MongoDB集群部署

1. 部署规划

服务器基础操作环境确认为RedHat7.6操作系统，主机名称统一以mongo前缀加编号的形式命名，编号连续并由01开始递增。服务器操作系统字符集为UTF8编码格式，时区为GMT标准时区。

1.1 节点规划

三台服务器，每台服务器上2个磁盘做RAID1，用作系统盘。4个磁盘做单盘RAID0，用作数据盘 ，数据盘采用xfs文件系统。集群采用副本集+分片部署方式。

节点部署结构规划如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **主机名** | **IP地址** | **数据磁盘挂载路径** | **用途** |
| mongo01 | 192.168.51.5 | /mnt/disk2  /mnt/disk3  /mnt/disk4  /mnt/disk5 | 配置副本集节点  数据副本集节点  路由副本集节点 |
| mongo02 | 192.168.51.6 | /mnt/disk2  /mnt/disk3  /mnt/disk4  /mnt/disk5 | 配置副本集节点  数据副本集节点  路由副本集节点 |
| mongo03 | 192.168.51.7 | /mnt/disk2  /mnt/disk3  /mnt/disk4  /mnt/disk5 | 配置副本集节点  数据副本集节点  路由副本集节点 |

1.2 组件规划

1.2.1 路由副本集节点

|  |  |  |
| --- | --- | --- |
| **主机名** | **IP:Port** | **目录** |
| mongo01 | 192.168.51.5:27017 | /mnt/disk2/mongodb/mongos |
| mongo02 | 192.168.51.6:27017 | /mnt/disk2/mongodb/mongos |
| mongo03 | 192.168.51.7:27017 | /mnt/disk2/mongodb/mongos |

1.2.2 配置副本集节点

|  |  |  |
| --- | --- | --- |
| **主机名** | **IP:Port** | **目录** |
| mongo01 | 192.168.51.5:27019 | /mnt/disk2/mongodb/config |
| mongo02 | 192.168.51.6:27019 | /mnt/disk2/mongodb/config |
| mongo03 | 192.168.51.7:27019 | /mnt/disk2/mongodb/config |

1.2.3 数据副本集节点

|  |  |  |
| --- | --- | --- |
| **Shard名** | **IP:Port** | **目录** |
| shard0 | 192.168.51.5:27020 | /mnt/disk3/mongodb/data |
| 192.168.51.6:27020 | /mnt/disk3/mongodb/data |
| 192.168.51.7:27020 | /mnt/disk3/mongodb/data |
| shard1 | 192.168.51.5:27021 | /mnt/disk4/mongodb/data |
| 192.168.51.6:27021 | /mnt/disk4/mongodb/data |
| 192.168.51.7:27021 | /mnt/disk4/mongodb/data |
| shard2 | 192.168.51.5:27022 | /mnt/disk5/mongodb/data |
| 192.168.51.6:27022 | /mnt/disk5/mongodb/data |
| 192.168.51.7:27022 | /mnt/disk5/mongodb/data |

1.2.4 用户与用户组

root用户用于服务器管理，以及集群部署。mongod用户用于MongoDB集群。

使用以下命令创建mongo用户/用户组：

# /usr/sbin/groupadd -r mongod

# /usr/sbin/useradd -M -r -g mongod -s /bin/false mongod

1.2.5 安全设置

需要关闭SELinux。防火墙也建议关闭，如果开启，需要为上述端口放行。

1. 系统环境检查
   1. CPU检查
      1. 是否开启超线程

查看物理cpu：

# cat /proc/cpuinfo | grep "physical id" | sort | uniq

physical id : 0

physical id : 1

单个CPU的逻辑核心数量：

# cat /proc/cpuinfo | grep "cores" | uniq

cpu cores : 12

系统CPU线程数：

# cat /proc/cpuinfo | grep "processor" | wc -l

24

# 这台机器拥有2颗物理CPU，每个CPU有12个逻辑核心，系统一共拥有24个CPU线程，所以没有开启超线程

* + 1. 检查节能模式是否关闭

查看CPU当前模式：

# cat /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor

设置CPU的模式：

# cpupower frequency-set -g performance

* 1. 网络检查

需要最低的网络为千兆以太网，并使用ping命令检查数据包接收延迟。

检查/etc/hosts文件，确保集群内部所有节点都已经配置。

* 1. 检测操作系统和内核版本

查看操作系统

# cat /etc/redhat-release

查看内核

# uname -a

* 1. 检查磁盘

使用lsblk命令查看所有磁盘和挂载情况。

检查/etc/fstab，确保所有使用的磁盘已经添加。

如果磁盘没有正确格式化并挂载，使用parted命令格式化磁盘，并使用mkfs.xfs初始化文件系统。然后使用blkid获取磁盘的UUID，并添加到/etc/fstab文件中。

使用badblocks 命令检查所有磁盘，确保没有坏道。

# nohup badblocks -v -b 8092 -c 1 /dev/sdb > /tmp/sdb.txt &

* 1. 检查系统时区及字符集

使用date命令，确保各节点的时区和时间一致。

执行“echo $LANG”命令，确保字符集为UTF8。

1. 安装准备
   1. 检查安装包

安装包transwarp-mongodb-4.2.14.tgz上传至各个节点的/opt目录下，并解压缩。

# tar zxf transwarp-mongodb-4.2.14.tgz

# ls /opt/transwarp-mongodb

bin conf mongo\_config.sh mongo\_mongos.sh mongo\_shard.sh

* 1. 修改相关配置

根据组件规划，修改mongo\_config.sh、mongo\_mongos.sh、mongo\_shard.sh脚本中的IP地址、端口和目录路径。根据shard的个数，需要为每个shard创建一个mongo\_shard.sh脚本。

3.2.1 mongo\_config.sh

如果使用默认端口27019，则不需要修改。

如果使用默认目录/mnt/disk2/mongodb/config，则不需要修改。

否则，需要修改如下配置：

PORT=27019

DB\_PATH=/mnt/disk2/mongodb/config

3.2.2 mongo\_mongos.sh

需要修改如下配置中的配置副本集节点的IP地址和端口：

$EXEC\_PATH --port 27017 --configdb TranswarpMongoConfigServer/172.22.51.5:27019,172.22.51.6:27019,172.22.51.8:27019

如果使用默认端口27017，则不需要修改端口。

3.2.3 mongo\_shard.sh

本示例中有3个shard，需要拷贝mongo\_shard.sh为mongo\_shard.0.sh、mongo\_shard.1.sh、mongo\_shard.2.sh。

修改各个脚本中的如下配置，：

PORT=27020

SHARDNAME="TranswarpMongoShard0"

SHARD\_DB\_PATH=/mnt/disk3/mongodb/data

PIDFILE=$PID\_PATH/mongodb-shard0.pid

LOG\_PATH=/var/log/mongodb/mongodb-shard0-$(hostname).log

LOCKFILE="$LOCKDIR/mongodb-shard0"

比如对于shard1，应该修改为：

PORT=27021

SHARDNAME="TranswarpMongoShard1"

SHARD\_DB\_PATH=/mnt/disk4/mongodb/data

PIDFILE=$PID\_PATH/mongodb-shard1.pid

LOG\_PATH=/var/log/mongodb/mongodb-shard1-$(hostname).log

LOCKFILE="$LOCKDIR/mongodb-shard1"

* 1. 创建相关目录

在各个节点上创建如下目录，并修改属主权限。

# mkdir -p /mnt/disk2/mongodb/{mongos,config}

# chown -R mongod:mongod mnt/disk2/mongodb/{mongos,config}

# mkdir -p /mnt/disk{3,4,5}/mongodb/data

# chown -R /mnt/disk{3,4,5}/mongodb/data

# mkdir -p /var/{run,log}/mongodb

# chown -R mongod:mongod /var/{run,log}/mongodb

# chown -R mongod:mongod /opt/transwarp-mongodb

1. 安装步骤
   1. 启动配置服务器

三个节点都需要执行该操作。

# /opt/transwarp-mongodb/mongo\_config.sh start

启动成功后，在任一节点上执行副本集初始化：

# /opt/transwarp-mongodb/bin/mongo --port 27019 --host 192.168.51.5

> MongoDB server version: 4.2.14

> cfg={

\_id:"TranswarpMongoConfigServer",

configsvr: true,

members:[

{\_id:0, host:'192.168.51.5:27019'},

{\_id:1, host:'192.168.51.6:27019'},

{\_id:2, host:'192.168.51.8:27019'}

]};

rs.initiate(cfg);

验证： rs.status();

* 1. 启动数据服务器

三个节点上都需要执行此操作。

以下操作中，以shard0为例：

# /opt/transwarp-mongodb/mongo\_shard.0.sh start

启动成功后，在任一节点上执行副本集初始化：

# /opt/transwarp-mongodb/bin/mongo --port 27020 --host 192.168.51.5

> MongoDB server version: 4.2.14

> cfg={

\_id:"TranswarpMongoShard0",

members:[

{\_id:0, host:'192.168.51.5:27020'},

{\_id:1, host:'192.168.51.6:27020'},

{\_id:2, host:'192.168.51.7:27020'}

]};

rs.initiate(cfg);

* 1. 启动路由服务器

三个节点上都需要执行此操作。

# /opt/transwarp-mongodb/mongo\_mongos.sh start

启动成功后，接入其中一个，添加分片：

# /opt/transwarp-mongodb/bin/mongo --port 27017 --host 192.168.51.5

mongos> MongoDB server version: 4.2.14

mongos> sh.addShard("TranswarpMongoShard0/192.168.51.5:27020")

mongos> sh.addShard("TranswarpMongoShard1/192.168.51.5:27021")

mongos> sh.addShard("TranswarpMongoShard2/192.168.51.5:27022")

* 1. 创建管理员账户

需要在admin库中创建一个名为admin的管理员用户，密码为admin。具有集群管理权限、所有数据库的操作权限。

use admin

db.createUser({

user:'admin',pwd:'admin',

roles:[

{role:'clusterAdmin',db:'admin'},

{role:'userAdminAnyDatabase',db:'admin'},

{role:'dbAdminAnyDatabase',db:'admin'},

{role:'readWriteAnyDatabase',db:'admin'}

]})

1. 安全认证
   1. 生成keyfile

# openssl rand -base64 90 > /opt/transwarp-mongodb/conf/mongodb-keyfile

# chown mongod.mongod /opt/transwarp-mongodb/conf/mongodb-keyfile

# chmod 700 /opt/transwarp-mongodb/conf/mongodb-keyfile

需要把文件mongodb-keyfile拷贝到各个节点的/opt/transwarp-mongodb/conf目录下，并修改属主权限。

* 1. 修改配置

修改/opt/transwarp-mongodb/conf/mongod.conf，添加如下配置：

security:

keyFile: /opt/transwarp-mongodb/conf/mongodb-keyfile

authorization: enabled

修改/opt/transwarp-mongodb/conf/mongos.conf，添加如下配置：

security:

keyFile: /opt/transwarp-mongodb/conf/mongodb-keyfile

* 1. 重启集群，使配置生效
  2. 创建一个普通用户

普通用户由管理员创建,通常需要指定某个数据库来操作。我们创建一个testdb数据库，给这个数据库添加一个用户，用户名为test01,密码为123456。并授予该用户对testdb数据库进行读写操作的权限。

普通用户需要由管理员创建并授权。所以，我们首先做的就是用管理员账户登录数据库。

use admin

db.auth('admin','admin')

创建testdb数据库

use命令切换数据库时如果该库不存在，那么则会创建该数据库

use testdb

创建用户

db.createUser({

user:"test01",

pwd:"123456",

roles:[

{role:"readWrite",db:"testdb"}

]

})