Lab solutions 02-16

1

Made bashrc function that spins up new instance with given name. Used this function to spin up new instance with name "backup".

Generated key in "db1" with `ssh-keygen`. Manually copied generated public key from db1 to authorized keys in backup.

Check that it works -> it does!

2

> Useful commands from Kyrre: `sfdisk -l`, `date`, `mkfs`, formattere disk: `<filsystem> /dev/vdb`

- Made volume "backupVolume" with size 5G and attached it to backup instance (device `/dev/backups`).
- formatted backup to mkfs.ext4 vdb with `mkfs.ext4 /dev/mkfs.ext4`
- Made backups directory and mounted with `mount /dev/vdb /backups`

TODO: Mount on startup

3

Stuff needed for a "full" backup of mysql database:

mysqldump (`mysqldump --opt --master-data=2 --flush-logs --all-databases > backup.sql`)

```
1  /var/log/syslog
2  /var/log/mysql/*
3  /etc/logrotate.d/mysql-server
```

4

Script can be found in this directory with the name "db1_fullBackup.sh"

5

1. What files are being rotated: `/var/log/mysql/mysql.log` `/var/log/mysql/mysql-slow.log` `/var/log/mysql/mariadb-slow.log` `/var/log/mysql/error.log`
2. When are the files rotated: Once a week. Daily backup is taken, and 7 archieved logs are kept.
3. How many rotated files we have: 7

6

TBA

7

As per right now we can't because we're missing keys going that way. If we added the key, we would also have to add a database user with the right privileges.

8

Taking backup directly from backup instance

Advantages:

- With multiple database-servers, the script will only live on one server
- The backup server decided when it wants to take backup

Disadvantage:

- Encryption not for free
- Database might not be ready for backup

9

The command for starting a new binary log file is `mysqladmin flush-logs`

10

From: https://dev.mysql.com/doc/refman/5.7/en/flush.html

If binary logging is enabled, the sequence number of the binary log file is incremented by one relative to the previous file. If relay logging is enabled, the sequence number of the relay log file is incremented by one relative to the previous file.

11

Do not stop at the end of requested binary log from a MySQL server, but rather continue printing to end of last binary log

12

```
mysqlbinlog --start-datetime="2018-02-14 09:00:00"--stop-datetime="2018-02-15
 23:59:59"
```

13

The way we interpreted the task was that we were supposed to show that we could figure out the exact commands that were run at a position in the binlogs. We chose position 32547436. The commands that were run were:

```
1  SET @@session.foreign_key_checks=1, @@session.unique_checks=1/*!*/;
2  SET @@session.sql_mode=1411383296/*!*/;
3  /*!\C utf8 *//*!*/;
4  SET @@session.character_set_client=33,@@session.collation_connection
      =33,@@session.collation_server=8/*!*
5  /;
6  GRANT SELECT, UPDATE, INSERT, CREATE, DROP, DELETE ON bookfacedb.* TO '
      harry'@'10.10.%' IDENTIFIED BY 'k
7  aboom'
8  /*!*/;
```

(The whole binlog entry:)

```
1  # at 32547436
2  #180316 12:32:58 server id 1  end_log_pos 32547625 CRC32 0xaf7a3b10
        Query   thread_id=16    exec_tim
```

```
 3  e=0      error_code=0
 4  SET TIMESTAMP=1521203578/*!*/;
 5  SET @@session.foreign_key_checks=1, @@session.unique_checks=1/*!*/;
 6  SET @@session.sql_mode=1411383296/*!*/;
 7  /*!\C utf8 *//*!*/;
 8  SET @@session.character_set_client=33,@@session.collation_connection
       =33,@@session.collation_server=8/*!*
 9  /;
10  GRANT SELECT, UPDATE, INSERT, CREATE, DROP, DELETE ON bookfacedb.* TO '
       harry'@'10.10.%' IDENTIFIED BY 'k
11  aboom'
12  /*!*/;
```

14

Backup Policy:

- A full backup is taken every monday midnight.
- Incremental backups are taken every days at midnight, except monday midnights.
- If the bandwidth is too low at monday midnights, an incremental backup is taken instead of a full backup.

15

In case someone does `delete * from posts` by accident:

- Copy latest mysqldump to db1 from backups
- Execute the commands in mysqldump on the database

(Could we have used bin-log files here? Sure, but we're not confident enough with them to trust that approach)

In case we need to insert everything on a new server because the old one broke:

- Make new database server using our osnew bash function
- Follow instructions from appropriate lab to properly install MariaDB
- Copy latest mysqldump to new database instance from backups
- Execute the commands in mysqldump on the database

Concatenated them all to a file with the command: `sudo cat mariadb-bin.* | sudo tee megafile > /dev/`**`null`**

Tried opening the new megafile with `mysqlbinlog megafile`. Contains this:

```
 1  /*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
 2  /*!40019 SET @@session.max_insert_delayed_threads=0*/;
 3  /*!50003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/
       ;
 4  DELIMITER /*!*/;
 5  # at 4
 6  #180316 11:31:18 server id 1  end_log_pos 256 CRC32 0x7f6e06c3  Start:
       binlog v 4, server v 10.2.13-MariaDB-10.2.13+maria~xenial-log
       created 180316 11:31:18 at startup
 7  ROLLBACK/*!*/;
 8  BINLOG '
 9  BqurWg8BAAAA/AAAAABAAAAAAQAMTAuMi4xMy1NYXJpYURCLTEwLjIuMTMrbWFyaWF+
       eGVuaWFs
10  LWxvZwAAAAAAAAAAAAGq6taEzgNAAgAEgAEBAQEEgAA5AAEGggAAAAICAgCAAAACgoKAAAAAAAA

11  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

12  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

13  AAAAAAAAAAEEwQADQgICAoKCgHDBm5/
14  '/*!*/;
15  # at 256
16  #180316 11:31:18 server id 1  end_log_pos 285 CRC32 0x37b97081  Gtid
       list []
17  # at 285
18  #180316 11:31:18 server id 1  end_log_pos 330 CRC32 0xb69d1942  Binlog
       checkpoint mariadb-bin.000001
19  # at 330
20  #180316 11:33:57 server id 1  end_log_pos 353 CRC32 0xfc6abcdf  Stop
21  ERROR: Error in Log_event::read_log_event(): 'Event invalid', data_len:
       1, event_type: 190
22  ERROR: Could not read entry at offset 353: Error in log format or read
       error.
23  DELIMITER ;
24  # End of log file
25  ROLLBACK /* added by mysqlbinlog */;
```

```
26  /*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
27  /*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;
```

Our conclusion is that this did not work. Why? We don't know, but we're guessing it has to do with the format of the bin-log files, as that's what the output is pointing towards.