

# SLB+nginx 负载均衡宕机模拟

作者: [jyl](#)

原文链接: <https://ld246.com/article/1621930636812>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



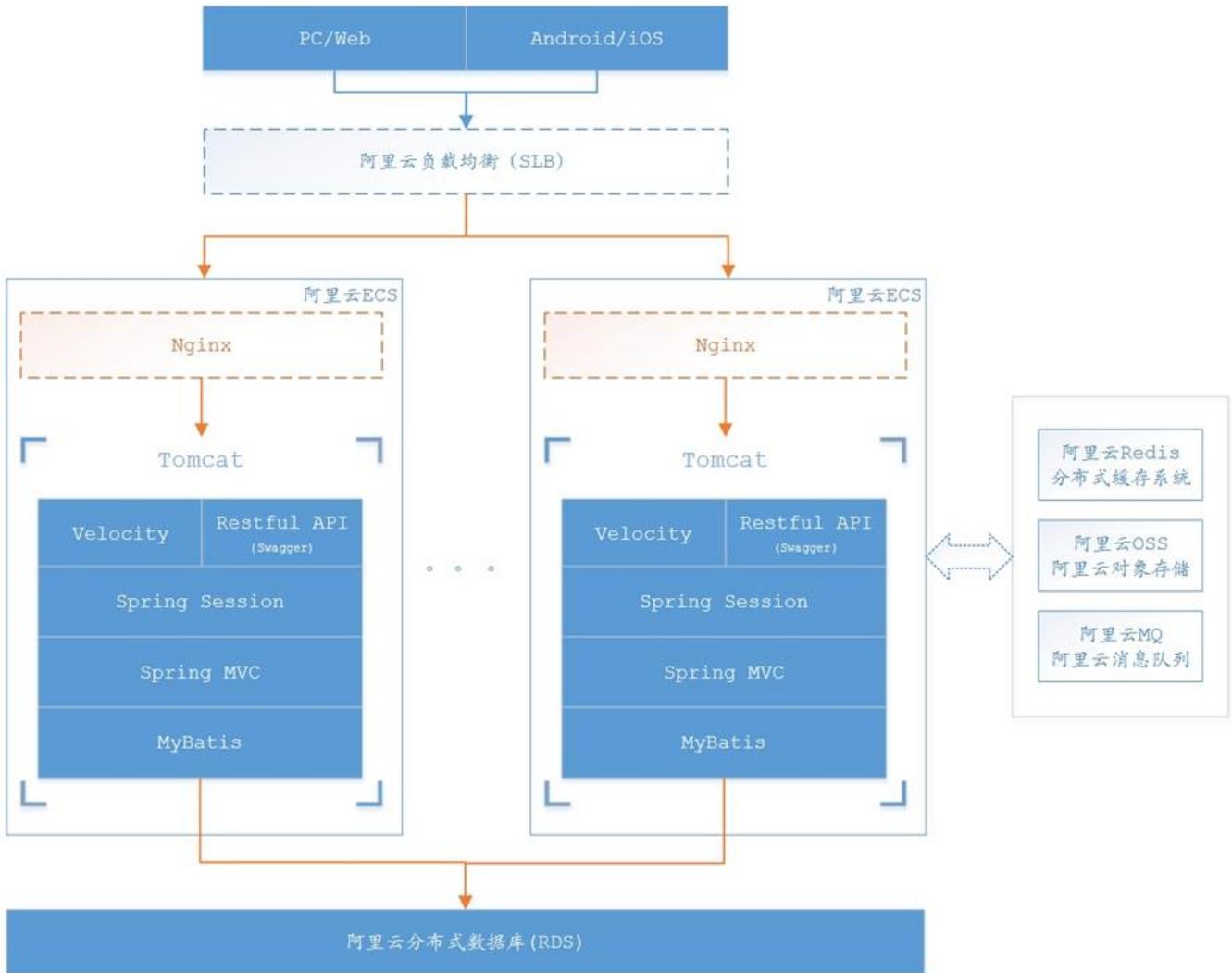
上篇[Nginx原理探究](#)这篇文章主要探索了ng的几个核心点，分别是:1、反向代理 2、动静分离 3、负载均衡 4、跨域配置。今天主要探索下ng上的负载均衡,与其说是探索，不如说是进行一次实验(模拟服务器)。

首先，参考了《[nginx和阿里云的SLB的区别](#)》和[掘金](#)上的一张负载架构图。

简单分析ng上的SLB(负载均衡) 和阿里云SLB 的区别

1、"Nginx是一个有负载均衡web环境，阿里云SLB是一个单独的负载均衡系统"

2、如图分析，阿里云SLB更多的应用于服务器间的负载均衡，然后ng多应用于单一服务器间的相同用的负载均衡(当然人家也能做服务器负载)。本质没有区别，都是做负载均衡的，但是阿里SLB偏应于系统端，ng偏应用于服务端。可以参考"[阿里云负载均衡SLB是什么，如何使用](#)"



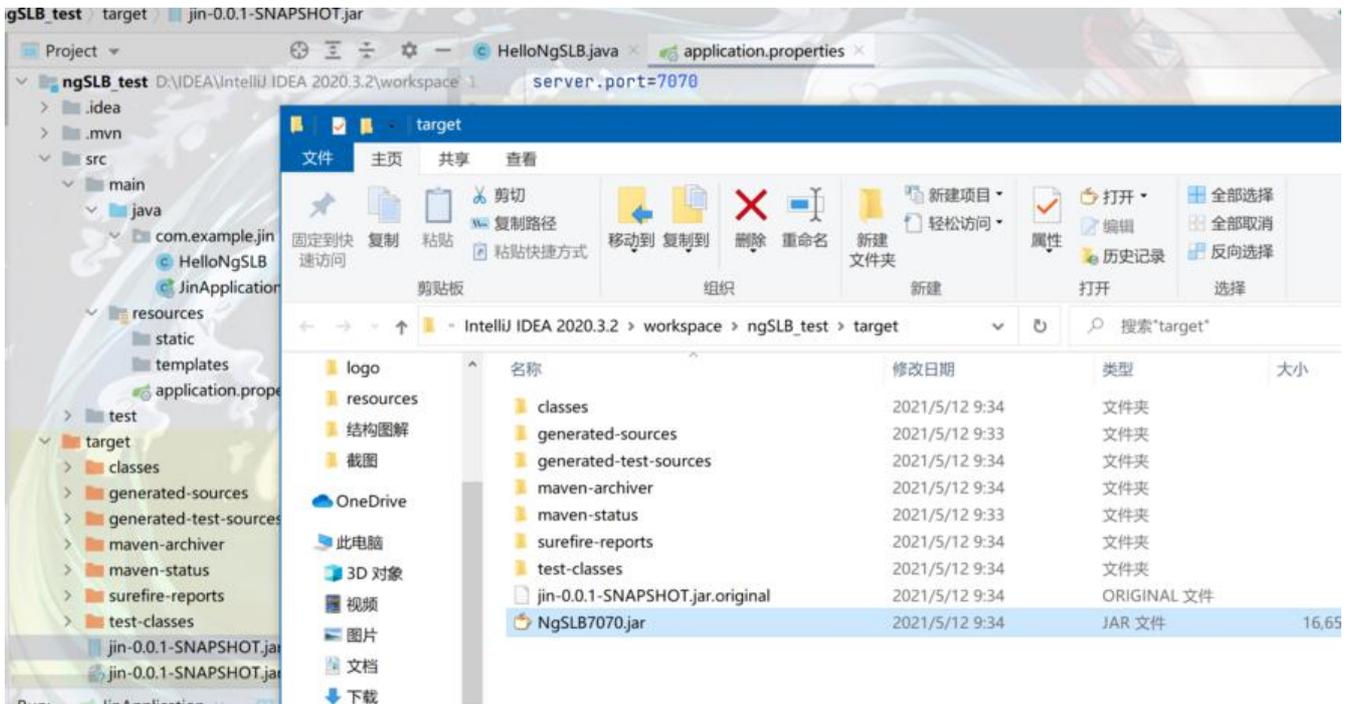
补一下SLB这个概念，百度百科：“**服务器负载均衡 (Server Load Balancing)**，可以看作HSRP (热备份**路由器**协议) 的扩展，实现多个服务器之间的负载均衡”。

阿里云SLB的实验是做不了了，我只有**一台服务器**，模拟一下ng的SLB吧。首先我先随便写两个Web目(内容一样)先挂在服务器上。(写的是真的简单。。。)两个相同的项目，访问请求放回"hello NgSLB"，分别用7070、6060端口启动。(用这个makedown上传图片不老好传的，6060那个和7070图一样)

```

D:\IDEA\IntelliJ IDEA 2020.3.2\workspace 1 package com.example.jin;
2
3 import org.springframework.web.bind.annotation.RequestMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 /**
7  * @author jinyunlong
8  * @date 2021/5/12 9:25
9  * @profession ICBC 锅炉房保安
10 */
11 @RestController
12 public class HelloNgSLB {
13     @RequestMapping(value = "/NgSLB")
14     public String hellongsלב(){
15         String src = "hello NgSLB";
16         return src;
17     }
18 }
19

```



启动一下项目，监听端口。

```

next path
2021-05-12 09:48:43.263 INFO 15847 --- [main] com.example.jin.JinApplication : Started JinAp
unning for 5.275)
C
root@hecs-x-medium-2-linux-20200324093024 ngSLB_test]# netstat -an|grep 6060
cp6 0 0 :::6060 :::* LISTEN
cp6 0 0 :::1:50348 :::1:6060 TIME_WAIT
root@hecs-x-medium-2-linux-20200324093024 ngSLB_test]# netstat -an|grep 7070
cp6 0 0 :::7070 :::* LISTEN
root@hecs-x-medium-2-linux-20200324093024 ngSLB_test]#

```

浏览器都可以正常访问到了，然后在ng上配置一下反向代理和负载均衡。

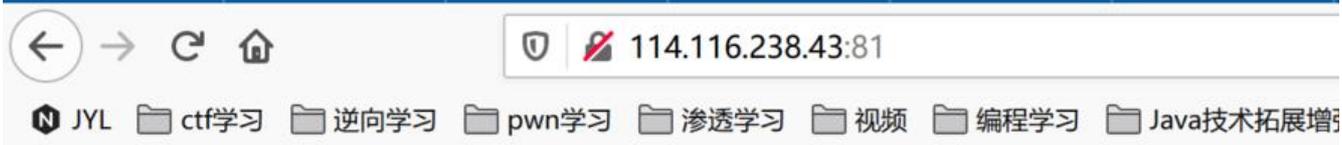
```
upstream gggg {
    #ip_hash;
    server 114.116.238.43:7070;
    server 114.116.238.43:6060;
}

server {
    listen 81;
    server_name 114.116.238.43;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location ^~/ {
        proxy_pass http://gggg/NgSLB;
        proxy_connect_timeout 1;
        proxy_read_timeout 1;
        proxy_send_timeout 1;
    }
}
```



hello NgSLB

一通转转转然后可以正常访问到接口。其实这里有个不直观的地应该是把两个项目输出值稍作修改的不过模拟的是企业级接口，原理应该没问题。

然后我把6060端口项目关了，默认他宕了。

```
[root@hecs-x-medium-2-linux-20200324093024 work01]# cd ngSLB_test/
[root@hecs-x-medium-2-linux-20200324093024 ngSLB_test]# ls
NgSLB6060.jar NgSLB7070.jar
[root@hecs-x-medium-2-linux-20200324093024 ngSLB_test]# ps -ef|grep NgSLB6060.jar
root      15744 16239  0 09:48 pts/0    00:00:07 java -jar NgSLB6060.jar
root      31658 16239  0 10:26 pts/0    00:00:00 grep --color=auto NgSLB6060.jar
[root@hecs-x-medium-2-linux-20200324093024 ngSLB_test]# kill -9 15744
[root@hecs-x-medium-2-linux-20200324093024 ngSLB_test]# ps -ef|grep NgSLB6060.jar
root      31723 16239  0 10:26 pts/0    00:00:00 grep --color=auto NgSLB6060.jar
[1]-  Killed                  java -jar NgSLB6060.jar
[root@hecs-x-medium-2-linux-20200324093024 ngSLB_test]# netstat -an|grep 6060
[root@hecs-x-medium-2-linux-20200324093024 ngSLB_test]# netstat -an|grep 7070
tcp6      0      0 :::7070          :::*              LISTEN
[root@hecs-x-medium-2-linux-20200324093024 ngSLB_test]#
```

服务还在，那么这时ng就只能去轮询7070的项目了



现在把两个端口都停了，就都访问不到了。至此，ng的负载均衡模拟一个服务宕掉后另一个能轮询到一个正常访问的实验就成功了。



## An error occurred.

Sorry, the page you are looking for is currently unavailable.  
Please try again later.

If you are the system administrator of this resource then you should check [error log](#) for details.

*Faithfully yours, nginx.*

其实ng负载均衡配置项还有很多别的指令这次都没有用到，比如weight(权重)、down(暂时不参与负载)、ackup(备机?)等等，以后再做研究吧。

ng探索基本也就到这了，下午去帮同事看看PUBLICCMS这个开源系统的代码逻辑(他偏偏用ssh，我学学ssm，现在用jdbc、jpa。。。)