

# 09.LNMT动态网站

- 09.LNMT动态网站
  - JVM故障排查思路

徐亮伟, 江湖人称标杆徐。多年互联网运维工作经验, 曾负责过大规模集群架构自动化运维管理工作。擅长Web集群架构与自动化运维, 曾负责国内某大型电商运维工作。

个人博客"[徐亮伟架构师之路](#)"累计受益数万人。

笔者Q:552408925、572891887

架构师群:471443208

## Nginx + Tomcat

Tomcat默认监听在8080端口

Tomcat依赖于java

### 1.安装 jdk

```
[root@tomcat-node1-20 ~]# yum install java -y
[root@tomcat-node1-20 ~]# java -version
openjdk version "1.8.0_161"
OpenJDK Runtime Environment (build 1.8.0_161-b14)
OpenJDK 64-Bit Server VM (build 25.161-b14, mixed mode)
```

### 2.安装 tomcat

```
[root@tomcat-node1-20 ~]# mkdir /soft/src -p
[root@tomcat-node1-20 ~]# cd /soft/src
[root@nginx ~]# wget http://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-9/v9.0.7/bin/apache-tomcat-9.0.7.tar.gz
[root@tomcat-node1-20 src]# tar xf apache-tomcat-9.0.7.tar.gz -C /soft
[root@tomcat-node1-20 soft]# cp -r apache-tomcat-9.0.7/ tomcat-8080
[root@tomcat-node1-20 bin]# /soft/tomcat-8080/bin/startup.sh
```

### 3.配置 Nginx proxy 负载均衡

```
[root@nginx nginx]# yum install nginx -y
[root@nginx nginx]# systemctl enable nginx
[root@nginx nginx]# systemctl start
[root@nginx nginx]# cat jsp.conf
```

```
upstream java_prod {
    server 192.168.56.20:8080;
}
server {
    listen 80;
    server_name localhost;
    location / {
        proxy_pass http://java_prod;
        include proxy_params;
    }
}
```

## JVM故障排查思路

- 1.jps获取java进程的PID。
- 2.jstack pid >> java.txt 导出CPU占用高进程的线程栈。
- 3.top -H -p PID 查看对应进程的哪个线程占用CPU过高。
4. echo "obase=16;989"|bc 将线程的PID转换为16进制。
- 5.在第二步导出的java.txt中查找转换成为16进制的线程PID。找到对应的线程栈。
- 6.分析负载高的线程栈都是什么业务操作。优化程序并处理问题。

```
#最大线程数
maxThreads="600"
#初始化时创建的线程数
minSpareThreads="100"
#一旦创建的线程超过这个值，Tomcat就会关闭，不再需要的socket线程。
maxSpareHtreads="500"
#指定当所有可以使用的处理请求的线程数量都被使用时，可以放到处理队列中的请求数，超过这个数的请求将不予处理。
acceptCount="700" />
```