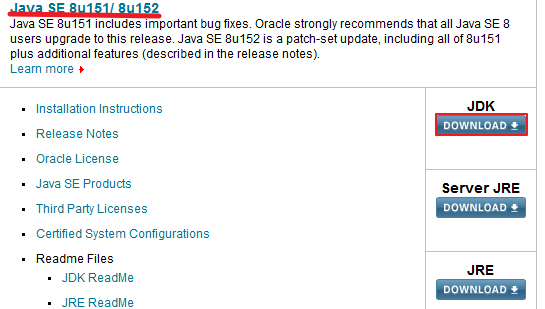
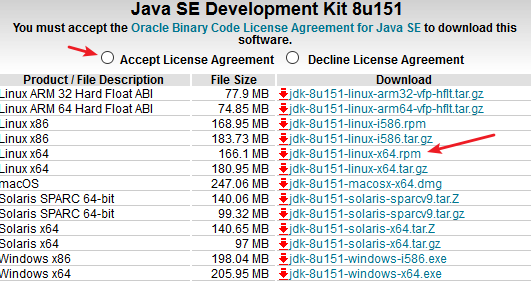
# Tomcat服务部署

## 软件清单

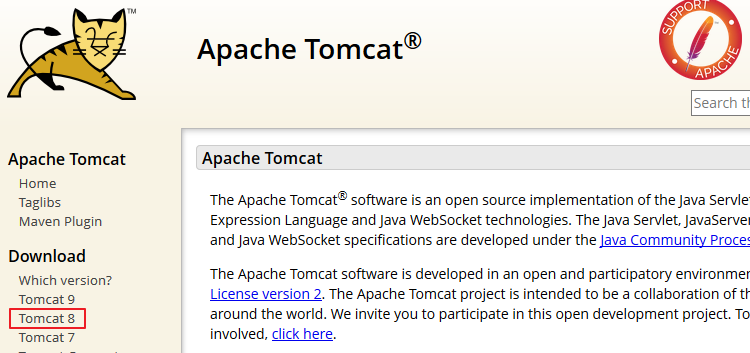
软件清单一：JDK官网下载地址：

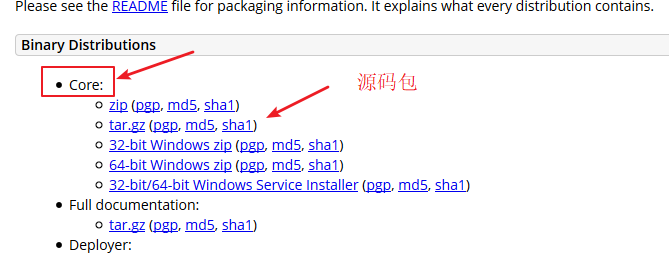
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>





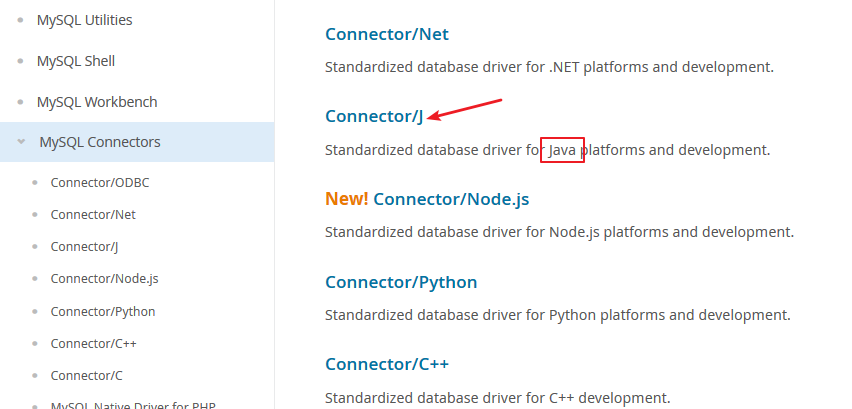
软件清单二：Tomcat官网下载地址<http://tomcat.apache.org/>

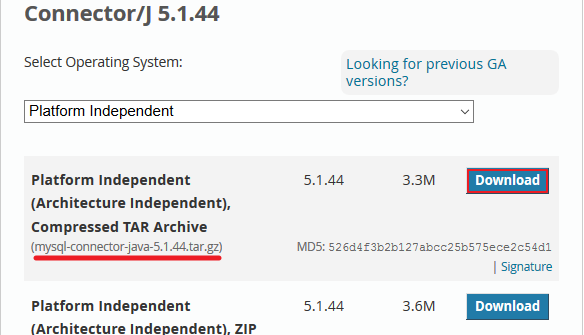


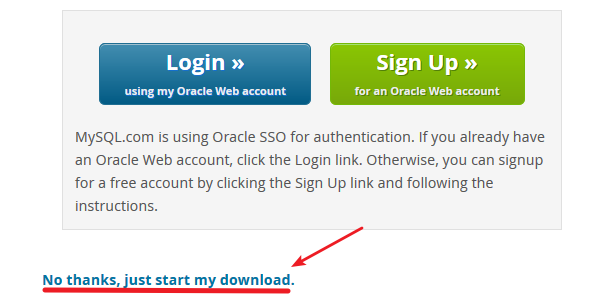


软件清单三：中间件Mysql-Connector-Java 用于连接MYSQL 和 JAVA

<http://dev.mysql.com/downloads/connector>







软件清单四：Tomcat-native

Tomcat Native 这个项目让Tomcat使用使用Apache的apr 包处理文件和网络IO提高性能，这个软件不用下载，在tomcat的bin目录下已经有了，可以使tomcat处理静态的性能略逊于apache

软件清单五：MYSQL

官网地址：<http://dev.mysql.com/downloads/mysql/>

因为Mysql源码包安装时间太长，这里为了简单起见，使用yum安装

## 安装环境

在安装之前查看是否有java相关的软件包

[root@demon ~]# rpm -qa | grep java

java-1.6.0-openjdk-1.6.0.0-1.66.1.13.0.el6.x86\_64

java-1.6.0-openjdk-devel-1.6.0.0-1.66.1.13.0.el6.x86\_64

libvirt-java-0.4.9-1.el6.noarch

java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.x86\_64

tzdata-java-2013g-1.el6.noarch

java-1.7.0-openjdk-devel-1.7.0.45-2.4.3.3.el6.x86\_64

libvirt-java-devel-0.4.9-1.el6.noarch

把这些卸载了，装oracle的jdk，卸载命令如下：

[root@demon ~]# rpm -e --nodeps java-1.6.0-openjdk-1.6.0.0-1.66.1.13.0.el6.x86\_64

卸载干净后，安装刚刚下载的jdk

[root@demon ~]# rpm -ivh jdk-8u151-linux-x64.rpm

Preparing... ########################################### [100%]

1:jdk1.8 ########################################### [100%]

Unpacking JAR files...

tools.jar...

plugin.jar...

javaws.jar...

deploy.jar...

rt.jar...

jsse.jar...

charsets.jar...

localedata.jar...

查看版本

[root@demon ~]# java -version

java version "1.8.0\_151"

Java(TM) SE Runtime Environment (build 1.8.0\_151-b12)

Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)

查看java相关文件存储位置

[root@demon ~]# rpm -qpl jdk-8u151-linux-x64.rpm | head -5

/usr

/usr/java

/usr/java/jdk1.8.0\_151

/usr/java/jdk1.8.0\_151/.java

/usr/java/jdk1.8.0\_151/.java/.systemPrefs

可以看到java文件都被存放到 /usr/java目录下面

若系统之前装了旧版jdk，如jdk-1.7.x，可以直接rpm安装1.8.x的jdk

但还需要修改/etc/profile文件的环境变量，如果没有装过旧版jdk可以直接跳过此步

export JAVA\_HOME = /usr/java/jdk1.8.0\_151/

export JAVA\_BIN = /usr/java/jdk1.8.0\_151/bin

export PATH = ${JAVA\_HOME}bin:$PATH

export CLASSPATH=.:${JAVA\_HOME}/lib/dt.jar:${JAVA\_HOME}/lib/tools.jar

[root@demon ~]# source /etc/profile

[root@demon ~]# java -version

java version "1.8.0\_151"

Java(TM) SE Runtime Environment (build 1.8.0\_151-b12)

Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)

## 安装tomcat

[root@demon ~]# tar xf apache-tomcat-8.5.23.tar.gz -C /usr/local/src/

[root@demon ~]# ll /usr/local/src/apache-tomcat-8.5.23/

drwxr-x--- 2 root root 4096 Nov 15 12:48 bin

drwx------ 2 root root 4096 Sep 28 18:31 conf

drwxr-x--- 2 root root 4096 Nov 15 12:48 lib

-rw-r----- 1 root root 57092 Sep 28 18:31 LICENSE

drwxr-x--- 2 root root 4096 Sep 28 18:30 logs

-rw-r----- 1 root root 1723 Sep 28 18:31 NOTICE

-rw-r----- 1 root root 7064 Sep 28 18:31 RELEASE-NOTES

-rw-r----- 1 root root 15946 Sep 28 18:31 RUNNING.txt

drwxr-x--- 2 root root 4096 Nov 15 12:48 temp

drwxr-x--- 7 root root 4096 Sep 28 18:30 webapps

drwxr-x--- 2 root root 4096 Sep 28 18:30 work

bin存放执行文件，conf配置文件，logs日志文件，webapps网站根目录

[root@demon ~]# ll /usr/local/src/apache-tomcat-8.5.23/bin/

-rw-r----- 1 root root 34813 Sep 28 18:30 bootstrap.jar

-rw-r----- 1 root root 14505 Sep 28 18:30 catalina.bat

-rwxr-x--- 1 root root 21793 Sep 28 18:30 catalina.sh

-rw-r----- 1 root root 1664 Sep 28 18:31 catalina-tasks.xml

-rw-r----- 1 root root 24283 Sep 28 18:30 commons-daemon.jar

-rw-r----- 1 root root 204944 Sep 28 18:30 commons-daemon-native.tar.gz

-rw-r----- 1 root root 2040 Sep 28 18:30 configtest.bat

-rwxr-x--- 1 root root 1922 Sep 28 18:30 configtest.sh

在bin目录下面的脚本中既有windows的bat 也有linux的shell，该包支持跨平台

这种包解压后不需要编译，称为二进制的源码包。

为了方便日后使用，更换一下tomcat的目录名和位置

[root@demon ~]# mv /usr/local/src/apache-tomcat-8.5.23/bin/ /usr/local/tomcat

[root@demon ~]# vim /etc/init.d/tomcat

#!/bin/bash

# Tomcat init script for linux

# chkconfig: 2345 96 14

# description: The Apache Tomcat Server/JSP container

JAVA\_OPTS='-server -Xms64m -Xmx128m'

JAVA\_HOME=/usr/java/jdk1.8.0\_151

CATALINA\_HOME=/usr/local/tomcat

$CATALINA\_HOME/bin/catalina.sh $\*

[root@demon ~]# chmod +x /etc/init.d/tomcat

[root@demon ~]# /etc/init.d/tomcat start

Using CATALINA\_BASE: /usr/local/tomcat

Using CATALINA\_HOME: /usr/local/tomcat

Using CATALINA\_TMPDIR: /usr/local/tomcat/temp

Using JRE\_HOME: /usr

Using CLASSPATH: /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar

Tomcat started.

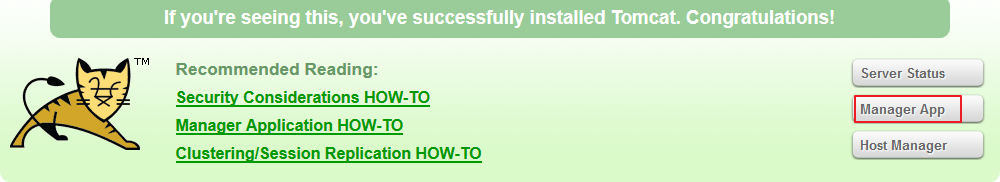
也可以不编写该脚本，直接使用tomcat/bin下的 startup.sh 和 shutdown.sh

[root@demon ~]# netstat -nlpt | grep 8080

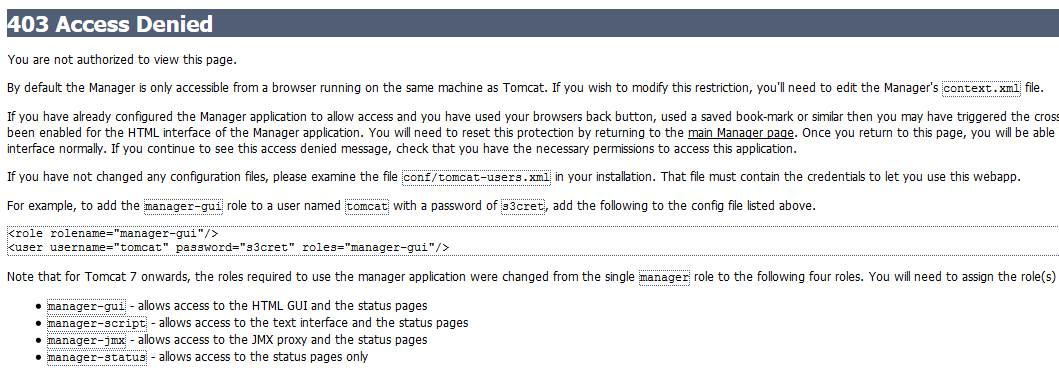
tcp 0 0 :::8080 :::\* LISTEN 3911/java

使用宿主机的浏览器访问8080

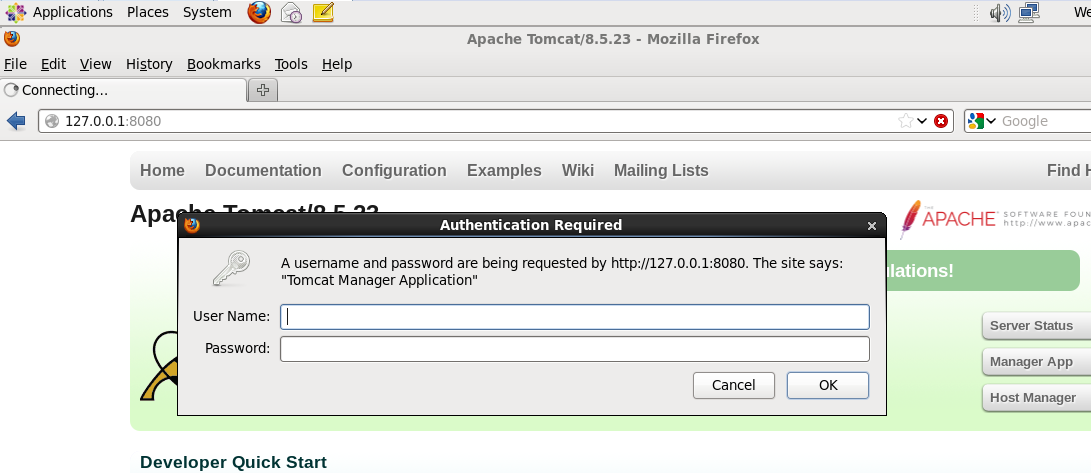




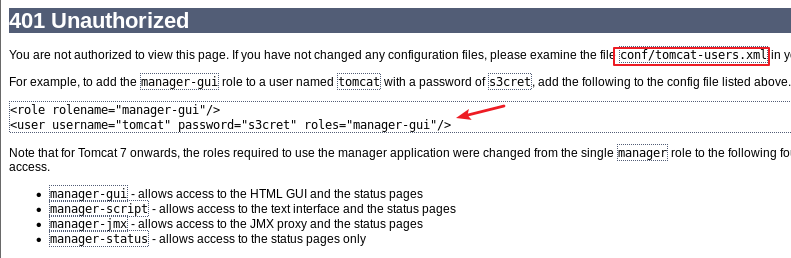
发现403访问拒绝



这是因为管理界面不允许远程访问，只能在搭建tomcat的localhost来访问它如下在Linux里访问它



这里需要输入账号密码，但是我们没有配置，点击取消显示提示界面

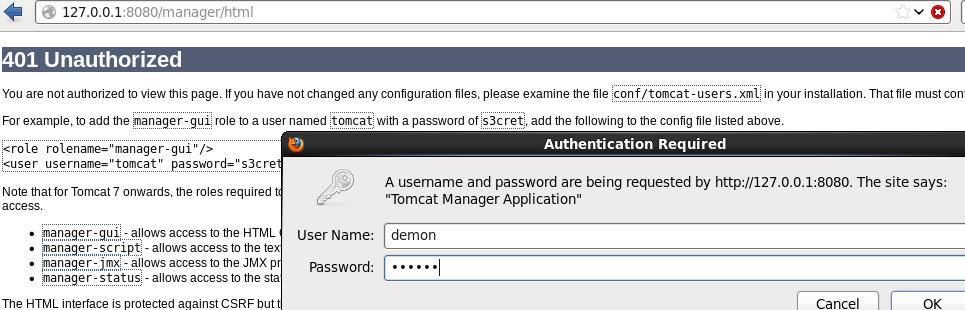


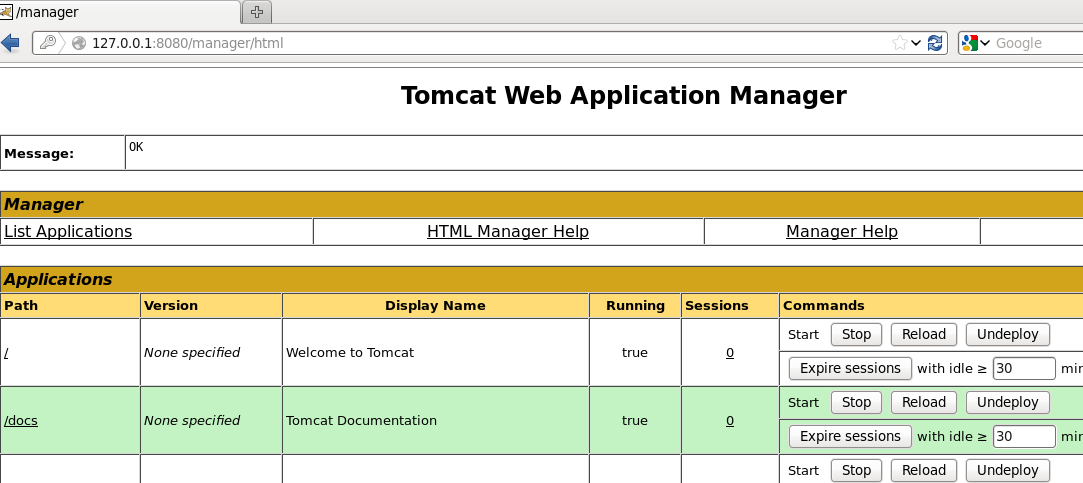
然后根据提示对 /usr/local/tomcat/conf/tomcat-users.xml配置文件进行修改，在<tomcat-users>标签里添加两行

<role rolename="manager-gui"/>

<user username="demon" password="123456" roles="manager-gui"/>

重启服务后继续使用linux的浏览器访问localhost:8080，输入如上创建的账号密码





然后就可以进入web应用管理界面了

## 虚拟主机

创建虚拟主机，编辑配置文件conf/server.xml，在<engine>标签内添加如下两个标签块

</Host>

<Host name="demon.cn" appBase="/var/www/html">

<Context path="" docBase="/var/www/html/web1" />

</Host>

<Host name="demon.com" appBase="/var/www/html">

<Context path="hello" docBase="/var/www/html/web2" />

</Host>

name表示虚拟主机名，不同的域名对应不同的站点

appBase表示web应用存放的根目录

path表示虚拟路径，如path=”hello”，那么访问它时就是 <http://demon.com:8080/hello>

docBase表示该虚拟主机的家目录

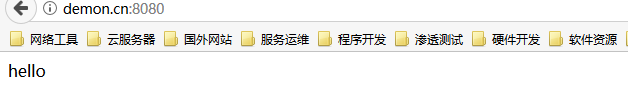
创建对应的文件于填写的web根目录下

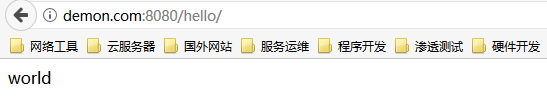
[root@demon conf]# mkdir -p /var/www/html/web1 /var/www/html/web2

[root@demon conf]# echo hello > /var/www/html/web1/index.html

[root@demon conf]# echo world > /var/www/html/web2/index.html

重启服务，再宿主机上访问测试.





## Tomcat优化

Tomcat优化包Tomcat-native ，安装相关的依赖包

[root@demon conf]# yum update

[root@demon conf]# yum -y install apr-devel gcc gcc-c++ openssl-devel openssl

之前说过native插件是Tomcat软件自带的，存放于 /usr/local/tomcat/bin下面

[root@demon bin]# cd /usr/local/tomcat/bin/

[root@demon bin]# tar xf tomcat-native.tar.gz -C /usr/local/src/

[root@demon native]# cd /usr/local/src/tomcat-native-1.2.14-src/native/

然后参考目录下的BUILDING文件进行编译安装

1. Prerequisites

Install OpenSSL version 1.0.2 or higher

Install APR version 1.4.0 or higher.

Download and expand the source package or use an svn checkout

> cd native

2. Configure build environment

Note: This step is only required if you are building from an svn checkout. It

is not required when building from a source package.

> sh buildconf --with-apr=apr\_source\_location.

3. Build

> configure --with-apr=apr\_install\_location --with-ssl=openssl\_install\_location

> make

This should produce a file named libtcnative-1.so

Note: To build without SSL support use:

> configure --disable-openssl --with-apr=apr\_install\_location

[root@demon native]# ./configure --with-apr=/usr/ --with-java-home=/usr/java/jdk1.8.0\_151/ --with-ssl

做到这里，demon是失败的，原因是apr版本太低，无法配置，先放过。

## Tomcat与mysql连接

[root@demon native]# yum -y install mysql-server

[root@demon native]# service mysqld start

[root@demon native]# mysql\_secure\_installation

[root@demon native]# mysql -uroot -p123456

mysql> create database tomcat;

mysql> use tomcat;

mysql> create table users(id int,name varchar(128));

mysql> insert into users value(1,'Come on'),(2,'Go Go!!');

mysql> grant all on tomcat.\* to tomcat@'localhost' identified by '123456';

mysql> flush privileges;

Ctrl+C

[root@demon native]# mysql -utomcat -p123456

[root@demon ~]# tar xf mysql-connector-java-5.1.44.tar.gz -C /usr/local/src/

[root@demon mysql-connector-java-5.1.44]# cp mysql-connector-java-5.1.44-bin.jar /usr/local/tomcat/lib/

[root@demon ~]# /etc/init.d/tomcat stop && /etc/init.d/tomcat start

[root@demon ~]# cd /usr/local/tomcat/webapps/ROOT/

[root@demon ROOT]# vim mysql.jsp

<%@ page import="java.sql.\*"%>

<html>

<body>

<%

Class.forName("org.gjt.mm.mysql.Driver").newInstance();

String url = "jdbc:mysql://localhost/tomcat?user=tomcat&password=123456&useUnicode=true&characterEncoding=utf-8";

Connection conn = DriverManager.getConnection(url);

Statement stmt = conn.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,ResultSet.CONCUR\_UPDATABLE);

String sql = "select \* from users";

ResultSet rs = stmt.executeQuery(sql);

while(rs.next()){%>

step:<%=rs.getString(1)%>

context:<%=rs.getString(2)%><br><br>

<%}%>

<%out.print("Jsp connect Mysql is OK!!!");%>

<%rs.close();

stmt.close();

conn.close();

%>

</body>

</html>

