

GNU/Linux-Tomcat



Tomcat 配置与应用

GNU/Linux-Tomcat

A traditional Chinese ink wash painting is visible in the background. In the upper right corner, a dragon's head is depicted with sharp teeth and whiskers. In the lower right corner, a crane stands on a small patch of ground, facing left.

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、 Sun 和其他一些公司及个人共同开发而成。

GNU/Linux-Tomcat

Tomcat

由于 Sun 的参与和支持，最新的 Servlet 和 JSP 规范总是能在 Tomcat 中得到体现，Tomcat 5 支持最新的 Servlet 2.4 和 JSP 2.0 规范。因为 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。

GNU/Linux-Tomcat

A traditional Chinese ink wash painting is visible in the background. It features a bird, possibly a crane or heron, standing on a rocky outcrop. A bare, gnarled branch extends from the upper right corner towards the center of the frame. The painting is executed in black ink on a light, textured background.

Tomcat

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，在中小型系统和并发访问用户不是很多的情况下被普遍使用，是开发和调试 JSP 程序的首选。

GNU/Linux-Tomcat

Tomcat

可以这样认为，当在一台机器上配置好 Apache 服务器，可利用它响应 HTML（标准通用标记语言下的一个应用）页面的访问请求。实际上 Tomcat 部分是 Apache 服务器的扩展，但它是独立运行的，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。

GNU/Linux-Tomcat

Tomcat

当配置正确时，Apache 为 HTML 页面服务，而 Tomcat 实际上运行 JSP 页面和 Servlet。另外，Tomcat 和 IIS 等 Web 服务器一样，具有处理 HTML 页面的功能，另外它还是一个 Servlet 和 JSP 容器，独立的 Servlet 容器是 Tomcat 的默认模式。不过，Tomcat 处理静态 HTML 的能力不如 Apache 服务器。

GNU/Linux-Tomcat

Tomcat

同时 tomcat 属于 web 服务器的中间件



GNU/Linux-Tomcat

中间件

中间件是一种独立的系统软件或服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源。中间件位于客户机 / 服务器的操作系统之上，管理计算机资源和网络通讯。

是连接两个独立应用程序或独立系统的软件。相连接的系统，即使它们具有不同的接口，但通过中间件相互之间仍能交换信息。执行中间件的一个关键途径是信息传递。通过中间件，应用程序可以工作于多平台或 OS 环境。

GNU/Linux-Tomcat

中间件

中间件是一类连接软件组件和应用的计算机软件，它包括一组服务。以便于运行在一台或多台机器上的多个软件通过网络进行交互。该技术所提供的互操作性，推动了一致分布式体系架构的演进，该架构通常用于支持并简化那些复杂的分布式应用程序，它包括 web 服务器、事务监控器和消息队列软件。

GNU/Linux-Tomcat

A traditional Chinese ink wash painting is visible in the background. It features a bird, possibly a crane or heron, standing on a rocky outcrop in the lower right. A bare, gnarled branch extends from the upper right towards the center. The painting is executed in black ink on a light, textured background.

中间件

中间件 (middleware) 是基础软件的一大类，属于可复用软件的范畴。顾名思义，中间件处于操作系统软件与用户的应用软件的中间。

中间件 GNU/Linux-Tomcat

中间件在操作系统、网络和数据库之上，应用软件的下层，总的作用是为处于自己上层的应用软件提供运行与开发的环境，帮助用户灵活、高效地开发和集成复杂的应用软件。在众多关于中间件的定义中。

比较普遍被接受的是 IDC 表述的：中间件是一种独立的系统软件或服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源，中间件位于客户机服务器的操作系统之上，管理计算资源和网络通信。

GNU/Linux-Tomcat

A traditional Chinese ink wash painting of a rooster, positioned in the upper right corner of the slide. The rooster is depicted with expressive brushstrokes, showing its head, neck, and tail feathers. The background of the slide is a light, textured grey.

中间件

IDC 对中间件的定义表明，中间件是一类软件，而非一种软件；中间件不仅仅实现互连，还要实现应用之间的互操作；中间件是基于分布式处理的软件，最突出的特点是其网络通信功能。

GNU/Linux-Tomcat

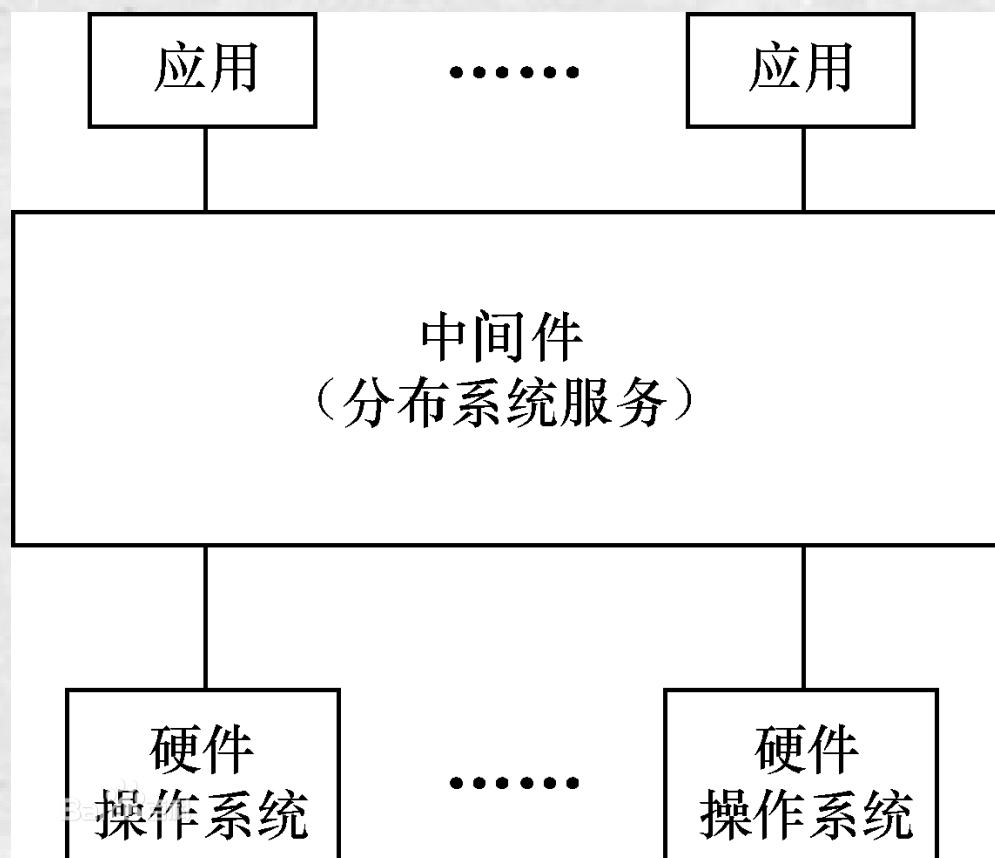


中间件

中间件是位于平台（硬件和操作系统）和应用之间的通用服务，如图 1 所示，这些服务具有标准的程序接口和协议。针对不同的操作系统和硬件平台，它们可以有符合接口和协议规范的多种实现。

GNU/Linux-Tomcat

中间件



GNU/Linux-Tomcat

中间件特点

中间件应具有如下一些特点：

1. 满足大量应用的需要；
2. 运行于多种硬件和 OS 平台；
3. 支持分布计算，提供跨网络、硬件和 OS 平台的透明性的应用或服务的交互；
4. 支持标准的协议；
5. 支持标准的接口。



GNU/Linux-Tomcat

安装 Tomcat

1. 安装 JDK

#wget

<http://192.168.188.2/softs/JDK/jdk-8u40-linux-x64.rpm>

#rpm -ivh jdk-8u40-linux-x64.rpm



GNU/Linux-Tomcat

安装 Tomcat

2. 配置工作环境

```
#vi /etc/profile
```

//* 在最后追加

```
export JAVA_HOME=/usr/java/default  
export PATH=$PATH:$JAVA_HOME/bin  
export CLASSPATH=.:$JAVA_HOME/jre/lib:  
$JAVA_HOME/lib:$JAVA_HOME/lib/tools.jar
```



GNU/Linux-Tomcat

安装 Tomcat

3. 应用工作环境

```
#source /etc/profile
```



GNU/Linux-Tomcat

安装 Tomcat

4. 测试JDK 环境

```
#vi day.java
```

```
import java.util.Calendar;
```

```
class day {
```

```
    public static void main(String[] args) {
```

```
        Calendar cal = Calendar.getInstance();
```

```
        int year = cal.get(Calendar.YEAR);
```

```
        int month = cal.get(Calendar.MONTH) + 1;
```



GNU/Linux-Tomcat

安装 Tomcat

4. 测试JDK 环境

```
int day = cal.get(Calendar.DATE);  
int hour = cal.get(Calendar.HOUR_OF_DAY);  
int minute = cal.get(Calendar.MINUTE);  
System.out.println(year + "/" + month + "/" +  
day + " " + hour + ":" + minute);  
}  
}
```

GNU/Linux-Tomcat

安装 Tomcat

4. 测试JDK 环境

```
#javac day.java
```

```
#jdava day ← 应能看到当前时间
```



GNU/Linux-Tomcat

安装 Tomcat

5. 下载 tomcat 软件包

#wget

<http://192.168.188.2/softs/Tomcat/apache-tomcat-8.0.30.tar.gz>



GNU/Linux-Tomcat

安装 Tomcat

5. 安装 tomcat 软件包

```
#tar xvfz apache-tomcat-8.0.30.tar.gz  
#mkdir -v /usr/tomcat8  
#mv apache-tomcat-8.0.30/* /usr/tomcat8/  
#useradd -M -d /usr/tomcat8 tomcat8  
#chown -R tomcat8. /usr/tomcat8
```



GNU/Linux-Tomcat



安装 Tomcat

6. 配置 tomcat 服务

#vi /usr/lib/systemd/system/tomcat.service



GNU/Linux-Tomcat

安装 Tomcat

6. 配置 tomcat 服务

[Unit]

Description=Apache Tomcat 8

After=network.target



GNU/Linux-Tomcat

安装 Tomcat

6. 配置 tomcat 服务

[Service]

Type=oneshot

ExecStart=/usr/tomcat8/bin/startup.sh

ExecStop=/usr/tomcat8/bin/shutdown.sh

RemainAfterExit=yes

User=tomcat8

Group=tomcat8



GNU/Linux-Tomcat

安装 Tomcat

6. 配置 tomcat 服务

[Install]

WantedBy=multi-user.target



GNU/Linux-Tomcat

安装 Tomcat

7. 启动 tomcat 服务

```
#systemctl start tomcat
```

```
#systemctl enable tomcat
```



GNU/Linux-Tomcat

安装 Tomcat

8. 测试 tomcat 服务

打开浏览器，输入：
http://tomcat_srv_ip:8080



GNU/Linux-Tomcat

测试 Tomcat

9. 测试 tomcat 服务是否可以开启 Java 支持

```
# mkdir /usr/tomcat8/webapps/ROOT/WEB-INF/classes
```

```
# chown tomcat8.
```

```
/usr/tomcat8/webapps/ROOT/WEB-INF/classes
```

GNU/Linux-Tomcat



测试 Tomcat

9. 测试 tomcat 服务是否可以开启 Java 支持

```
# cd /usr/tomcat8/webapps/ROOT/WEB-INF/classes
```



GNU/Linux-Tomcat

测试 Tomcat

9. 测试 tomcat 服务是否可以开启 Java 支持

```
# vi daytime.java
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.Calendar;
```




GNU/Linux-Tomcat



测试 Tomcat

9. 测试 tomcat 服务是否可以开启 Java 支持

```
public class daytime extends HttpServlet {  
    public void doGet(HttpServletRequest request  
        ,HttpServletResponse response)
```



GNU/Linux-Tomcat

测试 Tomcat

9. 测试 tomcat 服务是否可以开启 Java 支持

```
throws IOException, ServletException{  
    response.setContentType("text/html");  
    PrintWriter out = response.getWriter();  
    Calendar cal = Calendar.getInstance();
```

GNU/Linux-Tomcat

测试 Tomcat

9. 测试 tomcat 服务是否可以开启 Java 支持

```
out.println("<html>\n<head>\n<title>DayTime</ti  
tle>\n</head>\n<body>");
```

```
    out.println("<div style=\"font-size: 40px; text-  
align: center; font-weight: bold\">");
```

GNU/Linux-Tomcat

测试 Tomcat

9. 测试 tomcat 服务是否可以开启 Java 支持

```
        cal.get(Calendar.DATE) + " " +  
cal.get(Calendar.HOUR_OF_DAY) + ":" +  
cal.get(Calendar.MINUTE));  
        out.println("</div>\n</body>\n</html>");  
    }  
}
```

GNU/Linux-Tomcat

测试 Tomcat

9. 测试 tomcat 服务是否可以开启 Java 支持

```
# javac -classpath /usr/tomcat8/lib/servlet-api.jar  
daytime.java
```

```
# vi /usr/tomcat8/webapps/ROOT/WEB-INF/web.xml
```



GNU/Linux-Tomcat

测试 Tomcat

9. 测试 tomcat 服务是否可以开启 Java 支持
/* 在 <web-app> 至 </web-app> 之间添加

```
<servlet>
```

```
  <servlet-name>daytime</servlet-name>
```

```
  <servlet-class>daytime</servlet-class>
```

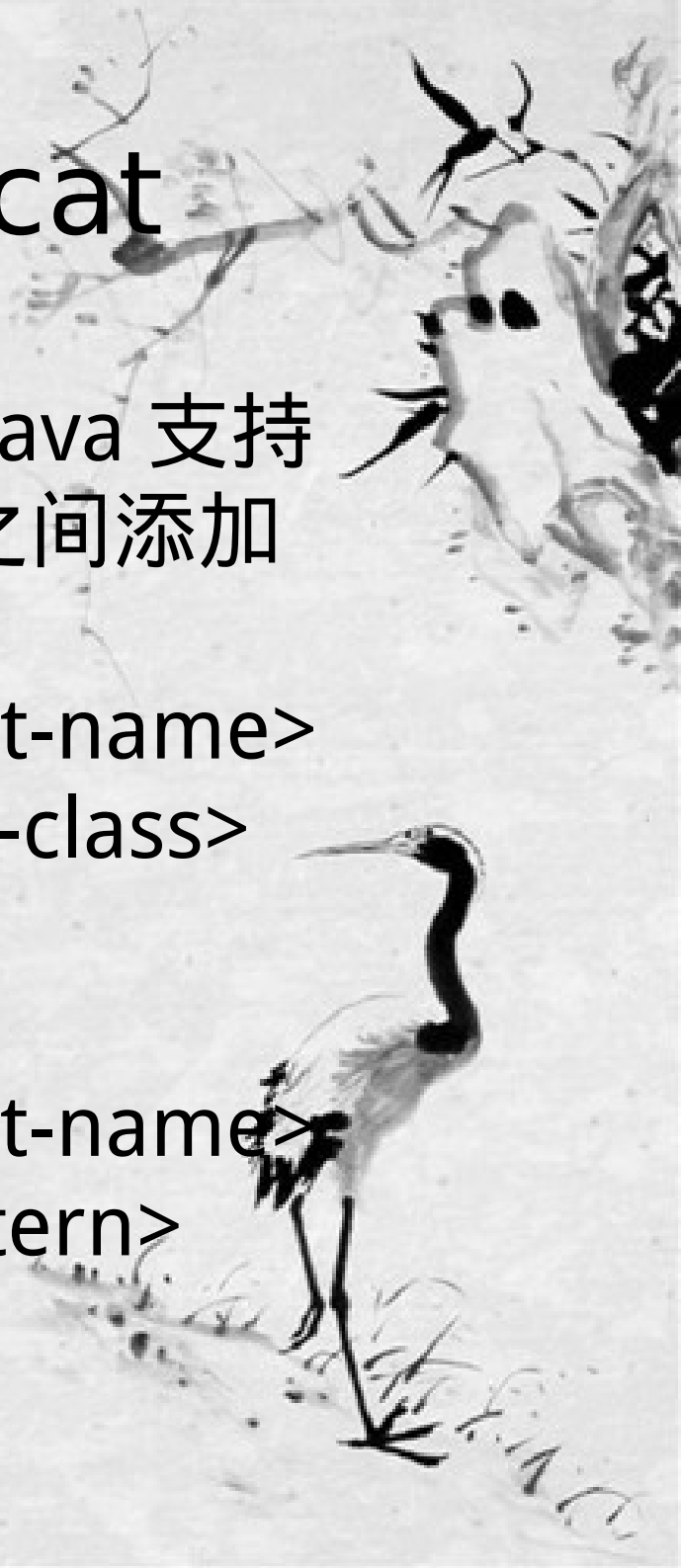
```
</servlet>
```

```
<servlet-mapping>
```

```
  <servlet-name>daytime</servlet-name>
```

```
  <url-pattern>/daytime</url-pattern>
```

```
</servlet-mapping>
```



GNU/Linux-Tomcat



测试 Tomcat

10. 访问测试 tomcat 服务是否可以开启 Java 支持

打开浏览器，输入

`http://tomcat_srv_ip:8080/daytime`



GNU/Linux-Tomcat

隐藏 Tomcat 的访问端口 8080

1. 安装 apache 并启动

2. 编写 apache 的 proxy_ajp.conf

```
# vi /etc/httpd/conf.d/proxy_ajp.conf
```

/* 当访问 http://httpd_srv_ip/tomcat8 时，准备
ajp 协议。 ajp 协议默认监听 8009

```
ProxyPass /tomcat8/ ajp://192.168.188.111:8009/
```

```
#systemctl restart httpd
```


GNU/Linux-Tomcat

隐藏 Tomcat 的访问端口 8080

/*AJP 是定向包协议。因为性能原因，使用二进制格式来传输可读性文本。WEB 服务器通过 TCP 连接和 SERVLET 容器连接。为了减少进程生成 socket 的花费，WEB 服务器和 SERVLET 容器之间尝试保持持久性的 TCP 连接，对多个请求 / 回复循环重用一个连接。一旦连接分配给一个特定的请求，在请求处理循环结束之前不会再分配。

GNU/Linux-Tomcat

隐藏 Tomcat 的访问端口 8080

3. 测试

打开浏览器，输入：

http://apache_srv_ip/tomcat/

