# PXC源码安装

一、关闭防火墙

1、关闭系统防火墙

service iptables stop

chkconfig iptables off

2、关闭SELinux

sed -i 's/SELINUX=.\*/SELINUX=disabled/g' /etc/selinux/config

setenforce 0

二、下载源码包

1、下载Percona-XtraDB-Cluster-5.7.17-29.20.tar.gz源码包

wget -P /opt/src/ https://www.percona.com/downloads/Percona-XtraDB-Cluster-57/Percona-XtraDB-Cluster-5.7.17-29.20/source/tarball/Percona-XtraDB-Cluster-5.7.17-29.20.tar.gz

2、下载percona-xtrabackup-2.4.9.tar.gz源码包

wget -P /opt/src/ https://www.percona.com/downloads/XtraBackup/LATEST/source/tarball/percona-xtrabackup-2.4.9.tar.gz

3、下载boost\_1\_59\_0.tar.gz源码包

mkdir -p /usr/local/boost

wget -P /usr/local/boost/ https://sourceforge.net/projects/boost/files/boost/1.59.0/boost\_1\_59\_0.tar.gz

注意：一定要下载boost\_1\_59，其他版本会出现问题。

三、安装、编译源码包

1、安装依赖包

# yum -y install libtool ncurses-devel libgcrypt-devel libev-devel git scons gcc gcc-c++ openssl check cmake bison boost-devel asio-devel libaio-devel ncurses-devel readline-devel pam-devel socat libaio automake autoconf vim redhat-lsb check-devel curl curl-devel xinetd

2、编译安装percona-xtrabackup-2.4.9

#解压percona-xtrabackup-2.4.9.tar.gz到/opt/src/目录

tar -xvf /opt/src/percona-xtrabackup-2.4.9.tar.gz -C /opt/src/

cd /opt/src/percona-xtrabackup-2.4.9

mkdir debug #目的是为了防止污染源码

cd debug

#cmake编译 ..指上一级目录

cmake .. -DBUILD\_CONFIG=xtrabackup\_release \

-DWITH\_MAN\_PAGES=OFF \

-DDOWNLOAD\_BOOST=1 \

-DWITH\_BOOST="/usr/local/boost"

#安装

make -j4 && make install

#建立软连接

ln -sf /usr/local/xtrabackup/bin/\* /usr/sbin/

-j指定作业数，作业数是指编译时指定主机的CPU个数。

3、创建MySQL用户、数据目录和日志目录

#创建数据目录及日志目录

mkdir -p /data/mysqldata

mkdir -p /data/logs/mysql

#创建mysql用户组及用户，如果mysql用户存在的话，就不需要了

groupadd mysql

useradd mysql -s /sbin/nologin -M

#赋权限

chown -R mysql.mysql /data/mysqldata

chown -R mysql.mysql /data/logs/mysql

4、解压Percona-XtraDB-Cluster-5.7.17-29.20.tar.gz，使用scons生成garbd、libgalera\_smm.so

#解压Percona-XtraDB-Cluster-5.7.17-29.20.tar.gz：

tar -xvf /opt/src/Percona-XtraDB-Cluster-5.7.17-29.20.tar.gz -C /opt/src/

cd /opt/src/Percona-XtraDB-Cluster-5.7.17-29.20

cd percona-xtradb-cluster-galera

#查看revno值：

cat GALERA-REVISION

#生成libgalera\_smm.so和garbd：

scons -j4 psi=1 --config=force revno="7e383f7" boost\_pool=0 libgalera\_smm.so

scons -j4 --config=force revno="7e383f7" garb/garbd

#创建PXC安装目录：

mkdir -p /usr/local/mysql/{bin,lib}

#拷贝libgalera\_smm.so和garbd到PXC安装目录：

cp garb/garbd /usr/local/mysql/bin/

cp libgalera\_smm.so /usr/local/mysql/lib/

5、编译、安装Percona-XtraDB-Cluster

cd /opt/src/Percona-XtraDB-Cluster-5.7.17-29.20

#为不污染源码，创建debug目录

mkdir debug

cd debug

#编译

cmake .. -DCMAKE\_INSTALL\_PREFIX="/usr/local/mysql" \

-DDEFAULT\_CHARSET=utf8 \

-DDEFAULT\_COLLATION=utf8\_general\_ci \

-DSYSCONFDIR=/etc \

-DMYSQL\_DATADIR="/data/mysqldata" \

-DMYSQL\_UNIX\_ADDR=/data/mysqldata/mysql.sock \

-DMYSQL\_TCP\_PORT=3306 \

-DWITH\_INNOBASE\_STORAGE\_ENGINE=1 \

-DWITH\_CSV\_STORAGE\_ENGINE=1 \

-DWITH\_ARCHIVE\_STORAGE\_ENGINE=1 \

-DWITH\_BLACKHOLE\_STORAGE\_ENGINE=1 \

-DWITH\_MYISAM\_STORAGE\_ENGINE=1 \

-DWITH\_DEBUG=ON \

-DWITH\_WSREP=ON \

-DWITH\_INNODB\_DISALLOW\_WRITES=ON \

-DDOWNLOAD\_BOOST=1 \

-DWITH\_BOOST="/usr/local/boost"

#安装

make -j4 && make install

#拷贝文件到安装目录

cp -R /opt/src/Percona-XtraDB-Cluster-5.7.17-29.20/percona-xtradb-cluster-tests /usr/local/mysql/

#创建软连接mysql

ln -sf /usr/local/mysql/bin/\* /usr/sbin/

#拷贝启动文件到系统启动目录

cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysql

#拷贝监控文件到xinetd管理目录

cp /usr/local/mysql/xinetd.d/mysqlchk /etc/xinetd.d/

#加入动态库

echo "/usr/local/mysql/lib" >> /etc/ld.so.conf

ldconfig

注意：

编译PXC时，-DWITH\_WSREP=ON

编译Percona-Server时，此参数不用设置

开启debug模式，-DWITH\_DEBUG=ON

6、配置MySQL健康检查

vim /etc/xinetd.d/mysqlchk

将

server = /usr/bin/clustercheck

改为

server = /usr/sbin/clustercheck

#添加mysqlchk服务

vim /etc/services

添加

mysqlchk 9200/tcp #mysql

四、数据库配置

1、对my.cnf配置详见wiki上[Percona Xtradb Cluster 5.7 my.cnf 标准格式](/pages/viewpage.action?pageId=5931303)

[mysqld\_safe]

pid-file = /data/mysqldata/mysql.pid

[mysqld]

# GENERAL

#basedir = /usr/local/mysql

datadir = /data/mysqldata

tmpdir = /tmp

socket = /data/mysqldata/mysql.sock

pid\_file = /data/mysqldata/mysql.pid

user = mysql

port = 3306

character-set-server = utf8

bind-address = 0.0.0.0

server-id = 172161105 #自定义

skip-name-resolve = 1

# INNODB

# This changes how |InnoDB| autoincrement locks are managed and is a requirement for Galera

default\_storage\_engine = InnoDB

innodb\_buffer\_pool\_size = 6G #80%Memory

innodb\_autoinc\_lock\_mode = 2

innodb\_buffer\_pool\_instances = 8

innodb\_thread\_concurrency = 16 #cpu cores

innodb\_log\_buffer\_size = 32M

innodb\_log\_file\_size = 1024M

innodb\_online\_alter\_log\_max\_size = 512M

innodb\_open\_files = 1024

innodb\_purge\_threads = 2

innodb\_data\_home\_dir = /data/mysqldata

#innodb\_data\_file\_path = ibdata1:256M:autoextend

innodb\_read\_io\_threads = 8 #20%innodb\_thread\_concurrency

innodb\_write\_io\_threads = 8 #20%innodb\_thread\_concurrency

innodb\_file\_per\_table = 1

innodb\_flush\_method = O\_DIRECT

innodb\_flush\_log\_at\_trx\_commit = 2

innodb\_max\_dirty\_pages\_pct = 70

innodb\_file\_format = Barracuda

innodb\_file\_format\_max = Barracuda

innodb\_buffer\_pool\_dump\_at\_shutdown = OFF

innodb\_buffer\_pool\_load\_at\_startup = OFF

innodb\_undo\_log\_truncate = ON

innodb\_undo\_tablespaces = 4

innodb\_strict\_mode = OFF

# MyISAM

key\_buffer\_size = 32M

# LOGS

#general\_log = 1

#general\_log\_file = /data/logs/mysql/mysql\_general.log

log\_timestamps = system

log\_warnings = 2

log\_error = /data/logs/mysql/mysql\_error.log

slow\_query\_log = ON

slow\_query\_log\_file = /data/logs/mysql/mysql\_slow.log

log\_queries\_not\_using\_indexes = 0

long\_query\_time = 1

expire\_logs\_days = 15

log-bin = mysql-bin.log

innodb\_print\_all\_deadlocks = 1

relay-log = relay-log

relay-log-index = relay-log

# BINLOG

# In order for Galera to work correctly binlog format should be ROW

binlog\_format = ROW

binlog\_cache\_size = 16M

max\_binlog\_size = 512M

# OTHER

tmp\_table\_size = 32M

max\_heap\_table\_size = 128M

query\_cache\_type = 0

query\_cache\_size = 0M

max\_connections = 1024

thread\_cache\_size = 200

open\_files\_limit = 65535

optimizer-switch = "mrr=on,mrr\_cost\_based=off,batched\_key\_access=on" #开启mmr功能

auto\_increment\_offset = 1

join\_buffer\_size = 5M

sort\_buffer\_size = 5M

sql\_mode = STRICT\_TRANS\_TABLES,NO\_ENGINE\_SUBSTITUTION

performance\_schema = ON

default\_password\_lifetime = 0

# WSREP

wsrep\_auto\_increment\_control = OFF

# Path to Galera library

#wsrep\_provider = /usr/lib64/libgalera\_smm.so

wsrep\_provider = /usr/local/mysql/lib/libgalera\_smm.so

# Cluster connection URL

wsrep\_cluster\_address = gcomm://172.16.1.102:4567,172.16.1.103:4567 #集群内主机

# Node #1 address

wsrep\_node\_address = 172.16.1.102 # 本机IP地址

# SST method

wsrep\_sst\_method = xtrabackup-v2

# Cluster name

wsrep\_cluster\_name = Yooli\_Hadoop\_Cluster #自定义

# Authentication for SST method

wsrep\_sst\_auth = "sstuser:sstuser"

wsrep\_max\_ws\_rows = 0

wsrep\_max\_ws\_size = 2147483647

wsrep\_slave\_threads = 16

wsrep\_provider\_options = "base\_port=4567;gcache.size=20G;gcache.page\_size=512M;gcs.fc\_limit=256;gcs.fc\_master\_slave=yes"

pxc\_strict\_mode = PERMISSIVE

#Replication

log\_slave\_updates = 1

gtid-mode = on

enforce\_gtid\_consistency = true

binlog\_checksum = CRC32

slave\_allow\_batching = 1

master\_verify\_checksum = 1

slave\_sql\_verify\_checksum = 1

master\_info\_repository = TABLE

relay\_log\_info\_repository = TABLE

[client]

socket = /data/mysqldata/mysql.sock

port = 3306

[mysql]

default-character-set = utf8

prompt ="\\u@\\h : \\d \\R:\\m:\\s>"

no-auto-rehash

2、启动MySQL

集群内第一个节点应以bootstrap-pxc的方式启动

service mysql bootstrap-pxc

3、修改root密码，根据错误日志中的初始化密码登陆，并创建sstuser账户

# mysql -uroot -p

> alter user 'root'@'localhost' identified by '';

#创建sstuser账户，并赋权

> create user 'sstuser'@'localhost' identified by 'sstuser';

> grant reload,lock tables,process,replication client on \*.\* to 'sstuser'@'localhost';

> flush privileges;

安装完毕！

当进行gdb调试时，可能会遇到debuginfo-install没有安装的问题，如下：

Program exited with code 01.

Missing separate debuginfos, use: debuginfo-install glibc-2.12-1.192.el6.x86\_64 keyutils-libs-1.4-5.el6.x86\_64 krb5-libs-1.10.3-57.el6.x86\_64 libaio-0.3.107-10.el6.x86\_64 libcom\_err-1.41.12-22.el6.x86\_64 libgcc-4.4.7-18.el6.x86\_64 libselinux-2.0.94-7.el6.x86\_64 libstdc++-4.4.7-18.el6.x86\_64 nss-softokn-freebl-3.14.3-23.3.el6\_8.x86\_64 openssl-1.0.1e-57.el6.x86\_64 zlib-1.2.3-29.el6.x86\_64

解决方法为：

1、修改文件/etc/yum.repos.d/CentOS-Debuginfo.repo中的enabled参数，将其值改为1

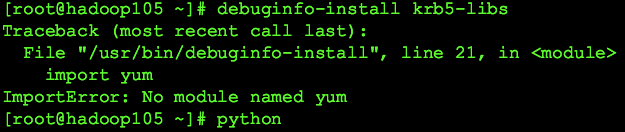
即：enabled=1

2、使用命令安装：

yum install nss-softokn-debuginfo -nogpgcheck

yum install yum-utils

3、安装debuginfo-install后，在执行debuginfo-install时，会遇到如下问题：



解决方法为（大家都知道的）：

vim /usr/bin/debuginfo-install

将

#!/usr/bin/python

改为

#!/usr/bin/python2.6