

# Linux高级运维

**NSD OPERATION**

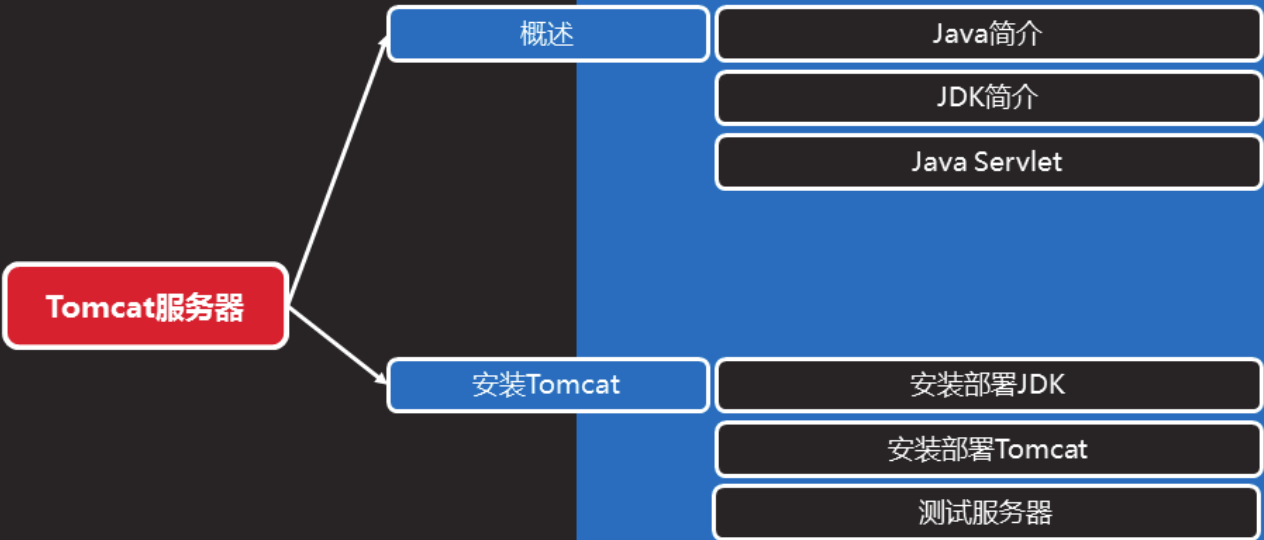
**DAY05**

# 内容

|    |               |              |
|----|---------------|--------------|
| 上午 | 09:00 ~ 09:30 | 作业讲解和回顾      |
|    | 09:30 ~ 10:20 | Tomcat服务器    |
|    | 10:30 ~ 11:20 | Tomcat应用案例   |
|    | 11:30 ~ 12:20 |              |
| 下午 | 14:00 ~ 14:50 | Varnish代理服务器 |
|    | 15:00 ~ 15:50 |              |
|    | 16:00 ~ 16:50 |              |
|    | 17:00 ~ 17:30 | 总结和答疑        |



## Tomcat服务器



# 概述



## Java简介

- Java
  - java是一种跨平台的、面向对象的程序设计语言，Java技术具有卓越的通用性、高效性、**平台移植性**和安全性。
- Java体系
  - Java SE（标准版）
  - Java EE（企业版）
  - Java ME（移动版）



# JDK简介

知识讲解

- JDK(Java Development Kit)是Sun针对Java开发者推出的**Java语言的软件开发工具包**
- JDK是整个Java的核心
  - 包括了Java运行环境
  - Java工具（如编译、排错、打包等工具）
  - Java基础的类库



## JDK简介（续1）

知识讲解

- JRE（Java Runtime Environment，Java运行环境），**JRE是JDK的子集**
- JRE包括
  - Java虚拟机（jvm）
  - Java核心类库和支持文件
  - 不包含开发工具(JDK)--编译器、调试器和其它工具



# Java Servlet

- Servlet是java扩展Web服务器功能的组件规范

## 知识讲解

- 常见Servlet容器
  - IBM websphere
  - Oracle weblogic
  - Apache tomcat
  - RedHat Jboss



## 安装Tomcat

# 安装部署JDK

- 安装jdk1.8

知识讲解

```
[root@svr5 ~]# yum -y install java-1.8.0-openjdk
[root@svr5 ~]# yum -y install java-1.8.0-openjdk-headless
```



# 安装部署Tomcat

- 安装Tomcat

知识讲解

```
[root@svr5 ~]# tar -xzf apache-tomcat-8.0.30.tar.gz
[root@svr5 ~]# mv apache-tomcat-8.0.30 /usr/local/tomcat
[root@svr5 ~]# 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
```



## 安装部署Tomcat ( 续1 )

- 启动Tomcat

```
[root@svr5 ~]# /usr/local/tomcat/bin/startup.sh
Using CATALINA_BASE: /usr/local/tomcat
Using CATALINA_HOME: /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME: /usr/java/default
Using CLASSPATH:
/usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
Tomcat started.
```

知识讲解



## 测试服务器

- 生成测试页面

```
[root@svr5 ~]# # vim /usr/local/tomcat/webapps/ROOT/test.jsp
<html>
<body>
<center>
Now time is: <%=new java.util.Date()%>
</center>
</body>
</html>
```

- 客户端测试

- firefox http://localhost:8080
- firefox http://localhost:8080/test.jsp

知识讲解



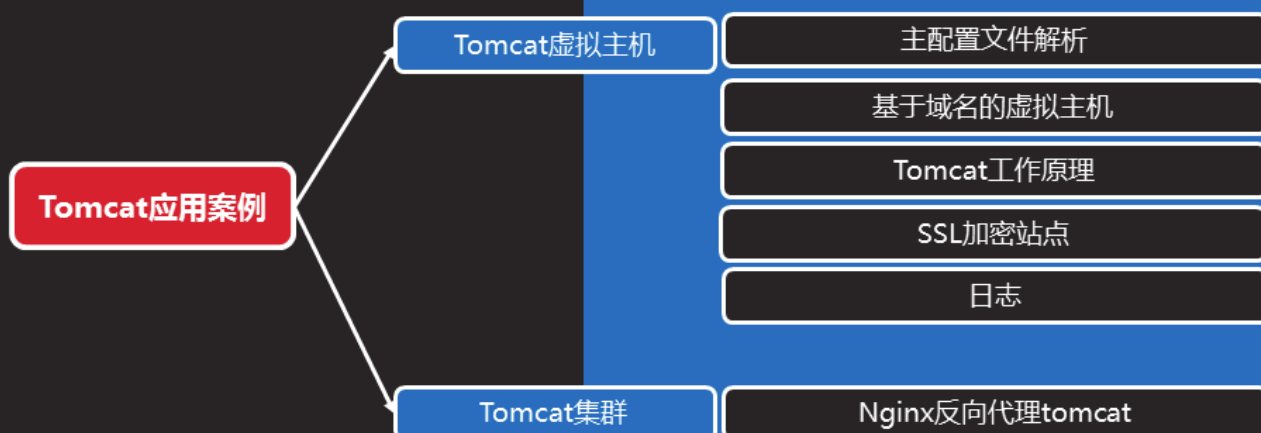
# 案例1：安装部署Tomcat服务器

课堂练习

- 安装部署JDK环境
- 安装部署Tomcat服务器
- 创建JSP测试页面，文件名为test.jsp，显示服务器当前时间



## Tomcat应用案例





# Tomcat虚拟主机

## 主配置文件解析

- server.xml配置文件框架

```
<?xml version='1.0' encoding='utf-8'?>
<Server port="8005" shutdown="SHUTDOWN">
  <Service name="Catalina">
    <Connector port="8080" protocol="HTTP/1.1"
      connectionTimeout="20000"
      redirectPort="8443" />
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
    <Engine name="Catalina" defaultHost="localhost">
      <Host name="localhost" appBase="webapps"
        unpackWARs="true" autoDeploy="true">
      </Host>
    </Engine>
  </Service>
</Server>
```

# 基于域名的虚拟主机

- 修改server.xml文档，添加host虚拟主机

```
#vim /usr/local/tomcat/conf/server.xml
... ..
<Host name="www.a.com" appBase="a" unpackWARS="true"
autoDeploy="true">
</Host>
<Host name="www.b.com" appBase="b" unpackWARS="true"
autoDeploy="true">
</Host>
#mkdir -p /usr/local/tomcat/{a,b}/ROOT
#echo "A" > /usr/local/tomcat/a/ROOT/index.html
#echo "B" > /usr/local/tomcat/b/ROOT/index.html
#/usr/local/tomcat/bin/shutdown.sh
#/usr/local/tomcat/bin/startup.sh
```

知识讲解



## 基于域名的虚拟主机（续1）

- 修改server.xml文档，给host添加context

```
#vim /usr/local/tomcat/conf/server.xml
... ..
<Host name="www.a.com" appBase="a" unpackWARS="true"
autoDeploy="true">
<Context path="" docBase="base" reloadable="true"/>
</Host>
<Host name="www.b.com" appBase="b" unpackWARS="true"
autoDeploy="true"> </Host>
#mkdir -p /usr/local/tomcat/a/base
#echo "base" > /usr/local/tomcat/a/base/index.html
#/usr/local/tomcat/bin/shutdown.sh
#/usr/local/tomcat/bin/startup.sh
```

知识讲解



//appBase定义基础目录，基础目录下可以有很多项目，默认项目ROOT  
//docBase定义首页路径，默认为ROOT

## 基于域名的虚拟主机（续2）

- 修改server.xml文档，给host添加context

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

```
... ..
```

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

```
<Context path="/test" docBase="/var/www/html/" />
```

```
</Host>
```

```
<Host name="www.b.com" appBase="b" unpackWARS="true"
autoDeploy="true"> </Host>
```

```
#echo "test" > /var/www/html/index.html
```

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

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

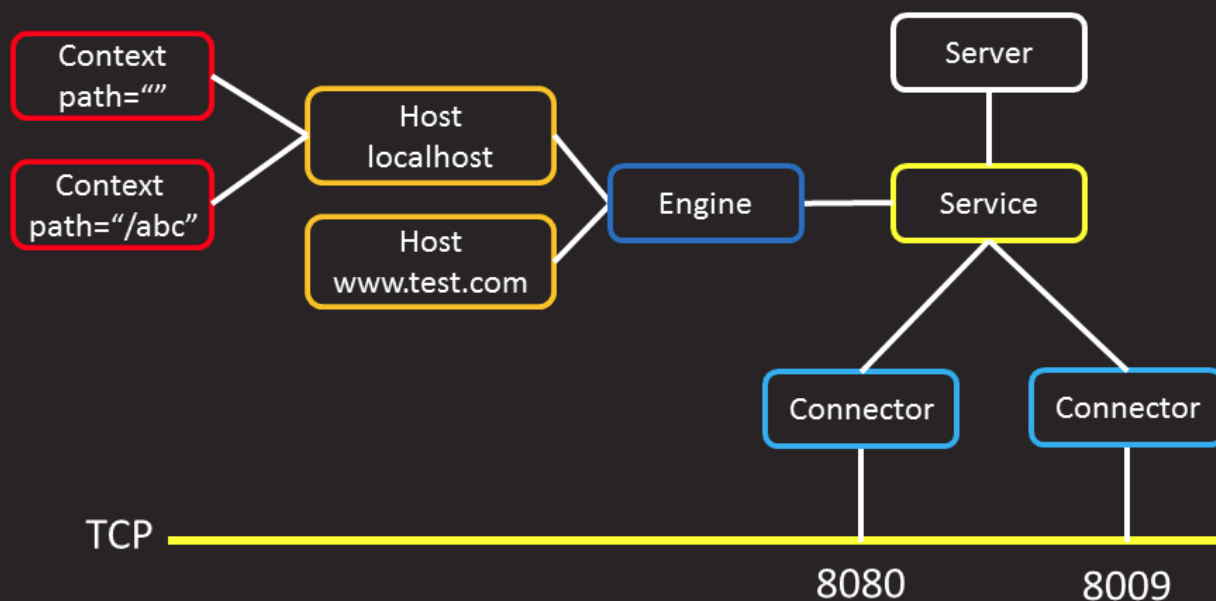
//path指定用户访问的URL,docBase指定页面存储的位置

验证：http://IP:8080/test/

知识讲解



## Tomcat工作原理



知识讲解



# SSL加密站点

知识讲解

- 生产私钥证书文件

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

- 修改server.xml配置文件

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

```
... ..
```

```
<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" />
```



## SSL加密站点（续1）

- 客户端测试

```
# firefox https://localhost:8443
```

```
# firefox https://www.a.com:8443
```

知识讲解



# 日志

- 虚拟主机创建独立日志文件

知识讲解

```
#vim /usr/local/tomcat/conf/server.xml
<Host name="www.a.com" appBase="a" unpackWARs="true"
autoDeploy="true">
  <Context path="" docBase="base" />
  <Valve className="org.apache.catalina.valves.AccessLogValve"
    prefix="wwwacom_access_log." suffix=".txt"
    pattern="common"/>
</Host>
```



## Tomcat集群

# Nginx反向代理tomcat

- 修改nginx配置

```
#vim /usr/local/nginx/conf/nginx.conf
http{
    upstream toms {
        server 192.168.2.100:8080;
        server 192.168.2.200:8080;
    }
    server {
        listen 80;
        server_name localhost;
        location / {
            proxy_pass http://toms;
        }
    }
}
```

知识讲解



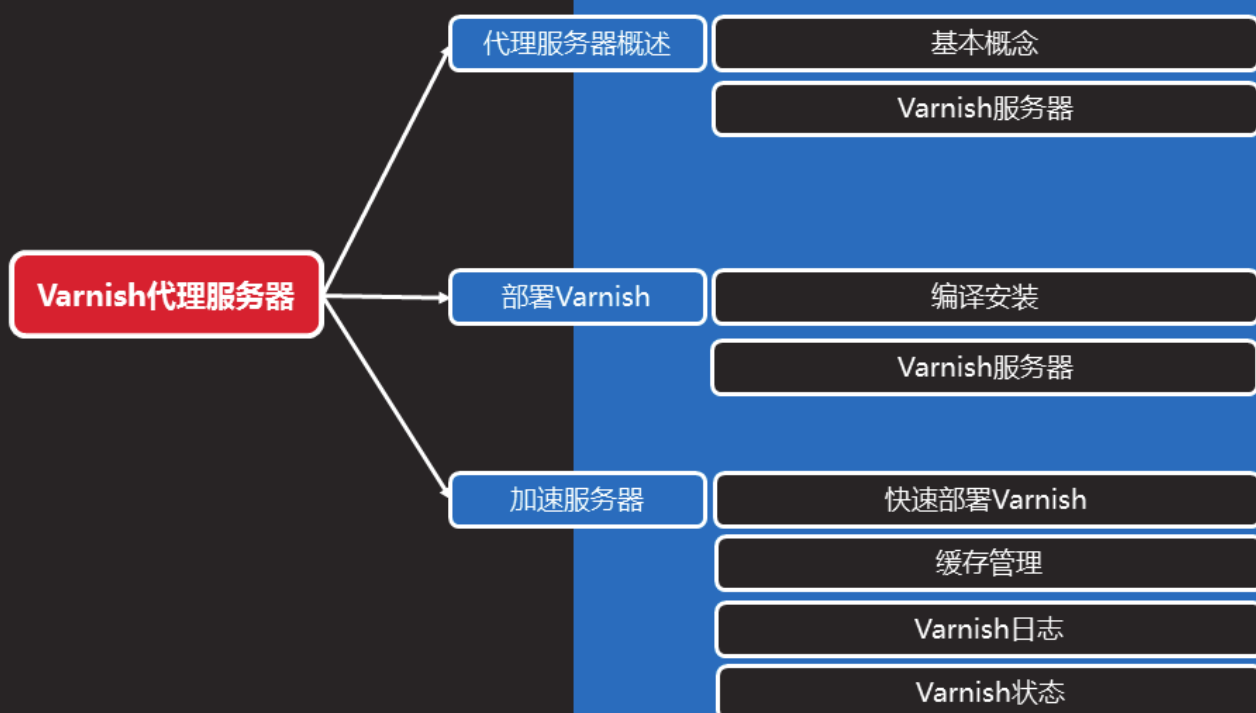
## 案例2：使用Tomcat部署虚拟主机

- 设置基于域名的虚拟主机，要求如下：
  - 域名：www.aa.com www.bb.com
- www.aa.com的根路径为/usr/local/tomcat/aa/ROOT
- www.bb.com的根路径为/usr/local/tomcat/bb/base
- 访问页面时支持SSL加密通讯
- 私钥、证书存储路径为/usr/local/tomcat/conf/cert
- 每个虚拟主机都拥有独立的访问日志文件
- 配置tomcat集群环境

课堂练习



# Varnish代理服务器



## 代理服务器概述

## 基本概念

知识讲解

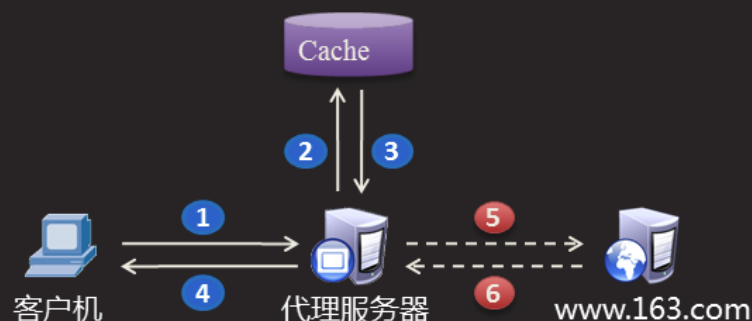
- Proxy , 代理人、经纪人、中介
- 代替用户处理事务
  - 能提供正常情况下不便或无法访问的资源
- 应用场景：
  - Web访问加速（正/反向）
  - IP伪装、“翻墙”



## 基本概念（续1）

知识讲解

- 代理服务器工作机制
  - Cache池的复用
  - 线路转移

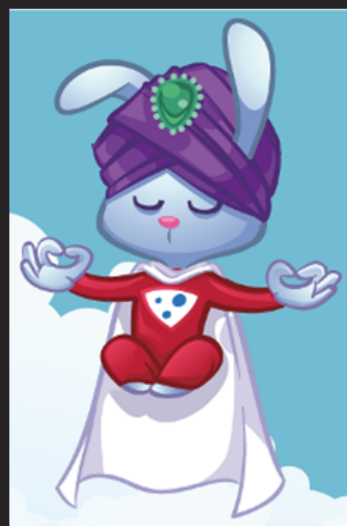




# Varnish服务器

- Varnish一款高性能且开源的反向代理服务器。
- Varnish具有性能高、速度更快、管理更方便等诸多优点。

知识讲解



## 部署Varnish

## 编译安装

- 编译安装

知识讲解

```
[root@svr5 ~]# yum -y install gcc readline-devel
[root@svr5 ~]# yum -y install ncurses-devel pcre-devel
[root@svr5 ~]# yum localinstall python-docutils-0.11-0.2.20130715svn7687.el7.noarch.rpm
[root@svr5 ~]# useradd -s /sbin/nologin varnish //创建账户
[root@svr5 ~]# tar -xf varnish-5.2.1.tar.gz
[root@svr5 ~]# cd varnish-5.2.1
[root@svr5 varnish-5.2.1]# ./configure
[root@svr5 varnish-5.2.1]# make && make install
[root@svr5 varnish-5.2.1]# cp etc/example.vcl /usr/local/etc/default.vcl
```



## Varnish服务器

- 服务器软件

知识讲解

```
[root@svr5 ~]# ls /usr/local/bin/
varnishadm varnishlog varnishstat varnishtop
varnishhist varnishncsa varnishtest
[root@svr5 ~]# ls /usr/local/sbin/
varnishd
```



# 加速服务器

## 快速部署Varnish

- 修改配置文件

```
[root@svr5 ~]# vim /usr/local/etc/default.vcl
```

```
backend default {
```

```
    .host = "192.168.2.100";           //后端服务器的IP地址
```

```
    .port = "80";
```

```
}
```

## 快速部署Varnish ( 续1 )

- 启动服务

```
[root@svr5 ~]# varnishd -f /usr/local/etc/default.vcl
```

知识讲解

- 其他选项

```
#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
```



## 缓存管理

- 清除缓存内容的命令格式：

```
varnishadm ban req.url ~ .*
```

知识讲解



# Varnish日志

- varnishlog [-w file] //共享内存的日志
- varnishncsa [-w file] //类Apache日志

知识讲解



# Varnish状态

- varnishstat
  - Client connections accepted:  
表示客户端成功发送连接总数量
  - Client requests received:  
客户端发送HTTP请求的总数
  - Cache hits:  
!重要：命中缓存的次数
  - Cache misses  
!重要：缓存非命中的个数。

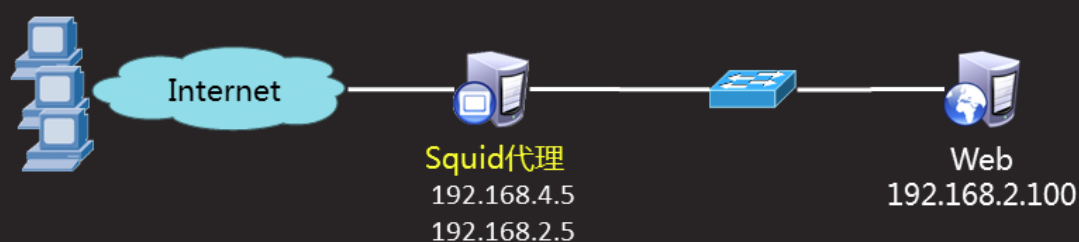
知识讲解



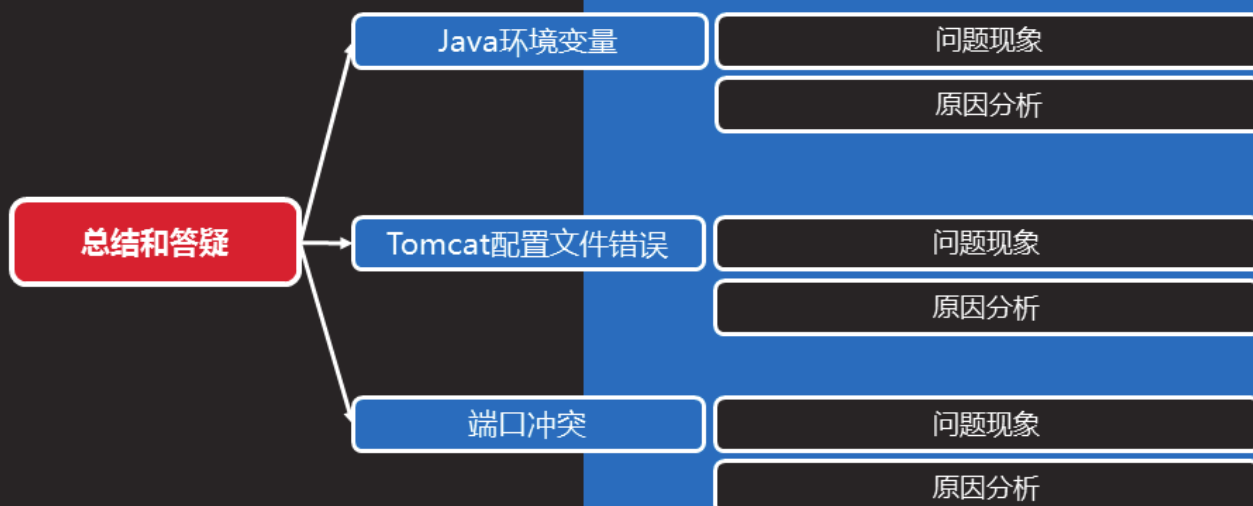
## 案例3：使用Varnish加速Web

课堂练习

- 使用Varnish加速后端Apache Web服务
- 客户端访问192.168.4.5时，varnish转发请求给源站Web
- 使用varnishadm管理缓存页面
- 使用varnishstat查看Varnish状态



## 总结和答疑



# Java环境变量

## 问题现象

- 故障错误信息

```
[root@svr5 ~]# /usr/local/tomcat/bin/catalina.sh start
```

```
Neither the JAVA_HOME nor the JRE_HOME environment variable is defined
```



## 原因分析

- 分析故障信息
  - Tomcat启动时无法找到java
- 分析故障原因
  - 未安装jdk
  - 或者安装jdk后没有设置正确的环境变量
  - 使用命令java -version查看版本

知识讲解



## Tomcat配置文件错误



## 问题现象

- 故障错误信息

```
[root@svr5 nginx-1.8.0]# vim /usr/local/tomcat/conf/server.xml
```

该文件语法严格，容易出错

知识讲解



## 原因分析

- 分析故障
  - 大小写错误，如<host> </Host>
  - 关键词不匹配，如<Host>无结束的</Host>
  - 位置错误，如将<Host>写到<Engine>的上面
  - 默认的localhost站点名称被修改
- 分析故障原因
  - Tomcat严格区分大小写
  - <Host>和</Host>为一对，不可缺少
  - 每个容器仅可以包含有效的信息，注意位置问题
  - 修改localhost站点名称后，访问服务时会找不到默认站点

知识讲解



# 端口冲突



## 问题现象

- 故障错误信息

```
[root@svr5 ~]# varnishd  
bind(): Address already in use
```



# 原因分析

## 知识讲解

- 分析故障
- 故障信息：bind(): Address already in use
- 分析故障原因
  - 端口已经被其他服务所占用
  - 找到占用80端口的应用并关闭即可

