Incremental Backup in MySQL

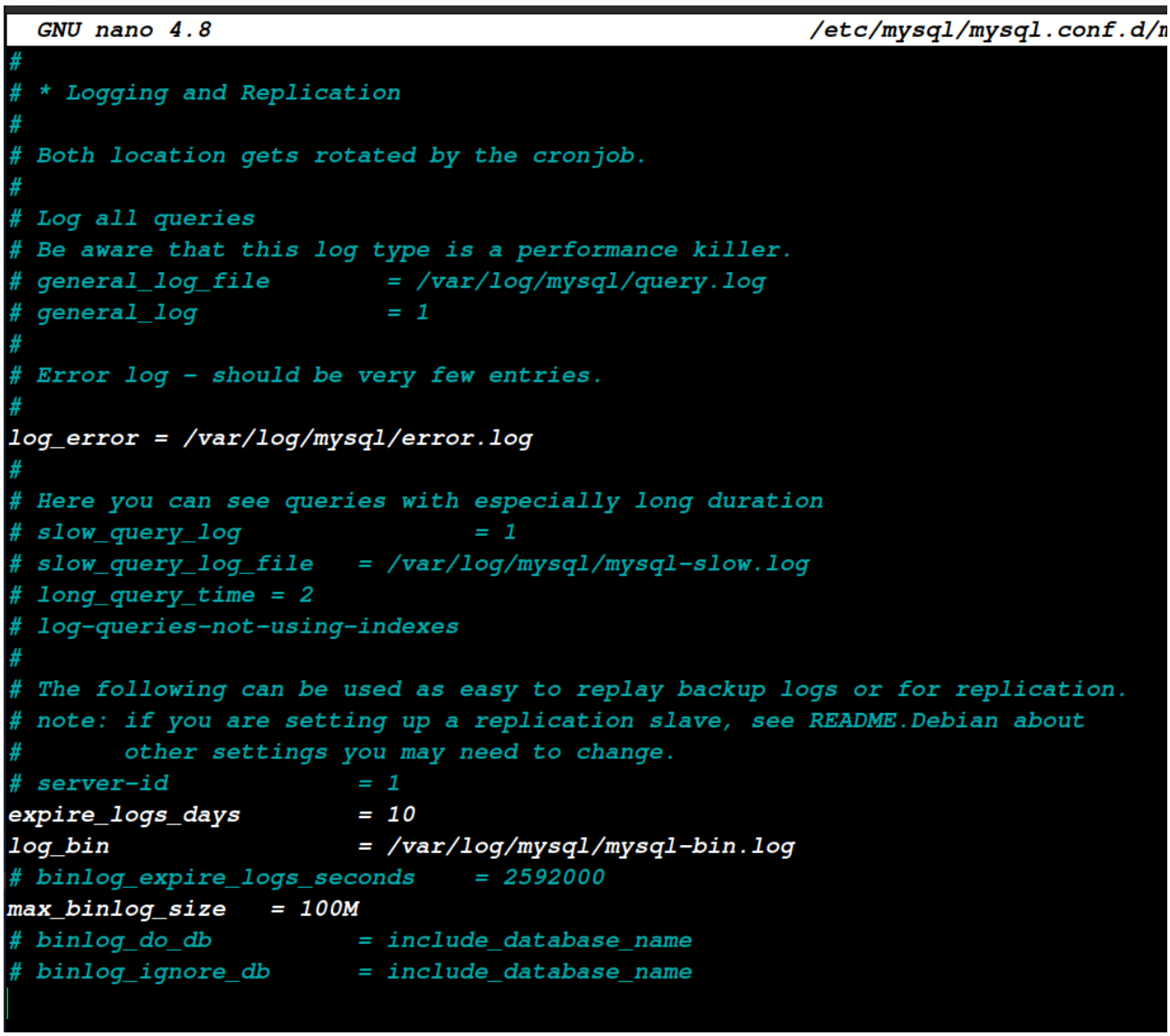
Backing up a MySQL database regularly is an important task of any system administrator to prevent data loss in case of application bugs or system failure. There are two types of backup full and incremental. A full backup contains a collection of all MySQL queries. A full backup is not a recommended solution for a large backup. An incremental backup is a type of backup that only saves data that has been changed or created since the previous backup activity was conducted. incremental backup saves the storage space and uses fewer resources. To setup incremental backup in MySQL we have to follow below steps:-

Step -I : open and modify mysqld.cnf file

nano /etc/mysql/mysql.conf.d/mysqld.cnf

Add or modify the following lines:

```
log_bin                 = /var/log/mysql/mysql-bin.log
expire_logs_days        = 10
```



Save and close the file then restart the MySQL service to apply the changes:

systemctl restart mysql

Now, check the MySQL binary log with the following command:

ls -l /var/log/mysql/

```
bibek  bibek-GF63-Thin-9SC  ~  ls -l /var/log/mysql/
total 116
-rw-r----- 1 mysql adm   19257 May 17 08:53 error.log
-rw-r----- 1 mysql adm    1149 May 10 22:05 error.log.1.gz
-rw-r----- 1 mysql adm     540 May  7 10:47 error.log.3.gz
-rw-r----- 1 mysql adm    2032 May  6 21:21 error.log.4.gz
-rw-r----- 1 mysql mysql  1644 May 11 12:23 mysql-bin.000001
-rw-r----- 1 mysql mysql  1098 May 11 12:28 mysql-bin.000002
-rw-r----- 1 mysql mysql  1825 May 11 12:54 mysql-bin.000003
-rw-r----- 1 mysql mysql  2905 May 11 13:05 mysql-bin.000004
-rw-r----- 1 mysql mysql  4750 May 11 15:52 mysql-bin.000005
-rw-r----- 1 mysql mysql   180 May 11 17:54 mysql-bin.000006
-rw-r----- 1 mysql mysql   180 May 12 20:58 mysql-bin.000007
-rw-r----- 1 mysql mysql   180 May 13 11:47 mysql-bin.000008
-rw-r----- 1 mysql mysql   180 May 13 13:09 mysql-bin.000009
-rw-r----- 1 mysql mysql   180 May 13 18:23 mysql-bin.000010
-rw-r----- 1 mysql mysql   180 May 14 01:43 mysql-bin.000011
-rw-r----- 1 mysql mysql  1825 May 17 07:38 mysql-bin.000012
-rw-r----- 1 mysql mysql 12115 May 17 08:13 mysql-bin.000013
-rw-r----- 1 mysql mysql  1098 May 17 08:14 mysql-bin.000014
-rw-r----- 1 mysql mysql  1423 May 17 08:19 mysql-bin.000015
-rw-r----- 1 mysql mysql   157 May 17 08:53 mysql-bin.000016
-rw-r----- 1 mysql mysql   512 May 17 08:53 mysql-bin.index
```

As you can see, mysql-bin.000016 is a MySQL latest binary log file. All changes in all MySQL databases will be stored in this file.

Step -II :- Create a Database and Table

Next, we will create a test database, table and insert some rows in the table.

First, connect to MySQL with the following command:

---

mysql

Once you are connected, create a database name mydb with the following command:

```
------------------------------*------------------------*--------------------*-------------
mysql> CREATE DATABASE mydb;
------------------------------*------------------------*--------------------*-------------
```

Next, change the database to mydb and create a new table named my_tbl:

```
------------------------------*------------------------*--------------------*-------------
mysql> USE mydb;
mysql> create table my_tbl(
my_id INT NOT NULL AUTO_INCREMENT,
my_field VARCHAR(100) NOT NULL,
submission_date DATE,
time_created TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY ( my_id )
);
------------------------------*------------------------*--------------------*---------------
```

Next, insert some rows with the following command:

```
------------------------------*------------------------*--------------------*----------------
mysql> INSERT into my_tbl (my_field) VALUES ('val1');
mysql> INSERT into my_tbl (my_field) VALUES ('val2');
mysql> INSERT into my_tbl (my_field) VALUES ('val3');
```

Next, exit from the MySQL shell:

```
mysql> exit;

------------------------------*------------------------*--------------------*----------------
```

```
mysql> CREATE DATABASE mydb;
Query OK, 1 row affected (0.03 sec)

mysql>  USE mydb;
Database changed
mysql> create table my_tbl(
    -> my_id INT NOT NULL AUTO_INCREMENT,
    -> my_field VARCHAR(100) NOT NULL,
    -> submission_date DATE,
    -> time_created TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    -> PRIMARY KEY ( my_id )
    -> );
Query OK, 0 rows affected (0.08 sec)

mysql> INSERT into my_tbl (my_field) VALUES ('val1');
Query OK, 1 row affected (0.03 sec)

mysql> INSERT into my_tbl (my_field) VALUES ('val2');
Query OK, 1 row affected (0.03 sec)

mysql> INSERT into my_tbl (my_field) VALUES ('val3');
Query OK, 1 row affected (0.02 sec)

mysql> exit
Bye
```

Step III :- Take a Full MySQL Database Backup

---

mysqldump -uroot -p mydb --flush-logs > full.sql

---

where,

--flush-logs will close current logs (mysql-bin.000016) and create a new one (mysql-bin.000017).

```
bibek  bibek-GF63-Thin-9SC  ~  mysqldump -uroot -p mydb --flush-logs > full.sql

bibek  bibek-GF63-Thin-9SC  ~  ls
anaconda3   crontab_test  Desktop    Downloads  Music     Public  tds_fdw-2.0.2  tmp            Videos
a.sql       dead.letter   Documents  full.sql   Pictures  snap    Templates      v2.0.2.tar.gz

bibek  bibek-GF63-Thin-9SC  ~
```

You can check the new MySQL binary log file with the following command:

---

ls -l /var/log/mysql/

```
bibek  bibek-GF63-Thin-9SC  ~  ls -l /var/log/mysql/
total 120
-rw-r----- 1 mysql adm   19257 May 17 08:53 error.log
-rw-r----- 1 mysql adm    1149 May 10 22:05 error.log.1.gz
-rw-r----- 1 mysql adm     540 May  7 10:47 error.log.3.gz
-rw-r----- 1 mysql adm    2032 May  6 21:21 error.log.4.gz
-rw-r----- 1 mysql mysql  1644 May 11 12:23 mysql-bin.000001
-rw-r----- 1 mysql mysql  1098 May 11 12:28 mysql-bin.000002
-rw-r----- 1 mysql mysql  1825 May 11 12:54 mysql-bin.000003
-rw-r----- 1 mysql mysql  2905 May 11 13:05 mysql-bin.000004
-rw-r----- 1 mysql mysql  4750 May 11 15:52 mysql-bin.000005
-rw-r----- 1 mysql mysql   180 May 11 17:54 mysql-bin.000006
-rw-r----- 1 mysql mysql   180 May 12 20:58 mysql-bin.000007
-rw-r----- 1 mysql mysql   180 May 13 11:47 mysql-bin.000008
-rw-r----- 1 mysql mysql   180 May 13 13:09 mysql-bin.000009
-rw-r----- 1 mysql mysql   180 May 13 18:23 mysql-bin.000010
-rw-r----- 1 mysql mysql   180 May 14 01:43 mysql-bin.000011
-rw-r----- 1 mysql mysql  1825 May 17 07:38 mysql-bin.000012
-rw-r----- 1 mysql mysql 12115 May 17 08:13 mysql-bin.000013
-rw-r----- 1 mysql mysql  1098 May 17 08:14 mysql-bin.000014
-rw-r----- 1 mysql mysql  1423 May 17 08:19 mysql-bin.000015
-rw-r----- 1 mysql mysql  1825 May 17 09:50 mysql-bin.000016
-rw-r----- 1 mysql mysql   157 May 17 09:50 mysql-bin.000017
-rw-r----- 1 mysql mysql   544 May 17 09:50 mysql-bin.index
```

Now, all database changes will be written in **mysql-bin.000017** file.

Next, login to MySQL again and insert more rows:

```
mysql
mysql> USE mydb;
mysql> INSERT into my_tbl (my_field) VALUES ('val4');
mysql> INSERT into my_tbl (my_field) VALUES ('val5');
mysql> INSERT into my_tbl (my_field) VALUES ('val6');
mysql> exit;
```

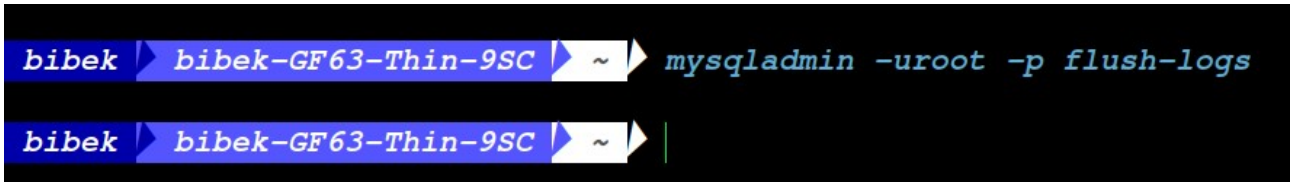
At this point, we have new database changes saved in the file **mysql-bin.000017** after the full backup.

Step IV :- Take an Incremental MySQL Backup

In order to take an incremental backup. You will need to flush the binary log again and save binary logs created from the last full backup.

To flush the binary log, run the following command:

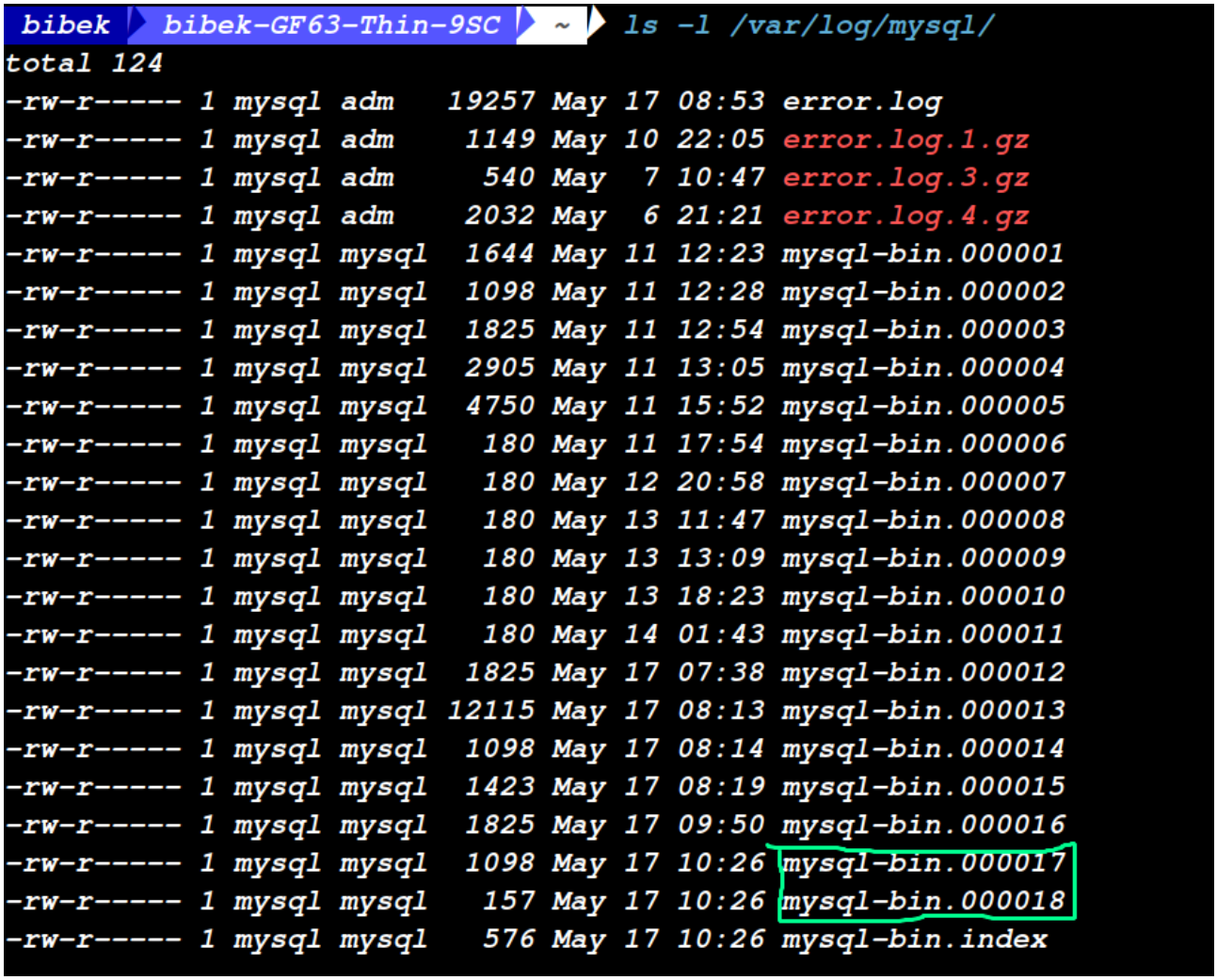
```
mysqladmin -uroot -p flush-logs
```



This will close the **mysql-bin.000017** file and create a new one. You can check it with the following command:

```
------------------------------*------------------------*--------------------*-----------------
ls -l /var/log/mysql/
------------------------------*------------------------*--------------------*-----------------
```

You should see the following output:



Step V :- Delete a MySQL Database Table Records and Restore it From Incremental Backup

Before Delete

```
Database changed
mysql> select * from my_tbl;
+-------+----------+-----------------+---------------------+
| my_id | my_field | submission_date | time_created        |
+-------+----------+-----------------+---------------------+
|     1 | val1     | NULL            | 2022-05-17 09:39:48 |
|     2 | val2     | NULL            | 2022-05-17 09:39:54 |
|     3 | val3     | NULL            | 2022-05-17 09:39:59 |
|     4 | val4     | NULL            | 2022-05-17 10:16:58 |
|     5 | val5     | NULL            | 2022-05-17 10:17:02 |
|     6 | val6     | NULL            | 2022-05-17 10:17:07 |
+-------+----------+-----------------+---------------------+
6 rows in set (0.00 sec)
```

Now, Let’s Delete latest inserted records.

---*---*---*---

mysql

use database mydb;

delete from my_tbl where my_id > 3;

---*---*---*---

After Delete

```
mysql> select * from my_tbl;
+-------+----------+-----------------+---------------------+
| my_id | my_field | submission_date | time_created        |
+-------+----------+-----------------+---------------------+
|     1 | val1     | NULL            | 2022-05-17 09:39:48 |
|     2 | val2     | NULL            | 2022-05-17 09:39:54 |
|     3 | val3     | NULL            | 2022-05-17 09:39:59 |
+-------+----------+-----------------+---------------------+
3 rows in set (0.01 sec)
```

Now Let’s Restore form Binary Log files.

---*---*---*---

sudo mysqlbinlog /var/log/mysql/mysql-bin.000017 | mysql -uroot -p mydb

---*---*---*---

Step IV :- Login to MySQL again and Verify

You should see that all rows are restored:

```
mysql> use mydb;
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 * from my_tbl;
+-------+----------+-----------------+---------------------+
| my_id | my_field | submission_date | time_created        |
+-------+----------+-----------------+---------------------+
|     1 | val1     | NULL            | 2022-05-17 09:39:48 |
|     2 | val2     | NULL            | 2022-05-17 09:39:54 |
|     3 | val3     | NULL            | 2022-05-17 09:39:59 |
|     4 | val4     | NULL            | 2022-05-17 10:16:58 |
|     5 | val5     | NULL            | 2022-05-17 10:17:02 |
|     6 | val6     | NULL            | 2022-05-17 10:17:07 |
+-------+----------+-----------------+---------------------+
6 rows in set (0.01 sec)
```