Tomcat部署文档

目录

**[Tomcat部署文档 1](#_Toc3537)**

**[一、 什么是tomcat 1](#_Toc10662)**

**[名称由来 1](#_Toc9556)**

**[二、 Tomcat工作模式 2](#_Toc31923)**

**[三、 Tomcat文件介绍 3](#_Toc12175)**

**[四、 Tomcat配置 6](#_Toc25477)**

[步骤一、环境准备 6](#_Toc15284)

[步骤二、更改配置文件 6](#_Toc2533)

[关闭tomcat的端口（可不改，默认） 7](#_Toc16962)

[Web访问端口（可不改，默认） 7](#_Toc31435)

[AJP端口（可不改，默认） 8](#_Toc3327)

**[五、 tomcat管理 10](#_Toc16614)**

1. 什么是tomcat

Tomcat是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，由[Apache](https://baike.baidu.com/item/Apache/6265" \t "https://baike.baidu.com/item/tomcat/_blank)、Sun 和其他一些公司及个人共同开发而成。

Tomcat 服务器是一个免费的开放源代码的Web 应用服务器，属于轻量级应用[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "https://baike.baidu.com/item/tomcat/_blank)，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP 程序的首选。实际上Tomcat是Apache 服务器的扩展，但运行时它是独立运行的，所以当你运行tomcat 时，它实际上作为一个与Apache 独立的进程单独运行的。

另外Tomcat还是一个Servlet和JSP容器，让java web app跑在里面的东西，独立的Servlet容器是Tomcat的默认模式。Tomcat 技术先进、性能稳定，而且免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。

名称由来Tomcat最初是由Sun的软件架构师詹姆斯·邓肯·戴维森开发的。后来他帮助将其变为开源项目，并由Sun贡献给Apache软件基金会。由于大部分开源项目O'Reilly都会出一本相关的书，并且将其封面设计成某个动物的素描，因此他希望将此项目以一个动物的名字命名。因为他希望这种动物能够自己照顾自己，最终，他将其命名为Tomcat（英语公猫或其他雄性猫科动物）。而O'Reilly出版的介绍Tomcat的书籍（ISBN 0-596-00318-8）[1]的封面也被设计成了一个公猫的形象。而Tomcat的Logo兼吉祥物也被设计为一只公猫。

下载tomcat可从官网上下载

Tomcat官网：http://tomcat.apache.org/index.html

蔡卓贤:

200 请求已成功，请求所希望的响应头或数据体将随此响应返回。

301 被请求的资源已永久移动到新位置。

302 请求的资源现在临时从不同的 URI 响应请求。

400 1、语义有误，当前请求无法被服务器理解。2、请求参数有误。

401 当前请求需要用户验证。

403 服务器已经理解请求，但是拒绝执行它。

404 请求失败，请求所希望得到的资源未被在服务器上发现。

500 服务器遇到了一个未曾预料的状况，无法完成对请求的处理，会在程序码出错时出现。

501 服务器不支持当前请求所需要的某个功能。无法识别请求的方法。

502 作为网关或者代理工作的服务器尝试执行请求时，从上游服务器接收到无效的响应。

503 由于临时的服务器维护或者过载，服务器当前无法处理请求。

大韦铜:

[图片]

1. **Tomcat工作模式**

Tomcat作为servlet容器，有**三种工作模式**：

* 1、独立的servlet容器，servlet容器是web服务器的一部分；
* 2、进程内的servlet容器，servlet容器是作为web服务器的插件和java容器的实现，web服务器插件在内部地址空间打开一个jvm使得java容器在内部得以运行。反应速度快但伸缩性不足；
* 3、进程外的servlet容器，servlet容器运行于web服务器之外的地址空间，并作为web服务器的插件和java容器实现的结合。反应时间不如进程内但伸缩性和稳定性比进程内优；

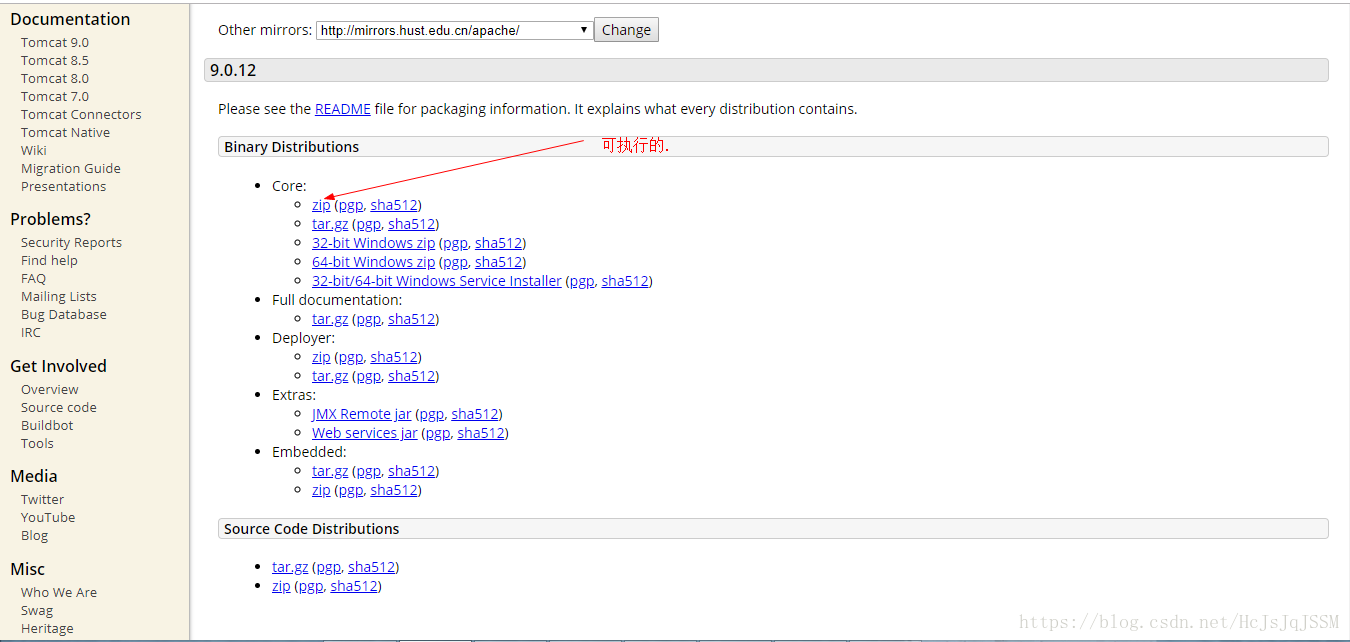
#### **运行模式**

常见于server.xml中的连接器类型通常有4种：  
1) HTTP连接器 2) SSL连接器 3) AJP 1.3连接器 4) proxy连接器

Tomcat Connector(连接器)有三种运行模式：

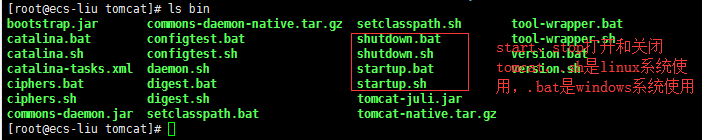
* 1、bio(blocking I/O)，即阻塞式I/O操作，表示Tomcat使用的是传统的Java I/O操作(即java.io包及其子包)；一个线程处理一个请求，缺点：并发量高时，线程数较多，浪费资源。
* 2、nio(new I/O)，Java nio是一个基于缓冲区、并能提供非阻塞I/O操作的Java API，因此nio也被看成是non-blocking I/O的缩写。它拥有比传统I/O操作(bio)更好的并发运行性能，利用 Java 的异步请求 IO 处理，可以通过少量的线程处理大量的请求。
* 3、apr(Apache Portable Runtime/Apache可移植运行时)，Tomcat将以JNI的形式调用Apache HTTP服务器的核心动态链接库来处理文件读取或网络传输操作，从而大大地提高Tomcat对静态文件的处理性能。Tomcat apr也是在Tomcat上运行高并发应用的首选模式。

1. **Tomcat文件介绍**

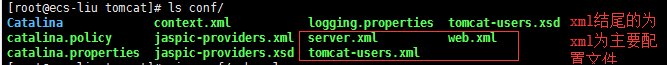
**在官网上下载tar包解压后，会有tomcat目录****Tomcat目录文件**IMG_256

**Tomcat各目录含义**：

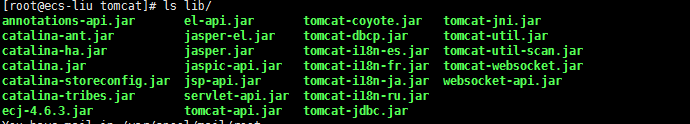
1、**Bin目录**：存放windows或Linux平台上启动和关闭Tomcat的[脚本文件](https://baike.baidu.com/item/%E8%84%9A%E6%9C%AC%E6%96%87%E4%BB%B6" \t "https://baike.baidu.com/item/tomcat/_blank)



2、**Conf目录**：存放Tomcat服务器的各种全局[配置文件](https://baike.baidu.com/item/%E9%85%8D%E7%BD%AE%E6%96%87%E4%BB%B6" \t "https://baike.baidu.com/item/tomcat/_blank)，其中最重要的是server.xml和web.xml



3、**Lib目录**：存放Tomcat服务器所需的各种JAR文件，一些函数接口



4、**Logs目录**：存放Tomcat执行时的日志文件

IMG_256

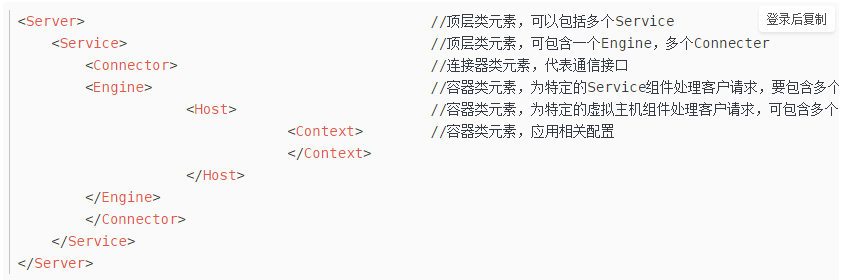
5、**Webapps目录**：Tomcat的主要Web发布目录，默认情况下把Web应用文件放于此目录（应用程序部署的位置），平时war包也放在该目录，启动tomcat时会自动解压

IMG_256

6、**Work目录**：存放JSP编译后产生的[class文件](https://baike.baidu.com/item/class%E6%96%87%E4%BB%B6" \t "https://baike.baidu.com/item/tomcat/_blank)（系统工作过程中jsp文件执行生成对应的文件目录）

7、**Temp目录**：临时文件

**Tomcat配置文件中的格式**



1. **Tomcat配置**

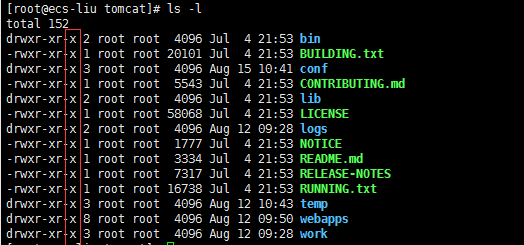
**步骤一、环境准备**

Tomcat由于是java编写，运行tomcat前需要安装jdk环境

Yum -y install install java-1.8.0-openjdk install java-1.8.0-openjdk-devel

给Tomcat下的文件赋予可执行权限,a+x是Linux下的所有用户,-R是当前目录下,\*表示所有文件

Chmod a+x /usr/local/tomcat/\*



**步骤二、更改配置文件**

**Tomcat默认端口：**

**8005**--关闭tomcat进程所用。当执行shutdown.sh关闭tomcat时就是连接8005端口执行“SHUTDOWN”命令--由此，我们直接telnet8005端口执行“SHUTDOWN”（要大写，小写没用；不运只能telnet 127.0.0.1 8005其他地址telnet都不能连接）也可以成功关闭tomcat。同时反之如果8005端口未监听那么tomcat无法用shutdown.sh关闭。

**8009**--httpd等反向代理tomcat时就可以使用使用[ajp协议反向代理](http://www.cnblogs.com/lsdb/p/8087185.html" \t "https://www.cnblogs.com/lsdb/p/_blank)到该端口。虽然我们经常都是使用http反向代理到8080端口，但由于ajp建立tcp连接后一般长时间保持，从而减少了http反复进行tcp连接和断开的开销，所以反向代理中ajp是比http高效的。

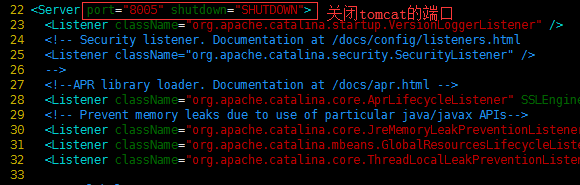
**8080**--默认的http监听端口。

8443--默认的https监听端口。默认未开启，如果要开启由于tomcat不自带证书所以除了取消注释之外，还要自己生成证书并在<Connector>中指定方可。

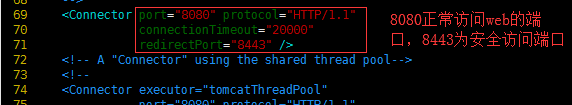
1. 、**更改server.xml主配置文件**

vim /usr/local/tomcat/conf/server.xml

**关闭tomcat的端口（可不改，默认）**



**Web访问端口（可不改，默认）**



**设置最大连接数：**

<Connector   port="8080"     
      maxThreads="150"       
      minSpareThreads="25"       
      maxSpareThreads="75"     
      acceptCount="100"     
      />   #可设置最大连接数   
maxThreads="150"     表示最多同时处理150个连接     
minSpareThreads="25"     表示即使没有人使用也开这么多空线程等待     
maxSpareThreads="75"     表示如果最多可以空75个线程，例如某时刻有80人访问，之后没有人访问了，则tomcat不会保留80个空线程，而是关闭5个空的。     
acceptCount="100"   当同时连接的人数达到maxThreads时，还可以接收排队的连接，超过这个连接的则直接返回拒绝连接。

**AJP端口（可不改，默认）**

IMG_256

148 <Host name="localhost" appBase="webapps"

149 unpackWARs="true" autoDeploy="true">

150

Name为本机网页的域名（需要在/etc/hosts文件做域名解析或者在/etc/resolv.conf写DNS服务器）

appBase为网页或者应用程序war包的目录（默认webapps目录）

unpackWARs为是否开启war包解压功能，默认开启

IMG_256

更改网站路径：

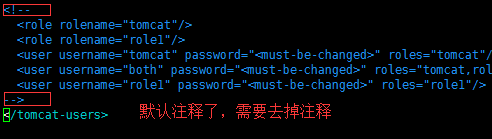
<Contest path="/yxxzsTest" docBase="/home/hejw/tomcat\_test/code/yxxzd\_war\_exploded"

reloadable="false" />

1. **、更改web.xml文件**（web.xml文件称为部署描述文件（Deployment Descriptor,简称DD文件），该文件可以覆盖Servelt中的标识设置，可以不更改） vim /usr/local/tomcat/conf/web.xml

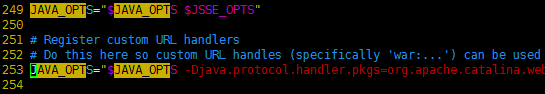


1. **、更改tomcat-users.xml**（设置tomcat管理账户）和context.xml（两个文件可不改） tomcat-users.xml文件



**4）、增加tomcat最大使用内存**

tomcat默认可以使用的内存为128MB，在较大型的应用项目中，这点内存是不够的，需要调大。文件目录：/usr/local/tomcat/bin/catalina.sh



加上下面这句：   
JAVA\_OPTS='$JAVA\_OPTS -server -Xmx800m -Xms512m -XX:MaxNewSize=256m -XX:MaxPermSize=256m -Djava.awt.headless=true'

#ms是最小内存，mx是最大内存。这里设置最小内存为512M,最大内存为 800M。$JAVA\_OPTS是保留先前设置。

1. **tomcat管理**

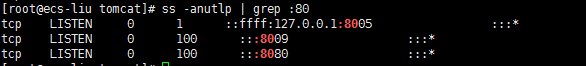
版本查看：/usr/local/tomcat/bin/versio.sh



/usr/local/tomcat/bin/startup.sh

/usr/local/tomcat/bin/shutdown.sh #启动和关闭tomcat,启动成功后会有8005、8080、80093个端口，可通过如下命令查询:

Netstat(或者ss) -anutlp | grep :80



# 升级tomcat

1，停【旧】tomcat,/usr/local/tomcat/bin/shutdown.sh

2，备份【旧】tomcat文件夹，文件夹改个名即可:cp -r /usr/local/tomcat /opt/tomcatold

3，下载传【新】tomcat压缩文件：wget 网址

4，解压【新】tomcat压缩文件，放在之前tomcat同样路径下，改名文件夹为旧版tomcat名，并增加执行权限a+x：tar -xf /root/tomcat-8.5.43.tar.gz -C /usr/local/tomcat

chmod a+x /ust/local/tomcat/\*

（这样环境变量就不需要改了）

5，拷贝【旧】webapps下的程序包文件夹到【新】webapps下:cp -r /usr/local/tomcatold/webapps /usr/local/tomcat/webapps/

6，备份【新】server.xml，改名备份即可cp /usr/local/tomcat/conf/server.xml /usr/local/tomcat/conf/serverback.xml

7，拷贝【旧】server.xml到【新】tomcat相同目录下

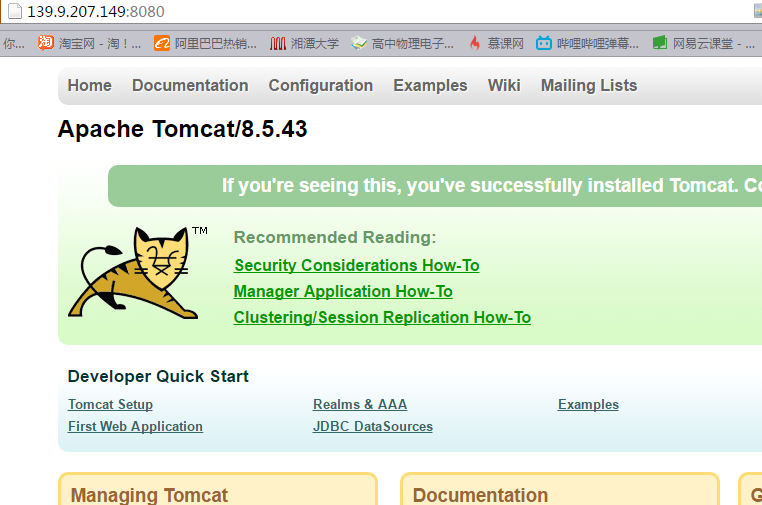
（6和7两步 有的可能其他配置文件做过修改的也类似处理）

cp /usr/local/tomcatold/conf/server.xml /usr/local/tomcat/conf/

1. 删除【新】webapps下的examples : rm /usr/local/tomcat/webapps/examples，若项目不再webapps目录下，webapps下的文件都可以删掉

# 删除系统自带的小猫和例子  
cd /usr/local/tomcat7/webapps  
rm -rf  host-manager  
rm -rf examples  
rm -rf manager/  
rm -rf docs

9，启动【新】tomcat:/usr/local/tomcat/bin/startup.sh



第二种升级方法：

备份 老版本tomcat下的bin 文件夹 和lib文件夹（一定要备份）

步骤六：  
复制 新版本的tomcat下的bin目录下的文件夹到 老版本bin目录下  
复制 新版本的tomcat下的lib目录下的文件夹到 老版本lib目录下

##### 步骤七：

到进程管理器中找到tomcat的进程 开启服务  
查看tomcat下logs中的日志信息，观察日志中的信息， 上报版本是否是新版本的版本号。服务是否正常启动。

#### 注意

这种tomcat 的升级方式 优点是：  
不用担心自己在tomcat中部署的项目，  
tomcat的升级只涉及到两个文件夹  
一个是lib 一个是bin 只要将新版本的lib 和bin 复制到需要升级的lib 和bin下就可以了

**--制作人 刘兆臣**

**时间：2019.08.15**