patroni 高可用搭建

神州飞象（北京）数据科技有限公司

修订记录

日期	版本	变更标识 缺陷标识	修改描述	作者
2018 年 04 月 02 日	0.1			

目录

Patroni 安装

1 安装环境说明

本例中安装 patroni 开源软件和 patroni 运行所需的中间件和数据库 zookeeper、patroni、 postgresql 软件版本信息如下：

软件	版本
patroni	patroni-1.4.3
zookeeper	zookeeper-3.3.6
postgresql	postgresql-10.1

Python,pip 版本：

Python	Python 2.7.5
pip	pip 9.0.1 (python 2.7)

主机信息如下

系统非最小化安装，默认安装

IP 地址	主机名	操作系统
198.168.191.140	test2	CentOS Linux release 7.4.1708 (Core)
198.168.191.143	test3	CentOS Linux release 7.4.1708 (Core)
198.168.191.142	test1	CentOS Linux release 7.4.1708 (Core)

所有用到的包均在百度云盘的压缩包中（patroni 安装部署需要的包），请解压并按照步骤使用里面的各个包

下载地址: https://pan.baidu.com/s/1nKT5nb180WlEDufj3pKS5Q **提取密码:** wsll

2 运行环境搭建

2.1 安装 openssl-devel、python-devel、libnl、jq、libnfnetlink-devel、haproxy、watchdog（root）

三个节点均需执行

解压:

```
tar -xvf package.tar.gz
cd ./package/package
```

分别进入 openssl-devel、python-devel、libnl、jq、libnfnetlink-devel、watchdog 目录，

用 yum 安装 6 个目录中的包:

```
yum install ./*
```

2.2 Python2.7 的搭建（root）

test3，test1 节点执行

运行 patroni 需要 python2.7 以上版本。

查看 python 版本信息命令:

```
python --version
```

在 centos7.4 下，python 的版本是 2.7.5，不需要进行这一步！

这里用的是 python2.7.1 版本，下载地址

Python-2.7.1 下载: https://www.python.org/ftp/python/2.7.1/Python-2.7.1.tar.bz2

安装 2.7.1 替换掉 2.6.6

解压:

```
tar -jxvf Python-2.7.1.tar.bz2
```

进入目录：

```
cd Python-2.7.1
```

安装：

```
./configure
make all
make install
make clean
make distclean
```

查看版本信息:

```
/usr/local/bin/python2.7 -V
```

建立软连接，使系统默认的 python 指向 python2.7：

```
mv /usr/bin/python /usr/bin/python2.6.6
ln -s /usr/local/bin/python2.7 /usr/bin/python
```

重新检验 Python 版本

```
python -V
```

解决系统 Python 软链接指向 Python2.7 版本后，因为 yum 是不兼容 Python 2.7 的，所以 yum 不能正常工作，我们需要指定 yum 的 Python 版本：

```
vi /usr/bin/yum
```

将文件头部的

`#!/usr/bin/python`

改成

`#!/usr/bin/python2.6.6`

保存&退出！

2.3 pip 的安装（root）

test3,test1 执行

安装 setuptools：

```
cd /home/postgres/package/setuptools
python ./setup.py install
```

安装 pip：

```
cd /home/postgres/package/pip
chmod +x setup.py
python ./setup.py install
pip --version
pip 9.0.1 from /usr/lib/python2.7/site-packages/pip-9.0.1-py2.7.egg
(python 2.7)
```

2.4 创建系统用户（以下的普通用户均为这个用户）（root）

三个节点均需执行

创建系统普通用户（**后文标注的普通用户均为这个用户**），定义密码：

```
useradd postgres
passwd postgres
```

2.5 建立日志目录（普通用户）

三个节点均需创建

```
mkdir -p /home/postgres/logfile/
mkdir -p /home/postgres/etcd
```

3 安装 zookeeper

3.1 解压（root）

三个节点均需执行

解压安装文件：

```
tar -xzvf zookeeper-3.3.6.tar.gz
```

3.2 安装（root）

三个节点均需执行

修改 ZooKeeper 配置文件

在其中一台机器上192.168.0.2，解压缩zookeeper-3.3.4.tar.gz，把conf目录下的zoo_sample.cfg 复制成zoo.cfg文件，修改配置文件conf/zoo.cfg，内容如下所示：

```
tickTime=2000
dataDir=/root/zookeeper
dataLogDir=/home/postgres/logfile
clientPort=2181
initLimit=5
syncLimit=2
server.1=192.168.0.2:2888:3888
server.2=192.168.0.3:2888:3888
server.3=192.168.0.4:2888:3888
```

initLimit：这个配置项是用来配置 Zookeeper 接受客户端（这里所说的客户端不是用户连接 Zookeeper 服务器的客户端，而是 Zookeeper 服务器集群中连接到 Leader 的 Follower 服务器）初始化连接时最长能忍受多少个心跳时间间隔数。当已经超过 10 个心跳的时间（也就是 tickTime）长度后 Zookeeper 服务器还没有收到客户端的返回信息，那么表明这个客户端连接失败。总的时间长度就是 5*2000=10 秒

syncLimit：这个配置项标识 Leader 与 Follower 之间发送消息，请求和应答时间长度，最长不能超过多少个 tickTime 的时间长度，总的时间长度就是 2*2000=4 秒

server.A=B：C：D：其中 A 是一个数字，表示这个是第几号服务器；B 是这个服务器的 ip 地址；C 表示的是这个服务器与集群中的 Leader 服务器交换信息的端口；D 表示的是万一集群中的 Leader 服务器挂了，需要一个端口来重新进行选举，选出一个新的 Leader，而这个端口就是用来执行选举时服务器相互通信的端口。如果是伪集群的配置方式，由于 B 都是一样，所以不同的 Zookeeper 实例通信端口号不能一样，所以要给它们分配不同的端口号。

第二步：远程复制分发安装文件

上面已经在一台机器192.168.0.2上配置完成ZooKeeper，现在可以将该配置好的安装文件远程拷贝到集群中的各个结点对应的目录下

第三步：设置 myid

在我们配置的dataDir指定的目录下面，创建一个myid文件，里面内容为一个数字，用来标识当前主机，conf/zoo.cfg文件中配置的server.X中X为什么数字，则myid文件中就输入这个数字。

第四步：启动 ZooKeeper 集群

在名节点的bin目录下执行：./zkServer.sh start

tailf zookeeper.out 可能发现如下异常

ARN [WorkerSender Thread:QuorumCnxManager@384] - Cannot open channel to 2 at election address slave-02/192.168.0.178:3888

```
java.net.ConnectException: Connection refused
at sun.nio.ch.SocketChannelImpl.checkConnect(Native Method)
at
sun.nio.ch.SocketChannelImpl.finishConnect(SocketChannelImpl.java:567
)
at sun.nio.ch.SocketAdaptor.connect(SocketAdaptor.java:100)
at
org.apache.zookeeper.server.quorum.QuorumCnxManager.connectOne(Quorum
CnxManager.java:371)
at
org.apache.zookeeper.server.quorum.QuorumCnxManager.toSend(QuorumCnxM
anager.java:340)
at
org.apache.zookeeper.server.quorum.FastLeaderElection$Messenger$Worke
rSender.process(FastLeaderElection.java:360)
at
org.apache.zookeeper.server.quorum.FastLeaderElection$Messenger$Worke
rSender.run(FastLeaderElection.java:333)
at java.lang.Thread.run(Thread.java:662)
2012-01-08 06:51:19,420 - WARN [WorkerSender
Thread:QuorumCnxManager@384] - Cannot open channel to 3 at election address
slave-03/192.168.0.177:3888
java.net.ConnectException: Connection refused
at sun.nio.ch.SocketChannelImpl.checkConnect(Native Method)
at
sun.nio.ch.SocketChannelImpl.finishConnect(SocketChannelImpl.java:567
)
at sun.nio.ch.SocketAdaptor.connect(SocketAdaptor.java:100)
at
org.apache.zookeeper.server.quorum.QuorumCnxManager.connectOne(Quorum
CnxManager.java:371)
at
org.apache.zookeeper.server.quorum.QuorumCnxManager.toSend(QuorumCnxM
anager.java:340)
at
org.apache.zookeeper.server.quorum.FastLeaderElection$Messenger$Worke
rSender.process(FastLeaderElection.java:360)
at
org.apache.zookeeper.server.quorum.FastLeaderElection$Messenger$Worke
rSender.run(FastLeaderElection.java:333)
at java.lang.Thread.run(Thread.java:662)
```

由于ZooKeeper集群启动的时候，每个结点都试图去连接集群中的其它结点，先启动的肯定连不上后面还没启动的，所以上面日志前面部分的异常是可以忽略的。通过后面部分可以看到，集群在选出一个Leader后，最后稳定了。

其他结点可能也出现类似问题，属于正常。

第五步：安装验证

可以通过ZooKeeper的脚本来查看启动状态，包括集群中各个结点的角色（或是Leader，或是Follower），如下所示，是在ZooKeeper集群中的每个结点上查询的结果：
./bin/zkServer.sh status
另外，可以通过客户端脚本，连接到ZooKeeper集群上。对于客户端来说，ZooKeeper是一个整体（ensemble），连接到ZooKeeper集群实际上感觉在独享整个集群的服务，所以，你可以在任何一个结点上建立到服务集群的连接，例如
./bin/zkCli.sh -server 192.168.0.2:2181

4 patroni 安装与配置

4.1 安装 python 模块包（root）

test3,test1 节点进行

```
cd /home/postgres/package/python_package
source /home/postgres/.bash_profile #source 数据库环境变量
pip install ./*
```

4.2 安装 patroni（普通用户）

test3,test1 节点进行

这里只需要解压文件：

```
unzip patroni-1.4.3.zip
```

4.3 配置 patroni 参数文件（普通用户）

/home/postgres/patroni-1.4.3/postgres0.xml

/home/postgres/patroni-1.4.3/postgres1.xml

/home/postgres/patroni-1.4.3/postgres2.xml

三个模板参数文件中各项参数的意义:

```
scope: batman ##集群名，二级目录名Etcd:/<namespace>/<scope>/config
#namespace: /service/  ##一级目录名Etcd:/<namespace>/<scope>/config
name: postgresql0 ##patroni节点名

restapi: ##haproxy的监听端口,8008,8009,8010……
  listen: 192.168.191.143:8008
  connect_address: 192.168.191.143:8008
#  certfile: /etc/ssl/certs/ssl-cert-snakeoil.pem
#  keyfile: /etc/ssl/private/ssl-cert-snakeoil.key
#  authentication:
#    username: username
```

```
#    password: password

etcd: ##本数据库节点指向的etcd节点的位置
  host: 192.168.191.143:2379

bootstrap:
  # this section will be written into Etcd:/<namespace>/<scope>/config
after initializing new cluster
  # and all other cluster members will use it as a `global configuration`
  dcs:
    ttl: 30 ##当一段时间内没有人更新dcs中leader key，则视作到期，删除并重新选举新的主节点。 默认 30 seconds
    loop_wait: 10 ## Patroni多久循环一次HA loop。 默认 10 seconds
    retry_timeout: 10 ##如果更新dcs中leader key失败，Patroni 将在这段时间内尝试再更新。默认 10 seconds
    maximum_lag_on_failover: 1048576 ##每一次的HA loop，主节点把wal_position写入到dcs中．而从属于这个主的每个从节点将会用自己的最后一个wal_position与主节点的进行对比。如果主从的差距大于maximum_lag_on_failover ，patroni便不会再让这个差距变的更大。默认 1048576bit
#    master_start_timeout: 300 ##主节点启动超时。默认 300 seconds
#    synchronous_mode: false ##同步流复制配置选项。默认 false
    postgresql:
      use_pg_rewind: true ##pg_rewind开关
#      use_slots: true
      parameters: ##pg参数文件配置
#        wal_level: hot_standby
#        hot_standby: "on"
#        wal_keep_segments: 8
#        max_wal_senders: 5
#        max_replication_slots: 5
#        wal_log_hints: "on"
#        archive_mode: "on"  ##若要开启归档，归档参数默认不写入数据目录里面的参数文件，要取消注释
#        archive_timeout: 1800s ##若要开启归档，归档参数默认不写入数据目录里面的参数文件，要取消注释

#        archive_command: mkdir -p ../wal_archive && test ! -
f ../wal_archive/%f && cp %p ../wal_archive/%f  ##若要开启归档，归档参数默认不写入数据目录里面的参数文件，要取消注释

#      recovery_conf:
#        restore_command: cp ../wal_archive/%f %p  ##恢复参数默认不会写入recovery_conf，要取消注释
```

```
  # some desired options for 'initdb'
  initdb:  # Note: It needs to be a list (some options need values, others
are switches) ##initdb命令参数
  - encoding: UTF8
  - data-checksums

  pg_hba:  # Add following lines to pg_hba.conf after running 'initdb'
## pg_hba文件参数
  - host replication replicator 0.0.0.0/0 md5
  - host all all 0.0.0.0/0 md5
#  - hostssl all all 0.0.0.0/0 md5

  # Additional script to be launched after initial cluster creation (will
be passed the connection URL as parameter)
# post_init: /usr/local/bin/setup_cluster.sh ##在初始群集创建后将要启动的脚
本

  # Some additional users users which needs to be created after initializing
new cluster
##在初始群集创建后将要附加的用户
  users:
    admin:
      password: admin
      options:
        - createrole
        - createdb

postgresql:
  listen: 0.0.0.0:5432 ##数据库监听范围
  connect_address: 192.168.191.143:5432 ##数据库ip::端口
  data_dir: data/postgresql0 ##data目录位置,相对、绝对路径均可以
#  bin_dir:
  pgpass: /tmp/pgpass0 ##pgpass文件位置
  authentication: ##数据库流复制用户和超级用户
    replication:
      username: replicator
      password: rep-pass
    superuser:
      username: postgres
      password: zalando
  parameters:
    unix_socket_directories: '.' ##指定Unix域套接字（S）的目录

watchdog:  ## watchdog默认开启
```

```
  mode: automatic # Allowed values: off, automatic, required
  device: /dev/watchdog
  safety_margin: 5

tags:
    nofailover: false ##不进行故障转移，单个节点设置没影响
    noloadbalance: false
    clonefrom: false
    nosync: false
    replicatefrom: postgres0 ##级联流复制用到，默认不是级联
```

分别在 test3 节点的**/home/postgres/patroni-1.4.2/postgres0.yml** 与 test2 节点的 **/home/postgres/patroni-1.4.2/postgres1.yml** 进行配置，必需指定的参数为下面的参数（**下面针对于测试，只做部分参数的调整,不要直接把下面的参数覆盖到参数文件中,请根据上面的各项参数意义，来配置以下的内容**）：

test3 下的 postgres0.yml

vi /home/postgres/patroni-1.4.2/postgres0.yml

请将以下参数在 postgres0.yml 文件中做相应的修改

```
restapi:
  listen: 192.168.191.143:8008
  connect_address: 192.168.191.143:8008
zookeeper: ##这里的etcd改成zookeeper
  hosts: 192.168.1.140:2181,192.168.1.142:2181,192.168.1.143:2181
      parameters:
        wal_level: logical
  pg_hba:  # Add following lines to pg_hba.conf after running 'initdb'
  - host replication replicator 127.0.0.1/32 md5
  - host all all 0.0.0.0/0 trust
  - host replication ruser 192.168.191.142/32 trust
  - host replication ruser 192.168.191.143/32 trust
##流复制用户加入认证
postgresql:
  listen: 0.0.0.0:5432
  connect_address: 192.168.191.143:5432
  data_dir: /home/postgres/pg10/data
  pgpass: /home/postgres/.pgpass
      username: ruser
      password: '123456'
    superuser:
      username: postgres
      password: '123456'
```

```
unix_socket_directories: '/tmp'
```

test2 下 postgres1.yml

vi /home/postgres/patroni-1.4.2/postgres0.yml

请将以下参数在 postgres1.yml 文件中做相应的修改

```
restapi:
  listen: 192.168.191.142:8009
  connect_address: 192.168.191.142:8009
zookeeper: ##这里的etcd改成zookeeper
  hosts: 192.168.1.140:2181,192.168.1.142:2181,192.168.1.143:2181
     parameters:
       wal_level: logical
  pg_hba:  # Add following lines to pg_hba.conf after running 'initdb'
  - host replication replicator 127.0.0.1/32 md5
  - host all all 0.0.0.0/0 trust
  - host replication ruser 192.168.191.142/32 trust
  - host replication ruser 192.168.191.143/32 trust
##流复制用户加入认证
postgresql:
  listen: 0.0.0.0:5432
  connect_address: 192.168.191.142:5432
  data_dir: /home/postgres/pg10/data
pgpass: /home/postgres/.pgpass
     username: ruser
     password: '123456'
    superuser:
     username: postgres
     password: '123456'
    unix_socket_directories: '/tmp'
```

5 更改数据库参数

详情查看《Patroni 使用维护手册》参数的更改章节

6 加入自动中间件和复制程序的切换

详情查看《中间件与复制程序切换脚本安装部署指南》

7 集群的启停

7.1 关闭现有流复制集群主备库（普通用户）

test3，test1 执行

```
pg_ctl -D /home/postgres/pg10/data/ stop
```

7.2 修改 pg_hba 文件（普通用户）

test3，test1 执行,先关备，再关主

pg_hba.conf 必须有主备的 replication user 的权限，添加:

```
host    replication     ruser          192.168.191.142/32      trust
host    replication     ruser          192.168.191.143/32      trust
```

7.3 启动

zookeeper ⇨ Patroni

7.4 修改 pg_hba 文件（普通用户）

test3，test1 执行

pg_hba.conf 必须有主备的 replication user 的权限，添加:

```
host    replication     ruser          192.168.191.142/32      trust
host    replication     ruser          192.168.191.143/32      trust
```

7.5 启动 zookeeper（root）

```
开启命令
三个zookeeper节点均需执行：
/root/zookeeper-3.3.6/bin/start
查看zookeeper运行状态：
/root/zookeeper-3.3.6/bin/status
```

7.6 启动 Patroni（普通用户）

Patroni 所在节点均需执行:

输出到屏幕运行

```
/home/postgres/patroni-1.4.3/patroni.py /home/postgres/patroni-
1.4.3/postgresmq.yml
```

后台运行

```
nohup /home/postgres/patroni-1.4.3/patroni.py
/home/postgres/patroni-1.4.3/postgresmq.yml  >
/home/postgres/logfile/patroni_log 2>&1 &
```