Tomcat实现session共享

一、如何保持session会话

目前，为了使web能适应大规模的访问，需要实现应用的集群部署。集群最有效的方案就是负载均衡，而实现负载均衡用户每一个请求都有可能被分配到不固定的服务器上，这样我们首先要解决session的统一来保证无论用户的请求被转发到哪个服务器上都能保证用户的正常使用，即需要实现session的共享机制。

在集群系统下实现session统一的有如下几种方案：

1、请求精确定位：session sticky，例如基于访问ip的hash策略，即当前用户的请求都集中定位到一台服务器中，这样单台服务器保存了用户的session登录信息，如果宕机，则等同于单点部署，会丢失，会话不复制。

2、session复制共享：session replication，如tomcat自带session共享，主要是指集群环境下，多台应用服务器之间同步session，使session保持一致，对外透明。 如果其中一台服务器发生故障，根据负载均衡的原理，调度器会遍历寻找可用节点，分发请求，由于session已同步，故能保证用户的session信息不会丢失，会话复制,。

此方案的不足之处：

必须在同一种中间件之间完成(如:tomcat-tomcat之间).

session复制带来的性能损失会快速增加.特别是当session中保存了较大的对象,而且对象变化较快时, 性能下降更加显著，会消耗系统性能。这种特性使得web应用的水平扩展受到了限制。

Session内容通过广播同步给成员，会造成网络流量瓶颈，即便是内网瓶颈。

在大并发下表现并不好

3、基于cache DB缓存的session共享

基于 memcache/redis缓存的 session 共享

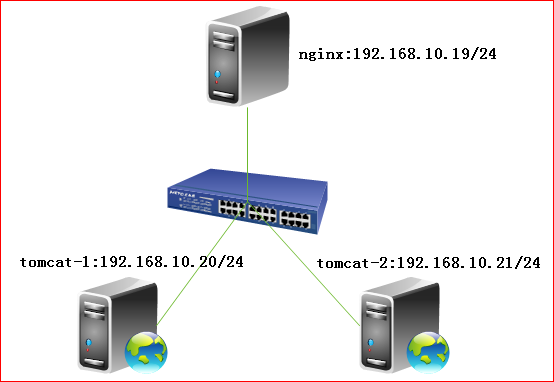
即使用cacheDB存取session信息，应用服务器接受新请求将session信息保存在cache DB中，当应用服务器发生故障时，调度器会遍历寻找可用节点，分发请求，当应用服务器发现session不在本机内存时，则去cache DB中查找，如果找到则复制到本机，这样实现session共享和高可用。

二、本配置基于tomcat7环境，通过tomcat集群自带的session复制，session信息将会被自动复制到各个节点。

案例环境：

|  |  |  |  |
| --- | --- | --- | --- |
| 主机 | 操作系统 | IP地址 | 主要软件 |
| Tomcat-1 | Centos6.5 x86\_64 | 192.168.10.20 | jdk-7u65-linux-x64.gz  apache-tomcat-7.0.54.tar.gz |
| Tomcat-2 | Centos6.5 x86\_64 | 192.168.10.21 |

实验拓扑：



2、在tomcat-1和tomcat-2节点上安装JDK

在安装tomcat之前必须先安装JDK，JDK的全称是java development kit,是sun公司免费提供的java语言的软件开发工具包，其中包含java虚拟机（JVM），编写好的java源程序经过编译可形成java字节码，只要安装了JDK，就可以利用JVM解释这些字节码文件，从而保证了java的跨平台性

安装JDK，配置java环境：

将jdk-7u65-linux-x64.gz解压



将解压的jdk1.7.0\_65目录移致动到/usr/local/下并重命名为java



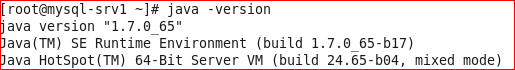
在/etc/profile文件中添加内容如下：



通过source命令执行profile文件，使其生效。



分别在在tomcat-1和tomcat-2节点运行java -version命令查看java版本是否和之前安装的一致。



至此java环境已经配置完成

3、在tomcat-1和tomcat-2节点安装配置tomcat

解压apache-tomcat-7.0.54.tar.gz包



将解压生成的文件夹移动到/usr/local/下，并改名为tomcat7

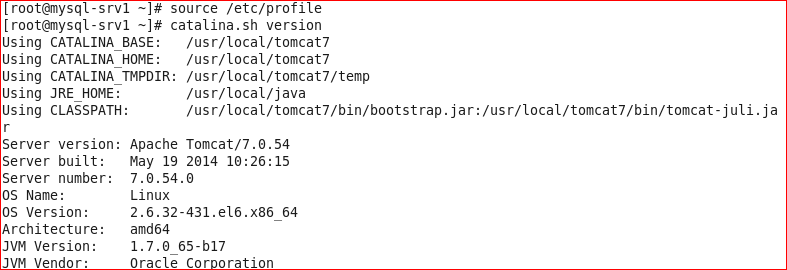


配置tomcat环境变量

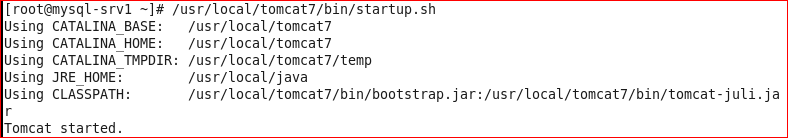
/etc/profile文件内容如下：



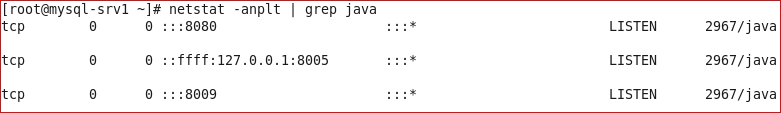
通过source命令执行profile文件，使其生效。



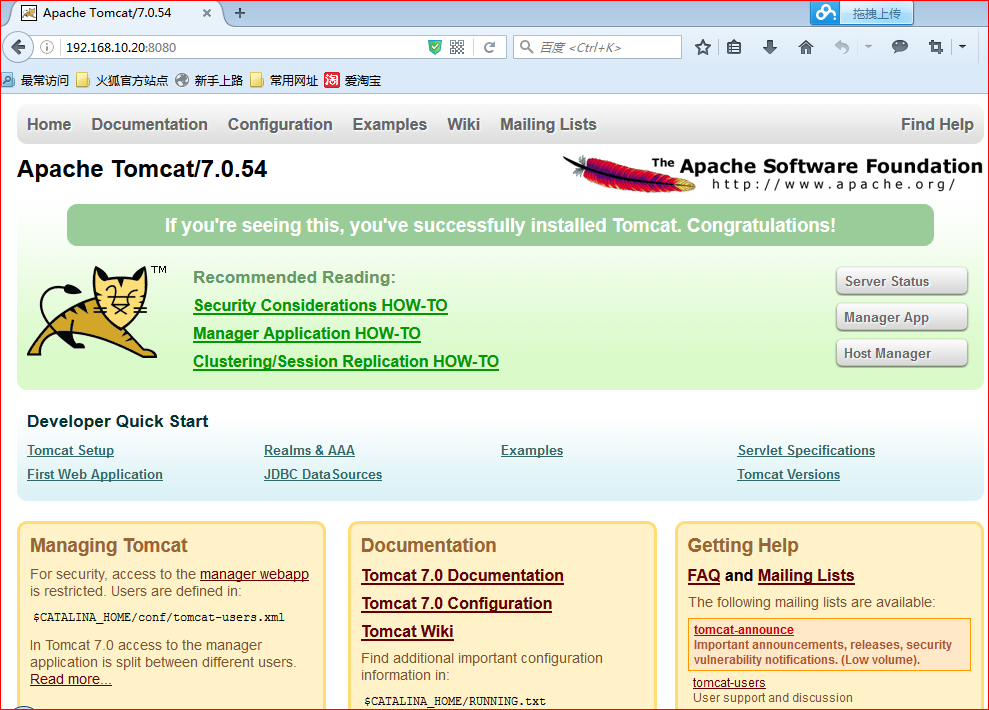
启动tomcat



Tomcat默认运行在8080端口，运行netstat命令查看8080端口监听的信息



打开浏览器分别对tomcat-1和tomcat-2访问测试



如果想关闭tomcat则运行/usr/local/tomcat7/bin/shutdown.sh命令

好了，大家可以看到访成功。说明我们的tomcat安装完成，下面我们来配置一下

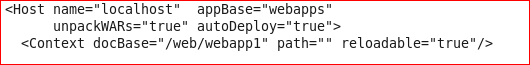
4、修改配置文件

# vim /usr/local/tomcat7/conf/server.xml

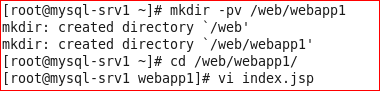
设置默认主机，并增加jvmRoute



定义一个虚拟主机，并将网站文件路径指向/web/webapp1，在host段增加context段



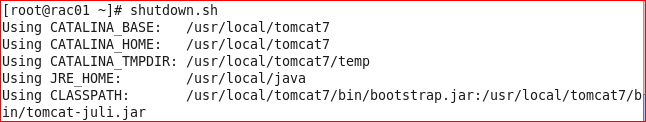
增加文档目录与测试文件

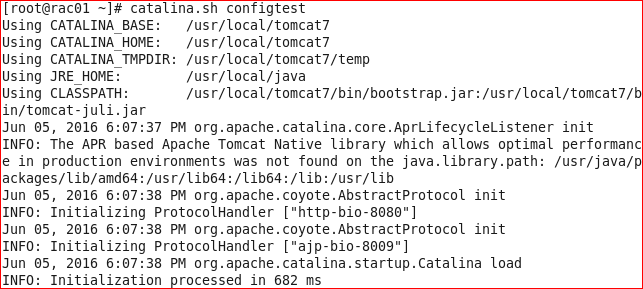


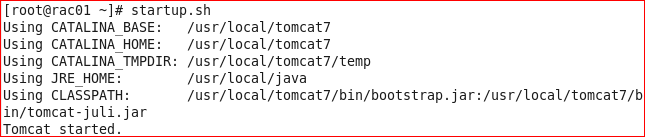
index.jsp内容如下：



停止tomcat运行，检查配置文件并启动tomcat

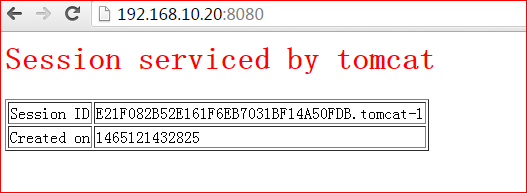


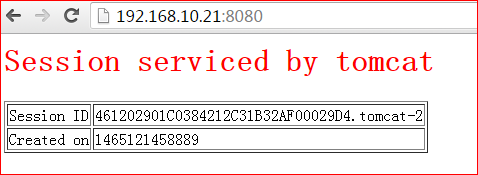




Tomcat-2节点与tomcat-1节点配置基本类似，只是jvmRoute不同，另外为了区分由哪个节点提供访问，测试页标题也不同。其他的配置都相同。过程略

再次测试访问一下





可以看到session会话不同

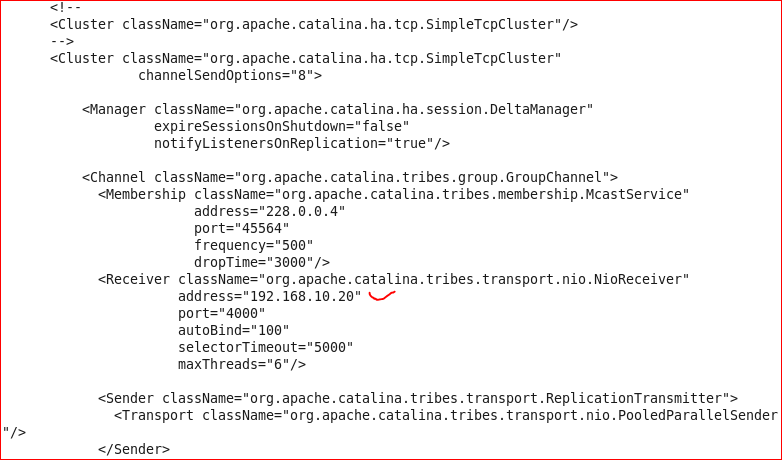
好了，到这里我们准备工作就全部完成了，下面我们来配置tomcat的负载均衡，通过session复制实现会话保持。

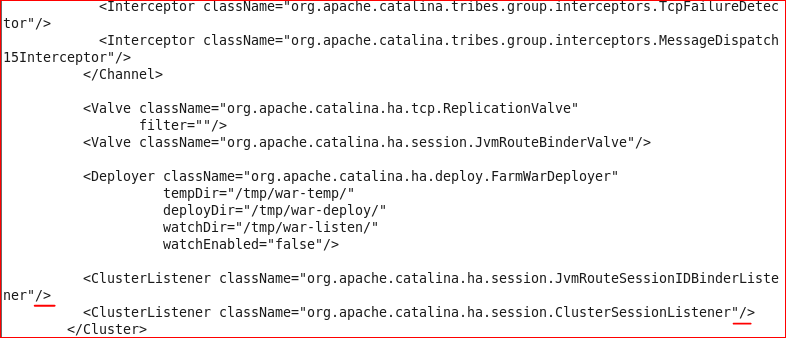
5、配置会话共享集群，分别在tomcat-1和tomcat-2完成下面的操作。

配置server.xml文件

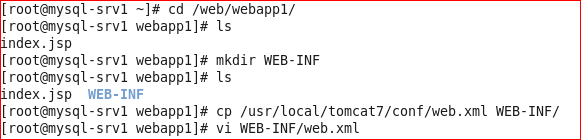


在Server.xml中，找到被注释<Cluster/>节点，在下面添加如下内容：

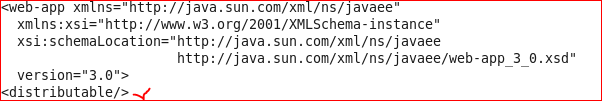




修改应用的web.xml文件，加入标签 <distributable**/**> ，如下所示：

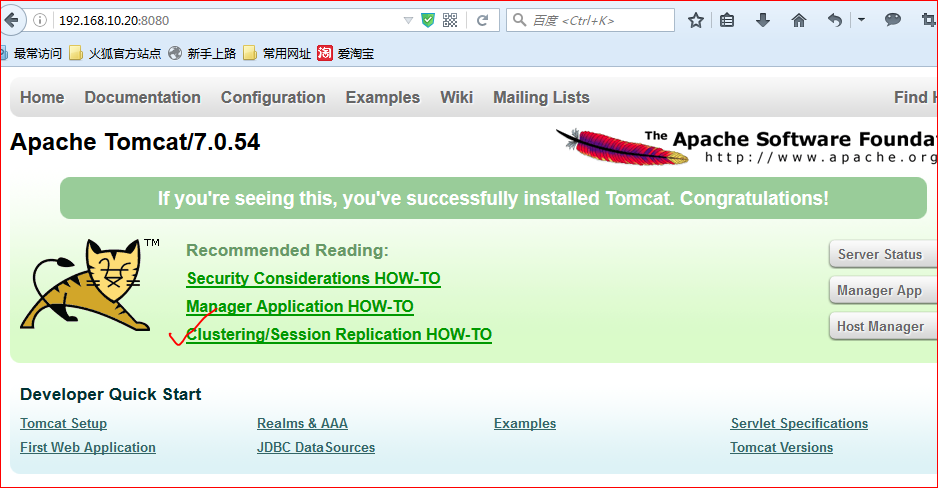


添加标签<distributable**/**>



直接加在**</web-app>** 之前就可以了这个是加入tomcat的session 复制的，做tomcat集群必须需要这一步，否则用户的session 就无法正常使用。

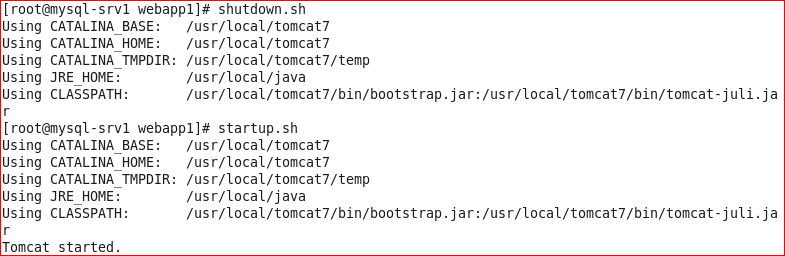
注意：以上内容可以参考clustering/session replication how-to，如下图所示：



注意：tomcat主机必须要指向正确的网关，否则tomcat会启动失败，报如下错误



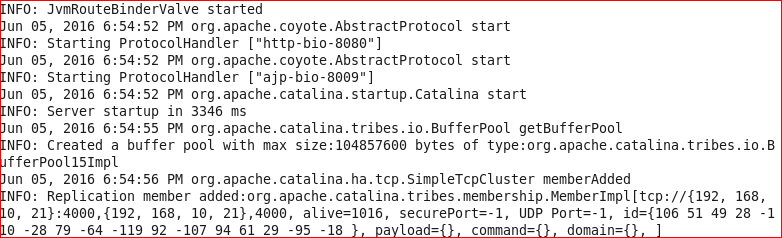
重启tomcat服务并查看端口的监听状态。





Tomcat-2节点与tomcat-1配置类似，只是address要写成tomcat-2的ip。

查看tomcat日志：/usr/local/tomcat7/logs/catalina.yyyy-mm-dd.log



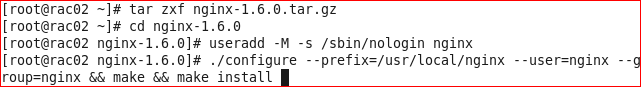
6、nginx服务器配置

用nginx实现tomcat的负载均衡

关闭防火墙

安装相关软件包：#yum –y install pcre-devel zlib-devel

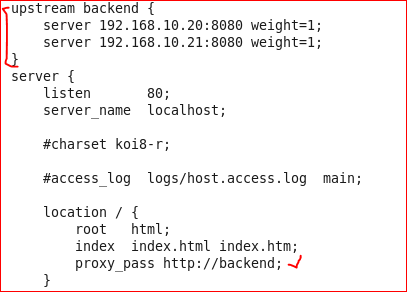
解压并安装nginx



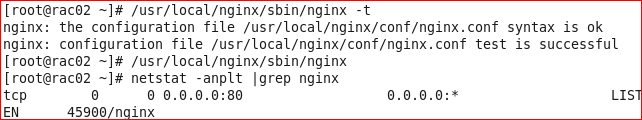
修改nginx.conf文件



修改内容如下：

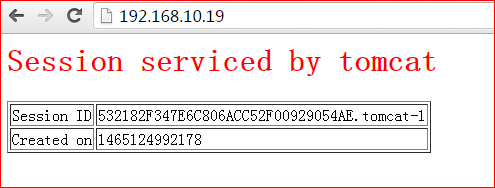


启动nginx服务



7、下面开始测试

在客户端使用浏览器访问nginx的地址



刷新页面

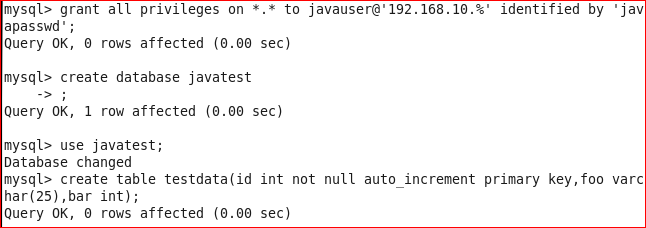


大家可以从图中看到，不管你怎么刷新SessionID都不会变，说明我们的Tomcat的DeltaManager集群配置完成，实现了多台主机之间会话共享

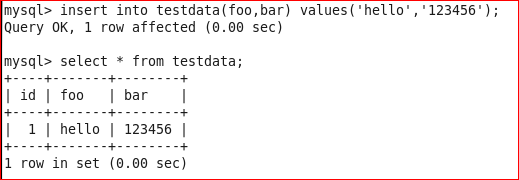
8、tomcat连接mysql数据库

192.168.10.22作为mysql数据库服务器

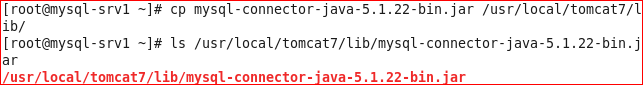
配置mysql



插入些数据



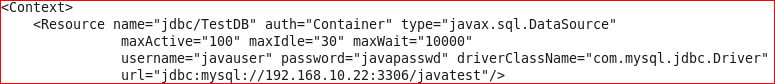
下载mysql-connector-java-5.1.22-bin.jar并复制到$CATALINA\_HOME/lib目录下



context configuration

configure the JNDI datasource in tomcat by adding a declaration for your resource to your context





保存修改并退出

web.xml configuration





保存修改并退出，重启tomcat服务



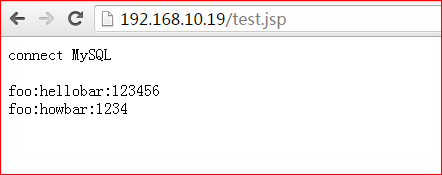
Test code

Now create a simple test.jsp page,内容如下：





测试访问



看到上面的结果说明访问成功

注：

以上配置可以参考tomcat docs

