

Table of Contents

- [1. 环境讲解](#)
- [2. 部署运行流程](#)
 - [2.1. mha-manager介绍](#)
 - [2.1.1. 镜像构建](#)
 - [2.2. mysql-57-master](#)
 - [2.3. mysql-57-slave1 & mysql-57-slave2](#)
- [3. 需要注意的点](#)
 - [3.1. perl脚本有问题](#)
 - [3.2. 所有节点均需要打开 bin log](#)

1 环境讲解

采用docker进行构建，包含如下4个服务

服务名	hostname/容器名	root密码	mysql密码	3306映射的外部端口
mha-manager	mysql-mha-manager	root4manager		
mysql-57-master	mysql-57-master	root4master	MyMaster	13570
mysql-57-slave1	mysql-57-slave1	root4slave1	MySlave1	13571
mysql-57-slave2	mysql-57-slave2	root4slave2	MySlave2	13572

各个节点之间通信全采用hostname查找的方式，配置中不指定IP

2 部署运行流程

直接运行start.sh start.sh会做如下工作

- 编译服务需要的镜像
- 启动容器
- sleep 10s
- 收集各个容器的ssh /root/.ssh/id_rsa.pub 收集并生成 authorized_keys 文件
- 将生成的 authorized_keys 文件拷贝到各个容器中，并修改所有者

2.1 mha-manager介绍

主要配置为 [mha config file](#) 需要注意 password 是数据库密码， repl_password 是linux下的用户

密码，可以理解为ssh密码

2.1.1 镜像构建

参考 [manager Dockerfile](#) 中的内容

2.2 mysql-57-master

通过 [my.cnf](#) 加载semi插件，并开启配置

```
# 加载semi master插件
plugin-load-add=semisync_master.so
rpl_semi_sync_master_enabled=1
rpl_semi_sync_master_timeout=1000
```

2.3 mysql-57-slave1 & mysql-57-slave2

除 server-id 两者的配置几乎相同，

- 在 [slave1 my.cnf](#) [slave2 my.cnf](#)中开启semi插件

```
# 加载semi slave插件
plugin-load-add=semisync_slave.so
rpl_semi_sync_slave_enabled=1
```

- 配置初始化脚本，进行master的连接 [slave1 init.sql](#) 和 [slave2 init.sql](#)

```
change master to master_host='mysql-57-master',
               master_user='root',
               master_password='MyMaster',
               master_log_file='mysql-bin.000001';

start slave;
```

3 需要注意的点

3.1 perl脚本有问题

报错如下

```
Sun Mar 14 22:45:07 2021 - [error][/usr/share/perl5/vendor_perl/MHA/MasterMonitor.pm, 1n4
Sun Mar 14 22:45:07 2021 - [error][/usr/share/perl5/vendor_perl/MHA/MasterMonitor.pm, 1n5
```

NodeUtil.pm:201行会报错，需要修改为：

```
$str =~ /(\d+)\.(\d+)/;  
my $strmajor = "$1.$2";  
my $result = sprintf( '%03d%03d', $strmajor =~ m/(\d+)/g );
```

3.2 所有节点均需要打开 **bin log**

slave节点my.cnf添加bin log 配置且所有的bin log配置需要相同，例如过滤的数据库，需要记录bin log的数据库