Data Replication of OpenMRS Database

Introduction

- This is a tutorial for setting up a master-slave replication of the OpenMRS database.
- This tutorial uses the AMPATH Medical Record System (AMRS) implementation of OpenMRS as an example.
    - Change names accordingly to suit your use.

Notes

- The host name of our Master Server is called AMRSPROD (as set by an alias in the host file).
- The host name of our Slave Server is called AMRSRESEARCH.
- It is recommended to use the same name for the replicated OpenMRS database on the Master as for the slave.
    - Our replicated database is called 'amrs' on both AMRSPROD and AMRSRESEARCH.

References

- MySQL Reference Manual Chapter on Replication
    - (I used this when setting up AMRS replication.)
- Conference 2008 Replication Tutorial
    - (I used MySQL Replication Tutorial Presentation (pdf) from here when setting up AMRS replication.)
- MySQL Conference 2009 Replication Tutorial

Tutorial

1. Ensure that each mysql have bind the host ip adddress
2. Create a replication user on the master database.
3. :

```
#: mysql -u root -p
#: mysql> GRANT REPLICATION SLAVE on *.* TO 'rep_user'@'amrsresearch' IDENTIFIED BY 'this-is-the-
password';
#:
```

4. Stop the mysqld on the master server (AMRSPROD).
5. Edit my.cnf (or my.ini in Windows) on the master server (AMRSPROD):
6. :

```
#: [mysqld]
#: ## Replication of AMRS database:
#:
#: # An ID number to give the master (required):
#: server_id=1
#:
#: # The location and name of the binlog log files (required):
#: log-bin="E:/MySQL Data/binlog/amrs-bin"
#:
#: # This will limit replication to only the amrs database:
#: binlog-do-db=amrs
#:
#: # This keeps only two weeks of binlog files:
#: expire_logs_days=14
#:
#: # Set to '1' to write to binlog for every SQL statement instead of every transaction
#: # (Safest option in case of server crash):
#: sync_binlog=1
#:
```

7. Restart mysqld on master (AMRSPROD).
8. Obtain the master replication information (AMRSPROD).
9. :

```
mysql -u root -p
```

- Lock tables so no one can write to them.

10. :

```
mysql> FLUSH TABLES WITH READ LOCK;
```

- Then the replication information:

11. :

```
#: mysql > SHOW MASTER STATUS;
#: +----------------+----------+--------------+------------------+
#: | File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
#: +----------------+----------+--------------+------------------+
#: | amrs-bin.00001 | 97       | amrs         |                  |
#: +----------------+----------+--------------+------------------+
#:
```

12. Create the backup of the master (AMRSPROD).
13. *__Either__ stop mysqld on master and manually copy the data files (not covered here) __or__...
    - Create a mysqldump file (takes longer than copying data files):
14. :

```
mysqldump -u root -p -q -e --single-transaction -r"amrs_backup.sql" amrs
```

15. Restore the backup onto the slave (AMRSRESEARCH)
    - If you chose the mysqldump route, then run this on the slave:
16. :

```
#: mysql -u root -p
#: mysql> source amrs_backup.sql
```

    - Depending on the size of the database, it may be a good time for lunch.
17. Grant the OpenMRS webapp_user for the slave server (AMRSRESEARCH) read-only access to all database tables except those that will not be replicated.
18. Stop the mysqld on the slave server (AMRSRESEARCH)
19. Edit my.cnf (or my.ini in Windows) on the slave server (AMRSRESEARCH):
20. :

```
#: [mysqld]
#: ## Replication for AMRS:
#: # ID for slave server must be different than master (required):
#: server-id=3
#:
#: # Ignore these tables to run AMRS on slave server:
#: replicate-ignore-table=amrs.global_property
#: replicate-ignore-table=amrs.scheduler_task_config
#: replicate-ignore-table=amrs.scheduler_task_config_property
#:
#: # Ignore these tables to run reports using AMRS on slave server:
#: replicate-ignore-table=amrs.cohort
#: replicate-ignore-table=amrs.cohort_member
#: replicate-ignore-table=amrs.report_object
#: replicate-ignore-table=amrs.report_schema_xml
#:
#: # Ignore any module tables you want on slave but not on master:
#: replicate-ignore-table=amrs.versionedfileupload_versioned_file
#: replicate-ignore-table=amrs.reporttemplate_report_template
#:
```

21. Restart the mysqld on the slave (AMRSRESEARCH).
22. Configure the slave (AMRSRESEARCH):
23. :

```
#: mysql -u root -p
#: mysql> CHANGE MASTER TO master_host='amrsprod',
#:     -> master_user='rep_user',
#:     -> master_password='this-is-the-password',
#:     -> master_log_file = â€œamrs-bin-00001â€?,
#:     -> master_log_pos = 97;
#:
```

#: info| **IMPORTANT**: Change the *master_log_file* and *master_log_pos* to the values copied from the master (AMRSPROD).

24. Start the slave process.
25. :

```
mysql> START SLAVE;
```

26. Check the slave status.
27. :

```
mysql> SHOW SLAVE STATUS\G
```

28. On master (AMRSPROD) unlock tables so they can be written.
29. :

```
mysql> UNLOCK TABLES;
```