# 数据库服务器部署规范

规范见附件：

规范中可能会存在没有考虑到的问题，如遇到，请大家自行修改。

[Edit Document](/plugins/servlet/confluence/editinword/8094602/attachments/数据库服务器部署规范.docx)

数据库服务器部署规范

有利网MySQL DBA

2018年03月20日

**目录**

一 硬件及版本               3

二 程序组件               4

2.1               python 组件               4

2.2               salt-minion 组件               4

2.3               系统工具组件               5

2.4               yum 软件源组件               5

2.5               PXC 组件               5

2.6               Percona-Server 组件               5

2.7               mysql 工具组件               6

2.8               keepalived 组件               6

2.9               zabbix 组件               6

三 配置               7

3.1               OS 配置               7

3.2               账户               7

3.3               salt-minion 配置               8

3.4               my.cnf 配置               8

3.5               keepalived 配置               9

3.6               crontab 设置               9

3.7               zabbix 主要参数设置               10

四 安装               11

4.1 关闭防火墙               11

4.2 卸载已安装 MySQL               11

4.3 安装 yum 源               11

4.4 安装系统工具               11

4.5 安装 python2.7               12

4.6 安装 Percona-XtraDB-Cluster 或 Percona-Server               13

4.7 安装 keepalived               20

4.8 安装 zabbix-agent               22

4.9 安装 salt-minion               23

4.10 创建系统用户               23

4.11 设置定时任务               24

4.12 修改 numa 及 elevvator 并重启服务器               25

以下内容根据当前线上数据库服务器配置为参考。我们将从 三个方面来规范数据库服务器的 部署，具体如下：

## **一 硬件及** **版本**

以当前线上服务器为 参考 ， 服务器硬件要求见下表：

|  |  |  |
| --- | --- | --- |
| 服务器硬件要求 | 参数 | 参考值 |
| CPU核数 | >=32 |
| Memory | >=64G |
| 磁盘空间 ( 数据目录 ) | =1 .2T(Min) |
| >=1.8T |
| 磁盘整列 | RAID10 |
| RAID5 |
| PCIE-SSD |

系统及相关软件 版本见下 表 ：

|  |  |
| --- | --- |
| 名称 | 版本 |
| 操作系统版本 | CentOS release 6.9 (Final) |
| python | 2.7.13 |
| salt-minon | 2016.11.6 |
| e pel-release | 6-8 |
| percona-release | 0.1-4 |
| Percona-XtraDB-Cluster | 5.7.17 |
| Percona-Server | >=5.7.17 |
| percona-xtrabackup | >=2.4.7 |
| keepalived | >=1.2.13 |
| sysbench | >=1.0.13 |
| zabbix | >=3.4.4-2 |

## **二 程序组件**

### **2.1** **python** **组件**

|  |  |
| --- | --- |
| python | 依赖包 |
| paramiko |
| psutil |
| pexpect |
| simplejosn |
| MySQL-python |

使用 pip install 命令进行安装依赖包。

### **2.2** **salt-minion** **组件**

|  |  |  |
| --- | --- | --- |
| salt-minion | 依赖包 | 版本 |
| cffi | 11.5 |
| jinja2 | 2.10 |
| msgpack-python | 0.5.5 |
| mysql-python | 1.2.5 |
| pycparser | 2.18 |
| pycrypto | 2.6.1 |
| PyPAML | 3.12 |
| PyZMQ | 17.0.0 |
| Tornado | =4.5.3 |
| ZMQ | 4.1.6 |

安装 salt-minion 时，依赖包会自动安装，安装后使用 salt-minion --versions-report 命令进行检测，其中，红色部分的 Tornado 版本为指定版本，如大于本版本，请卸载后安装指定版。

### **2.3** **系统工具组件**

|  |
| --- |
| 系统工具 |
| sysbench、screen、innotop、sysstat、iftop、socat 、 wget、xz、curl、tree、gdb、perf、glibc、make、autoconf、openssl、gcc、gcc-c++ 、 mai l x 、ntp、vixie-cron |

### **2.4** **yum** **软件源组件**

|  |  |  |
| --- | --- | --- |
| yum源组件 | 组件 | 版本 |
| epel-release | 6-8 |
| percona-release | 0.1-4 |

### **2.5** **PXC** **组件**

|  |  |  |
| --- | --- | --- |
| PXC 组件 | 组件 | 版本 |
| Percona-XtraDB-Cluster | 5.7.17  或  5.7.21 |
| Percona-XtraDB-Cluster-server |
| Percona-XtraDB-Cluster-client |
| Percona-XtraDB-Cluster-shared |
| Percona-XtraDB-Cluster-devel |
| Percona-XtraDB-Cluster-garbd |

### **2.6** **Percona-Server** **组件**

|  |  |  |
| --- | --- | --- |
| Percona-Server | 组件 | 版本 |
| Percona-Server-server | >=5.7.17 |
| Percona-Server-client |
| Percona-Server-shared |
| Percona-Server-devel |

### **2.7** **mysql** **工具组件**

|  |  |
| --- | --- |
| 组件 名称 | 组件 功能 |
| percona-toolkit | pt工具 |
| percona-xtrabackup-24 | 备份工具 |

### **2.8** **keepalived** **组件**

|  |  |
| --- | --- |
| keepalived | 组件功能 |
| 实现高可用 |

### **2.9** **zabbix** **组件**

|  |  |  |
| --- | --- | --- |
| zabbix | 组件 | 版本 |
| zabbix-agent | >= 3.4 .4-2 |
| zabbix-sender | >= 3.4 .4-2 |

## **三 配置**

### **3.1** **OS** **配置**

|  |  |
| --- | --- |
| 名称 | 参考 值 |
| iptables | off |
| selinux | off |
| elevator | deadline |
| n oop( 适用于PCIE-SSD ) |
| numa | off |

### **3.2** **账户**

|  |  |
| --- | --- |
| 名称 | 账户 |
| system user | dan.su |
| zhaojing.peng |
| qihang.li |
| jianwei.zheng |
| xihui.zhang |
| mysql user | dan.su@% |
| zhaojing.peng@% |
| qihang.li@% |
| jianwei.zheng@% |
| xihui.zhang@% |
| cactiuser@%     # (zabbix) |
| replication@%     # (replication) |
| backupuser@localhost     # (backup) |
| backupuser@127.0.0.1     # (backup) |
| sstuser@localhost     # (PXC sst) |

### **3.3** **salt-minion** **配置**

配置文件所在目录为/etc/salt/minion，需要配置的参数如下：

|  |  |  |
| --- | --- | --- |
| 文件位置 | 参数 | 参考 值 |
| /etc/salt/minion | default\_include | minion.d/\*.conf |
| master | salt-master IP |
| id | hostname |
| log\_file | /var/log/salt/minion |
| key\_logfile | /var/log/salt/key |

### **3.4** **my.cnf** **配置**

MySQL 配置文件所在位置为/etc/my.cnf，一部分为通用配置，如下：

|  |  |  |
| --- | --- | --- |
| 文件名称 | 参数 | 参考值 |
| /etc/my.cnf | server-id | 自定义( 唯一 ) |
| innodb\_buffer\_pool\_size | =80% Memory |
| innodb\_thread\_concurrency | <=Processor(Cores) |
| innodb\_write\_io\_threads | 20%innodb\_thread\_concurrency |
| innodb\_read\_io\_threads | 20%innodb\_thread\_concurrency |
| optimizer-switch | >5.7.17(use\_index\_extensions=off)  <=5.7. 17 (use\_index\_extensions=on) |
| innodb\_io\_capacity | sas= 200 ( 默认值 ) |
| PCIE-SSD=2048 |
| innodb\_io\_capacity\_max | sas=20 00 ( 默认值 ) |
| PCIE-SSD =4096 |
| innodb\_purge\_threads | sas= 2-4 |
| PCIE-SSD>= 8 |
| innodb\_max\_dirty\_pages\_pct | 70% |

另一部分为PXC 集群 专有配置，如下：

|  |  |  |
| --- | --- | --- |
| 文件名称 | 参数 | 参考值 |
| /etc/my.cnf | wsrep\_cluster\_address | gcomm:// |
| wsrep\_node\_address | 本地IP |
| wsrep\_cluster\_name | 集群名称需 自定义 |
| wsrep\_sst\_auth | sstuser:sstuser |
| wsrep\_max\_ws\_rows | 0 |
| wsrep\_max\_ws\_size | 2147483647 |
| wsrep\_provider\_options | 其中 gcache.size=20G |
| wsrep\_slave\_threads | 20 ( 与物理CPU核数保持一致 ) |

### **3.5** **keepalived** **配置**

|  |  |
| --- | --- |
| 文件名称 | 参数 |
| /etc/keepalived/keepalived.conf | router\_id |
| --vip |
| interface |
| virtual\_router\_id |
| priority |
| virtual\_server |
| real\_server |
| connect\_port |

### **3.6** **crontab** **设置**

|  |  |  |
| --- | --- | --- |
| 名称 | 节点 | 定时任务 |
| crontab | Primary | ntpdate |
| mysql\_flush\_logs .py |
| Secondary | ntpdate |
| mysql\_flush\_logs.py |
| db\_backup.py |
| backup\_binlog.py |
| Readonly | ntpdate |
| All node | agent\_update.sh  agent\_update\_exception.sh  agent\_monitor.sh |

### **3.7** **zabbix** **主要参数设置**

|  |  |  |
| --- | --- | --- |
| 文件名称 | 参数 | 参考值 |
| /etc/zabbix/zabbix-agentd.conf | Server | 192.168.1.179 |
| StartAgents | 5 |
| ServerActive | 192.168.1.179:10051 |
| Hostname | hostname |

## **四 安装**

### **4.1 关闭防火墙**

**4.1.1** **查看当前操作系统发行版本** **、CPU、Memory、磁盘空间及磁盘阵列信息**

**4.1.2 关闭防火墙并设置为开机不自启**

# service iptables stop

# chkconfig iptables off

**4.1.3 关闭selinux** **：**

# sed -i 's/SELINUX=enable/SELINUX=disabled/g' /etc/selinux/config

# setenforce 0

### **4.2 卸载已安装** **MySQL**

**4.2.1 查看系统是否已安装MySQL，如果存在，将其删除**

# rpm –qa | grep mysql

# yum erase -y mysql-libs-5 \*

### **4.3 安装** **yum** **源**

**4.** **3.1** **安装epel-release**

# yum install epel-release

**4.** **3.2** **安装percona-release**

# rpm –ivh https://www.percona.com/downloads/percona-release/redhat/0.1-4/percona-release-0.1-4.noarch.rpm

### **4.** **4** **安装系统工具**

# yum install –y screen innotop sysstat iftop socat wget xz curl tree gdb perf glibc make autoconf openssl openssl-devel gcc gcc-c++ vixie-cron ntp mailx

### **4.5 安装** **python2.7**

**4.5.1 进入** **/opt/src** **目录下，下载源码包**

# cd /opt/src

# wget https://www.python.org/ftp/python/2.7.13/Python-2.7.13.tar . xz

**4.5.2 解压** **Python-2.7.13.tar** **.** **xz**

# xz –df Python-2.7.13.tar . xz && tar –xvf Python-2.7.13.tar

**4.5.3 进入** **Python-2.7.13** **目录**

# cd Python-2.7.13

# ./configure

# make && make altinstall

# echo /usr/local/lib/python2.7 > /etc/ld.so.conf.d/python2.7.conf

# ldconfig

# restorecon –RF /

# mv /usr/bin/python /usr/bin/python.bak

# ln –sf /usr/local/bin/python2.7 /usr/bin/python

4.5.4 修改yum

#vi m /usr/bin/yum

#!/usr/bin/python

改为

#!/usr/bin/python2.6

4.5.5 安装setuptools

# wget https://bootstrap.pypa.io/ez\_setup.py -O - | python

4.5.6 安装pip

# curl https://bootstrap.pypa.io/get-pip.py | python

4.5.7 安装python需要的依赖包

#pip install paramiko

#pip install psutil

#pip install pexpect

#pip install simplejson

#pip install MySQL-python

### **4.6 安装** **Percona-XtraDB-Cluster** **或** **Percona-Server**

**4.6.1 下载Percona-XtraDB-Cluster相关 rpm包**

# wget -P /opt/src https://www.percona.com/downloads/Percona-XtraDB-Cluster-57/Percona-XtraDB-Cluster-5.7.17-29.20/binary/redhat/6/x86\_64/Percona-XtraDB-Cluster-57-5.7.17-29.20.3.el6.x86\_64.rpm

# wget -P /opt/src https://www.percona.com/downloads/Percona-XtraDB-Cluster-57/Percona-XtraDB-Cluster-5.7.17-29.20/binary/redhat/6/x86\_64/Percona-XtraDB-Cluster-shared-57-5.7.17-29.20.3.el6.x86\_64.rpm

# wget -P /opt/src https://www.percona.com/downloads/Percona-XtraDB-Cluster-57/Percona-XtraDB-Cluster-5.7.17-29.20/binary/redhat/6/x86\_64/Percona-XtraDB-Cluster-devel-57-5.7.17-29.20.3.el6.x86\_64.rpm

# wget -P /opt/src https://www.percona.com/downloads/Percona-XtraDB-Cluster-57/Percona-XtraDB-Cluster-5.7.17-29.20/binary/redhat/6/x86\_64/Percona-XtraDB-Cluster-server-57-5.7.17-29.20.3.el6.x86\_64.rpm

# wget -P /opt/src https://www.percona.com/downloads/Percona-XtraDB-Cluster-57/Percona-XtraDB-Cluster-5.7.17-29.20/binary/redhat/6/x86\_64/Percona-XtraDB-Cluster-client-57-5.7.17-29.20.3.el6.x86\_64.rpm

# wget -P /opt/src https://www.percona.com/downloads/Percona-XtraDB-Cluster-57/Percona-XtraDB-Cluster-5.7.17-29.20/binary/redhat/6/x86\_64/Percona-XtraDB-Cluster-garbd-57-5.7.17-29.20.3.el6.x86\_64.rpm

**4.6.2 安装PXC**

# yum localinstall –y Percona-XtraDB-Cluster-57-5.7.17-29.20.3.el6.x86\_64.rpm Percona-XtraDB-Cluster-shared-57-5.7.17-29.20.3.el6.x86\_64.rpm Percona-XtraDB-Cluster-devel-57-5.7.17-29.20.3.el6.x86\_64.rpm Percona-XtraDB-Cluster-server-57-5.7.17-29.20.3.el6.x86\_64.rpm Percona-XtraDB-Cluster-client-57-5.7.17-29.20.3.el6.x86\_64.rpm Percona-XtraDB-Cluster-garbd-57-5.7.17-29.20.3.el6.x86\_64.rpm

**4.6.3 下载Percona-Server 相关rpm包**

**# wget** -P /opt/src **https://www.percona.com/downloads/Percona-Server-LATEST/Percona-Server-5.7.17-13/binary/redhat/6/x86\_64/Percona-Server-client-57-5.7.17-13.1.el6.x86\_64.rpm**

**# wget** -P /opt/src **https://www.percona.com/downloads/Percona-Server-LATEST/Percona-Server-5.7.17-13/binary/redhat/6/x86\_64/** Percona-Server-server-57-5.7.17-13.1.el6.x86\_64.rpm

**# wget** -P /opt/src **https://www.percona.com/downloads/Percona-Server-LATEST/Percona-Server-5.7.17-13/binary/redhat/6/x86\_64/** Percona-Server-shared-57-5.7.17-13.1.el6.x86\_64.rpm

**# wget** -P /opt/src **https://www.percona.com/downloads/Percona-Server-LATEST/Percona-Server-5.7.17-13/binary/redhat/6/x86\_64/** Percona-Server-devel-57-5.7.17-13.1.el6.x86\_64.rpm

**4.6.4 安装Percona-Server**

# yum localinstall –y **Percona-Server-client-57-5.7.17-13.1.el6.x86\_64.rpm** Percona-Server-server-57-5.7.17-13.1.el6.x86\_64.rpm Percona-Server-shared-57-5.7.17-13.1.el6.x86\_64.rpm Percona-Server-devel-57-5.7.17-13.1.el6.x86\_64.rpm

4. 6 . 5 创建数据目录及日志目录

# mkdir –p /data/mysqldata

# mkdir –p /data/logs/mysql

4. 6 . 6 更改/data /mysqldata 与 /data/logs/mysql 目录的所属用户为mysql，并赋权

# chown –R mysql:mysql /data/mysqldata

# chown –R mysql:mysql /data/logs

# chmod –c 755 /data/mysqldata –R

# chmod –c 755 /data/logs –R

4.6.7 配置my.cnf，配置文件位置 /etc/my.cnf ，其中 # WSREP 部分为Percona-XtraDB-Cluster专有配置，将其去掉为Percona-Server的配置，如下

[mysqld\_safe]

pid-file                                                                                     = /data/mysqldata/mysql.pid               #pi d所在目录

[mysqld]

# GENERAL

#basedir                                                                                     = /usr/local/mysql

datadir                                                                                     = /data/mysqldata                                           #数据目录

tmpdir                                                                                     = /tmp

socket                                                                                     = /data/mysqldata/mysql.sock               #socket所在目录

pid\_file                                                                                     = /data/mysqldata/mysql.pid               #pi d所在目录

user                                                                                                   = mysql

port                                                                                                   = 3306

character-set-server                                                         = utf8

bind-address                                                                       = 0.0.0.0

server-id                                                                                     = 1518319998                                           #可通过 select unix\_timestamp (now()) 获取

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                                           = 50 G                                                                                     #=80%的内存空间

innodb\_autoinc\_lock\_mode                                           = 2

innodb\_buffer\_pool\_instances                             = 8

innodb\_thread\_concurrency                                           = 64                                                                                     #=CPU线程数

innodb\_log\_buffer\_size                                           = 32M

innodb\_log\_file\_size                                                         = 1024M

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

innodb\_open\_files                                                         = 1024

innodb\_purge\_threads                                                         = 8                                                                       #可调整 ( 建议 : 机械硬盘2-4, PCIE-SSD 为8 )

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

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

innodb\_read\_io\_threads                                           = 16                                                                                     #20%的 innodb\_thread\_concurrency

innodb\_write\_io\_threads                                           = 16                                                                                     #20%的 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                                                                                     #可调整 , 建议为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

innodb\_doublewrite                                                         = OFF

innodb\_io\_capacity                                                         = 2048               #机械硬盘默认 (200) 即可，PCIE-SSD建议为2048

innodb\_io\_capacity\_max                                           = 4096               #机械硬盘默认 (2000) 即可，PCIE-SSD建议为4096

# 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" #如果 5.7.17<version<5.7.21, 将 use\_index\_extensions 设置为off

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

#interactive\_timeout                                                         = 300

#wait\_timeout                                                                       = 300

#max\_user\_connections                                           = 50

# WSREP

wsrep\_auto\_increment\_control                             = OFF

# Path to Galera library

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

# Cluster connection URL

wsrep\_cluster\_address                                           = gcomm://192.168.3 4 .30:4567,192.168.3 4 .31:4567 #集群内数据库地址

# Node #1 address

wsrep\_node\_address                                                         = 192.168.1.30               #本机IP地址

# SST method

wsrep\_sst\_method                                                         = xtrabackup-v2

# Cluster name

wsrep\_cluster\_name                                                         = Yooli\_Fuscent\_Cluster               #自定义集群Name

# Authentication for SST method

wsrep\_sst\_auth                                                                       = "sstuser:sstuser"               #设置用户及密码，

wsrep\_max\_ws\_rows                                                         = 0

wsrep\_max\_ws\_size                                                         = 2147483647

wsrep\_slave\_threads                                                         = 32

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

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

4.6.8 启动mysql服务

当启动PXC集群中的第一个节点的mysql服务时，执行：

# service mysql bootstrap-pxc

4.6.9 创建用户

# mysql –uroot –p

Enter password:

> CREATE USER ‘sstuser’@’localhost’ IDENTIFIED BY ‘sstuser’;

> GRANT RELOAD,PROCESS,LOCK TABLES,REPLICATION CLIENT ON \*.\* TO ‘sstuser’@’localhost’;

> CREATE USER IF NOT EXISTS 'replication'@'%';

> ALTER USER 'replication'@'%' IDENTIFIED WITH 'mysql\_native\_password' AS '\*9B03F857FF4C36B08406FED4A0115A4300A093F0';

> GRANT REPLICATION CLIENT, REPLICATION SLAVE ON \*.\* TO 'replication'@'%';

> CREATE USER IF NOT EXISTS 'cactiuser'@'%';

> ALTER USER 'cactiuser'@'%' IDENTIFIED WITH 'mysql\_native\_password' AS '\*43DD7940383044FBDE5B177730FAD3405BC6DAD7';

> GRANT PROCESS, SELECT, SUPER ON \*.\* TO 'cactiuser'@'%';

> FLUSH PRIVILEGES;

备注：个人MySQL账户可自行定义。

4.6.10 启动PXC集群中的第二个节点，数据会从第一个节点拉过来

# service mysql start

4.6.1 1 配置见证服务器garb，位置在/etc/sysconfig/garb

# Copyright (C) 2012 Codershipy

# This config file is to be sourced by garb service script.

# A comma-separated list of node addresses (address[:port]) in the cluster

GALERA\_NODES ="192.168.35.26:4567 192.168.35.27:4567"                             #需要见证的集群内节点IP地址及端口

# Galera cluster name, should be the same as on the rest of the nodes.

GALERA\_GROUP ="CARD\_CLUSTER"                                                                                     # 与需要见证的集群中 wsrep\_cluster\_name 相同

# Optional Galera internal options string (e.g. SSL settings)

# see http://galeracluster.com/documentation-webpages/galeraparameters.html

# GALERA\_OPTIONS=""

# Log file for garbd. Optional, by default logs to syslog

# Deprecated for CentOS7, use journalctl to query the log for garbd

LOG\_FILE="/tmp/garb.log"                                                                                                   #日志所在位置

4.6.1 2 修改Linux用户nobody

# vim /etc/passwd

nobody:x:99:99:Nobody:/:/sbin/nologin

改为

nobody:x:99:99:Nobody:/:/sbin/ bash

4.6.13 启动garb服务

# service garb start

**4.6.** **14** **安装percona-toolkit与percona-xtrabackup：**

# yum install –y percona-xtrabackup

# yum install –y percona-toolkit

### **4.7 安装** **keepalived**

4.7.1 yum安装keepalived

# yum install –y keepalived

4.7.2 配置keepalived，配置文件所在目录/ etc/keepalived/keepalived.conf

#! Configuration File for keepalived

global\_defs {

   notification\_email {

        dbagroup@yooli.com

   }

   notification\_email\_from dba@yooli.com

   smtp\_server smtp.mail.yooli.com

   smtp\_connect\_timeout 5

   router\_id KP\_FUSCENT\_HA                             #集群内相同的一个名称，需自定义

}

vrrp\_script check\_run {

     script "/etc/keepalived/check\_mysql\_cluster.sh --vip =192.168. 1 .230"               # V IP需自定义

     interval 15               #15秒执行一次

     timeout 10                             #脚本执行超过10秒即为超时,触发keepalived ip切换动作

}

vrrp\_sync\_group VG1 {

     group {

          VI\_1

     }

}

vrrp\_instance VI\_1 {

    state backup

    interface bond0               #网络接口

    virtual\_router\_id 230               #集群内需相同 , 需要修改#

    priority 100     #主服务器值大于备服务器,需要修改#

    nopreempt        #不主动重新取得IP地址

    advert\_int 1

    authentication {        #验证类型与密码 集群内相同

        auth\_type PASS

        auth\_pass 1111

    }

    track\_script {

       check\_run

    }

    virtual\_ipaddress {

        192.168.1.230 #与-- vip 相同，需要修改

    }

    notify\_master "/etc/keepalived/keepalive\_notify.sh --stateinfo=Master"  #状态触发后执行的脚本

    notify\_backup "/etc/keepalived/keepalive\_notify.sh --stateinfo=Backup"

    notify\_fault "/etc/keepalived/keepalive\_notify.sh --stateinfo=fault"

}

virtual\_server 192.168.1.230 3306{ #根据实际分配的IP和端口进行设置

    delay\_loop 6

    lb\_algo wcl

    lb\_kind DR

    persistence\_timeout 30

    protocol TCP

real\_server 192.168.1.30 3306 { #根据实际IP ( 本地IP ) 和端口进行修改

        weight 200

        TCP\_CHECK {

            connect\_timeout 10       #(10秒无响应超时)

            connect\_port 3306

        }

    }

}

4.7.3 配置smtp邮箱，修改 /etc/mail.rc

# cat   << EOF >>   /etc/mail.rc

set   nss-config-dir=/etc/pki/nssdb

set   from="dba@yooli.com"

set   smtp=smtps://smtp.exmail.qq.com:465

set   smtp-auth=login

set   smtp-auth-user=dba@yooli.com

set   smtp-auth-password="urpass"

set   ssl-verify=ignore

EOF

4.7.4 配置keepalived日志，修改 /etc/sysconfig/keepalived

# sed   '/KEEPALIVED\_OPTIONS="-D"/s/"-D"/"-D -d -S 0"/'   /etc/sysconfig/keepalived

4.7.5 在 /etc/keepalived/ 目录下创建脚本 check\_mysql\_cluster.sh 、 keepalive\_notify.sh ，脚本内容见 http://wiki.yooli-in.com/pages/viewpage.action?pageId=1540575 ：

# cd /etc/keepalived/ && touch check\_mysql\_cluster.sh && chmod +x check\_mysql\_cluster.sh

# cd /etc/keepalived/ && touch keepalive\_notify.sh && chmox +x keepalive\_notify.sh

4.7.6 启动rsyslog与keepalived服务

# service rsyslog restart

# service keepalived start

### **4.8 安装** **zabbix-agent**

4.8.1 安装zabbix的yum源

# rpm –ivh http://repo.zabbix.com/zabbix/3.4/rhel/6/x86\_64/zabbix-release-3.4-1.el6.noarch.rpm

4.8.2 yum安装zabbix-agent、zabbix-sender

# yum install –y zabbix-agent zabbix-sender

4.8.3 配置zabbix-agent，文件位置为 /etc/zabbix/zabbix\_agentd.conf

PidFile=/var/run/zabbix/zabbix\_agentd.pid

LogFile=/var/log/zabbix/zabbix\_agentd.log

LogFileSize=128

Server =192.168.1.179               #zabbix server

StartAgents=5

ServerActive =192.168.1.179:10051

Hostname =db5-31.yooli-idc.net               #hostname

BufferSend=10

BufferSize=1024

MaxLinesPerSecond=256

Timeout=15

Include=/etc/zabbix/zabbix\_agentd.d/

UnsafeUserParameters=1

4.8.4 将一台服务器中/etc/zabbix/目录下的scripts、 zabbix\_agentd.d 文件夹拷贝到当前服务器的/etc/zabbix/目录下

4.8.5 启动zabbix-agent

# service zabbix-agent start

### **4.9 安装** **salt-minion**

4.9.1 安装salt-minion

# pip install – U salt==2016.11.6

4.9. 2 配置salt-minion，文件位置 /etc/salt/minion

default\_include: minion.d/\*.conf

master : 192.168.1.179               #salt-master

id : db5-30.yooli-idc.net               #hostname

log\_file: /var/log/salt/minion

key\_logfile: /var/log/salt/key

4.9.3 查看salt-minion依赖包版本，其中注意 Tornado版本，要为4.5.3

# salt-minion --versions-report

4.9.4 启动salt-minion

# service salt-minion start

### **4.10 创建系统用户**

4.10.1 创建sudoers文件（如果不存在），并修改权限

# touch /etc/sudoers

# chmod 640 /etc/sudoers

4.10.2 在 /etc/sudoers 文件末尾添加，并修改权限

# vim /etc/sudoers

Defaults:salt     !requiretty

%Sudoers ALL=(ALL)       ALL

%Sudoers ALL=(ALL)      NOPASSWD: ALL

################### sudo ######################

Defaults logfile=/var/log/sudo.log

Defaults loglinelen=0

Defaults !syslog

# chmod 440 /etc/sudoers

4.10.3 创建/var /log/sudo.log

# touch /var /log/sudo.log

4.10. 4 修改/etc/rsyslog.conf其权限及内容

# chmod 640 /etc/rsyslog.conf

# vim /etc/rsyslog.conf

local2.debug    /var/log/sudo.log

# chmod 440 /etc/rsyslog.conf

4.10.5 创建系统个人账户并加入sudo组，模版如下

# userdel -r testuser -r;useradd -N -p `echo " password "|openssl passwd -1 -salt $(< /dev/urandom tr -dc '[:alnum:]' | head -c 32) -stdin` testuser ; usermod -G Sudoers testuser

### **4.1** **1** **设置定时任务**

4.12.1 设置 Primary 节点定时任务

# crontab – e

\*/15 \* \* \* \* /usr/sbin/ntpdate -u pool.ntp.org >/dev/null 2>&1

10 1 \* \* 0 /usr/bin/python /data/scriptdir/mysql\_flush\_logs.py

4.12.2 设置 Secondary 节点定时任务

# crontab – e

\*/15 \* \* \* \* /usr/sbin/ntpdate -u pool.ntp.org >/dev/null 2>&1

10 1 \* \* 0 /usr/bin/python /data/scriptdir/mysql\_flush\_logs.py

0 3 \* \* \* /usr/bin/python /data/scriptdir/db\_backup.py FULL >/data/scriptdir/mysql\_full\_backup.log;

0 2 \* \* \* /usr/bin/python /data/scriptdir/backup\_binlog.py >/data/scriptdir/backup\_binlog\_log.log

4.12.3 配置 readonly 节点定时任务

# crontab – e

\*/15 \* \* \* \* /usr/sbin/ntpdate -u pool.ntp.org >/dev/null 2>&1

28 \*/2 \* \* \* /bin/bash /etc/titanagent/agent\_update.sh >> /var/log/titanagent/check.o.log 2>> /var/log/titanagent/check.e.log

\*/2 \* \* \* \* /bin/bash /etc/titanagent/agent\_update\_exception.sh >> /var/log/titanagent/check.o.log 2>> /var/log/titanagent/check.e.log

\*/2 \* \* \* \* /bin/bash /etc/titanagent/agent\_monitor.sh >> /var/log/titanagent/edog.o.log 2>> /var/log/titanagent/edog.e.log

备注：安全漏洞检测定时任务后期会在所有服务器上进行部署。

### **4.1** **2** **修改** **numa** **及** **elevvator** **并重启服务器**

在 /etc/grub.conf 文件中kernel所对应行的结尾添加：

# vim /etc/grub.conf

kernel /vmlinuz-2.6.32-696.13.2.el6.x86\_64 ro root=UUID=b96a5d9c-9038-4e1a-a4e9-f3427bdbb5f8 rd\_NO\_LUKS  KEYBOARDTYPE=pc KEYTABLE=us rd\_NO\_MD crashkernel=auto LANG=zh\_CN.UTF-8 rd\_NO\_LVM rd\_NO\_DM rhgb quiet numa=off elevator=deadline

重启服务器：

# reboot