数据库高可用-Percona-XtraDB-Cluster安装手册

|  |  |  |  |
| --- | --- | --- | --- |
| 文档版本号： | V1.0 | 文档编号： |  |
| 文档密级： |  | 归属部门/项目： |  |
| 产品名： | 贵州大医疗平台 | 子系统名： | 基础平台 |
| 编写人： | 雷少杰 | 编写日期： | 2016-11-09 |

目录

[MySQL数据库高可用-Percona Cluster安装手册 1](#_Toc466473532)

[1 Percona-XtraDB-Cluster简介 3](#_Toc466473533)

[1.1 介绍 3](#_Toc466473534)

[1.2 Percona XtraDB Cluster与MySQL Replication区别 3](#_Toc466473535)

[1.3 架构图 4](#_Toc466473536)

[1.4 数据复制图示 4](#_Toc466473537)

[2 部署安装 5](#_Toc466473538)

[2.1 安装环境 5](#_Toc466473539)

[2.2 安装配置 5](#_Toc466473540)

[2.2.1 安装前准备 5](#_Toc466473541)

[2.2.2 Yum安装[推荐] 5](#_Toc466473542)

[2.2.3 配置 6](#_Toc466473543)

[2.3 启动 7](#_Toc466473544)

[2.3.1 首次启动第一个节点 7](#_Toc466473545)

[2.3.2 启动其他节点 7](#_Toc466473546)

[2.3.3 重启第一个节点 8](#_Toc466473547)

[2.4 测试 8](#_Toc466473548)

# Percona-XtraDB-Cluster简介

## 介绍

Percona XtraDB Cluster是MySQL高可用性和可扩展性的解决方案.

集群是有节点组成的，推荐配置至少3个节点，但是也可以运行在2个节点上。

每个节点都是普通的mysql/percona服务器，可以将现有的数据库服务器组成集群，反之，也可以将集群拆分成单独的服务器。

每个节点都包含完整的数据副本。

* Percona XtraDB Cluster特性：

同步复制，事务要么在所有节点提交或不提交。

多主复制，可以在任意节点进行写操作。

在从服务器上并行应用事件，真正意义上的并行复制。

节点自动配置。

数据一致性，不再是异步复制。

* Percona XtraDB Cluster完全兼容MySQL和Percona Server，表现在：

数据的兼容性

应用程序的兼容性：无需更改应用程序

* 优点

当执行一个查询时，在本地节点上执行。因为所有数据都在本地，无需远程访问。

无需集中管理。可以在任何时间点失去任何节点，但是集群将照常工作。

良好的读负载扩展，任意节点都可以查询。

* 缺点如下

加入新节点，开销大。需要复制完整的数据。

不能有效的解决写缩放问题，所有的写操作都将发生在所有节点上。

有多少个节点就有多少重复的数据。

## Percona XtraDB Cluster与MySQL Replication区别

分布式系统的CAP理论。

C---一致性，所有节点的数据一致。

A---可用性，一个或多个节点失效，不影响服务请求。

P---分区容忍性，节点间的连接失效，仍然可以处理请求。

任何一个分布式系统，需要满足这三个中的两个。

MySQL Replication: 可用性和分区容忍性

Percona XtraDB Cluster: 一致性和可用性

因此MySQL Replication并不保证数据的一致性，而Percona XtraDB Cluster提供数据一致性。

Percona XtraDB Cluster组件：

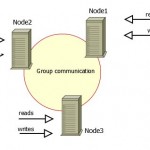
Percona XtraDB Cluster基于XtraDB的Percona Server以及包含写复制集补丁。使用Galera library，事务型应用下的通用的多主同步复制插件。

Galera 新特性有：

1.IST(Incremental State Transfer)增量状态传输。对于WAN特别有用。

2.RSU(Rolling Schema Update)旋转更新架构。不会阻止对表进行操作。

## 架构图

[](http://www.ttlsa.com/html/1360.html/xx)

## 数据复制图示



# 部署安装

## 安装环境

|  |  |  |
| --- | --- | --- |
| IP | 操作系统 | 机器名 |
| 10.12.8.111 | Centos 7.2 | pxc-node111 |
| 10.12.8.112 | Centos 7.2 | pxc-node112 |
| 10.12.8.113 | Centos 7.2 | pxc-node113 |

推荐三个几点，2个节点也可以。

## 安装配置

### 安装前准备

* 关闭SeLinux

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

setenforce 0

重启服务器

确认状态：sestatus

* 关闭放火墙

systemctl start firewalld.service#启动firewall

systemctl stop firewalld.service#停止firewall

systemctl disable firewalld.service#禁止firewall开机启动

* 修改主机名

分别在三台机器上运行

hostnamectl set-hostname [pxc-node111| pxc-node112| pxc-node113] -static

* 删除mysql的安装

yum remove mysql mysql-server mysql-libs mysql-server

### Yum安装[推荐]

#### 配置yum源

* 配置EPEL源[用于安装socat]

将文件放入/etc/yum.repos.d目录中

* 配置Percona-XtraDB-Cluster源

wget https://www.percona.com/redir/downloads/percona-release/redhat/0.1-4/percona-release-0.1-4.noarch.rpm[新版本可能会变化]

rpm -ivh percona-release-0.1-4.noarch.rpm

* yum更新

yum clean all

yum makecache

* 安装

yum install socat

yum install Percona-XtraDB-Cluster-57

### RPM安装

文件下载：<http://10.1.3.180:8803/common/tools/percona/>

<https://www.percona.com/downloads/>

yum install perl-DBD-MySQL

rpm -ivh socat-1.7.2.2-5.el7.x86\_64.rpm

rpm -ivh Percona-XtraDB-Cluster-client-57-5.7.14-26.17.1.el7.x86\_64.rpm

rpm -ivh Percona-XtraDB-Cluster-shared-57-5.7.14-26.17.1.el7.x86\_64.rpm

rpm -ivh libev-4.15-3.el7.x86\_64.rpm

rpm -ivh Percona-XtraDB-Cluster-shared-compat-57-5.7.14-26.17.1.el7.x86\_64.rpm

rpm -ivh perl-DBD-MySQL-4.023-5.el7.x86\_64.rpm

rpm -ivh percona-xtrabackup-24-2.4.4-1.el7.x86\_64.rpm

rpm -ivh Percona-XtraDB-Cluster-server-57-5.7.14-26.17.1.el7.x86\_64.rpm

rpm -ivh Percona-XtraDB-Cluster-57-5.7.14-26.17.1.el7.x86\_64.rpm

### 配置

#### 配置文件[/etc/my.cnf]

每台服务器上都要配置

|  |
| --- |
| [mysqld]  #111机器为pxc-node111.pid，112机器为：pxc-node112.pid，113机器为：pxc-node113.pid  pid-file="pxc-node111.pid"  datadir=/opt/mysql/data  binlog\_format=ROW  #存储引擎设置为innodb，myisam仅仅测试支持  default-storage-engine=innodb  #指定innodb 自增长列锁模式，2为交叉锁模式， 多个语句能同时执行  innodb\_autoinc\_lock\_mode=2  innodb\_locks\_unsafe\_for\_binlog=1  query\_cache\_size=0  query\_cache\_type=0  bind-address=0.0.0.0  log\_error = error.log  wsrep\_cluster\_name="my\_wsrep\_cluster"  #指定集群每个节点的连接地址  wsrep\_cluster\_address = gcomm://10.12.8.111,10.12.8.112,10.12.8.113  wsrep\_node\_address = 10.12.8.111  wsrep\_provider = /usr/lib64/libgalera\_smm.so  wsrep\_slave\_threads=1  wsrep\_certify\_nonPK=1  wsrep\_max\_ws\_rows=131072  wsrep\_max\_ws\_size=1073741824  wsrep\_debug=1  wsrep\_convert\_LOCK\_to\_trx=0  wsrep\_retry\_autocommit=1  wsrep\_auto\_increment\_control=1  wsrep\_drupal\_282555\_workaround=0  wsrep\_causal\_reads=0  wsrep\_notify\_cmd=  #指定SST方式，支持rsync(最快，需要锁表)，mysqldump和xtrabackup，从5.5.33-23.7.6起支持xtrabackup-v2  wsrep\_sst\_method=xtrabackup-v2  #SST的认证方式  wsrep\_sst\_auth=sstuser:sstuser  lower\_case\_table\_names=1  pxc\_strict\_mode =PERMISSIVE |

## 启动

### 首次启动第一个节点

* 启动

systemctl start [mysql@bootstrap.service](mailto:mysql@bootstrap.service)

* 查找密码

more /opt/mysql/data/error.log|grep password

出现内容：

[Note] A temporary password is generated for root@localhost: sp(viVQ5Q0Jg

红色部分为密码

* 登陆MySQL

mysql -u root –p’ sp(viVQ5Q0Jg’

修改ROOT密码：set password for 'root'@'localhost'=password('root');

创建SST用户：

grant all privileges on \*.\* to 'sstuser'@'%' identified by 'sstuser' with grant option;

flush privileges;

### 启动其他节点

运行命令：systemctl start [mysql.service](mailto:mysql@bootstrap.service)

数据库会自动从第一个节点上同步。

### 重启第一个节点[可选]

systemctl stop [mysql@bootstrap.service](mailto:mysql@bootstrap.service)

systemctl start [mysql.service](mailto:mysql@bootstrap.service)

## 测试

在一个数据库上创建数据、表，增删改查记录

登陆其他节点查询数据，看是否正常自动同步。