

MySQL Fabric HA

mysql 安装（Windows）

1. 准备

分别在4台服务器上进行MySQL的安装，服务器应用场景如下

IP	PORT	场景
192.168.1.1	3306	Fabric
192.168.1.2	3306	APP
192.168.1.3	3306	APP
192.168.1.4	3306	APP

2. 安装和初始化

mysql官网下载所使用的mysql版本，本文以绿色版安装方式进行讲解，下载文件为.zip压缩包，解压zip包，进入到bin目录下执行以下命令安装

```
mysql --install <服务名> // 安装服务
```

安装完成后服务是启动不成功的，需在mysql根目录新增my.ini文件，文件中内容为

```
[mysqld]
port=3306 // mysql服务的端口
datadir= 当前目录/Data // 数据库存放地址
```

在mysql的bin目录下执行下面命令来初始化数据

```
mysql --initialize --console //--console在当前控制台打印初始化日志
```

从打印的日志中找到下面一行信息，此信息为初始化完成首次登录信息， root@localhost:后面内容为root账号的密码（密码为随机生成，以本机生成为准）。

[Note] A temporary password is generated for root@localhost:hiyk+om5Vmpq

进入系统的服务管理列表，找到安装时使用的服务名，启用mysql服务；或使用如下命令启动

```
net start|stop 服务名    // start: 开启服务; stop: 关闭服务。
```

服务开启成功后，在mysql的bin目录下执行以下命令登录mysql

```
mysql -uroot -p
Enter password: *****    //密码就是初始化时打印的密码
```

首次登录成功，需要修改密码

```
mysql> set password=password('123456');    //修改密码
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

目前为止mysql安装完成

配置支持Fabric

MySQL服务安装完成修改my.ini文件，在文件中[mysqld] 模块中添加以下项，修改完成重启MySQL服务

```
server_id=1    // 每台机器server_id必须唯一
log-bin
gtid-mode=ON
enforce-gtid-consistency
log_slave_updates
```

4台MySQL都需增加Fabric使用账号

```
CREATE USER 'fabric'@'%' IDENTIFIED BY 'fabric'; // 创建fabric使用用户
GRANT ALL ON *.* TO 'fabric'@'%' IDENTIFIED BY 'fabric'; // 给fabric赋所有权
flush privileges; // 刷新权限
```

Fabric安装

安装

mysql官网下载所使用的Fabric版本(MySQL Utilities1.6.1 及之前版本包含Fabric), 本文以绿色版安装方式进行讲解, 下载文件为.zip压缩包, Fabric使用python编写, 安装前需先安装python环境, 将下载的压缩包进行解压, 进入根目录执行以下命令进行安装

```
python setup.py install
```

配置

修改etc/mysql/fabric.cfg文件

```
[DEFAULT]
prefix = G:\Program Files (x86)\MySQL\MySQL Utilities    #Fabric的安装目录
sysconfdir = G:\Program Files (x86)\MySQL\MySQL Utilities\etc\mysql    #Fabric的配置文件存放目录
logdir = G:\Program Files (x86)\MySQL\MySQL Utilities\log    #Fabric log文件的存放目录

[storage]    #Fabric应用连接数据库信息
address = localhost:3306    #数据库url
user = fabric    #数据库连接账号
password = fabric    #数据库连接密码
database = fabric    #数据库名
auth_plugin = mysql_native_password    #设置使用的认证插件
connection_timeout = 6    #中断请求之前等待的最大时间, 单位秒
connection_attempts = 6    #创建连接的最大尝试次数
connection_delay = 3    #连续尝试创建连接之间的延时时间, 默认3s

[servers]
user = fabric
password = root
backup_user = fabric
backup_password = root
restore_user = fabric
restore_password = root
unreachable_timeout = 20

[protocol.xmlrpc]    #该段定义Fabric接收通过XML-RPC协议的请求
address = 0.0.0.0:32274    #标识Fabric使用的主机和端口(0.0.0.0所有ip,可指定ip地址), 接收XML-RPC请求
threads = 5    #XML-RPC会话线程的并发创建数, 决定多少并发请求Fabric能接受, 与数据库连接池数量一致
user = admin    #用户名, 认证命令行请求, jdbc连接使用此账号进行登录
password = admin    #用户密码, 认证命令行请求, jdbc连接使用此密码进行登录
disable_authentication = no    #是否启用命令行请求需要认证, 默认要认证
realm = MySQL Fabric
```

```

ssl_ca =      #使用ssl认证方式, 指定PEM格式文件, 包含信任SSL证书的列表
ssl_cert =    #SSL认证文件, 用于创建安全的连接
ssl_key =     #SSL key文件

[executor]    #通过XML-RPC接收到的请求, 映射到程序能立即执行或通过队列执行者, 保证冲突的请求处理
              按序执行。通常读操作立即执行通过XML-RPC会话线程, 写操作通过执行者
executors = 5  #多少线程用于执行者

[logging]     #设置Fabric日志信息记录到哪里, 如果不是开启为后台进程, 将打印日志到标准输出
level = INFO  #日志级别, 支持DEBUG, INFO, WARNING, ERROR, CRITICAL
url = file:///var/log/fabric.log  #存储日志的文件, 能为绝对或相对路径(如是相对路径, 将参照
default段logdir参数指定的日志目录)

[sharding]    #Fabric使用mysqldump和mysql客户端程序, 执行移动和分离shards, 指定程序的路径
mysqldump_program = /usr/bin/mysqldump
mysqlclient_program = /usr/bin/mysql

[statistics]
prune_time = 3600  #删除大于1h的条目

[failure_tracking]  #连接器和其他外部实体能报告错误, fabric保持跟踪服务器健康状态和采取相应的
                  行为, 如提升一个新的master, 如果一个服务器时不稳定的, 但不是master, 将简单的标记为错误。
notifications = 300  #多少次报告错误后, 将标志服务器不可用
notification_clients = 50  #多少不同源报告错误
notification_interval = 60  #评估错误数的统计时间
failover_interval = 0  #为了避免整个系统不可用, 自上次提升间隔多少秒后, 新master才能选取
detections = 3  #为了缓解fabric, 提供内建的错误检查, 如果错误检查启动监控一个组, 需要连续尝试
3(默认)次访问当前master都错误后, 才能提升新master,
detection_interval = 6  #连续检查之间的间隔时间
detection_timeout = 1  #错误检查程序尝试连接到一个组中服务器的超时时间
prune_time = 3600  #在错误日志中保留多久的错误信息

[connector]   #Fabric-aware连接器连接到Fabric, 获取组、shards、服务器的信息, 缓存结果到本地的
              时长, 以提高性能。
ttl = 1  #缓存生存时间, 单位s, 决定多长时间, 连接器考虑一个信息从Fabric获取是有效的

```

初始化数据

开启Fabric应用使用数据库服务, 执行以下命令初始化数据。

```
mysqlfabric manage setup
```

开启服务

在根目录下执行以下命令开启Fabric服务

```
mysqlfabric manage start    #stop为关闭
```

创建集群组、增加应该服务器信息

```
mysqlfabric group create my_group    #创建集群组my_group，my_group为自定义名字
mysqlfabric group add my_group 192.168.1.2:3306    #添加组成员，将应用数据库连接地址加入
mysqlfabric group add my_group 192.168.1.3:3306
mysqlfabric group add my_group 192.168.1.4:3306
```

开启选举功能

```
mysqlfabric group promote my_group
```

激活故障自动切换

```
mysqlfabric group activate my_group
```

查看Fabric状态

```
mysqlfabric group lookup_servers my_group
```

完成。

JAVA JDBC连接方式

使用MySQL Fabric后，需修改数据库连接驱动，mysql的驱动包中已包含Fabric连接驱动，只需修改驱动名为以下，不需要新增加jar包

```
com.mysql.fabric.jdbc.FabricMySQLDriver
```

数据库连接信息URL地址使用Fabric的地址，Fabric使用XML-RPC协议进行通信，所以端口为fabric中配置的[protocol.xmlrpc]模块中的address信息（0.0.0.0:32274, 4个0改成Fabric服务器的ip），URL信息中增加fabricServiceGroup(创建的组)、fabricUsername（[protocol.xmlrpc]中的user）、fabricPassword信息([protocol.xmlrpc]中的password)，数据库名使用Fabric前后保持一致。格式如下

```
db.url=jdbc:mysql:fabric://192.168.15.200:32274/shopweb?  
fabricServerGroup=my_group&fabricUsername=admin&fabricPassword=123456  
db.user=数据库应用账号（使用Fabric前后保持一致）  
db.password=数据库应用密码（使用Fabric前后保持一致）
```