Docker 搭建 Redis 主从复制 + 哨兵

作者：724555508
原文链接：https://ld246.com/article/1594025620292
来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="1917630972.png" data-src=
https://b3logfile.com/file/2020/07/1917630972-6aef28ea.png?imageView2/2/interlace/1/for
at/jpg">
<h3 id="一-安装Redis">一、安装 Redis</h3>
<h4 id="1-拉取官方的镜像-标签为-3-2">1.拉取官方的镜像，标签为 3.2</h4>
<pre><code class="highlight-chroma"><span class="highlight
cl"> [root@localhost /]# docker pull redis:3.2
</code></pre>
<h4 id="2-下载完成后-我们就可以在本地镜像列表里查到-REPOSITORY-为-redis-标签为-3-2-的
像-">2.下载完成后，我们就可以在本地镜像列表里查到 REPOSITORY 为 redis,标签为 3.2 的镜像。<
h4>
<pre><code class="highlight-chroma"><span class="highlight
cl"> [root@localhost /]# docker images redis
  REPOSITORY
  TAG   IMAGE ID   CREATED   SIZE
  redis   3
2   43c923d57784   2 weeks ago   193.9 MB
</code></pre>
<h4 id="3-运行容器">3.运行容器</h4>
<pre><code class="highlight-chroma"><span class="highlight
cl"> [root@localhost /]# docker run -p 6379:6379 -v $PWD/data:/data -d redis:3.2 redis-ser
er --appendonly yes
  43f7a65ec7f8b
64eb1c5d82bc4fb60e5eb31915979c4e7821759aac3b62f330
</code></pre>
命令说明：
-p 6379:6379：将容器的 6379 端口映射到主机的 6379 端口
-v $PWD/data:/data：将主机中当前目录下的 data 挂载到容器的/data
redis-server --appendonly yes：在容器执行 redis-server 启动命令，并打开
edis 持久化配置
<h4 id="4-连接-查看容器">4.连接、查看容器</h4>
<pre><code class="highlight-chroma"><span class="highlight
cl"> [root@localhost /]# docker exec -it 43f7a65ec7f8 redis-cli
  172.17.0.1:637
> info
  # Server
  redis_version:3.
.0
  redis_git_sha1:
0000000
  redis_git_dirty:0
  redis_build_id:f
49541256e7d446
  redis_mode:sta
dalone
  os:Linux 4.2.0-1
-generic x86_64
  arch_bits:64
  multiplexing_api
epoll
  ...
</code></pre>
<h3 id="二-主从-集群">二、主从、集群</h3>
<h4 id="1-运行-Redis-镜像">1.运行 Redis 镜像</h4>

首先使用 docker 启动 3 个 Redis 容器服务，分别使用到 6379、6380、6381 端口</
trong>
<pre><code class="highlight-chroma"><span class="highlight
cl">docker run --name redis-6379 -p 6379:6379 -d redis:3.2 redis-server
docker run --nam
redis-6380 -p 6380:6379 -d redis:3.2 redis-server
docker run --nam
redis-6381 -p 6381:6379 -d redis:3.2 redis-server
</code></pre>
<h4 id="2-配置-Redis-集群">2.配置 Redis 集群</h4>
使用如下命令查看容器内网的 ip 地址等信息
<pre><code class="highlight-chroma"><span class="highlight
cl">docker inspect containerid（容器ID）
</code></pre>
3 个 Redis 的内网 ip 地址为：
<pre><code class="highlight-chroma"><span class="highlight
cl">redis-6379：172.17.0.3:6379
redis-6380：172.1
.0.4:6379
redis-6381：172.1
.0.5:6379
</code></pre>
进入 docker 容器内部，查看当前 Redis 角色（主 master 还是从 slave）（命令：info replicati
n）
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@localhost /]# docker exec -it 007f7ab412b9 redis-cli
127.0.0.1:6379>
info replication
># Replication
role:master
connected_slaves:

master_repl_offset
3860
repl_backlog_acti
e:1
repl_backlog_size:
048576
repl_backlog_first
byte_offset:2
repl_backlog_histl
n:3859

127.0.0.1:6379>

</code></pre>
可以看到当前 3 台 Redis 都是 master 角色，使用 redis-cli 命令修改 redis-6380、redis-6381
的主机为 172.17.0.3:6379
<pre><code class="highlight-chroma"><span class="highlight
cl">SLAVEOF 172.17.0.3 6379
</code></pre>
再次查看主机 info，已经有两个从机了（.4 和 .5）
<pre><code class="highlight-chroma"><span class="highlight

cl">127.0.0.1:6379> info replication
Replication
role:master
connected_slaves:

slave0:ip=172.17.0
4,port=6379,state=online,offset=3860,lag=0
slave1:ip=172.17.0
5,port=6379,state=online,offset=3860,lag=0
master_repl_offset
3860
repl_backlog_acti
e:1
repl_backlog_size:
048576
repl_backlog_first
byte_offset:2
repl_backlog_histl
n:3859
</code></pre>
至此，Redis 下的主从配置就 ok 了。
<h3 id="三-Sentinel-哨兵">三、Sentinel 哨兵</h3>
Redis 的 Sentinel 系统用于管理多个 Redis 服务器（instance）， 该系统执行以下三个任务（
体介绍可参考 <a href="https://ld246.com/forward?goto=http%3A%2F%2Fredisdoc.com%2Ft
pic%2Fsentinel.html" target="_blank" rel="nofollow ugc">链接）：
<pre><code class="highlight-chroma"><span class="highlight
cl">监控（Monitoring）： Sentinel 会不断地检查你的主服务器和从服务器是否运作正常。
提醒（Notificatio
）： 当被监控的某个 Redis 服务器出现问题时， Sentinel 可以通过 API 向管理员或者其他应用程序
送通知。
自动故障迁移（Au
omatic failover）： 当一个主服务器不能正常工作时， Sentinel 会开始一次自动故障迁移操作， 它
将失效主服务器的其中一个从服务器升级为新的主服务器， 并让失效主服务器的其他从服务器改为复
新的主服务器； 当客户端试图连接失效的主服务器时， 集群也会向客户端返回新主服务器的地址，
得集群可以使用新主服务器代替失效服务器。
</code></pre>
接下来直接进入 3 台 Redis 容器内部进行配置
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@localhost /]# docker exec -ti 容器id /bin/bash
</code></pre>
进入根目录创建 sentinel.conf 文件
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@localhost /]# cd / && touch sentinel.conf && touch log.txt
</code></pre>
修改 sentinel.conf 文件内容为：
<pre><code class="highlight-chroma"><span class="highlight
cl">sentinel monitor mymaster 172.17.0.3 6379 1
#添加为后台运行
daemonize yes
#指定日志目录
logfile "/log.txt"
</code></pre>
最后，启动 Redis 哨兵：
<pre><code class="highlight-chroma"><span class="highlight

cl">[root@localhost /]# redis-sentinel /sentinel.conf
</code></pre>
至此，Sentinel 哨兵配置完毕。
测试验证，如图
<img src="https://ld246.com/images/img-loading.svg" alt="FgHLpd.png" data-src="http
://b3logfile.com/file/2020/02/FgHLpd-8fea0d15.png?imageView2/2/interlace/1/format/jpg"