

duobei mysql主从部署文档

- 1 部署mysql主从服务
 - master 配置

```
1 master 数据库全量备份
2 copy备份到从库
3 锁库
4 增量备份
5 copy增量备份到从库
6 修改my.cnf
7 重启
8 授权用户及IP
```

* slave配置

```
1 应用备份数据
2 合并数据还原
3 修改my.cnf
4 重启
5 切换到主log点
6 开启主从服务
7 查看那状态
```

- 2 部署mysql主从延时程序
- 3 部署mysql延时监控程序[]
- 4 部署mysql状态服务程序[发微信通知]
- 5 操作步骤

1 部署mysql主从服务

```
1 ssh 登录mysql 主服务器 数据全量备份
```

```
innobackupex --user=$user --password=$passwd /var/nebula/backup/bak_db/
```

```
2 拷贝全量数据到从库
```

```
scp -r /var/nebula/backup/bak_db/2016-xxx
root@192.168xxx:/var/nebula/backup/bak_db/
```

```
3 锁表
```

```
mysql -uroot -p
```

```
flush tables with read lock;
```

4 增量备份

```
innobackupex --user=$user --password=$passwd --incremental --incremental-basedir=  
/var/nebula/backup/bak_db/$全备目录 /var/nebula/backup/db/incremental/
```

##增量恢复注意：最后一步的增量备份没有--redo-only选项！还有，可以使用--use_memory提高性能。

5 拷贝增量备份到从库

```
scp -r /var/nebula/backup/incremental/  
root@192.168xxx:/var/nebula/backup/incremental/
```

6 修改master my.cnf

(1) 修改数据库配置文件，指明 server_id，开启二进制日志(log-bin)；

```
## 设置 server_id，一般设置为 IP  
server_id=1
```

复制过滤：需要备份的数据库，输出 binlog

```
binlog-do-db=nebula_meeting
```

复制过滤：不需要备份的数据库，不输出(mysql 库一般不同步)

```
binlog-ignore-db=mysql
```

开启二进制日志功能，可以随便取，最好有含义

```
log-bin=mysql-master-bin
```

为每个 session 分配的内存，在事务过程中用来存储二进制日志的缓存

```
binlog_cache_size=1M
```

主从复制的格式(mixed,statement,row，默认格式是 statement)

```
binlog_format=mixed
```

二进制日志自动删除/过期的天数。默认值为 0，表示不自动删除。

```
expire_logs_days=7
```

跳过主从复制中遇到的所有错误或指定类型的错误，避免 slave 端复制中断。

如:1062 错误是指一些主键重复，1032 错误是因为主从数据库数据不一致
slave_skip_errors=1062

(2) 启动数据库, 查看当前是哪个日志, position 号是多少;

重启数据库

```
sudo /etc/init.d/mysqld status
```

```
sudo /etc/init.d/mysqld restart
```

```
show master status;
```

7 登录数据库, 授权数据复制用户(IP 地址为从机 IP 地址, 如果是双向主从, 这里的 还需要授权本机的 IP 地址, 此时自己的 IP 地址就是从 IP 地址);

```
grant replication slave, replication client on *.* to 'repl'@'192.168.0.62' identified by 'Duobei@123';
```

```
flush privileges;
```

(4) 备份数据库(记得加锁和解锁);[这里用的策略是停掉数据库, 拷贝数据文件]

拷贝数据文件, 同城专线

(5) 传送备份数据到 Slave 上;

2 slave 上配置

1 应用备份数据[保持slave mysql关闭状态]

****恢复基础备份**

```
innobackupex --user=$user --password=$passwd --defaults-file=/etc/my.cnf --use-memory=4G --apply-log --redo-only /var/nebula/$全备目录
```

****应用增量备份**

```
innobackupex --user=$user --password=$passwd --defaults-file=/etc/my.cnf --use-memory=4G --apply-log /var/nebula/$全备目录 --incremental-dir=/var/nebula/$增备目录
```

2 合并数据还原

```
innobackupex --user=$user --password=$passwd --defaults-file=/etc/my.cnf --use-memory=4G --apply-log /var/nebula/$全备目录
```

```
innobackupex --user=$user --password=$passwd --defaults-file=/etc/my.cnf --copy-back /var/nebula/$全备目录
```

3 修改my.cnf

(1) 修改数据库配置文件, 指明 `server_id`(如果是搭建双向主从的话, 也要开启二进制 日志 `log-bin`);

```
#erver_id, 一般设置为 IP  
server_id=2
```

复制过滤:需要备份的数据库, 输出 `binlog`

```
binlog-do-db=nebula_meeting
```

复制过滤:不需要备份的数据库, 不输出(mysql 库一般不同步)

```
binlog-ignore-db=mysql  
binlog-ignore-db = information_schema  
binlog-ignore-db = test
```

开启二进制日志功能, 可以随便取, 最好有含义

```
log-bin=mysql-slave-bin
```

为每个 session 分配的内存, 在事务过程中用来存储二进制日志的缓存

```
binlog_cache_size=1M
```

主从复制的格式(`mixed`,`statement`,`row`, 默认格式是 `statement`)

```
binlog_format=mixed
```

二进制日志自动删除/过期的天数。默认值为 0, 表示不自动删除。

```
expire_logs_days=7
```

跳过主从复制中遇到的所有错误或指定类型的错误, 避免 slave 端复制中断。

如:1062 错误是指一些主键重复, 1032 错误是因为主从数据库数据不一致

```
slave_skip_errors=1062
```

4 启动数据库;

```
/etc/init.d/mysqld status
```

```
/etc/init.d/mysqld restart
```

5 查看当前是哪个日志, `position` 号是多少(单向主从此步不需要, 双向主从需要);

5 指定 Master 的地址、用户、密码等信息;

```
change master to master_host='192.168.0.65', master_user='repl', master_password='  
zzyhappy123', master_port=5110, master_log_file='duobei_bin_log_master.000001' , m  
aster_log_pos=1969, master_connect_retry=30;
```

(5) 开启同步，查看状态。

```
show slave status\G;
```

可看到 Slave_IO_State 为空， Slave_IO_Running 和 Slave_SQL_Running 是 No，表明 Slave 还没有开始复制过程。

(6) 开启同步状态

```
mysql> start slave;
```

```
mysql> show slave status\G;
```

(7) 插入一条数据测试...

1.2 mysql 主从同步延时

```
1 mkdir /usr/local/mysql-tools
```

```
2 wget https://www.percona.com/downloads/percona-toolkit/2.2.20/deb/percona-toolkit_2.2.20-1.tar.gz
```

```
3 tar -xvf percona-toolkit_2.2.20-1.tar.gz
```

```
4 cd percona>>>
```

```
5 perl Makefile.PL 【在64上少一堆perl插件】
```

```
6 make && make test
```

```
7 make install
```

如果是非源码安装或源码安装是未指定路径，缺省情况下所有的pt相关的工具位于/usr/bin目录下

```
8 --ask-pwd 会出错
```

9 主上执行

```
pt-heartbeat --user=root --password='zzyhappy' --host=127.0.0.1 -P5110 --create-table -D duobei --interval=1 --update --replace --daemonize
```

```
10 【如果未指定create-table】建立表
```

```
CREATE TABLE heartbeat (  
ts varchar(26) NOT NULL,
```

```

server_id      int unsigned NOT NULL PRIMARY KEY,
file           varchar(255) DEFAULT NULL,      -- SHOW MASTER STATUS
position       bigint unsigned DEFAULT NULL,    -- SHOW MASTER STATUS
relay_master_log_file varchar(255) DEFAULT NULL, -- SHOW SLAVE STATUS
exec_master_log_pos  bigint unsigned DEFAULT NULL -- SHOW SLAVE STATUS
);

```

11 查看 pgrep -fl pt-heartbeat

12 从上执行[执行一次]

```

pt-heartbeat --user=root --password=zzyhappy --host=127.0.0.1 --port=5110 -D duobe
i --check

```

13 一直执行不退出

```

pt-heartbeat --user=root --password=zzyhappy --host=127.0.0.1 --port=5110 -D duobe
i --monitor

```

```

0.00s [ 0.00s, 0.00s, 0.00s ]

```

```

0.00s [ 0.00s, 0.00s, 0.00s ]

```

```

0.00s [ 0.00s, 0.00s, 0.00s ]

```

14 数据保存在文件里面

```

mkdir output.txt

```

```

pt-heartbeat --user=root --password=zzyhappy --host=127.0.0.1 --port=5110 -D duob
ei --interval=1 --frames==1s,5s,10s --print-master-server-id --file=/var/mysql/mys
ql-replication.log --monitor

```

在文件output.txt看到只有一行输出记录，因为标准输出默人行缓冲

```

pt-heartbeat --user=root --password=zzyhappy --host=127.0.0.1 --port=5110 -D duob
ei --interval=1 --frames==5s,15s,30s --print-master-server-id --log=mysql-replicat
ioin.log --monitor

```

1.3 部署延时监控程序[slave]

1 上传文件 mysql-delay.py 到/usr/src

2 启动python程序 python mysql-delay.py

[如果mysqldb 模块没有 需要安装]

3 grafana查看是否有数据

[centos安装升级Python--2.6 安装pip包管理工具]

python 查看版本

cd /usr/local

mkdir python-2.7

wget http://www.python.org/ftp/python/2.7.10/Python-2.7.10.tar.xz

unxz Python-2.7.10.tar.xz

tar -vxf Python-2.7.10.tar

./configure --enable-shared --enable-loadable-sqlite-extensions --with-zlib

make && make install

安装好Python2.7之后我们需要先把Python2.6备份起来，然后再对yum的配置进行修改，如果不进行这一步操作的话，执行yum命令将会提示你Python的版本不对。

mv /usr/bin/python /usr/bin/python2.6.6

ln -s /usr/local/bin/python2.7 /usr/bin/python

cd /usr/local

mkdir python-pip

wget --no-check-certificate https://github.com/pypa/pip/archive/1.5.5.tar.gz

tar zvxf 1.5.5.tar.gz #解压文件

cd pip-1.5.5/

python setup.py install

1.4 部署mysql状态服务程序[slave]

1 上传 `mysql-status-monitor.py` 到 `/usr/src`

2 启动python程序 `python mysql-delay.pyt`

[如果mysqldb 模块没有 需要安装]

3 运行时否有数据[暂时没有介入微信通知]

mysql 主从方案

- 1、全量备份
- 2、copy全量备份到从数据库服务器
- 3、锁库
- 4、增量备份
- 5、copy 增量备份到从数据库服务器
- 6、还原数据
- 7、启动从服务器mysql
- 8、更新主服务器my.cnf
- 9、iptables 加入规则，拒绝除从服务器外所有3306端口请求
- 10、关闭异常web服务
- 11、重启主服务器mysql
- 12、测试主从
- 13、恢复业务