## Redis文档

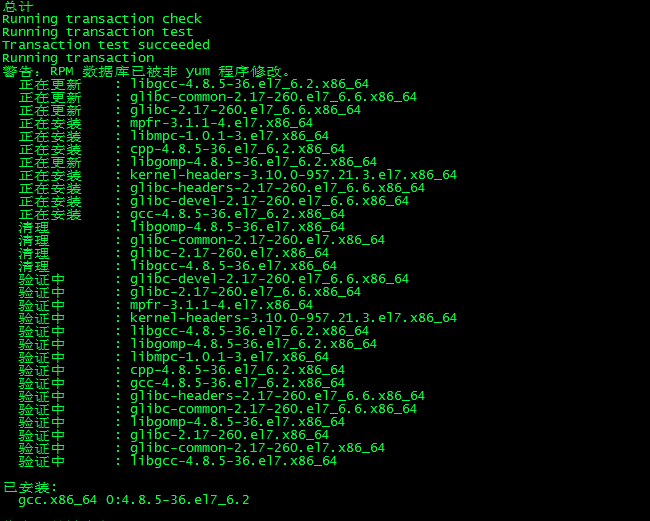
### Redis安装文档

#### 1.1启动虚拟机

#### 1.2.用第三方软件连接操作

#### 1.3要是用C语言的gcc，

所以要先安装gcc语句：yum -y install gcc 回车



#### 1.4.然后将下载的redis压缩包（后缀是gz的）通过软件放入/usr/local路径下，然后运行

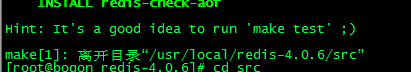




#### 1.5.运行 图片中语句来解压到当前文件夹



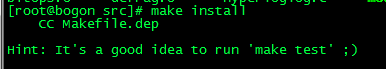
#### 1.6.然后cd /redis-4.0.6 进入到文件夹下然后直接make编译



出现如上hint语句表示成功

#### 1.7然后进入到src里面继续在src中执行make install





#### 1.8。替换文件配置

将cd /redis-4.0.6文件夹下的redis.conf和sentinel.conf替换成超哥提供的sentinel.conf和redis.conf。

#### 1.9.然后在cd /usr/local/bin中执行两句代码

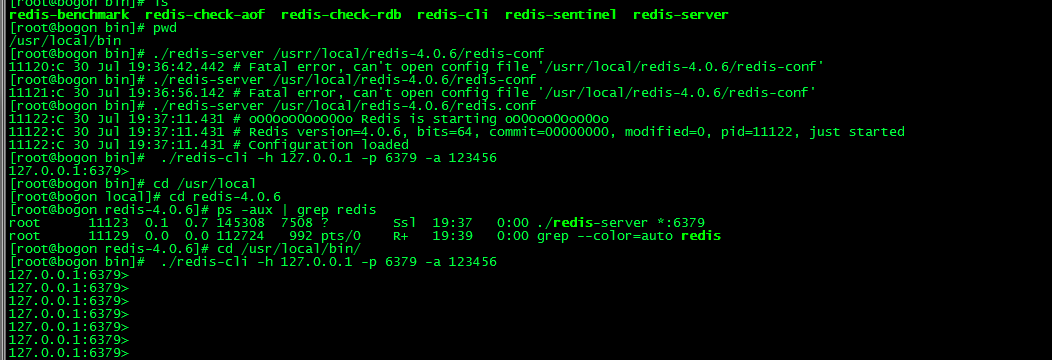
1. ./redis-server /usr/local/redis-4.0.6/redis.conf

2. 用ps -aux | grep redis查看进程是否启动



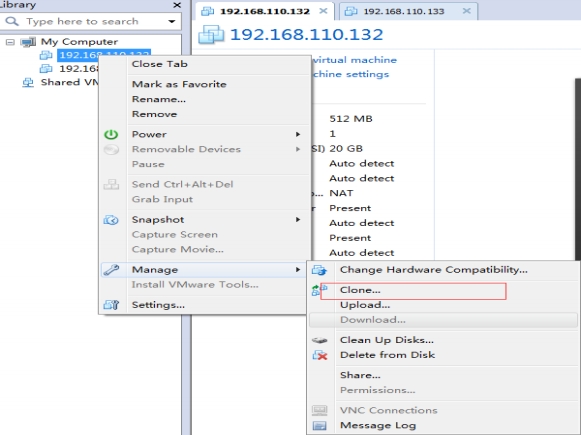
#### 2.0.登录进redis

./redis-cli -h 127.0.0.1 -p 6379 -a 123456

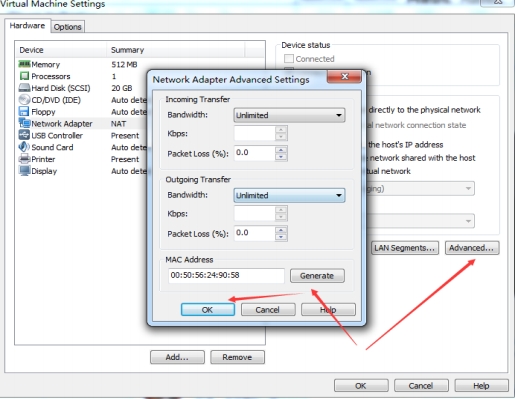


### Redis主从配置

#### 准备克隆3台linux主机，（2台也可以，3台能更好的演示）



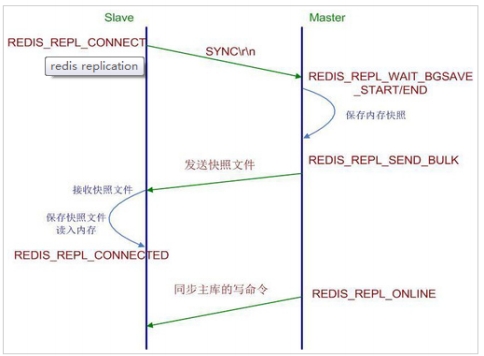
#### 2.2产生新的mac地址（及克隆的都要改IP地址）



#### 主从配置的原理

1. redis的复制功能是支持多个数据库之间的数据同步。一类是主数据库（master） 一类是从数据库（slave），主数据库可以进行读写操作，当发生写操作的时候自 动将数据同步到从数据库，而从数据库一般是只读的，并接收主数据库同步过来 的数据，一个主数据库可以有多个从数据库，而一个从数据库只能有一个主数据 库。
2. 、通过redis的复制功能可以很好的实现数据库的读写分离，提高服务器的负载能 力。主数据库主要进行写操作，而从数据库负责读操作
3. 主库有读写权限、从库只有读权限。
4. （自己理解）

原理就是从同步主，主发生的读写操作，保存内存快照，主发生快照文件给从，从接受快照文件，读取内容，然后同步主库的命令



#### 主从复制配置

1. 修改从的redis中的redis.conf文件，从同步主，所以只要配主的端口  
   2.找到redis.conf文件的文件 vim命令进入编辑，在最后加上  
   -slaveof 主的端口号 6379  
   -masterauth 123456 --主redis服务器配置了密码，则需要配置  
   3.一定记住2点：第一点：使用root账号  
    第二点：关闭防火器  
   4.如果是拷贝别人的redis.conf，必须进入命令符修改，不能在软件外修改，否则会报错误。  
   5.配置好后，必须关闭从的redis进程 ，用ps -aux | grep redis 查看进程  
    用 kill -9 进程端口号。  
   6.在从中测试写操作，会报错，只能读，keys \* 可以查看到主的key-value

### 3.哨兵机制文档

1. 拷贝超哥文件，替换主的sentinel.conf文件  
   2.找到该文件，然后进入修改  
   3.修改sentinel.conf配置，使用vim 命令进入

Sentinel monitor mymast 主的端口号 6379 1  
#主节点 名称 IP 端口号 选举次数(1票就升级主服务器，因为服务器不多)

1. 启动哨兵模式  
   ./redis-server /usr/local/redis/etc/sentinel.conf --sentinel &  
     
   5.配置以上基本就可以出来  
     
   6.还可以配置修改心跳检测5000毫秒  
   sentinel down-after-milliseconds mymaster 30(原值为30000 30毫秒)  
   7.配置合格从节点  
   sentinel parakkel-syncs mymaster 2  
   8.停止哨兵模式  
   ps -aux|grep ‘26379’  
   kill -9 哨兵进程号

### Redis事务文档

#### 4.1.Redis 事务可以一次执行多个命令， 并且带有以下两个重 要的保证

1. 隔离操作  
   2.原子操作

#### 4.2事务三阶段

1. 开始事务  
   2.命令入队  
   3.执行事务



#### 4.3.Redis事务命令

### Redis持久化文档

#### 5.1什么是Redis持久化

就是将内存数据保存到硬盘

#### 5.2Redis持久化存储

##### – RDB(默认方式)

• 二进制文件形式，不是实时存储，规定时间多少key变更才存储，体 积小  
  
 RDB 是在某个时间 点将数据写入一个临时文件，持久化结束后，用 这个临时文件替换上次持久化的文件，达到数据恢复。

优点：使用单独子进程来进行持久化，主进程不会进行任何 IO 操作， 保证了 redis 的高性能

缺点：RDB 是间隔一段时间进行持久化，如果持久化之间 redis 发生 故障，会发生数据丢失。所以这种方式更适合数据要求不严谨的时候

• 这里说的这个执行数据写入到临时文件的时间点是可以通过配置来自 己确定的，通过配置redis 在 n 秒内如果超过 m 个 key 被修改这执行 一次 RDB 操作。这个操作就类似于在这个时间点来保存一次 Redis 的 所有数据，一次快照数据。所有这个持久化方法也通常叫做 snapshots。

• RDB 默认开启，redis.conf 中的具体配置参数如下；

*#dbfilename：持久化数据存储在本地的文件*

dbfilename dump.rdb

*#dir：持久化数据存储在本地的路径，如果是在/redis/redis-3.0.6/src下启动 的redis-cli，则数据会存储在当前src目录下*

dir ./

*##snapshot触发的时机，save*

*##如下为900秒后，至少有一个变更操作，才会snapshot*

*##对于此值的设置，需要谨慎，评估系统的变更操作密集程度*

*##可以通过“save “””来关闭snapshot功能*

*#save时间，以下分别表示更改了1个key时间隔900s进行持久化存储；更改 了10个key300s进行存储；更改10000个key60s进行存储。*

save 900 1

save 300 10

save 60 10000

*##当snapshot时出现错误无法继续时，是否阻塞客户端“变更操作”，“错误” 可能因为磁盘已满/磁盘故障/OS级别异常等* stop-writes-on-bgsave-error yes

*##是否启用rdb文件压缩，默认为“yes”，压缩往往意味着“额外的cpu消耗”， 同时也意味这较小的文件尺寸以及较短的网络传输时间* rdbcompression yes



##### – AOF(推荐方式)

Append-only file，将“操作 + 数据”以格式化指令 的方式追加到操作日志文件的尾部，在 append 操 作返回后(已经写入到文件或者即将写入)，才进 行实际的数据变更，“日志文件”保存了历史所有 的操作过程

• 我们可以简单的认为 AOF 就是日志文件，此文件 只会记录“变更操作”(例如：set/del 等)

• AOF 默认关闭，开启方法，修改配置文件 reds.conf：appendonly yes



• 实时存储，日志文件方式存储，文件大，文件尾部进行累加。

##### -AOF与RDB区别

1) AOF 更加安全，可以将数据更加及时的同步到 文件中，但是 AOF 需要较多的磁盘 IO 开支，AOF 文件尺寸较大，文件内容恢复数度相对较慢。

• 2) snapshot，安全性较差，它是“正常时期”数据 备份以及 master-slave 数据同步的最佳手段，文 件尺寸较小，恢复数度较快。

推荐： master 通常使用 AOF，slave 使用 snapshot，

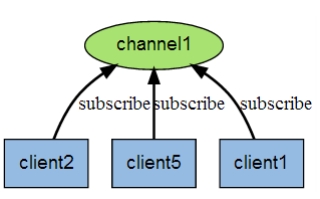
### Redis发布订阅文档

#### Redis 发布订阅(pub/sub)是一种消息通信模式

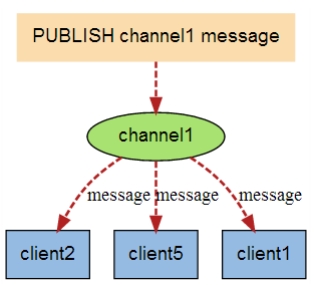
发送者 (pub)发送消息，订阅者(sub)接收消息。

• Redis 客户端可以订阅任意数量的频道。

• 下图展示了频道 channel1 ， 以及订阅这个频道的三个客 户端 —— client2 、 client5 和 client1 之间的关系：



当有新消息通过 PUBLISH 命令发送给频道 channel1 时， 这个消息就会被发送给订阅它的三个客户端：



#### 实例

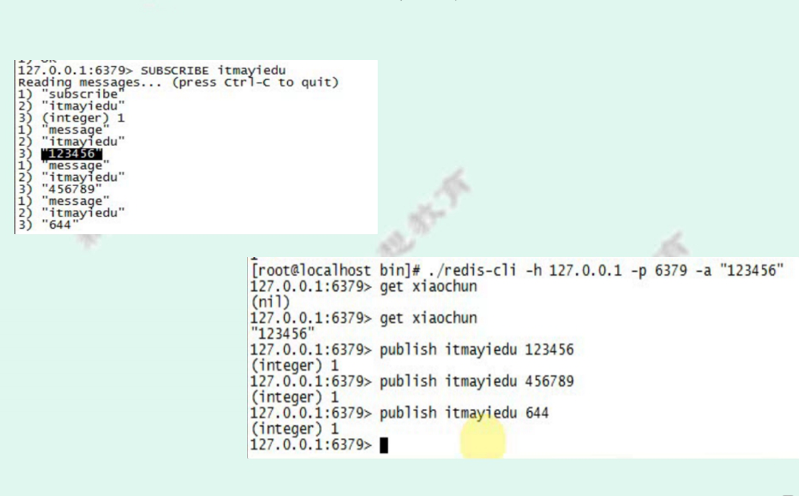




#### 发布订阅命令



#### 效果



#### 