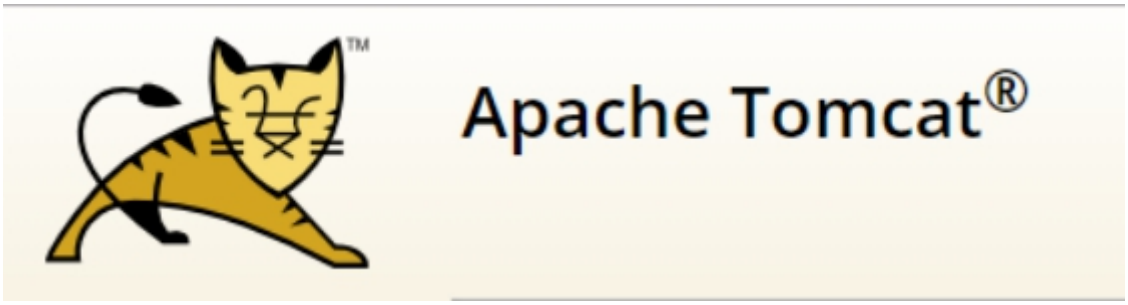
技能展示

- 理解Tomcat的应用场景
- 理解Tomcat的主配置文件
- 会安装配置Tomcat
- 会搭建Nginx+Tomcat负载均衡集群

一、Tomcat介绍



1.1.tomcat简介

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。由于有了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 和JSP 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。

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场合下被普遍使用，是开发和调试JSP 程序的首选。对于一个初学者来说，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是Apache 服务器的扩展，但运行时它是独立运行的，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，不过，Tomcat处理静态HTML的能力不如Apache服务器。

Java Servlet 是运行在 Web 服务器或应用服务器上的程序,它是作为来自 Web 浏览器或其他 HTTP 客户端的请求和 HTTP 服务器上的数据库或应用程序之间的中间层。

1.2.web服务对比

Java Web应用服务器的选择

> 性能高，但商业收费
>
> - Resin 服务器
>   Resin是Caucho公司的产品，是一个非常流行的支持Servlet和JSP的服务器，速度非常快。Resin本身包含了一个支持HTML的Web服务器，这使它不仅可以显示动态内容，而且显示静态内容的能力也毫不逊色，因此许多网站都是使用Resin服务器构建。
> - Jetty 服务器
>   Jetty是一个纯粹的基于Java的网页服务器和Java Servlet容器。尽管网页服务器通常用来为人们呈现文档，但是Jetty通常在较大的软件框架中用于计算机与计算机之间的通信。Jetty作为Eclipse基金会的一部分，是一个自由和开源项目。
> - JBoss服务器
>   JBoss是一个种遵从JavaEE规范的、开放源代码的、纯Java的EJB服务器，对于J2EE有很好的

> 支持。JBoss采用JML API实现软件模块的集成与管理，其核心服务又是提供EJB服务器，不包含Servlet和JSP的Web容器，不过它可以和Tomcat完美结合。
>
> - WebSphere 服务器
> WebSphere是IBM公司的产品，可进一步细分为 WebSphere Performance Pack、Cache Manager 和WebSphere Application Server等系列，其中WebSphere Application Server 是基于Java 的应用环境，可以运行于 Sun Solaris、Windows NT 等多种操作系统平台，用于建立、部署和管理Internet和Intranet Web应用程序。
>
> - WebLogic 服务器
> WebLogic 是BEA公司的产品，可进一步细分为 WebLogic Server、WebLogic Enterprise 和 WebLogic Portal 等系列，其中 WebLogic Server 的功能特别强大。WebLogic 支持企业级的、多层次的和完全分布式的Web应用，并且服务器的配置简单、界面友好。对于那些正在寻求能够提供Java平台所拥有的一切应用服务器的用户来说，WebLogic是一个十分理想的选择。
>
> 性价比高，开源免费
>
> - Tomcat

二、安装和配置

2.1.安装方式介绍

> yum方式安装 加载epel源，版本目前为tomcat7.0
>
> 二进制方式安装 直接下载，部署好java环境，即可运行（推荐）
>
> 源码方式安装　需要下载Apache Ant进行编译安装

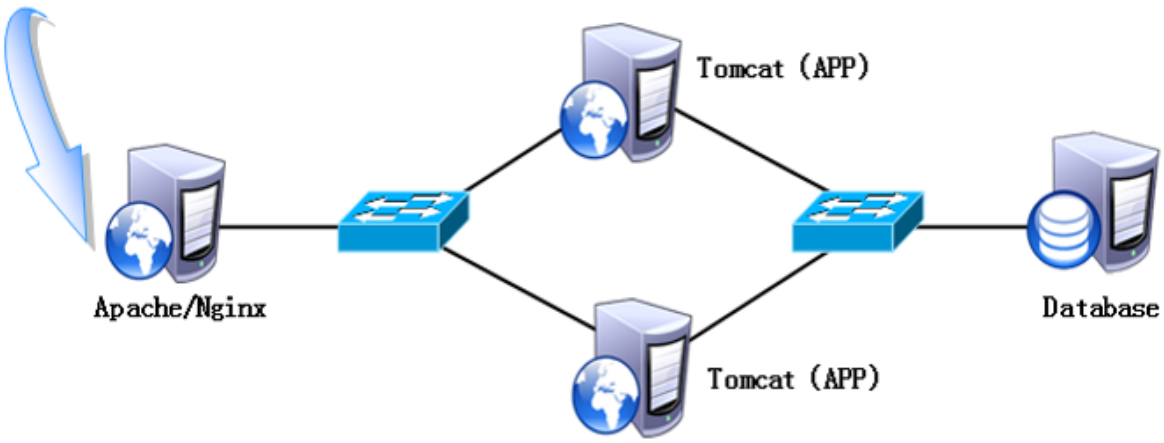
环境说明：

> 操作系统：CentOS Linux release 7.5.1804 (Core)
>
> JDK：jdk-8u261-linux-x64.tar.gz
>
> Tomcat：apache-tomcat-9.0.48.tar.gz

项目概述

> 京北点指科技有限公司发布V3版移联建站管理系统，该项目为Java语言开发的Web站点

Tomcat架构



2.2.二进制方式安装

安装JDK，配置Java环境

```
1、查看当前系统默认安装jdk版本
[root@myserver ~]# java -version

2、安装jdk
[root@jianfeng ~]# tar -zxvf jdk-8u261-linux-x64.tar.gz -C /usr/src
[root@jianfeng ~]# mv /usr/src/jdk1.8.0_261 /usr/local/java

3、配置环境变量
[root@jianfeng ~]# vim /etc/profile
export JAVA_HOME=/usr/local/java
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH

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

3、验证
[root@jianfeng ~]# source /etc/profile
[root@jianfeng ~]# java -version
java version "1.8.0_261"
Java(TM) SE Runtime Environment (build 1.8.0_261-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.261-b12, mixed mode)

注：卸载系统默认jdk,卸载openjdk相关软件包即可
```

安装Tomcat

```
1、安装
[root@jianfeng ~]# tar -zxvf apache-tomcat-9.0.48.tar.gz -C /usr/src
[root@jianfeng ~]# mv /usr/src/apache-tomcat-9.0.48 /usr/local/tomcat
[root@jianfeng ~]# ls /usr/local/tomcat/
bin  BUILDING.txt  conf  CONTRIBUTING.md  lib  LICENSE  logs  NOTICE  README.md
RELEASE-NOTES  RUNNING.txt  temp  webapps  work
```

Tomcat目录说明

> tomcat
> |---bin：存放启动和关闭tomcat脚本；
> |---configtest.sh： 检测配置文件
> |---startup.sh：启动tomcat
> |---shutdown.sh：关闭tomcat
> |---conf：存放Tomcat服务器的各种全局配置文文件，其中最重要的是server.xml和web.xml；
> |---**server.xml**： tomcat的主配置文件 host配置
> |---**web.xml**： tomcat的应用程序的部署描述符文件
> |---lib：存放Tomcat服务器所需的各种jar文件；
> |---logs：存放Tomcat执行时的日志文件；
> |---temp：此目录是tomcat存放的临时文件；
> |---webapps：tomcat默认存放应用程序的目录，好比apache的默认网页存放路径是/var/www/html一样；
> |---docs： tomcat文档
> |---examples： tomcat自带的一个独立的web应用程序例子

> |---host-manager： tomcat的主机管理应用程序
> |---manager： tomcat的管理应用程序
> |---**ROOT**： 指tomcat的应用程序的根，如果应用程序部署在ROOT中，则可直接通过http://ip:port 访问到
> |---work：用于存放JSP应用程序在部署时编译后产生的class文件

2.3.启动测试

> Tomcat服务器默认使用**8080**端口

开启tomcat

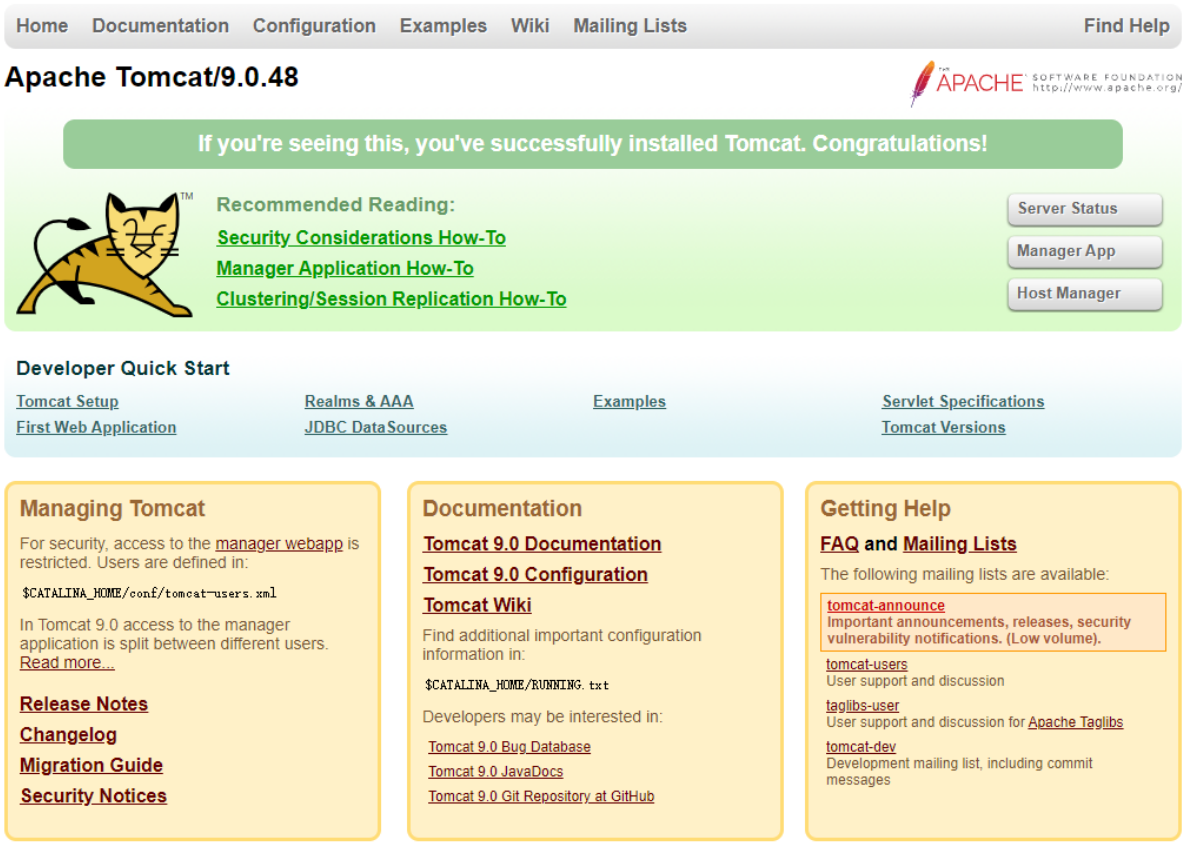
```
![tomcat3](images/3-3/tomcat3.png)[root@jianfeng ~]# cd /usr/local/tomcat/bin/
[root@jianfeng bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/tomcat
Using CATALINA_HOME:   /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME:        /usr/local/java/jre
Using CLASSPATH:
/usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

查看tomcat运行

```
[root@jianfeng ~]# netstat -anpt | grep 8080
tcp        0      0 :::8080                 :::*                    LISTEN      58409/java
```

测试

> 本地访问：http://localhost:8080



关闭tomcat

```
[root@jianfeng ~]# cd /usr/local/tomcat/bin/
[root@jianfeng bin]# ./shutdown.sh
```

2.4.创建服务脚本

```
1、创建服务脚本，可使用service命令，chkconfig命令对tomcat管理
[root@jianfeng ~]# vim /etc/init.d/tomcat
#!/bin/bash
#chkconfig: 35 86 89
#description: Apache Tomcat version 9.0 web applications and web services.

TOMCATBIN=/usr/local/tomcat/bin/catalina.sh
export JAVA_HOME=/usr/local/java

start(){
  $TOMCATBIN start
}
stop(){
  $TOMCATBIN stop
}
restart(){
 stop
 start
}
status(){
  if [ -f '/var/run/tomcat.pid' ];then
     echo 'tomcat is running'
  else
     echo 'tomcat is not run'
  fi
}
configtest(){
  $TOMCATBIN configtest
}

case "$1" in
start)
   start;;
stop)
   stop;;
restart)
   restart;;
status)
   status;;
configtest)
   configtest;;
*)
   echo '{start|stop|restart|configtest|status}'
esac

2、添加服务及管理
[root@jianfeng ~]# chmod +x /etc/init.d/tomcat
[root@jianfeng ~]# chkconfig --add tomcat
[root@jianfeng ~]# chkconfig tomcat on
[root@jianfeng ~]# service tomcat start
```

systemctl服务脚本适用centos7

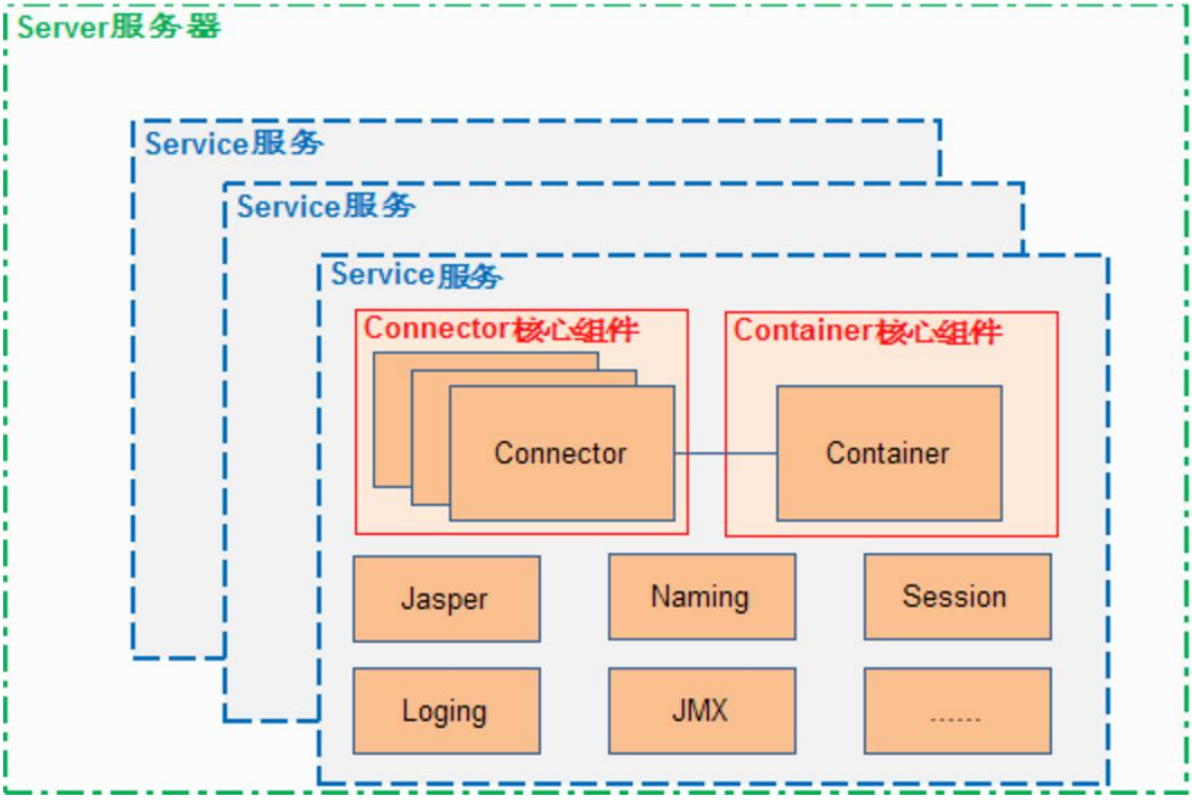
```
[root@jianfeng ~]# /usr/lib/systemd/system/tomcat.service
[Unit]
Description=Apache Tomcat 9
After=syslog.target network.target remote-fs.target nss-lookup.target

[Service]
Type=forking
PIDFile=/usr/local/tomcat/tomcat.pid    #需要在setenv文件内指定pid文件
ExecStart=/usr/local/tomcat/bin/catalina.sh start -DFOREGRAND
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s QUIT $MAINPID
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

2.5.Tomcat配置文件说明

Tomcat结构图



配置文件server.xml说明

> Server节点
>
> Tomcat实例的最顶层元素，对应Tomcat服务器，仅一个。
>
> port：指定监听关闭Tomcat服务器请求的端口。端口用于标记网络上的一个进程，由此一个端口只能唯一的标记一台主机上的一个进程，但是一个进程可以对应多个端口。
>
> shutdown：指定向上述端口发送的命令字符串。
>
> Service节点
>
> 监听来自客户端的http请求，一个Server节点可以有多个Service节点。个人认为所谓的多个Service是，服务器开启了多个线程去监听不同端口收到的来自客户端的http请求。
>
> name：指定Service的名字。

Connector节点

主要设置Service的监听端口。

客户端与Service之间的链接，配置客户端与服务端的链接信息。Tomcat默认监听的是8080端口，商业网站对外发布，若想用户不输入端口，使用http协议的默认端口，这里就必须将其改成80。一个Service节点可以有多个Connector节点。一个端口只能对应一个进程，但是一个进程可以对应多个端口。

port：指定监听客户端请求的端口。

protocol：指定传输协议。

connectionTimeout：指定链接超时时间，毫秒。

redirectPort：指定Service正在处理http请求时收到SSL请求后重定向的端口。

Engine节点

处理监听到的来自客户端的http请求，一个Service节点仅有一个Engine节点。

name：指定Engine的名字。

defaultHost：指定主机缺省时，默认的主机，必须是某个Host结点的name值。

一个完整的http请求网址为：协议://主机:端口/虚拟目录/站点根目录/资源，如http://localhost:8080/aa/examples/index.html。

主机（也可以是虚拟主机或者域名），对应着ip。网络上的一台服务器只有一个ip，但是一个ip可以对应多个逻辑主机，即一个ip可以有多个域名。这多个逻辑主机实际只对应一个真正的物理主机，由此称为虚拟主机。

虚拟目录，暂且认为是真实路径的一个别名，可以没有。

Host节点

主要设置Service的虚拟主机。

配置虚拟主机，一个Engine节点可以有多个Host节点，即多个虚拟主机。

name：指定虚拟主机的名字。该名字可以随便定义，但是必须能被DNS解析到，要对应到指定的ip地址。要定义虚拟主机，先要申　请域名。

appBase：指定存放web应用程序的根目录，如webapps，我们的站点根目录都应该放在此目录下。站点实际是一个有着特定结构的文件夹，该文件夹的结构有着自己的特点，如必须有一个WEB—INF子文件夹，该子文件夹又必须有一个web.xml文件。

Tomcat的conf下有个web.xml文件，用于配置web应用的的信息，被服务器上的全部站点共享，这里的改变会影响到所有的站点。通常我们都是配置web应用站点根目录下的web.xml文件，其改变只对该应用起作用。该文件可以用于设置web应用的默认资源

Context节点

主要设置Service的虚拟目录和默认站点。

配置虚拟目录和默认站点，一个Engin节点可以有多个Context节点，即多个虚拟目录。我们在访问http://locaohost:8080的时候实际访问的是http://locaohost:8080/ROOT/index.jsp。使用http协议，根据localhost定位到主机，根据端口8080定位到主机上的Tomcat服务器Service进程，此时因为用户没有指定站点和资源，则访问conf下的server.xml配置的默认站点和资源。在server.xml中找不到该配置，是因为ROOT是通过硬编码写好的，但是可以通过配置文件修改成其他的站点。

path：指定web应用程序的虚拟路径，如/aa

docBase：指定web应用程序的实际存放路径，即虚拟目录的实际路径，具体到站点目录

reloadable：若为true，Tomcat会自动检测项目WEB—INF目录下的lib和classes文件夹的内容，若有变化会自动重新加载项目而无需重启服务器。

Logger节点

Tomcat的log信息配置。

className：指定log使用的类名，必须实现org.apache.catalina.Logger接口。

prefix：指定log文件的前缀。

suffix：指定log文件的后缀。

timestamp：若为true，则log文件名中必须加入时间戳。

> Value节点
>
> 类似Logger。
>
> directory：指定log文件的存放位置。
>
> directory（指定log文件存放的位置）：
>
> pattern 有两个值，common方式记录远程主机名或ip地址，用户名，日期，第一行请求的字符串，HTTP响应代码，发送的字节数。combined方式比common方式记录的值更多。

2.6.基于域名虚拟web主机

配置域名

```
[root@jianfeng ~]# vim /etc/hosts
192.168.65.129  www.qingmei.com
```

建立java web站点目录

```
[root@jianfeng ~]# mkdir -pv /web/webapp
mkdir: 已创建目录 "/web"
mkdir: 已创建目录 "/web/webapp"
```

编辑java网页

```
[root@jianfeng ~]# vim /web/webapp/index.jsp
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<html>
<head>
    <title>java web page</title>
</head>

<body>
    <% out.println("welcom to test site,tomcat");%>
</body>
</html>
```

配置tomcat

```
#修改tomcat主配置文件server.xml文件
[root@jianfeng ~]# vim /usr/local/tomcat7/conf/server.xml
.........
#找到Host节点,修改主机名（name为域名）
<Host name="www.qingmei.com"  appBase="webapps" unpackWARs="true"
autoDeploy="true">
#添加下面内容
<Context docBase="/web/webapp" path="" reloadable="false"></Context>
```

重启tomcat

```
[root@jianfeng ~]# /usr/local/tomcat/bin/shutdown.sh
[root@jianfeng ~]# /usr/local/tomcat/bin/startup.sh
```

测试

> 本机访问： http://www.qingmei.com:8080

三、Nginx+Tomcat负载均衡架构

3.1.网络架构说明

使用3台虚拟机

主机	IP地址	主要软件
Nginx服务器	192.168.65.128	nginx-1.20.1.tar.gz
Tomcat服务器1	192.168.65.130	jdk-8u261-linux-x64.tar.gz apache-tomcat-9.0.48.tar.gz
Tomcat服务器2	192.168.65.131	jdk-8u261-linux-x64.tar.gz apache-tomcat-9.0.48.tar.gz

项目需求描述

> 随着业务的发展，单实例服务已经不能够满足业务使用需要
>
> pv 页面访问数量
>
> qps 每秒查询次数 并发量
>
> DAU 每日活跃用户数
>
> 单实例无法满足，扩容为多实例

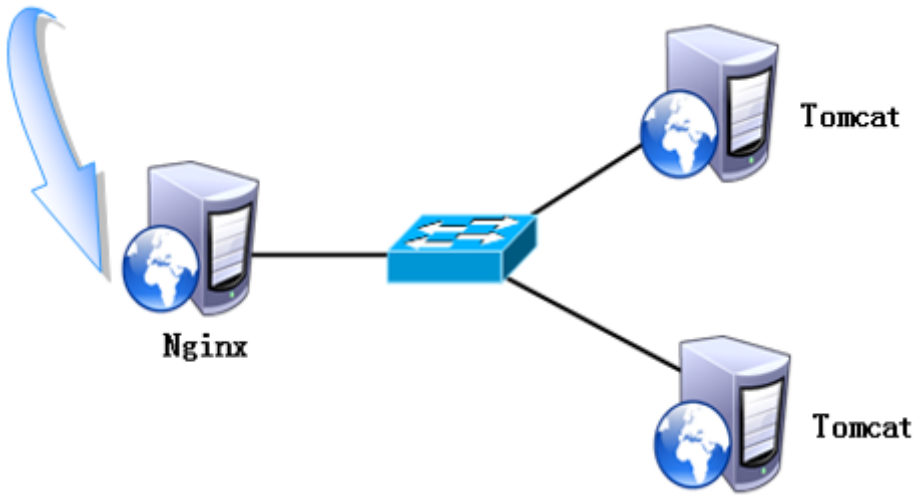
常见的业务运行方式

> 单实例单应用 一个tomcat一个web站点
>
> 单实例多应用 一个tomcat多个web站点
>
> 多实例单应用 多个tomcat同一个web站点

多实例实现方式：

> 多台服务器一个运行一个实例（性能好）

业务流程图示



3.2.Tomcat服务器安装配置

基本安装

> 参考上一节内容，也可将上节的服务器进行克隆
>
> 不必设置基于域名的web虚拟主机

差异化内容

> 在生产环境中通过两个tomcat服务器站点及设置相同，本例中我们为了测试效果所以需要对站点内容进行差异化编辑

```
1、Tomcat服务器1
[root@jianfeng ~]# vim /web/webapp/index.jsp
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<html>
<head>
    <title>java web page</title>
</head>

<body>
    <% out.println("welcom to test site,tomcat1");%>
</body>
</html>

2、Tomcat服务器2
[root@jianfeng ~]# vim /web/webapp/index.jsp
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<html>
<head>
    <title>java web page</title>
</head>

<body>
    <% out.println("welcom to test site,tomcat2");%>
</body>
</html>
```

3.3.Nginx服务器安装

nginx安装

> 参考上一章内容

3.4.Nginx负载均衡实现

> nginx可以做为负载均衡服务器使用，通过upstream和反向代理功能实现服务调度
>
> upstream模块主要用来配置上游服务器，（之所以称为上游服务器，是因为响应的数据是由upstream里配置的后端服务器传输过来的，属于响应数据的上游），**nginx配置集群就是由upstream完成的，upstream模块在http模块下进行配置。**

```
http{
    ........
    upstream tomcat_server {                    //添加,设定负载均衡的服务器列表
        server 192.168.65.130:8080 weight=1;    //添加,服务器IP及端口,weight是权值,数越大权值越高,优先级越高
```

```
        server 192.168.65.131:8080 weight=1;     //可以添加多台后端服务器
    }
    server {
        ......
        location / {
            ......
            proxy_pass http://tomcat;            //添加,代理tomcat_server服务器列表
        }
        .......
    }
    ........
}
```

注意：省略表示该位置可能有默认内容

负载均衡效果测试

通过浏览器访问：http://192.168.65.128
可看到tomcat服务器内容
反复刷新页面，可发现在两台服务器上实现了切换表示负载成功

小结思考

- Tomcat的配置文件有哪些?
- 如何在Nginx中设置负载均衡服务器列表?
- 在Nginx+Tomcat群集中，proxy_pass的作用是是什么?