21 21:43:11 |
|  10 | stephanieduncan | emily04@example.net           | 2024-04-21 21:43:11 |
```

Check slave status one more time:

Odne Rindheim

server@h214:~ — sudo mysql

```
+-------------------+-----------+
| server-bin.000001 |      1061 |
+-------------------+-----------+
1 row in set (0,000 sec)

MariaDB [ReplicationDB]> SHOW SLAVE STATUS\G
*************************** 1. row ***************************
                Slave_IO_State: Waiting for master to send event
                   Master_Host: 192.168.0.173
                   Master_User: replicator
                   Master_Port: 3306
                  Connect_Retry: 60
                Master_Log_File: client-bin.000001
            Read_Master_Log_Pos: 39014
                 Relay_Log_File: mariadb-relay-bin.000002
                  Relay_Log_Pos: 39266
        Relay_Master_Log_File: client-bin.000001
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
   Replicate_Wild_Ignore_Table:
                    Last_Errno: 0
                    Last_Error:
                   Skip_Counter: 0
          Exec_Master_Log_Pos: 39014
              Relay_Log_Space: 39577
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: 0
 Master_SSL_Verify_Server_Cert: No
                 Last_IO_Errno: 0
                 Last_IO_Error:
                Last_SQL_Errno: 0
                Last_SQL_Error:
   Replicate_Ignore_Server_Ids:
              Master_Server_Id: 2
                Master_SSL_Crl:
           Master_SSL_Crlpath:
                   Using_Gtid: No
                  Gtid_IO_Pos:
       Replicate_Do_Domain_Ids:
   Replicate_Ignore_Domain_Ids:
                  Parallel_Mode: optimistic
                      SQL_Delay: 0
          SQL_Remaining_Delay: NULL
      Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
              Slave_DDL_Groups: 0
Slave_Non_Transactional_Groups: 0
    Slave_Transactional_Groups: 5
1 row in set (0,000 sec)

MariaDB [ReplicationDB]>
```

client@h173:~ — sudo mysql

client@h173:~ — sudo mysql | client@h173:~ — sudo mysql

```
+-------------------+-----------+
| client-bin.000001 |     39014 |
+-------------------+-----------+
1 row in set (0.000 sec)

MariaDB [ReplicationDB]> SHOW SLAVE STATUS \G
*************************** 1. row ***************************
                Slave_IO_State: Waiting for master to send event
                   Master_Host: 192.168.0.214
                   Master_User: replicator
                   Master_Port: 3306
                  Connect_Retry: 60
                Master_Log_File: server-bin.000001
            Read_Master_Log_Pos: 1061
                 Relay_Log_File: mariadb-relay-bin.000002
                  Relay_Log_Pos: 1361
        Relay_Master_Log_File: server-bin.000001
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
   Replicate_Wild_Ignore_Table:
                    Last_Errno: 0
                    Last_Error:
                   Skip_Counter: 0
          Exec_Master_Log_Pos: 1061
              Relay_Log_Space: 1672
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: 0
 Master_SSL_Verify_Server_Cert: No
                 Last_IO_Errno: 0
                 Last_IO_Error:
                Last_SQL_Errno: 0
                Last_SQL_Error:
   Replicate_Ignore_Server_Ids:
              Master_Server_Id: 1
                Master_SSL_Crl:
           Master_SSL_Crlpath:
                   Using_Gtid: No
                  Gtid_IO_Pos:
       Replicate_Do_Domain_Ids:
   Replicate_Ignore_Domain_Ids:
                  Parallel_Mode: optimistic
                      SQL_Delay: 0
          SQL_Remaining_Delay: NULL
      Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
              Slave_DDL_Groups: 4
Slave_Non_Transactional_Groups: 0
    Slave_Transactional_Groups: 0
1 row in set (0.000 sec)

MariaDB [ReplicationDB]>
```