

# 安装和配置Canal

下面我们就开启mysql的主从同步机制，让Canal来模拟salve

## 1.开启MySQL主从

Canal是基于MySQL的主从同步功能，因此必须先开启MySQL的主从功能才可以。

这里以之前用Docker运行的mysql为例：

### 1.1.开启binlog

打开mysql容器挂载的日志文件，我的在 /tmp/mysql/conf 目录：

```
[root@node1 conf]# pwd
/tmp/mysql/conf
[root@node1 conf]# ll
总用量 4
-rw-r--r--. 1 root root 91 8月  9 18:22 my.cnf
```

修改文件：

```
vi /tmp/mysql/conf/my.cnf
```

添加内容：

```
log-bin=/var/lib/mysql/mysql-bin
binlog-do-db=heima
```

配置解读：

- `log-bin=/var/lib/mysql/mysql-bin`：设置binary log文件的存放地址和文件名，叫做mysql-bin
- `binlog-do-db=heima`：指定对哪个database记录binary log events，这里记录heima这个库

最终效果：

```
[mysqld]
skip-name-resolve
character_set_server=utf8
datadir=/var/lib/mysql
server-id=1000
log-bin=/var/lib/mysql/mysql-bin
binlog-do-db=heima
```

## 1.2.设置用户权限

接下来添加一个仅用于数据同步的账户，出于安全考虑，这里仅提供对heima这个库的操作权限。

```
create user canal@'%' IDENTIFIED by 'canal';
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT,SUPER ON *.* TO 'canal'@'%'
identified by 'canal';
FLUSH PRIVILEGES;
```

重启mysql容器即可

```
docker restart mysql
```

测试设置是否成功：在mysql控制台，或者Navicat中，输入命令：

```
show master status;
```

信息					结果 1					剖析					状态									
File					Position					Binlog_Do_DB					Binlog_Ignore_DB					Executed_Gtid_Set				
▶ mysql-bin.000001					1584					heima														

## 2.安装Canal

### 2.1.创建网络

我们需要创建一个网络，将MySQL、Canal、MQ放到同一个Docker网络中：

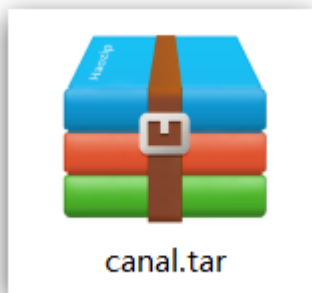
```
docker network create heima
```

让mysql加入这个网络：

```
docker network connect heima mysql
```

### 2.3.安装Canal

课前资料中提供了canal的镜像压缩包：



大家可以上传到虚拟机，然后通过命令导入：

```
docker load -i canal.tar
```

然后运行命令创建Canal容器：

```
docker run -p 11111:11111 --name canal \
-e canal.destinations=heima \
-e canal.instance.master.address=mysql:3306 \
-e canal.instance.dbusername=canal \
-e canal.instance.dbpassword=canal \
-e canal.instance.connectionCharset=UTF-8 \
-e canal.instance.tsdb.enable=true \
-e canal.instance.gtidon=false \
-e canal.instance.filter.regex=heima\\.* \
--network heima \
-d canal/canal-server:v1.1.5
```

说明：

- `-p 11111:11111`：这是canal的默认监听端口
- `-e canal.instance.master.address=mysql:3306`：数据库地址和端口，如果不知道mysql容器地址，可以通过 `docker inspect 容器id` 来查看
- `-e canal.instance.dbusername=canal`：数据库用户名
- `-e canal.instance.dbpassword=canal`：数据库密码
- `-e canal.instance.filter.regex=`：要监听的表名称

表名称监听支持的语法：

mysql 数据解析关注的表，Perl正则表达式。

多个正则之间用逗号(,)分隔，转义符需要双斜杠(\\)

常见例子：

1. 所有表：`.*` or `.*\\.*`
2. canal schema下所有表：`canal\\.*`
3. canal下的以canal打头的表：`canal\\.canal.*`
4. canal schema下的一张表：`canal.test1`
5. 多个规则组合使用然后以逗号隔开：`canal\\.*,mysql.test1,mysql.test2`