**Tomcat**

**简介**

Tomcat是java编写的软件

Tomcat是运行java代码的web服务器

Java是一种跨平台、面向对象的程序设计语言，

Java体系

1. java SE(标准版) 2.java EE(企业版) 3.java ME(移动版)

JDK(java-developmen-kit)是针对java开发者推出的java语言软件开发工具包

包括java运行环境

Java工具（编译、派错、打包等工具）

Java基础的类库

JRE是JDK的子集，可单独下载到

Open-JDK

常见的java servlet容器

IBM websphere

Oracle weblogic

Apache tomcat

Redhat jboss

红帽的光盘自带JDK

**部署Tomcat**

1.使用RPM安装JDK环境

web1：yum –y install java-1.8.0-openjdk                 //安装JDK

yum -y install java-1.8.0-openjdk-headless        //安装JDK

java -version                                     //查看JAVA版本

2.安装Tomcat

Web1：tar -xf apache-tomcat-8.0.30.tar.gz

mv apache-tomcat-8.0.30 /usr/local/tomcat

ls /usr/local/tomcat

bin/                              //主程序目录

lib/                       //库文件目录

logs                             //日志目录

temp/                          //临时目录

work/                           //自动编译目录jsp代码转换servlet

conf/                            //配置文件目录

webapps/                      //页面目录

启动服务：运行脚本/usr/local/tomcat/bin/startup.sh

检查启动情况：ss -ntulp | grep java

8080、8005、8009 应该启动了

8005端口启动非常慢，可用使用下面的命令用urandom替换random（非必须操作）

由取tomcat启动需要在内存调用大量的随机码，需要内存数据足够大，否则启动非常慢

无穷尽的取随即数/dev/random、/dev/urandom

mv /dev/random /dev/random.bak

ln -s /dev/urandom /dev/random

Tomcat的根文件夹：/usr/local/tomcat/webapps/ROOT/

Tomcat的配置文件：/usr/local/tomcat/conf/server.xml

**配置两个虚拟主机**

**改配置文件：**vim/usr/local/tomcat/conf/server.xml

Server>

<Service>

<Connector port=8080 /> //开启的端口

<Connector port=8009 /> //同上

<Engine name="Catalina" defaultHost="localhost"> //Engine默认的网

页

<Host name="www.a.com" appBase="a" unpackWARS="true"

autoDeploy="true"> //name虚拟机的域名 appBase虚拟机的根目录

</Host>

<Host name="www.b.com" appBase="b" unpackWARS="true"

autoDeploy="true">

</Host>

<Host name="www.b.com" appBase="b" unpackWARS="true"

autoDeploy="true">

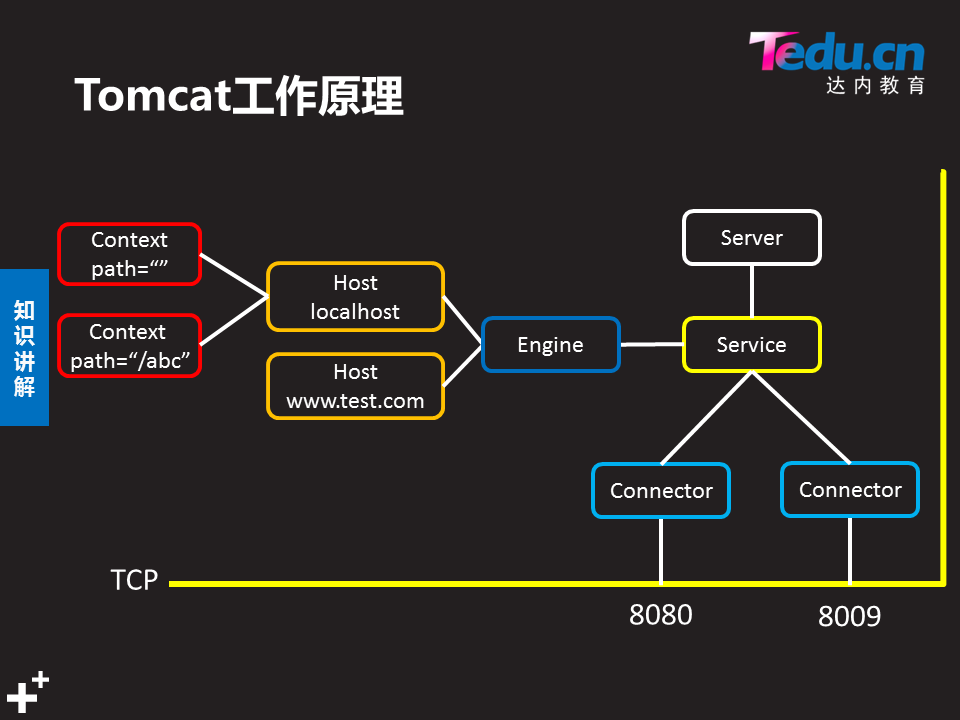
<Context path="" docBase="base" reloadable="true"/>

</Host> //docBase修改网站首页路径，默认为ROOT

</Server> appBase表示网站的根目录

</Server> //一个host表示一个虚拟主机,

所有的虚拟主机都必须放在Engine里面。Engine负责读取请求的头信息，然后Engine才把请求发给相应的虚拟主机。



Tomcat开启端口所有的请求都可以从这个端口机来,然后Engine

对请求头进行

设置虚拟主机根文件夹

mkdri /usr/local/tomcat/a/ROOT/

mkdri /usr/local/tomcat/a/ROOT/

Echo “I AM A” > /usr/local/tomcat/a/ROOT/index.html

Echo “I AM B” > /usr/local/tomcat/b/ROOT/index.html

重启tomcat，并检查端口启动情况

/usr/local/tomcat/bin/shutdown.sh

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

ss -ntulp | grep java

**配置Tomcat支持SSL加密网站**

**生成密匙**

keytool -genkeypair -alias tomcat -keyalg RSA -keystore /usr/local/tomcat/keystore

//提示输入密码为:123456

//-genkeypair     生成密钥对

//-alias tomcat     密钥别名

//-keyalg RSA     定义密钥算法为RSA算法

//-keystore         定义密钥文件存储在:/usr/local/tomcat/keystore

**修改文件，红色为添加项**

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

<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"

maxThreads="150" SSLEnabled="true"

scheme="https" secure="true"

keystoreFile="/usr/local/tomcat/keystore" keystorePass="123456"

clientAuth="false" sslProtocol="TLS" />

//keystoreFile指定加密文件位置 keystorePass指定加密密码

主日志：/usr/local/tomcat/logs/catalina.out

**为虚拟主机添加日志**

<Host name="www.b.com" appBase="b" unpackWARS="true"

autoDeploy="true">

<Valve className="org.apache.catalina.valves.AccessLogValve"

directory="logs" prefix="b\_log" suffix=".txt"

pattern="%h %l %u %t &quot;%r&quot; %s %b" />

</Host>

//红色为新添 加的日志行，directory为日志目录，相对于网站的根

prefix=为日志名 suffix为日志的扩展名

pattern=为日志的内容，均采用tomcat的内置变量

该内容可以在tomcat的默认的虚拟主机内找到模板

**代理服务器**

Varnish带缓存的代理服务器

一款高性能且开原的反向代理服务器

性能高、数度快、管理方便

总服务器与多台子缓存代理服务器之间的关系叫CND

CDN content delivery network

**源码安装**

yum -y install gcc readline-devel    //安装软件依赖包

yum -y install ncurses-devel         //安装软件依赖包

yum -y install pcre-devel             //安装软件依赖包

yum -y install python-docutils-0.11-0.2.20130715svn7687.el7.noarch.rpm

//安装软件依赖包

useradd -s /sbin/nologin varnish    //创建账户,安全，没有该用户，默认root用户启动

tar -xf varnish-5.2.1.tar.gz

cd varnish-5.2.1

./configure

make && make install

**复制启动脚本及配置文件**

cp /root/lnmp\_soft/varnish-5.2.1/etc/example.vcl /usr/local/etc/default.vcl

//软件在lnmp\_soft.tar.gz中相应的压缩包中

安装号后会提供varnishadm：守护进程可以接受来自客户端的HTTP请求，把他们传递到后 端服务器并缓存然后返回，以更好的满足后续同一地址的请求。

varnishlog：varnish日志

Varnishncsa：访问日志

varnishtest Varnishhist varnishd varnishstat varnishtop

**修改代理配置文件**

vim /usr/local/etc/default.vcl

backend default {

.host = "192.168.2.100";

.port = "80";

}

**启动服务**

varnishd -f /usr/local/etc/default.vcl

//varnishd 命令的其他选项说明如下：

//varnishd –s malloc,128M        定义varnish使用内存作为缓存，空间为128M

//varnishd –s file,/var/lib/varnish\_storage.bin,1G 定义varnish使用文件作为缓存