# mongodb副本集安装部署手册

# ****一、环境准备****

### 1、安装环境

2台linux服务器  
操作系统：CentOS release 6.5 (Final)  
IP地址：10.20.1.106（节点1）、10.20.1.107（节点2）

### 2、关闭iptables和selinux

# chkconfig iptables off   
# service iptables stop   
# vi /etc/selinux/config   
SELINUX=disabled

# ****二、安装mongodb（yum安装）****

注：在每一个节点上都安装

### 1.安装percona软件库

yum install <https://www.percona.com/redir/downloads/percona-release/redhat/latest/percona-release-0.1-4.noarch.rpm>

### 2.安装percona-server-mongodb

yum install Percona-Server-MongoDB-32  
yum install -y numactl

### 3.创建对应目录，修改权限

mkdir /data/mongodb/log -p  
mkdir /data/mongodb/data -p  
cd /data/  
chown [mongod:mongod](http://mongodmongod) -R mongodb/

### 4.修改系统配置

echo "never" > /sys/kernel/mm/transparent\_hugepage/enabled  
echo "never" > /sys/kernel/mm/transparent\_hugepage/defrag  
echo 0 > /proc/sys/vm/zone\_reclaim\_mode  
vim /etc/security/limits.conf  
mongod soft nofile 64000  
mongod hard nofile 64000  
mongod soft nproc 32000  
mongod hard nproc 32000

### 5.mongodb配置文件（两个节点分别配置）

节点1（PRIMARY）（10.20.1.106）

vim /etc/mongod.conf

# mongod.conf, Percona Server for MongoDB

# for documentation of all options, see:

# http://docs.mongo.org/manual/reference/configuration-options/

# Where and how to store data.

storage:

dbPath: /data/mongodb/data

journal:

enabled: true

engine: wiredTiger

storage.wiredTiger.engineConfig.cacheSizeGB: 8

systemLog:

destination: file

logAppend: true

path: /data/mongodb/log/mongod.log

timeStampFormat: iso8601-utc

quiet: true

processManagement:

fork: true

pidFilePath: /data/mongodb/data/mongod.pid

# network interfaces

net:

port: 27017

bindIp: 0.0.0.0

#security:

# keyFile: /data/mongodb/data/mongod.keyfile

# clusterAuthMode: keyFile

# authorization: enabled

#operationProfiling:

#replication:

# oplogSizeMB: 2048

# replSetName: "cashset"

# secondaryIndexPrefetch: "all"

#sharding:

## Enterprise-Only Options:

#auditLog:

#snmp:

节点2（SECONDARY）（10.20.1.107）  
vim /etc/mongod.conf

# mongod.conf, Percona Server for MongoDB

# for documentation of all options, see:

# http://docs.mongo.org/manual/reference/configuration-options/

# Where and how to store data.

storage:

dbPath: /data/mongodb/data

journal:

enabled: true

engine: wiredTiger

storage.wiredTiger.engineConfig.cacheSizeGB: 8

systemLog:

destination: file

logAppend: true

path: /data/mongodb/log/mongod.log

timeStampFormat: iso8601-utc

quiet: true

processManagement:

fork: true

pidFilePath: /data/mongodb/data/mongod.pid

# network interfaces

net:

port: 27017

bindIp: 0.0.0.0

#security:

# keyFile: /data/mongodb/data/mongod.keyfile

# clusterAuthMode: keyFile

# authorization: enabled

#operationProfiling:

#replication:

# oplogSizeMB: 2048

# replSetName: "cashset"

# secondaryIndexPrefetch: "all"

#sharding:

## Enterprise-Only Options:

#auditLog:

#snmp:

### 6.修改mongodb启动文件

vim /etc/init.d/mongod  
PIDFILEPATH=/data/mongodb/data/mongod.pid

### 7.启动mongodb服务（只启动节点1，节点2暂时不启动）

service mongod start

# ****三、开启登录验证模式****

注：只需要在节点1上操作

### 1、创建用户

#mongo  
MongoDB shell version: 3.2.10-3.0  
connecting to: test  
> use admin;  
switched to db admin  
> db.createUser({ "user" : "admin", "pwd": "admin",roles:[{role: "root",db:"admin"}]})

### 2、修改配置文件

vim /etc/mongod.conf

#security:

security:

keyFile: /data/mongodb/data/mongod.keyfile

clusterAuthMode: keyFile

authorization: enabled

replication:

oplogSizeMB: 51200 #集群oplog大小（50G）

replSetName: "repset" #集群名称，这个记住，下边集群配置会用到

secondaryIndexPrefetch: "all"

### 3、创建keyfile

openssl rand -base64 756 > /data/mongodb/data/mongod.keyfile

chmod 400 /data/mongodb/data/mongod.keyfile

chown /data/mongodb/data/mongod.keyfile

**4、拷贝keyfile到每一个**节点

 scp /data/mongodb/data/mongod.keyfile root@10.20.1.107:/data/mongodb/data/mongod.keyfile

chown mongod:mongod /data/mongodb/data/mongod.keyfile

### 5、重启mongodb（节点2第一次启动）

service mongod restart                --节点1重启

service mongod start                   --节点2启动

# ****四、配置见证服务****

注：见证服务配置在节点2（SECONDARY）（10.20.1.107）上

### 1、创建对应目录，修改权限

mkdir /data/mongodb/arbiter -p  
cd /data/mongodb  
chown [mongod:mongod](http://mongodmongod) -R arbiter/

cp /data/mongodb/data/mongod.keyfile /data/mongodb/arbiter/

chown mongod:mongod /data/mongodb/arbiter/mongod.keyfile

### 2、创建配置文件

cp /etc/mongod.conf /etc/mongod\_arbiter.conf  
vim /etc/mongod\_arbiter.conf

# mongod.conf, Percona Server for MongoDB

# for documentation of all options, see:

# http://docs.mongo.org/manual/reference/configuration-options/

# Where and how to store data.

storage:

dbPath: /data/mongodb/arbiter

journal:

enabled: true

# engine: mmapv1

# engine: rocksdb

# engine: wiredTiger

# engine: inMemory

# engine: PerconaFT

PerconaFT:

engineOptions:

cacheSize: 253687090 #注意内存分别，因为跟SECONDARY节点在同一机器，所以尽量少分配，此处分配了250M

journalCommitInterval: 100

collectionOptions:

compression: zlib

fanout: 128

readPageSize: 16384

indexOptions:

compression: zlib

fanout: 128

readPageSize: 16384

# Storage engine various options

# mmapv1:

# wiredTiger:

# where to write logging data.

systemLog:

destination: file

logAppend: true

path: /data/mongodb/arbiter/mongod\_arbiter.log #日志文件路径

timeStampFormat: iso8601-utc

processManagement:

fork: true

pidFilePath: /data/mongodb/arbiter/mongod.pid

# network interfaces

net:

port: 27018 #端口号，为与mongod服务区分，需要修改端口号

bindIp: 0.0.0.0

security:

keyFile: /data/mongodb/arbiter/mongod.keyfile

clusterAuthMode: keyFile

authorization: enabled

#operationProfiling:

replication:

oplogSizeMB: 512

replSetName: "repset"

secondaryIndexPrefetch: "all"

#sharding:

## Enterprise-Only Options:

#auditLog:

#snmp:

### 3、创建启动文件

vim /etc/init.d/mongod /etc/init.d/mongo-arbiter

#!/bin/bash

# mongod - Startup script for mongod

# chkconfig: 35 85 15

# description: Percona Server for MongoDB is a scalable, document-oriented database.

# processname: mongod

# config: /etc/mongod.conf

# pidfile: /var/run/mongod.pid

. /etc/rc.d/init.d/functions

# things from mongod.conf get there by mongod reading it

# NOTE: if you change any OPTIONS here, you get what you pay for:

# this script assumes all options are in the config file.

CONFIGFILE="/etc/mongod\_arbiter.conf" #注意配置文件路径

OPTIONS="-f $CONFIGFILE"

SYSCONFIG=""

STDOUT="/var/log/mongo/mongod\_arbiter.stdout"

STDERR="/var/log/mongo/mongod\_arbiter.stderr"

PIDFILEPATH=/data/mongodb/arbiter/mongod.pid #pid文件路径

PIDDIR=$(dirname $PIDFILEPATH)

MONGO\_USER=mongod

MONGO\_GROUP=mongod

mongod=${MONGOD:-/usr/bin/mongod}

if [ -f "$SYSCONFIG" ]; then

. "$SYSCONFIG"

fi

# Handle NUMA access to CPUs (SERVER-3574)

# This verifies the existence of numactl as well as testing that the command works

NUMACTL\_ARGS="--interleave=all"

if which numactl >/dev/null 2>/dev/null && numactl $NUMACTL\_ARGS ls / >/dev/null 2>/dev/null

then

NUMACTL="numactl $NUMACTL\_ARGS"

else

NUMACTL=""

fi

start()

{

# Make sure the default pidfile directory exists

if [ ! -d ${PIDDIR} ]; then

install -d -m 0755 -o ${MONGO\_USER} -g ${MONGO\_GROUP} ${PIDDIR}

fi

touch ${PIDFILEPATH}

chown ${MONGO\_USER} ${PIDFILEPATH}

/usr/bin/percona-server-mongodb-helper.sh

# Recommended ulimit values for mongod or mongos

# See http://docs.mongodb.org/manual/reference/ulimit/#recommended-settings

#

ulimit -f unlimited

ulimit -t unlimited

ulimit -v unlimited

ulimit -n 64000

ulimit -m unlimited

ulimit -u 64000

echo -n $"Starting mongod: "

daemon --user "$MONGO\_USER" --check ${mongod} "$NUMACTL $mongod $OPTIONS > ${STDOUT} 2> ${STDERR}"

#daemon --user "$MONGO\_USER" --check ${mongod} "$mongod $OPTIONS > ${STDOUT} 2> ${STDERR}"

RETVAL=$?

echo

[ $RETVAL -eq 0 ] && touch /var/lock/subsys/mongod

}

stop()

{

echo -n $"Stopping mongod: "

mongo\_killproc "$PIDFILEPATH" $mongod

RETVAL=$?

echo

[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/mongod

}

restart () {

stop

start

}

# Send TERM signal to process and wait up to 300 seconds for process to go away.

# If process is still alive after 300 seconds, send KILL signal.

# Built-in killproc() (found in /etc/init.d/functions) is on certain versions of Linux

# where it sleeps for the full $delay seconds if process does not respond fast enough to

# the initial TERM signal.

mongo\_killproc()

{

local pid\_file=$1

local procname=$2

local -i delay=300

local -i duration=10

local pid=`pidofproc -p "${pid\_file}" ${procname}`

kill -TERM $pid >/dev/null 2>&1

usleep 100000

local -i x=0

while [ $x -le $delay ] && checkpid $pid; do

sleep $duration

x=$(( $x + $duration))

done

kill -KILL $pid >/dev/null 2>&1

usleep 100000

rm -f "${pid\_file}"

checkpid $pid

local RC=$?

[ "$RC" -eq 0 ] && failure "${procname} shutdown" || success "${procname} shutdown"

RC=$((! $RC))

return $RC

}

RETVAL=0

case "$1" in

start)

start

;;

stop)

stop

;;

restart|reload|force-reload)

restart

;;

condrestart)

[ -f /var/lock/subsys/mongod ] && restart || :

;;

status)

#status $mongod

status -p $PIDFILEPATH

RETVAL=$?

;;

\*)

echo "Usage: $0 {start|stop|status|restart|reload|force-reload|condrestart}"

RETVAL=1

esac

exit $RETVAL

### 4、启动见证服务

service mongo-arbiter start

# ****五、集群配置****

注：此步骤只需再节点1（PRIMARY）（10.20.1.106）上操作  
#mongo  
>use admin  
>db.auth('admin','admin')  
>cfg={ \_id:"repset", members:[ {\_id:0,host:'10.20.1.106:27017'}, {\_id:1,host:'10.20.1.107:27017'},{\_id:2,host:'10.20.1.107:27018',[arbiterOnly:true](http://arbiterOnlytrue)}] };  
>rs.initiate(cfg);  
>rs.status();