MySQL Workshop 2

Overheads

Haim Tzadok

GRIGALE
www.grigale.com

Safe Usage

The material provided is not free of errors and should be treated as such.
If you find any error, please email me to haim.tzadok@grigale.com and I will try my best to correct any error.

Before using this material, always test and test again, before implementing !
The material may be subject to changes by MySQL and should be treated as a recommendation only.

Table Of Content

Session 1

Table Of Content

Session 2

Session 2

MySQL Monitoring

- Monitoring Tools
- Monitoring using benchmark
- MySQL Enterprise Monitor[MEM] – Installation brief
- Using MEM

Backup & Recovery

- MySQL backup methods
- MySQL Logical backup
- MySQL Physical backup
- MySQL Enterprise backup [MEB]
- Restoring using MEB
- Incremental and Differential backup using MEB

MySQL Advanced Security

- MySQL Firewall
- MySQL Auditing

MySQL Performance monitoring and tuning

- OS Performance tuning
- MySQL Thread handling
- MySQL Performance schema
- MySQL Optimization
- MySQL Query Cache
- Benchmrking using sysbench

MySQL Replication

- Replication methods
- Statement Based replication [SBR]
- Raw based replication [RBR]
- Replication chaining
- Semi-Synchronous replication
- Multi-Source replication
- Replication using binlog
- Replication using GTID's
- InnoDB Cluster

"The wolrd's most popular open source database"

Course Introduction

Course Time Frame

Training hours: 9:30-17:30

09:30- Start
10:45 – Short break (5 Min)
12:15 – Lunch(45 Min)
13:30 – Short break (5 Min)
15:00 – Long break (15 Min)
17:30 – Finish

MySQL Monitoring

GRIGALE
www.grigale.com
Grigale Ltd T +972 3 938 1169 F +972 3 938 0070
Lev Ha'aretz industrial zone, Rosh Ha'ayin Israel

MySQL Monitoring

- MySQL Monitoring Tools
- Monitoring MySQL using WorkBench
- Monitoring MySQL Server using CLI
- MySQL Enterprise Monitor – DEMO

MySQL Monitoring

MySQL Monitoring Tools

- MySQL Monitoring Tools
  - MySQL Logs
    - **Error log**: /var/log/mysqld.log – the first place to check if error occurs.
    - **General log**: When enabled contains connect/disconnect events, all issued queries. Should be used with care may slow the server by 15-20%.
    - **Slow query log**: When enabled, contains queries running longer than a threshold (long_query_time in seconds) $datadir/slow.log
  - 3rd party utilities: innotop, mytop, monyog, mysqltuner, sysbench

MySQL Monitoring

Monitoring MySQL using WorkBench

- Monitoring MySQL using WorkBench
- DEMO

MySQL Monitoring

- MySQL Enterprise Monitor – DEMO

MySQL Backup & Recovery

GRIGALE
www.grigale.com
Grigale Ltd T +972 3 938 1169 F +972 3 938 0070
Lev Ha'aretz industrial zone, Rosh Ha'ayin Israel

MySQL Backup and Restore

- MySQL Backup methods
- MySQL logical backup&restore
- MySQL physical backup&restore
- MySQL Backup and Restore using MEB

MySQL Backup and Restore

MySQL Backup methods

- Logical Backup
  - Storage engine independent
  - Architecture independent
  - Textual backup, easy to manipulate
  - Corruption is easily detected.
  - Can be very slow…
- Physical(raw) Backup
  - Consist of raw copies of directories and files
  - Suitable for large, important databases
  - Usually very fast…

MySQL Backup using mysqldump

MySQL logical backup&restore

- MySQL logical backup
  - Contains all sql statements needed to create db (with options --all-databases|--databases) ,create table and insert data.
- **Backup entire mysql:**
  - mysqldump --all-databases > mysql-dump-`date +%d.%m.%y`.sql
- **Backup specific databases**
  - mysqldump --databases db1 db2 db3 > mysql-dump-`date +%d.%m.%y`.sql
- **Backup specific tables from single database**
  - mysqldump db1 t1 t2 t3 >mysql-dump-`date +%d.%m.%y`.sql

MySQL Full Restore using mysqldump

MySQL logical backup&restore

- MySQL logical restore
  - Contains all sql statements needed to create db (if backup was with options --all-databases|--databases) ,create table and insert data.
- Full restore from full dump:
  - Create an empty /var/lib/mysql directory (datadir)
  - Empty the error log file echo > /var/log/mysqld.log
  - Initialize the database by starting the service.
  - Locate the temporary password
    - (grep temporary /var/log/mysqld.log)
  - Change the temporary password or root@localhost
    - mysql -u root –p<TMPPASS>
    - mysql> alter user root@'localhost' identified by 'MySQL5.7';
    - mysql> flush privileges;
  - Restory from the dump file
    - mysql –u root –pxxxxx < mysql-dump-`date +%d.%m.%y`.sql

    or
    - mysql> source mysql-dump-`date +%d.%m.%y`.sql

MySQL Restore using mysqldump

MySQL logical backup&restore

- MySQL logical restore
  - Contains all sql statements needed to create db (if backup was with options --all-databases|--databases) ,create table and insert data.
- Restore from the shell:
  - mysql < mysql-dump-`date +%d.%m.%y`.sql
- Restore within mysql shell:
  - mysql> source mysql-dump-`date +%d.%m.%y`.sql
- Restore when dump does not create db:
  - CREATE DATABASE IF NOT EXISTS db1;
  - USE db1;
  - source mysql-dump-`date +%d.%m.%y`.sql

MySQL Backup and Restore

MySQL physical backup

- MySQL physical backup
  - MySQL Enterprise Backup (Hot Backup)
  - File-system commands cp/tar/zip (Cold Backup)
  - ZFS/NetApp snapshots (Hot Backup)

MySQL Enterprise Backup

- MySQL Enterprise Backup is a RPM package
- MEB 3.x - Installed in /opt/mysql/meb-3.x for backup of 5.6 and bellow versions.
- MEB 4.x - Installed in /opt/mysql/meb-4.x for backup of 5.7 and above versions.
- Integrated in MySQL Enterprise Workbench.
- Allow: Hot backup, Full/Diffrential/Incremental backup
- Allow: Partial backup.

Command: /opt/mysql/meb-4.0/bin/mysqlbackup

Backup using MEB

Full Backup

```
/opt/mysql/meb-4.0/bin/mysqlbackup --defaults-file=/etc/my.cnf --host=localhost --port=3306 --socket=/u01/mysql/mysql01/mysql.sock --user=root --password=XXXXX --backup-dir=/u01/backup/mysql/`date +%d.%m.%y`.FULL --compress  --backup-image=mysql-backup.img --slave-info backup-to-image
```

Incremental backup

```
/opt/mysql/meb-4.0/bin/mysqlbackup --defaults-file=/etc/my.cnf --host=localhost --port=3306 --socket=/u01/mysql/mysql01/mysql.sock --user=root --password=XXXXX --backup-dir=/u01/backup/mysql --compress --with-timestamp --backup-image=mysql-backup.img --slave-info --incremental --incremental-base=history:last_backup backup-to-image
```

Differential backup

```
/opt/mysql/meb-4.0/bin/mysqlbackup --defaults-file=/etc/my.cnf --host=localhost --port=3306 --socket=/u01/mysql/mysql01/mysql.sock --user=root --password=XXXXX --backup-dir=/u01/backup/mysql --compress --with-timestamp --backup-image=mysql-backup.img --slave-info --incremental --incremental-base-dir=/var/backup/mysql-`date+%d.%m.%Y`.FULL   backup-to-image
```

Restore using MEB

Restore

/opt/mysql/meb-4.0/bin/mysqlbackup --defaults-file=/etc/my.cnf --host=localhost --port=3306 --socket=/apps/mysql/mysql01/mysql.sock --user=root --password=XXXXXX --uncompress --backup-dir=/root --backup-image=mysql-backup.img copy-back-and-apply-log

After the restore completed, we should perform the following:

- Restore ownerships by typing:
  - **chown -R mysql:mysql /u01/mysql/mysql01 /u01/redo**
- Delete the old mysql error log file by typing:
  - **rm /u01/mysql/mysql01/log/mysql.log**
- Perform SELinux security commands on /apps/mysql directory by typing:
  - **semanage fcontext -a -t mysqld_db_t "/u01/mysql(/.*)?"**
  - **restorecon -Rv /u01/mysql**
-

MySQL Enterprise Backup

DEMO

MySQL Advanced Security

GRIGALE
www.grigale.com
Grigale Ltd T +972 3 938 1169 F +972 3 938 0070
Lev Ha'aretz industrial zone, Rosh Ha'ayin Israel

MySQL Advanced Security

- MySQL Firewall
- MySQL Auditing
- MySQL encryption

MySQL Enterprise Firewall

- MySQL Firewall consists of several plugins and tables in information_schema and mysql db.
- MySQL Firewall is enabled globally but it will not work till it is implemented on users.
- Firewall mode on users –
  - **OFF** – Firewall is not implemented on the user.
  - **RECORDING** – Generates white list for the specific user which is being recorded.
  - **DETECTING** - detect and log any transaction which is not in the white list.
  - **PROTECTING** - enables the firewall to work in enforcing mode. (any statement that is not in the white list will be denied).

MySQL Firewall

- **MySQL Firewall installation**
  - mysql -u root -p mysql < /usr/share/mysql/linux_install_firewall.sql
- Enable Firewall mode staticlly -

  [mysqld]

  mysql_firewall_mode=ON

  mysql_firewall_trace=ON
- Enable Firewall mode dynamically -
  - SET GLOBAL mysql_firewall_mode=ON;
  - SET GLOBAL mysql_firewall_trace=ON;
- Disable Firewall immediately –
  - SET GLOBAL mysql_firewall_mode=OFF;

MySQL Firewall

Firewall is enabled per user base.

- **RECORDING mode –**
  - Generate white list for the specific user in a learning period.
  - If the server was rebooted in this mode all the white list that wasn't persistent will be deleted.
- **DETECTING mode** –
  - After finishing the learning period we can switch the firewall to detection mode where it will warn when the user is not obeying the white list rules.
- **PROTECTING mode** –
  - After finishing the learning period we can switch the firewall to protecting mode where it will block queries when the user is not obeying the white list rules.
- Reset Firewall mode on a user to be OFF –
  - Reset the user not to be filtered by the Firewall.

MySQL Firewall

- Implement Firewall on a user in RECORDING mode –
  - **CALL mysql.sp_set_firewall_mode('fwuser@localhost', 'RECORDING');**
- Implement Firewall on a user in DETECTING mode –
  - **CALL mysql.sp_set_firewall_mode('fwuser@localhost', 'DETECTING');**
- Implement Firewall on a user in PROTECTING mode –
  - **CALL mysql.sp_set_firewall_mode('fwuser@localhost', 'PROTECTING');**
- Reset Firewall mode on a user to be OFF –
  - **CALL mysql.sp_set_firewall_mode('fwuser@localhost', 'RESET');**

MySQL Enterprise Auditing

MySQL Auditing

MySQL Enterprise Audit provides an easy to use, policy-based auditing solution that helps organizations implement stronger security controls and satisfy regulatory compliance.

As more sensitive data is collected, stored and used online, database auditing becomes an essential component of any security strategy. To guard against the misuse of information, popular compliance regulations including HIPAA, Sarbanes-Oxley, and the PCI Data Security Standard require organizations to track access to information.

MySQL Auditing

Enabling MySQL Auditing

To check that the plugin is active -

- mysql> SELECT PLUGIN_NAME, PLUGIN_STATUS FROM INFORMATION_SCHEMA.PLUGINS WHERE PLUGIN_NAME LIKE 'audit%';

Edit /etc/my.cnf and add -

- [mysqld]
  audit-log=FORCE_PLUS_PERMANENT

MySQL Auditing

To enable users auditing -

- SELECT audit_log_filter_set_filter('log_all', '{ "filter": { "log": true } }');
- SELECT audit_log_filter_set_user('%', 'log_all');

MySQL Encryption

Encryption vs TDE

Enabling MySQL Encryption

Grigale Ltd T +972 3 938 1169 F +972 3 938 0070
Lev Ha'aretz industrial zone, Rosh Ha'ayin Israel

MySQL Encryption

Encryption vs TDE

Encryption – Data itself is encrypted

No indexes can be enabled on encrypted column.

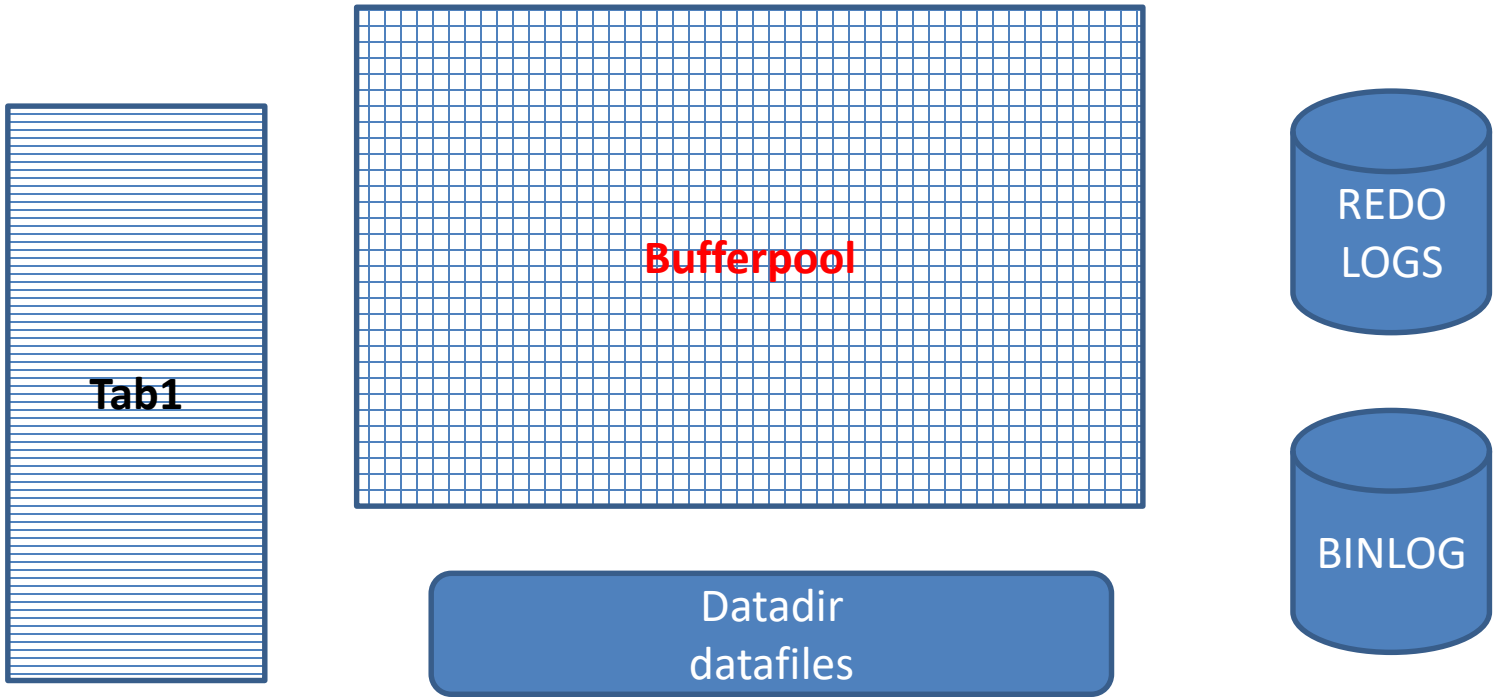
TDE – Transparent Data encryption is encryption in the file-system layer.

MySQL Performance Monitoring and Tuning

GRIGALE
www.grigale.com
Grigale Ltd T +972 3 938 1169 F +972 3 938 0070
Lev Ha'aretz industrial zone, Rosh Ha'ayin Israel

MySQL Performance Monitoring and Tuning

Buffer Pool, Redo Logs and Binlog



MySQL Performance Monitoring and Tuning

Performance schema

- A storage engine for collecting performance.
- Collection of sensors that inform about the time an access has occurred
- Collection of tables that stores all the timers.

MySQL Performance Monitoring and Tuning

SYS schema

- A storage engine that helps to interpret the performance schema.
- Collection of views that summarize performance schema data
- Stored procedures that enables performance schema operations such as configuration changes and diagnostic reports.

MySQL Performance Monitoring and Tuning

Indexes

- Performance report shows if we use indexes.
- No good indexes – it used an index but not the primary key.
- Primary key is a clustered index means that at the leaf of the cluster resides the information.
- IF the table does not have primary key InnoDB will create a shadow primary key which is not recommended.

MySQL Performance Monitoring and Tuning

Explain

- Explain can be executed in CLI
- Explain can be executed in workbench  to show a visualized execution plan.

MySQL Performance Monitoring and Tuning

Monitoring tools

- Workbench
  - Performance -> Performance Reports -> High Cost SQL Statements
- Mysqlslap – load emulation client
- HeidiSQL – SQL GUI

MySQL Tips

- MySQL one liner within the shell:
  - mysql –u root –p –e “source /tmp/dump.sql”
- MySQL diff to compare between databases –
  - mysqldiff --server1=user:pass@host:port:socket --server1=user:pass@host:port:socket db1:db2

MySQL Replication

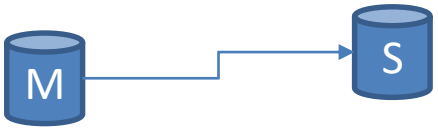
GRIGALE
www.grigale.com
Grigale Ltd T +972 3 938 1169 F +972 3 938 0070
Lev Ha'aretz industrial zone, Rosh Ha'ayin Israel

MySQL Replication

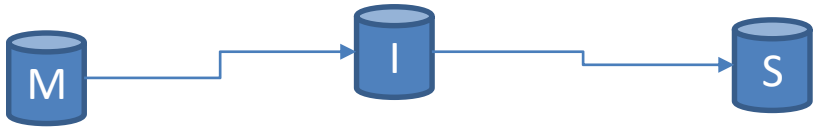
- **MySQL Replication formats**
  - **SBR** – Statement based Replication, replication ships the statements between replicated nodes.
  - **RBR** – Raw based replication, replication ships the outcome of the execution of statements between replicated nodes.
  - Mixed – Utilize both formats.
- **MySQL Replication methods**
  - Replication using Binlog – Slave fetches Transactions from the Master using binlog position mechanism.

  (The transactions are anonymous transactions)
  - Replication using GTID's – Slave fetches Transaction from the Master using Global Transaction ID's

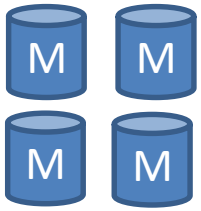
  (Each transaction has an ID)
- **MySQL Replication topologies**
  - Master/Slave



  - Chained (Master/Intermediate/Slave)



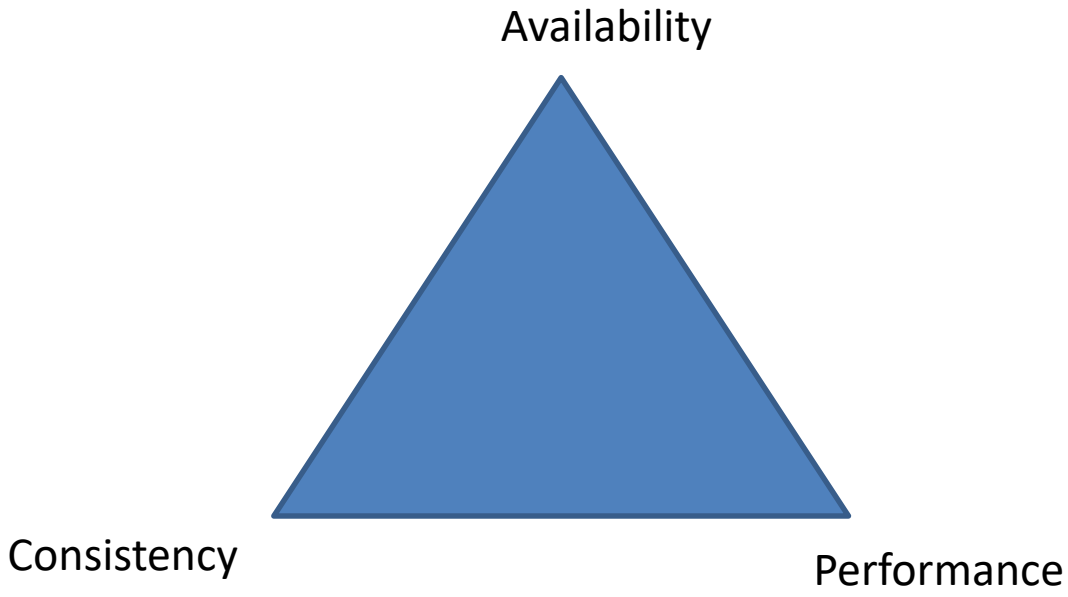
  - Group Replication



MySQL Replication

MySQL Replication formats

CAP Theorem – Consistency, Availability, Performance



We can only have two of the three.

MySQL Replication

MySQL Replication formats

- SBR – Statement based replication
  - Advantages: fast replication, less storage space for logs, allow tracking/auditing the db.
- RBR – Raw based replication
  - Advantages: All changes can be replicated, Safe, Allow handling non-deterministic statements without worry, better concurrency – less locks on the master and slave.
- Mixed – Utilize both formats

MySQL Replication

MySQL Replication tasks

Create replication user

```
mysql> create user replicator@'A' identified by 'replpassword' ;
mysql> grant replication slave on *.* to replicator@'A' ;
```

MySQL Replication Methods

- Binary log file position based replication
- Replication with GTID – Global Transaction identifiers

MySQL Replication

MySQL Replication methods

- Binary log file position based replication
  - The old and familiar replication method.
  - Asynchronous or semi-asynchronous
  - Both Master and Slave can be out of reach. Replication will resume automatically.
  - Slave must track the bin-log file and position
  - Client poll the master for bin-log entries.
  - Each bin-log entry is an anonymous transaction.

MySQL Replication

Binary log file position based replication

Master configuration

Edit /etc/my.cnf:

```
[mysqld]
server-id=1
log-bin
binlog_format=MIXED
# For greatest durability and consistency in InnoDB
sync_binlog=1
innodb_flush_log_at_trx_commit=1
master_info_repository=TABLE
relay_log_info_repository=TABLE
binlog-ignore-db=log
binlog-ignore-db=information_schema
binlog-ignore-db=performance_schema
replicate-wild-ignore-table=mysql.inventory
```

MySQL Replication

Binary log file position based replication

Master status

Restart mysqld service

```
mysql> show master status\G
*************************** 1. row ***************************
             File: mysql.000020
         Position: 154
     Binlog_Do_DB:
 Binlog_Ignore_DB:
Executed_Gtid_Set:
1 row in set (0.00 sec)
```

MySQL Replication

Binary log file position based replication

Slave configuration

Edit /etc/my.cnf:

```
[mysqld]
server-id=2
log-bin  # preparation to be master
binlog_format=MIXED
# For greatest durability and consistency in InnoDB
sync_binlog=1
innodb_flush_log_at_trx_commit=1
master_info_repository=TABLE
relay_log_info_repository=TABLE
binlog-ignore-db=log
binlog-ignore-db=information_schema
binlog-ignore-db=performance_schema
replicate-wild-ignore-table=mysql.inventory
```

MySQL Replication

Binary log file position based replication

Slave setup

- mysql> **CHANGE MASTER TO MASTER_HOST = 'A',**
  >MASTER_USER = 'replicator',
  > MASTER_PASSWORD = 'XXXXXXX',
  > MASTER_PORT = 3306,
  > MASTER_LOG_FILE = 'A_rep.000013',
  > MASTER_LOG_POS = 154
  FOR CHANNEL 'from_A';
- mysql> **start slave;**

MySQL Replication

Binary log file position based replication

Slave status

```
Mysql> show slave status\G
 Slave_IO_State: Connecting to master
                  Master_Host: mysql4
                  Master_User: repl
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql.000019
          Read_Master_Log_Pos: 154
               Relay_Log_File: mysql1-relay-bin-
from_mysql4.000002
                Relay_Log_Pos: 4
        Relay_Master_Log_File: mysql.000019
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              …
          Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 2003
                Last_IO_Error: error connecting to
master 'repl@mysql4:3306' - retry-time: 60  retries:
777
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 0
                  Master_UUID:
             Master_Info_File:
mysql.slave_master_info
                    SQL_Delay: 0
          SQL_Remaining_Delay: NULL
      Slave_SQL_Running_State: Slave has read all
relay log; waiting for more updates
           Master_Retry_Count: 86400
                  Master_Bind:
      Last_IO_Error_Timestamp: 161123 11:43:15
     Last_SQL_Error_Timestamp:
               Master_SSL_Crl:
               …
                Channel_Name: from_mysql4
          Master_TLS_Version:
3 rows in set (0.00 sec)
```

MySQL Replication

Binary log file position based replication

```
mysql> show binlog events\G
*************************** 283. row ***************************
   Log_name: olp1.000001
        Pos: 99862
 Event_type: Anonymous_Gtid
  Server_id: 11
End_log_pos: 99927
       Info: SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
*************************** 284. row ***************************
   Log_name: olp1.000001
        Pos: 99927
 Event_type: Query
  Server_id: 11
End_log_pos: 100070
       Info: use `mysql`; GRANT USAGE ON *.* TO ''@'','root'@'localhost' WITH GRANT OPTION
*************************** 285. row ***************************
   Log_name: olp1.000001
        Pos: 100070
 Event_type: Stop
  Server_id: 11
End_log_pos: 100093
       Info:
285 rows in set (0.00 sec)
```

MySQL Replication

Binary log file position based replication

```
mysql> show relaylog events [for channel 'from_mysql4'] \G
*************************** 1. row ***************************
   Log_name: mysql3-relay-bin-from_mysql4.000031
        Pos: 4
 Event_type: Format_desc
  Server_id: 3
End_log_pos: 123
       Info: Server ver: 5.7.13-enterprise-commercial-advanced-log, Binlog ver: 4
*************************** 2. row ***************************
   Log_name: mysql3-relay-bin-from_mysql4.000031
        Pos: 123
 Event_type: Previous_gtids
  Server_id: 3
End_log_pos: 154
       Info:
2 rows in set (0.00 sec)

mysql>
```

MySQL Replication

Binary log file position based replication

- Available commands

	Start	Stop	Status/Misc
MASTER			show master status; show binlog events\G
SLAVE	Start slave;	stop slave;	show slave status\G show relaylog events [for channel 'channel'] \G
	start slave for channel 'from_mysql1';	stop slave for channel 'from mysql1';	show slave status for channel 'from_mysql1'\G

- Is there a way to stop/start replication from the slave ?
- Change password of the user: replicator

MySQL Replication

Replication with GTID

- Replication with GTID – Global Transaction identifiers
  - A new replication method.
  - Asynchronous or semi-asynchronous
  - Both Master and Slave can be out of reach. Replication will resume automatically.
  - Slave don’t need to track the bin-log file and position
  - Client poll the master for bin-log entries.
  - Each bin-log entry is a known transaction.

MySQL Replication

Replication with GTID

- GTID = source id:transaction id
- Source – MySQL server uuid (from auto.cnf)
- Transaction – An identified bin-log entry
- E.G:

```
3E11FA47-71CA-11E1-9E33-C80AA9429562:23
```

- mysqlbinlog –base64-output=DECORE-ROWS
- show binlog events

MySQL Replication

Replication with GTID

- GTID = source id:transaction id
- Source – MySQL server uuid (from auto.cnf)
- Transaction – An identified bin-log entry
- E.G:

```
3E11FA47-71CA-11E1-9E33-C80AA9429562:23
```

- mysqlbinlog –base64-output=DECORE-ROWS
- show binlog events

MySQL Replication

Replication with GTID

Master Configuration

DATADIR/auto.cnf

```
[auto]
server-uuid=
```

/etc/my.cnf

```
[mysqld]
server-id=1
log-bin
# For greatest durability and consistency in InnoDB
sync_binlog=1
gtid-mode=on
enforce-gtid-consistency=1
innodb_flush_log_at_trx_commit=1
master_info_repository=TABLE
relay_log_info_repository=TABLE
binlog-ignore-db=log
binlog-ignore-db=information_schema
binlog-ignore-db=performance_schema
replicate-wild-ignore-table=mysql.inventory
```

MySQL Replication

Replication with GTID

Master configuration

After editing /etc/my.cnf

Restart MySQL

Log in to mysql shell

mysql> `show master status\G`

MySQL Replication

Replication with GTID

Slave configuration

DATADIR/auto.cnf

```
[auto]
server-uuid=
```

/etc/my.cnf

```
[mysqld]
server-id=2
log-bin
# For greatest durability and consistency in InnoDB
sync_binlog=1
innodb_flush_log_at_trx_commit=1
master_info_repository=TABLE
relay_log_info_repository=TABLE
binlog-ignore-db=log
binlog-ignore-db=information_schema
binlog-ignore-db=performance_schema
replicate-wild-ignore-table=mysql.inventory
```

MySQL Replication

Replication with GTID

Slave configuration

Slave setup

```
CHANGE MASTER TO
      > MASTER_HOST = 'A',
      > MASTER_USER = 'replicator',
      > MASTER_PASSWORD = 'XXXXXXX',
      > MASTER_AUTO_POSITION = 1
      > FOR CHANNEL 'from_A';
;
```

MySQL Replication

Replication with GTID

- Available commands

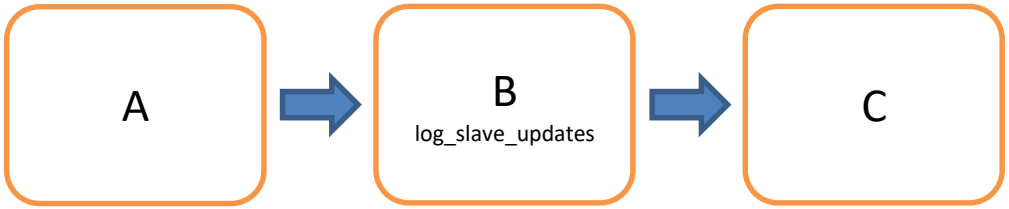
	Start	Stop	Status/Misc
MASTER			show master status;
SLAVE	Start slave;	stop slave;	show slave status\G
	Start slave for channel ‘from_mysql1’;	Stop slave for channel ‘from mysql1’;	show slave status for channel ‘from_mysql1’\G

- Is there a way to stop/start replication from the slave ?

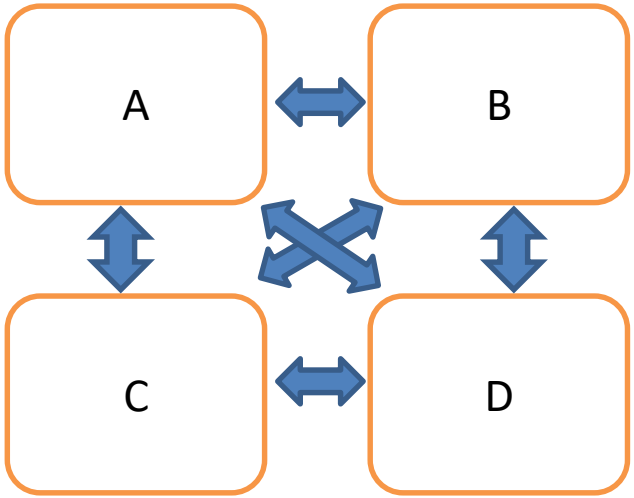
MySQL Replication

MySQL Replication topologies

- Chained Replication – MySQL 5.6



- Multi–Source Replication – MySQL 5.7



MySQL Replication

Chained Replication

- Master A – regular Master
- Master B – Master with log-slave-updates so relay-log transactions will be entered into the bin-log
- Slave C – Regular Slave

MySQL Replication

Multi–Source Replication

- Each server is maser and slave
- Every replication is called channel
- Each channel can be stopped or started independently.
- In case of any replication disruption replication will resume automatically

Thank You!

GRIGALE
www.grigale.com
Grigale Ltd T +972 3 938 1169 F +972 3 938 0070
Lev Ha'aretz industrial zone, Rosh Ha'ayin Israel