## Setup Mysql Replication

## Production Replication Setup

**+------------+ +------------+**

**+ + + +**

**+ master + + slave +**

**+ +<-------replication--------> + +**

**+ + + +**

**+------------ + +------------ +**

**192.168.100.123 192.168.100.124**

**Master Replication Setup**

The master (production server) needs to have binary logging turned on to facilitate replication. To turn on binary logging, we need to take the following steps:

* Create user relication:

CREATE USER 'repl'@'%.192.168.100.123l' IDENTIFIED BY '123456';

GRANT REPLICATION SLAVE ON \*.\* TO 'repl'@'%.192.168.100.123';

FLUSH PRIVILEGES;

OR: Using phpMyAdmin, add a new user called **replication\_user** which only has REPLICATION SLAVE access rights

* Add the following lines to the **/etc/my.cnf** file:

[mysqld]  
log-bin=mysql-bin  
server-id=1

* Restart mysql (as root)

[user@server]# service mysqld restart

* Stop the production server, and obtain its state information. Most importantly, note the **File** and **Position** values:

mysql> FLUSH TABLES WITH READ LOCK;  
mysql> SHOW MASTER STATUS;

+-----------------------+------------+----------------------+--------------------------+  
| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |  
+-----------------------+------------+----------------------+--------------------------+  
| mysql-bin.00003 | 240 | | |  
+-----------------------+------------+----------------------+---------------------------+

* While the database has the READ LOCK applied, export the data using mysqldump, including master data information:

mysqldump -uroot -p --all-databases --master-data > dbdump.db

* Then unlock the master database:

mysql> UNLOCK TABLES;

## Slave Replication Setup

To initialise the slave server, the following steps need to be taken:

* Add the following to the slave servers config file **/etc/my.cnf**:

# Replication settings  
log-bin  
replicate-ignore-db=mysql  
relay-log=triton-relay-bin  
log-slave-updates  
log-warnings  
server-id=2  
# Slave details  
report-host=slave.mydomain.com # YOUR SLAVE'S HOSTNAME

* Transfer the data (dbdump.db) file from Master and load it into the database:

shell> mysql < fulldb.dump

* Set the file information – this needs to match the values you observed on the master when you locked the tables above. In this example the filename was **mysql-bin.00003** and the log position was **240**:

mysql>change master to master\_host='192.168.100.123’, IP or domain name

master\_port=3306, master\_user='repl',

master\_password='123456',

master\_log\_file='mysql-bin.003',

master\_log\_pos=240;

* Start the slave:

mysql> start slave;

You should now have a working mysql replication database! .

## 

## Checking Replication is Working

mysql> show slave status\G;

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

Slave\_IO\_State: Waiting for master to send event

Master\_Host: 192.168.100.123

Master\_User: repl

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: mysql-bin.000003

Read\_Master\_Log\_Pos: 292

Relay\_Log\_File: triton-relay-bin.000002

Relay\_Log\_Pos: 437

Relay\_Master\_Log\_File: mysql-bin.000003

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

Replicate\_Do\_DB:

Replicate\_Ignore\_DB: mysql

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: 292

Relay\_Log\_Space: 593

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: