# Centos7 搭建mysql主从

CentOS7的yum源中默认是没有mysql的。为了解决这个问题，我们要先下载mysql的repo源。

**1. 下载mysql的repo源**

$ wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm

**2. 安装mysql-community-release-el7-5.noarch.rpm包**

$ sudo rpm -ivh mysql-community-release-el7-5.noarch.rpm

安装这个包后，会获得两个mysql的yum repo源：/etc/yum.repos.d/mysql-community.repo，/etc/yum.repos.d/mysql-community-source.repo。

**3. 安装mysql**

$ sudo yum install mysql-server

根据步骤安装就可以了，不过安装完成后，没有密码，需要重置密码。

**4. 重置密码**

重置密码前，首先要登录

$ mysql -u root

登录时有可能报这样的错：ERROR 2002 (HY000): Can‘t connect to local MySQL server through socket ‘/var/lib/mysql/mysql.sock‘ (2)，原因是/var/lib/mysql的访问权限问题。下面的命令把/var/lib/mysql的拥有者改为当前用户：

$ sudo chown -R mysql:mysql /var/lib/mysql

然后，重启服务：

$ service mysqld restart

接下来登录重置密码：

$ mysql -u root

mysql > use mysql;

mysql > update user set password=password(‘123456‘) where user=‘root‘;

mysql > exit;

**5. 开放3306端口(centos 7没有了iptables）**

|  |
| --- |
| centos7开放端口 |

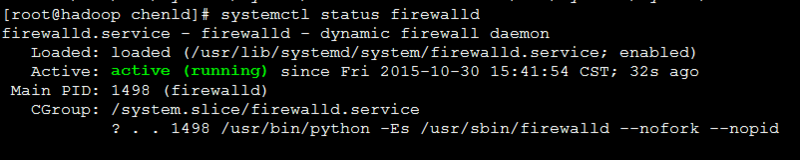
# FirewallD is not running

是你的防火墙还没开。 可以执行 systemctl start firewalld 开启防火墙。

## 相关命令 CentOS 7 上systemctl 的用法

<http://www.linuxidc.com/Linux/2014-11/109236.htm>

### systemctl status firewalld



# firewalld使用简介

<http://www.centoscn.com/CentOS/help/2015/0208/4667.html>

# Centos7 开放端口

Centos升级到7之后，发现无法使用iptables控制Linuxs的端口，google之后发现Centos 7使用firewalld代替了原来的iptables。下面记录如何使用firewalld开放Linux端口：

开启端口

firewall-cmd --zone=public --add-port=80/tcp --permanent

命令含义：

--zone #作用域

--add-port=80/tcp #添加端口，格式为：端口/通讯协议

--permanent #永久生效，没有此参数重启后失效

重启防火墙

firewall-cmd --reload

详细信息可以参考以下资料：

<http://stackoverflow.com/questions/24729024/centos-7-open-firewall-port>

<https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_Firewalls.html>

[centos7下使用yum安装mysql](http://www.mamicode.com/info-detail-503994.html)

在配置[**MySQL数据库**](http://lib.csdn.net/base/14)主从复制集群的时候要保证：

 1.主从服务器操作系统版本和位数一致。

  2.Mysql版本一致。

为了保证稳定性，最好服务器操作系统和Mysql数据库环境一致。

CentOS服务器上Mysql的安装方法可以参见博客：

<http://blog.csdn.net/jhq0113/article/details/43812895>

服务器配置：

Master:192.168.1.18（找到自己的主mysql服务器的ip）

Slave:192.168.1.16（找到自己的从服务器额）

Master(192.168.1.18)服务器:

1.编辑/etc/my.cnf

[root@jhq0229 ~]# vim /etc/my.cnf 

2.配置(写在[mysqld]下面)

**[plain]** [view plain](http://blog.csdn.net/jhq0113/article/details/44263367) [copy](http://blog.csdn.net/jhq0113/article/details/44263367) [print?](http://blog.csdn.net/jhq0113/article/details/44263367)[在CODE上查看代码片](https://code.csdn.net/snippets/619635)

1. [mysqld]
2. datadir=/data/mysql
3. socket=/var/lib/mysql/mysql.sock
4. user=mysql
6. #主从复制配置
7. innodb\_flush\_log\_at\_trx\_commit=1
8. sync\_binlog=1
9. #需要备份的数据库
10. binlog-do-db=orders
11. #不需要备份的数据库
12. binlog-ignore-db=mysql
14. #启动二进制文件
15. log-bin=mysql-bin
17. #服务器ID
18. server-id=1
20. # Disabling symbolic-links is recommended to prevent assorted security risks
21. symbolic-links=0
23. [mysqld\_safe]
24. log-error=/var/log/mysqld.log
25. pid-file=/var/run/mysqld/mysqld.pid

若没有配置binlog-do-db和binlog\_ignore\_db，表示备份全部数据库。

3.重启Mysqld服务

[root@jhq0229 ~]# service mysqld restart

4.为从Mysql创建用户

登录

[root@jhq0229 ~]# mysql -uroot -p  
        Enter password:

创建用户

mysql> create user 'mast'@'192.168.1.16(从服务器地址)' identified by '123456';

配置主从复制权限

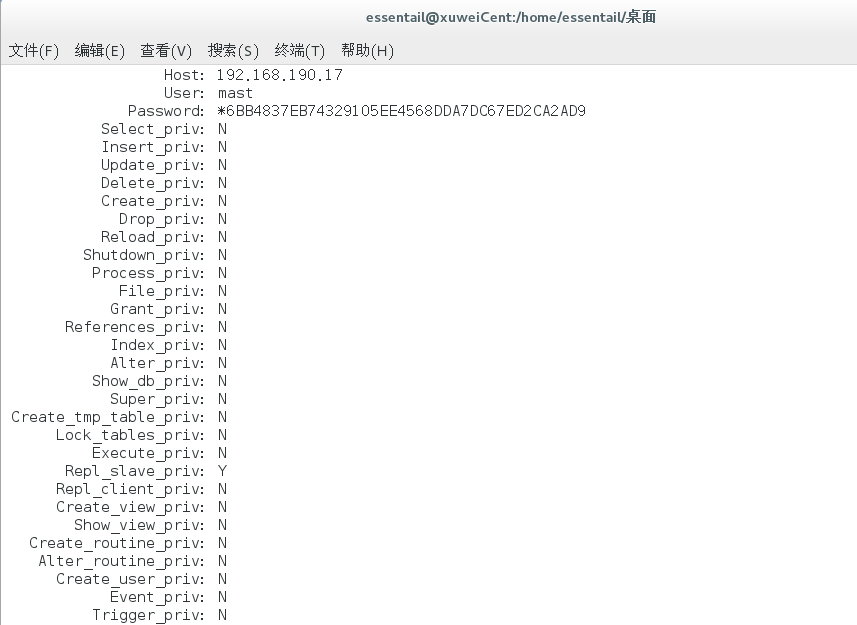
mysql> grant replication slave on \*.\* to 'mast'@'192.168.1.16' identified by '123456';

(数据库授予该用户权限操作 可以在此增加从服务器的权限)

SELECT DISTINCT CONCAT('User: ‘”,user,”'@'”,host,”';') AS query FROM mysql.user;查看是否有这个用户



查看该用户的权限 select \* from mysql.user where user='mast' \G（这里我只给了mast这个user 读的权限 所以只有一个y）



若orders数据库中已经有数据，还需要：

锁定数据

mysql> flush tables with read lock;

将数据导入到从数据库，方法有多种，我是用Navicat复制的，复制完成后：

查看master状态并解锁：

1. mysql**>** show master status;
2. +------------------+----------+--------------+------------------+-------------------+
3. | File      | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |
4. +------------------+----------+--------------+------------------+-------------------+
5. | mysql-bin.000003|  2005 | orders    | mysql    |     |
6. +------------------+------- ---+--------------+------------------+-------------------+
7. 1 row in set (0.00 sec)
9. mysql**>** unlock tables;

Slave(192.168.1.16)服务器：

1.配置服务ID

[root@jhq0113 ~]# vim /etc/my.cnf 

在[mysqld]下面加入

server-id=2

重启Mysql服务

[root@jhq0113 ~]# service mysqld restart

2.配置复制

登录mysql

[root@jhq0113 ~]# mysql -uroot -p

执行

mysql> change master to master\_host='192.168.1.18',

master\_user='mastj',

master\_password='123456',

master\_port=3306,

master\_log\_file='mysql-bin.000003',

master\_log\_pos=2005,

master\_connect\_retry=10;

参数详解：

  master\_host:主服务器的IP。  
  master\_user：配置主服务器时建立的用户名  
  master\_password：用户密码  
  master\_port：主服务器mysql端口，如果未曾修改，默认即可。

  master\_log\_file：日志文件名称，填写查看master状态时显示的File

  master\_log\_pos:日志位置，填写查看master状态时显示的Position

  master\_connect\_retry：重连次数

启动进程

 mysql> start slave;

      检查主从复制状态

**[plain]** [view plain](http://blog.csdn.net/jhq0113/article/details/44263367) [copy](http://blog.csdn.net/jhq0113/article/details/44263367) [print?](http://blog.csdn.net/jhq0113/article/details/44263367)

1. mysql> show slave status\G
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
3. Slave\_IO\_State: Waiting for master to send event
4. Master\_Host: 192.168.1.18
5. Master\_User: mastj
6. Master\_Port: 3306
7. Connect\_Retry: 10
8. Master\_Log\_File: mysql-bin.000003
9. Read\_Master\_Log\_Pos: 2369
10. Relay\_Log\_File: jhq0113-relay-bin.000002
11. Relay\_Log\_Pos: 647
12. Relay\_Master\_Log\_File: mysql-bin.000003
13. Slave\_IO\_Running: Yes
14. Slave\_SQL\_Running: Yes

若Slave\_IO\_Running和Slave\_SQL\_Running均为Yes，则表示连接正常。

此时就可以测试主从复制了。