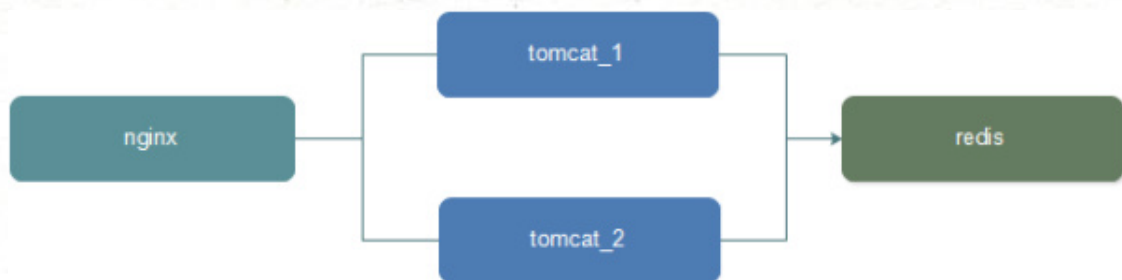


<https://lanjingling.github.io/2015/12/15/tomcat-redis-session/>



配置过程

0、实验环境介绍：

Centos6、nginx-1.6.0、apache-tomcat-7.0.27（两台）、
redis_version:2.8.23

1、安装nginx、tomcat、redis服务；

软件	version	ip : port
Nginx	1.6.0	192.168.137.122 : 84
Redis	2.8	192.168.137.122 : 6379
Tomcat_1	7.0.27	192.168.137.122 : 8080
Tomcat_2	7.0.27	192.168.137.101 : 8080

2、测试：

- 为tomcat_1编写测试页面（a.jsp），显示 “ response from tomcat_1 ”，同时页面提供按钮显示当前session值（<%=session.getId() %>），放到tomcat_1 服务器的webapps/ROOT目录下；
- 为tomcat_2编写测试页面（a.jsp），显示 “ response from tomcat_2 ”，同时页面提供按钮显示当前session值（<%=session.getId() %>），放到tomcat_2 服务器的webapps/ROOT目录下；

此时分别访问 [http://192.168.137.122 : 8080](http://192.168.137.122:8080) 和

[http://192.168.137.101 : 8080](http://192.168.137.101:8080) 地址，因为访问的是不同web服务器，所以各自显示不同的页面内容及session值肯定不同。

3、配置tomcat的session管理（持久化到redis中）：

- 添加jedis-2.0.0.jar、tomcat-redis-session-manager-1.2-

tomcat-7-java-7、commons-pool-1.3.jar 三个jar到tomcat的lib目录下；

- 修改tomcat的conf/context.xml 文件；

```
<Valve className="com.radiadesign.catalina.session.RedisSessionHandlerValve" />
<Manager className="com.radiadesign.catalina.session.RedisSessionManager"
    host="192.168.137.122"
    port="6379"
    database="0"
    maxInactiveInterval="60" />
```

- 重启tomcat；

4、配置nginx进行负载均衡（upstream）：

在nginx的conf目录下新建conf.d目录，然后在其中新建tomcat.conf文件，内容如下：

```
upstream tomcat {
    #ip_hash;
    server 127.0.0.1:8080;
    server 192.168.137.101:8080;
}
server {
    listen 84;
    server_name localhost;

    #动态页面交给http://tdt_wugk，也即我们之前在nginx.conf定义的upstream
    #tdt_wugk 均衡
    location ~ .*\.(\.do|jsp|action)?$ {
        proxy_pass http://tomcat;
        include proxy.conf;
    }

    #配置Nginx动静分离，定义的静态页面直接从Nginx发布目录读取。
    location ~ .*\.(\.html|htm|gif|jpg|jpeg|bmp|png|ico|txt|js|css)$ {
        root /data/;
    }
}
```

5、验证：

- 访问 `http://192.168.137.122 : 8080/a.jsp` 直接请求到tomcat_1服务器，显示 “ response from tomcat_1 ”，session 值为 ‘56E2FAE376A47F1C0961D722326B8423’ ；

- 访问 `http://192.168.137.101 : 8080/a.jsp` 直接请求到tomcat_2服务器，显示 “ response from tomcat_2 ” ， session 值为 ‘56E2FAE376A47F1C0961D722326B8423’ ；
- 访问 `http://192.168.137.122 : 84/a.jsp` 请求到 nginx 反向代理到指定Web服务器，由于默认使用轮询负载方式，反复刷新页面显示的内容在 “ response from tomcat_1 ” 和 “ response from tomcat_2 ” 之间切换，但 session 值保 ‘56E2FAE376A47F1C0961D722326B8423’ ；
- 使用 redis-cli 连接 redis 服务器，查看会显示有 “56E2FAE376A47F1C0961D722326B8423” key的 session 数据，value为序列化数据。

总结

- 1、使用tomcat7搭建时，redis的客户端jedis版本不能太高，否则启动tomcat时会报错；我使用的是tomcat7.0.27、jedis-2.0.0.jar、tomcat-redis-session-manager-1.2-tomcat-7-java-7、commons-pool-1.3.jar
- 2、Nginx+Redis+Tomcat集群实现session共享，配置过程简单如下：
 - 首先搭建好单机版的tomcat将session持久化到redis中；
 - 然后配置nginx轮询tomcat（负载均衡）；

```
location / {  
    # root html;  
    # index index.html index.htm;  
    proxy_pass http://tomcat;  
    include proxy.conf;  
}  
}
```