

一：安装前准备

1: DBProxy 依赖软件包

DBProxy 直接依赖的软件包主要包括 glib(>=2.34)、libevent(>=2.0)、mysqlclient(>=5.1)。间接依赖于 libxml、openssl、libffi 等。

2: yum 安装间接依赖:

```
yum install -y openssl-devel mysql-devel libxml2-devel libffi-devel
```

二：源代码安装

1: 源码安装 glib、libevent，默认都安装在/data/soft 目录下，此目录可根据实际环境自行配置:

a: 安装 libevent:

从 <http://libevent.org/>，下载 libevent-2.0.21-stable.tar.gz，解压缩;

进入源码目录，然后运行命令:

```
./configure --prefix=/data/soft/libevent/  
make && make install
```

b: 安装 glib 库

从 <http://ftp.gnome.org/pub/gnome/sources/glib/> 下载 glib-2.40.0.tar.xz，解压缩;

进入源码目录，运行命令:

```
./configure --prefix=/data/soft/glib  
make && make install
```

2: 安装 zabbix

DBProxy 通过单独部署的 zabbix agent 调用外部脚本(如果你的系统已经有 zabbix，和你系统的中的 zabbix 无关)，实现对后端数据库状态的检测。DBProxy 会向 zabbix agent 传输需要检测的数据库的 ip，port，用户名以及密码四个参数。zabbix 脚本依赖这四个参数实现对对应的数据库的状态的检测。同时，要求脚本返回“errno=0;status=up;errmsg=OK”这种格式的结果信息。DBProxy 根据 errno 的取值判定数据库的死活：errno=0 是活着的；errno=1 是死的。安装 zabbix 步骤如下:

a: 下载: http://www.zabbix.com/.../zabbix_agents.tar.gz。(HH)

b: 将该文件解压到/data/soft/zabbix 中，解压目录也可按照实际情况自行配置。

c: 新建 zabbix 用户和组:

```
groupadd zabbix
```

```
useradd -g zabbix zabbix
```

d: 更改 zabbix 目录属主: `chown -R zabbix:zabbix /data/soft/zabbix/`

e: 更改 zabbix 脚本的文件权限: `chmod 755 /data/soft/zabbix/pxc_check_status.in`

e: 启动 zabbix agent:

```
/data/soft/zabbix/sbin/zabbix_agentd -c /data/soft/zabbix/etc/zabbix_agentd.conf
```

/data/soft/zabbix/etc/zabbix_agentd.conf 是 zabbix_agent 的配置文件，需要注意的是，其中记录了脚本 pxc_check_status.in 的路径，如果 zabbix 解压目录不是

/data/soft/zabbix 的话，需要根据实际情况进行修改：

```
UserParameter=pxc.checkStatus[*],/data/soft/zabbix/pxc_check_status.in $1 $2 $3 $4
```

pxc_check_status.in，就是进行检查后端数据库状态的脚本文件。需要注意的是，该脚本文件的权限必须是：0755，否则，不会执行！而且，该脚本中，会对数据库做一些特殊检查，如果数据库无法满足要求，则会返回状态错误，所以，可以根据实际的情况，对该脚本做一定的修改。

3: 安装 DBProxy

下载源码，运行安装脚本 build.sh 即可，在安装脚本 build.sh 中，需要根据实际情况配置依赖库的路径，以及 DBProxy 的安装路径。默认安装目录都是在“/data/soft/”下，可根据实际情况进行修改。

三：二进制安装

1: 系统要求

Oracle Linux 5.7+/6.3+ x86_64

CPU 密集型(16CPU)，内存 1000M，硬盘空间 500M

2: 系统配置

a: 检查系统配置参数，打开文件的最大数量、内存转储文件的最大大小：

```
ulimit -c # 应该是 unlimited
```

```
ulimit -n # 应该大于等于 131072
```

如果不满足，则需要修改文件/etc/security/limits.conf，重启后生效：

```
*          soft    nofile   131072
*          hard    nofile   131072
*          soft    core     unlimited
```

```
sysctl fs.file-max # 应该大于等于 6815744
```

如果不满足，则修改/etc/sysctl.conf，并执行 sysctl -p 使该内核参数生效：

```
fs.file-max = 6815744
```

b: 如果 DBProxy 读写端口需要绑定在不同虚 IP 上，还要检查确认以下内核参数已打开：

```
sysctl net.ipv4.ip_nonlocal_bind # 应该等于 1
```

如果不满足，则修改/etc/sysctl.conf，并执行 sysctl -p 使该内核参数生效：

```
net.ipv4.ip_nonlocal_bind = 1
```

3: 安装二进制包

从…(HH) 下载二进制包，将二进制包手工解压缩安装至/data/soft 即可。默认安装目录/data/soft/dbproxy，不建议修改。

4: 配置

a: 修改配置文件和脚本，参照以下实例文件，按照实际情况编写配置文件和脚本，新编写的文件放在相同目录下，文件名不带.sample 后缀，配置文件中的配置信息含义，请看其他章节：

bin/pxc_check_status.sample	//zabbix agent 监测后端数据库的脚本
etc/mysql-proxy.cnf.sample	//DBProxy 的配置文件
etc/mysql-proxy.xml.sample	//DBProxy 使用的 XML 文件
etc/zabbix_agentd.cnf.sample	//zabbix agent 的配置文件

如果 dbproxy 安装目录不是 /data/soft 的话, 则需要修改 zabbix_agentd.cnf 中脚本的路径, 具体见下一章节。

b: 进入 INSTALL_DIR/bin 目录, 执行 ./mysql-proxyd init, 此命令执行创建 zabbix 操作系统账号, 设置文件属主和属性等操作。

四: 配置

dbproxy 需要调用安装包中自包含的 zabbix (如果你的系统已经有 zabbix, 和你系统的中的 zabbix 无关) 来得到后端 mysqld 的死活, 所以需要配置自包含的 zabbix。

1: zabbix 配置文件

zabbix_agent 的配置文件为 zabbix_agentd.cnf, 需要注意如下几个重要的参数:

zabbix_agent 的 ip 地址, 配置为 127.0.0.1, 不要修改此参数:

```
Server=127.0.0.1
```

检测脚本的路径, 需要根据实际情况进行修改:

```
userParameter=pxc.checkStatus[*],/data/soft/dbproxy/bin /pxc_check_status $1 $2 $3 $4
```

Timeout 为执行检查脚本的最大的超时时间 (秒), 目前 DBProxy 每 10 秒做一次健康检查, 此参数必须小于等于 DBProxy 检查间隔, 设置为 10 即可。

StartAgents 执行检查的工作进程的数量, 最好设置为大于所有后端数据库的数量。比如数据库 3 个, 此参数可设置为 10。现有的数据库集群最多 4, 5 个节点, 此参数需要设置为 10 或者更大。

```
Timeout=10
```

```
StartAgents=10
```

2: zabbix 检测脚本

zabbix 监测后端数据库状态的脚本为: pxc_check_status, 注意, 此脚本仅支持 PXC 类型数据库, 不适用于普通 MySQL 数据库。可根据实际情况, 自行修改脚本逻辑。当前检测脚本输入为四个参数:

\$1: 需要检测的 backend 的 ip

\$2: 需要检测的 backend 的 port

\$3: 检测用的用户名

\$4: 检测用户名对应的密码

检测脚本需要按如下的格式输出, errno:0 表示执行成功即存活的, 其他表示执行失败即 down:

```
errno=X;status=XXX;errmsg=XXXX
```

3: 账号配置

后端数据库上的账号，必须与 DBProxy 配置文件中的账号信息一致，比如配置文件 mysql-proxy.xml 中的用户名和密码均为 test，则需要在后端数据库上面添加相同账号。

mysql-proxy.xml 文件中的账号信息：

```
<user_info>
  <user name="用户名">
    <password>密码</password>
    <ip_ranges>
      <ip>本机地址</ip>
    </ip_ranges>
  </user>
</user_info>
```

注意，每增加一个账号，实际上需要增加 XML 中至少 5 条配置项：

user_info 账号的用户名、密码和 IP 地址；

conn_limit_rw 读写端口的前端访问限制；

conn_limit_ro 只读端口的前端访问限制；

pool_conf_rw 读写端口的后端连接池配置；

pool_conf_ro 只读端口的后端连接池配置；

五: 运行

1: 源码安装

源码安装完成，并且在进行相关配置之后，可以通过下面的命令启动 DBProxy：

```
/data/soft/DBProxy/mysql-proxy --defaults-file=/data/soft/DBProxy/etc/mysql-proxy.cnf
```

2: 二进制安装

使用安装目录下 bin 目录中的脚本启动即可：

DBProxy 启动：bin/mysql-proxyd start

DBProxy 停止：bin /mysql-proxyd stop

DBProxy 重启：bin /mysql-proxyd restart

DBProxy 运行状态查看：bin /mysql-proxyd status

在后端用户配置正确的情况下，登陆 DBProxy 管理端口，看到后端实例都是 up 状态。说明 DBProxy 配置完成了。

```
# mysql -uadtest -padtest -h127.0.0.1 -P4457
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 1
```

```
Server version: 5.0.99-agent-admin
```

```
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.
```

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> showbackends;

backend_ndx	address	status	type	connected_clients	rw_weight	ro_weight	
0	X.X.X.X:3401	up	rw	0	2	2	
1	X.X.X.X:3402	up	ro	0	4	4	
2	X.X.X.X:3403	up	ro	0	1	1	

3 rows in set (0.00 sec)