第一步、

选择一台linux服务器作为twemproxy代理中间件

下载安装twemproxy中间件

autoconf下载地址：<http://ftp.gnu.org/gnu/autoconf/autoconf-2.69.tar.gz>

twemproxy下载地址：<https://codeload.github.com/twitter/twemproxy/zip/master>

twemproxy的安装要求autoconf的版本在2.64以上，否则提示"error: Autoconf version 2.64 or higher is required"。

**安装autoconf**

查找旧版本 autoconf 并且卸载

rpm –qf autoconf

rpm -e --nodeps autoconf-2.63

下载autoconf进入目录

cd autoconf-2.69

指定安装目录

./configure --prefix=/usr

make && make install

安装twemproxy

unzip twemproxy-master.zip

unzip twemproxy-master.zip

生成configure

autoreconf –fvi

指定安装目录

./configure --prefix=/usr/local/twemproxy

make -j 8

make install

设置环境变量

echo "PATH=$PATH:/usr/local/twemproxy/sbin/" >> /etc/profile

source /etc/profile

创建相关目录（存放配置文件和pid文件）

cd /usr/local/twemproxy

mkdir run conf

添加proxy配置文件

vim /usr/local/twemproxy/conf/nutcracker.yml

内容如下（配置文件参考官方）

alpha:

listen: 127.0.0.1:22121

hash: fnv1a\_64

distribution: ketama

auto\_eject\_hosts: true

redis: true

server\_retry\_timeout: 2000

server\_failure\_limit: 1

servers:

- 127.0.0.1:7000:1

- 127.0.0.1:7001:1

- 127.0.0.1:7002:1

启动Twemproxy服务

nutcracker -t 测试配置文件

把conf目录复制到/usr/local/twemproxy/sbin/目录下，进行检测



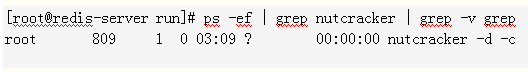
启动命令：

nutcracker -d -c /usr/local/twemproxy/conf/nutcracker.yml

-p /usr/local/twemproxy/run/redisproxy.pid

-o /usr/local/twemproxy/run/redisproxy.log

ps -ef | grep nutcracker | grep -v grep



接下来安装redis

安装redis选择一台服务器作为主节点

Linux下redis安装与使用

redis官网地址：<http://www.redis.io/>

最新版本：2.8.3

在Linux下安装Redis非常简单，具体步骤如下（官网有说明）：

cd /opt/

wget <http://download.redis.io/releases/redis-2.8.3.tar.gz>

tar xzf redis-2.8.3.tar.gz

cd redis-2.8.3

make 编译

mkdir /usr/local/redis

复制src目录下的redis-server、redis-benchmark、redis-cli和redis目录下的redis.conf/usr/local/redis

cp redis/src/redis-server redis-benchmark redis-cli /usr/local/redis

cp redis/src/redis.conf /usr/local/redis

cd /usr/local/redis

配置redis.conf

vim redis.conf

修改配置项

将daemonize no改为 改成yes 守护进程模式

如果需要远程连接修改bind 0.0.0.0

如果需要指定连接ip修改bind ip地址

修改dir存放数据目录

如 dir /usr/local/redis/data

将redis添加到自启动中

echo "/usr/local/bin/redis-server /etc/redis.conf" >> /etc/rc.d/rc.local

复制redis.conf和redis-server到/usr/local/bin

启动redis-server redis.conf

【安装redis从节点】

选择一台服务器作为redis从节点服务备份

如果主节点停止运行会自动选择其中有一台从节点作为主节点

登录从节点服务器

mkdir /usr/local/redis

cd /usr/local/redis

mkdir 7001（端口或者从节点名称）

cd 7001

远程复制redis.conf文件和redis/src目录到当前目录

远程复制redis目录中的sentinel.conf和src目录中的redis-sentinel

scp app@ipd地址:/usr/local/redis/redis.conf redis/src ./

scp app@ipd地址:/usr/local/redis/sentinel.conf redis/src/redis-sentinel ./

vim redis.conf

编辑从节点redis.conf文件

修改slaveof 127.0.0.1 6379（主节点ip地址） （主节点端口）

requirepass 设置密码

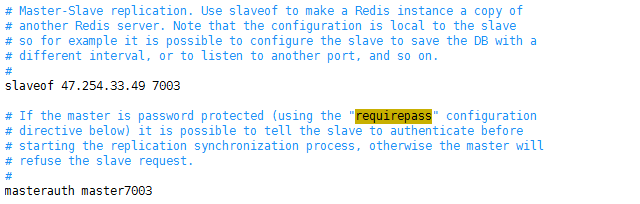
pidfile /var/run/redis\_7004.pid

pid主进程运行目录

dir 数据存放目录



设置从节点ip地址和主节点验证密码



将修改配置好的redis.conf复制到 /usr/local/bin目录下

cp redis.conf /usr/local/bin/

cp src/redis-server /usr/local/bin/

redis-server redis.conf

启动从节点redis服务

安装多台从节点以此类推

【安装redis哨兵】

一个主节点可以有多个从节点，每个从节点只能有一个主节点

哨兵是用来监视主节点是否在正常运行

如果主节点停止运行会按照权重提升从节点为主节点

cd /usr/local/redis/7001

使用三个哨兵监视主节点

mv sentinel.conf sentinel-1.conf

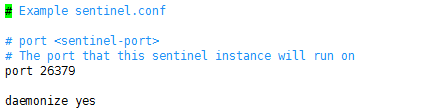
cp sentinel-1.conf sentinel-2.conf

cp sentinel-1.conf sentinel-3.conf

vim sentinel-1.conf 编辑哨兵配置文件

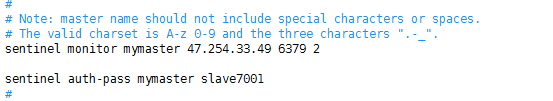
prot 26379(主节点端口)

daemonize yes 以守护进程方式运行

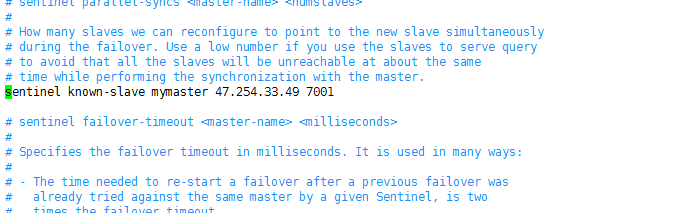


sentinel monitor mymaster ip 主节点端口 权重分配

sentinel auth-pass mymaster 从节点验证密码



sentinel known-slave mymaster ip 从节点端口



如果有多个从节点会在下面有多行这样的配置项

dir sentinel 工作目录

编辑2、3哨兵配置文件都是这样的

cp sentinel-1.conf sentinel-2.conf sentinel-2.conf /usr/local/bin

cp redis-sentinel /usr/local/bin

cd /usr/local/bin

redis-sentinel sentinel-1.conf

redis-sentinel sentinel-2.conf

redis-sentinel sentinel-2.conf

启动所有哨兵

ps aux | grep sentinel 查看哨兵进程

redis-cli –p 26379 测试查看哨兵监听的主从节点

sentinel masters查看当前的master主节点

sentinel slaves mymaster

查看所有从节点

sentinel get-master-addr-by-name mymaster

mymaster为节点起的名称

参考文档

<http://www.cnblogs.com/linuxbug/p/5131504.html>