

一、概述

1.web服务概述：web服务器也成为WWW（WORLD WIDE WEB,万维网）服务器，主要功能是提供网上信息浏览服务。

常见的web服务器有：

httpd)(apache)、nginx+php、

tomcat: jsp+html

weblogic: 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。

websphere: 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。

windows下的有：IIS（Internet Information Server）

2.客户端包括：IE、firefox、chrome、手机浏览器。

3.类linux系统

Linux+Apache+Mysql/Mariadb/Mangodb+Perl/PHP/Python 一组常用来搭建动态网站或者服务器的的开源软件。

本身都是各自独立的程序，但是因为常被放在一起使用，拥有了越来越高的兼容度，共同组成了一个强大的 Web 应用程序平台。随着开源潮流的蓬勃发展，开放源代码的 LAMP 已经与 J2EE 和.Net 商业软件形成三足鼎立之势，并且该软件开发的项目在软件方面的投资成本较低，因此受到整个 IT 界的关注。从网站的流量上来说，50%以上的访问流量是 LAMP 来提供的，LAMP 是最强大的网站解决方案。

4.什么是Apache

Apache Httpd Server 简称 **Apache**, 是**Apache**软件基金会的一个开源的网页服务器, 可以在大多数计算机系统中运行, 由于其多平台和安全性被广泛使用, 所以是最流行的**Web**服务器软件之一。

它快速、可靠并且可通过简单的 API 扩展, 将 Perl/Python 等解释器编译到服务器中! **Apache HTTP** 服务器是一个模块化的服务器, 各个功能使用模块化进行插拔! 目前支持 **Windows, Linux, Unix** 等平台!

Apache 软件基金会 (也就是 **Apache Software Foundation**, 简称为 **ASF**), 是专门为运作一个开源软件项目的 **Apache** 的团体提供支持的非盈利性组织, 这个开源软件项目就是 **Apache** 项目! 那么我们的 **HTTPD** 也只是 **Apache** 的开源项目之一!

主要的开源项目: **HTTP Server, Ant, DB, iBATIS, Jakarta, Logging, Maven, Struts, Tomcat, Tapestry, Hadoop** 等等。只是最有名的是 **HTTP Server**, 所以现在所说的 **Apache** 已经就是 **HTTPD Server** 的代号了! 我们还见的比较多的是 **Tomcat, Hadoop** 等项目。

官网网站: <http://www.apache.org/> <http://httpd.apache.org/>

图标:



5. 什么是Mysql

Mysql是一个关系型数据库管理系统, 由瑞典**MySQL AB** 公司开发, 目前属于**Oracle**旗下公司, 他也是最流行的关系型数据库管理系统! 特别在**WEB**应用发面!

关系型数据库: 它通过数据、关系和对数据的约束三者组成的数据模型来存放和管理数据!

官方网站:

<http://www.mysql.com/>

图标:



6.什么是php

PHP（外文名:**PHP: Hypertext Preprocessor**，中文名：“超文本预处理器”）是一种通 用开源脚本语言。

语法吸收了 C 语言、Java 和 Perl 的特点，利于学习，使用广泛，主要适用于 Web 开发领域。**PHP** 独特的语法混合了 C、Java、Perl 以及 **PHP** 自创的语法。它可以比 **CGI** 或者 Perl 更快速地执行动态网页。用 **PHP** 做出的动态页面与其他的编程语言相比，**PHP** 是将程序嵌入到 **HTML**（标准通用标记语言下的一个应用）文档中去执行，执行效率比 完全生成 **HTML** 标记的 **CGI** 要高许多；**PHP** 还可以执行编译后代码，编译可以达到加密和 优化代码运行，使代码运行更快。

简单的说，**PHP** 就是一个脚本解释器！

注意：

PHP 服务器安装好之后，升级切记注意，不要盲目升级！因为它会废弃很多废弃低效的语法！

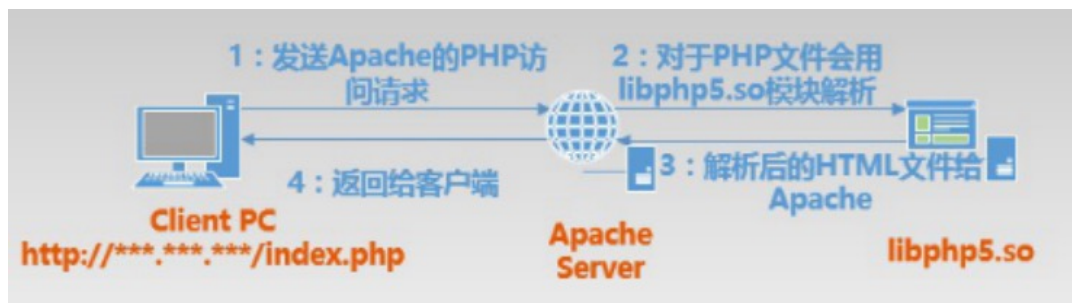
官方网站：

<http://php.net/>

图标：

图标就是 **php** 三个字母。

7.LAMP工作原理



8.web服务器的工作模式和端口

工作模式是：B/S模式

工作端口是：80/http 443/https SSL端口 tomcat是：8080/http

8443/https SSL端口。

安装服务器端：httpd

命令：yum install -y httpd 或者 yum install -y httpd-devel

httpd-2.4.6-88.el7.centos.x86_64.rpm：Apache服务的主程序包，服务器必须安装该软件包。

httpd-devel-2.4.6-88.el7.centos.x86_64.rpm：Apache开发程序包

httpd-manual-2.4.6-88.el7.centos.noarch.rpm：Apache手册文档，包含HTML格式的Apache计划的Apache User's Guide 说明指南

httpd-tools-2.4.6-88.el7.centos.x86_64.rpm：Apache工具包

注意：

源码编译需要手动把依赖的包安装好，才能编译通过。

安装字符界面下的浏览器：elinks

安装命令：

```
[root@kai ~]# yum install elinks -y
```

测试：

```
[root@kai ~]# elinks 192.168.0.122
```

启动服务和加入开机自启动：

```
[root@kai ~]# systemctl start httpd.service
```

```
[root@kai ~]# systemctl enable httpd.service
```

查看端口：

```
[root@kai ~]# netstat -lanutp | grep httpd
```

```
tcp6    0    0 :::80          :::*           LISTEN    35014/httpd
```

主配置文件介绍：

[root@kai ~]# vim /etc/httpd/conf/httpd.conf

ServerName 192.168.0.122:80 **# 服务器主机名**
PidFile /run/httpd/httpd.pid **# 运行进程ID存放**
Listen 80 **# 监听端口，默认本地 IP，如果指定 ip 写上 IP:80**
ServerRoot "/etc/httpd" **# apache配置文件的根目录**
IncludeOptional conf.d/*.conf **#conf.d 里面的 conf 文件也属有效配置文件**
User apache **# 以哪个身份运行**
Group apache **# 以哪个组运行**
ServerAdmin cat@xuegod.cn **#管理员邮箱**
DocumentRoot "/var/www/html/bbs" **# 默认的主目录 如果要更改 连同下面的Directory 目录一同修改，才有效**
AllowOverride None **#设置为 none，忽略.htaccess**
Options Indexes FollowSymLinks **# 目录权限 none : 为没有任何权限**
All:所有权限 Indexes: 没有找到默认文章（如：**index.php**）就会返回当前文件夹下的所以文件列表
AddDefaultCharset UTF-8 **#支持的语言，默认编码**

<IfModule dir_module> **# 默认首页 设置默认目录的默认文档**
 DirectoryIndex index.html
</IfModule>

<Directory /> **#这里是全局目录默认规则，拒绝所有**
 AllowOverride none
 Require all denied
</Directory>

<Files ".ht*"> **#防止 .htaccess 和 .htpasswd 文件被从Web上访问**
 Require all denied
</Files>

Require all granted **#允许所有访问**
Require all denied **#拒绝所有访问**
Require ip 192.168.1.0/24 **#仅允许192.168.1.0/24网络的主机访问**

Require not ip 192.168.1.2 #禁止192.168.1.2的主机访问，其它都可以
ErrorLog "logs/error_log" #错误日志的位置

9.虚拟主机的配置文件样例如下：

```
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot /var/www/html/bbs/
    ServerName www.lks.com
        <Directory "/app/www">      # 必须配置具体目录的访问授权
            Options FollowSymLinks
            AllowOverride All
            Require all granted      # 允许所有
        </Directory>
    ErrorLog logs/dummy-host.example.com-error_log
    CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>
```

10. yum 的方式搭建 LAMP 环境

LAMP=Linux（存在）+Apache（HTTPD 已安装）+Mysql+PHP

```
[root@kai ~]# yum -y install httpd mariadb mariadb-server php php-
mysql
```

12.测试数据库

启动数据库并加入开机自启动：

```
[root@kai ~]# systemctl start mariadb.service
[root@kai ~]# systemctl enable mariadb.service
```

安全初始化数据库：

```
[root@kai ~]# mysql_secure_installation
```

登录数据库：

```
[root@kai ~]# mysql -uroot -p123456
```

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 2

Server version: 5.5.64-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]>

13.测试apache 是否支持php

创建一个文件:

```
[root@kai ~]# vim /var/www/html/index.php
```

添加以下代码:

```
<?php
    phpinfo();
?>
```

然后访问主机IP, 如果出现以下界面, 说明LAMP环境搭建完成!



PHP Version 5.4.16	
System	Linux kai.kai.com 3.10.0-957.el7.x86_64 #1 SMP Thu Nov 8 23:39:32 UTC 2018 x86_64
Build Date	Oct 30 2018 19:31:42
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/fileinfo.ini, /etc/php.d/json.ini, /etc/php.d/mysql.ini, /etc/php.d/mysqli.ini, /etc/php.d/pdo.ini, /etc/php.d/pdo_mysql.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/phar.ini, /etc/php.d/sqlite3.ini, /etc/php.d/zip.ini
PHP API	20100412
PHP Extension	20100525

注意: 每次修改了配置文件httpd.conf中内容, 一定要重启apache服务! 命令如下:

```
[root@kai ~]# systemctl restart httpd.service
```

二.实战举例:

例1: 搭建一台测试web服务器

案例:

部门内部搭建一台WEB服务器, 采用的IP地址和端口为192.168.0.122:80, 首

页采用index.html文件。管理员E-mail地址为 cat@xuegod.cn，网页的编码类型采用UTF-8，所有网站资源都存放在/var/www/html目录下，并将Apache的配置文件根目录设置为/etc/httpd目录。

解答：

- 1.ServerAdmin cat@xuegod.cn # 管理员邮箱
- 2.ServerName 192.168.0.122:80 # 服务器主机名
- 3.DocumentRoot "/var/www/html" # 网站根目录
- 4.Options Indexes FollowSymLinks # 当一个网站没有默认首页的时候，允许显示此目录列表
- 5.DirectoryIndex index.html # 指定默认首页
- 6.AddDefaultCharset UTF-8 # 设置服务器默认编码为：utf-8

7.取消apache默认欢迎页面方法：

```
[root@kai httpd]# cd /etc/httpd/conf.d/
```

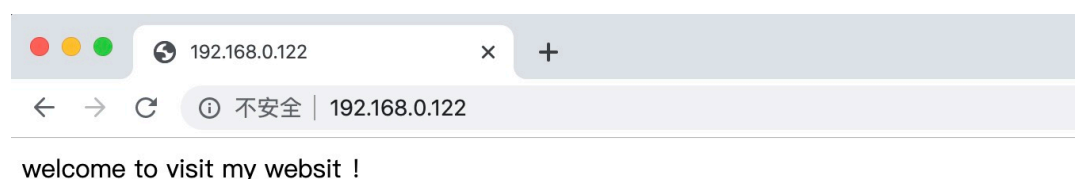
```
[root@kai conf.d]# mv welcome.conf welcome.conf.bak
```

三种方式，一个是删除welcome.conf或者注释里面的内容。建议使用mv 修改成不同名称，如上命令！

8.创建首页：

```
[root@kai ~]# echo "welcome to visit my websit ! " >  
/var/www/html/index.html
```

如下图：



三、LAMP环境下，如何设置集中不同的安全访问机制：

1.修改网站目录对应参数及权限

修改网站默认根目录，并且设置访问权限，只有在我允许范围之内或者指定ip才能访问：

修改网站默认更目录：

创建bbs目录：

```
[root@kai conf]# mkdir /var/www/html/bbs
```

把index文件移动到bbs目录下：

```
[root@kai conf]# mv /var/www/html/index.html /var/www/html/bbs/
```

修改httpd.conf主配置文件：

```
[root@kai conf]# vim httpd.conf
```

更改内容：

```
DocumentRoot "/var/www/html/bbs"
```

目录与访问控制：

```
<Directory "/var/www/html/bbs/">
```

```
Options Indexes FollowSymLinks
```

```
AllowOverride None
```

```
Order allow,deny
```

```
Allow from 192.168.0.0/24    # 允许访问的IP网络段
```

```
Deny from 192.168.0.103    # 不允许访问的ip 或者 ip段
```

```
# Allow from www.baidu.com  # 域的写法
```

```
# Require ip 192.168.1.0/24
```

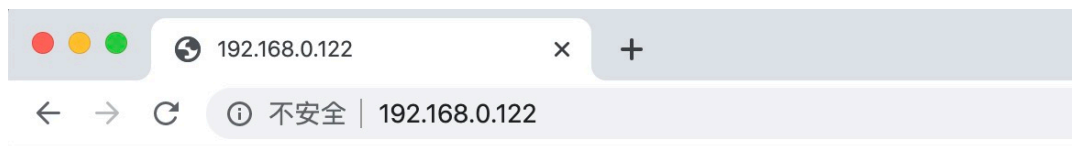
```
# Require not ip 192.168.0.103
```

```
# Require all granted
```

```
</Directory>
```

配置文件修改后，重启apache服务！

访问网站，如下图：



welcome to visit my websit ! hello

2.使用别名，引用网站根目录以外的路径配置：

注意：apache 的别名也叫虚拟目录

语法：

Alias /urlname/ "path物理路径"

例2：

将/usr/local/phpdata 目录通过虚拟目录功能添加到网站根目录。当访问 <http://192.168.0.122/phpdata/>时，就可以访问目录/usr/local/phpdata中的内容。

操作步骤：

1.创建目录：

```
[root@kai conf]# mkdir /usr/local/phpdata -p
```

2.添加index.html内容：

```
[root@kai conf]# echo "This is Alias Phpdata ! " >  
/usr/local/phpdata/index.html
```

3.配置httpd.conf文件：

```
[root@kai conf]# vim httpd.conf
```

配置的地方：

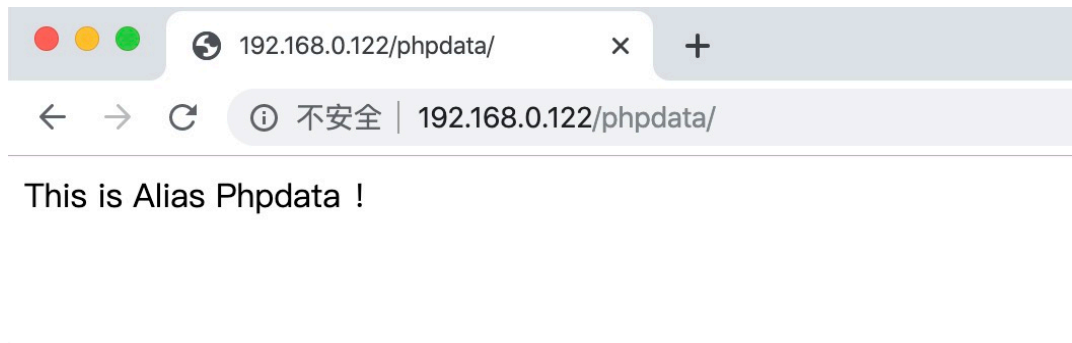
```
Alias /phpdata/ "/usr/local/phpdata/"  
<Directory "/usr/local/phpdata/">  
    AllowOverride None  
    Options Indexes FollowSymLinks  
    Require all granted  
</Directory>
```

保存文件，退出。

4.重启apache服务：

```
[root@kai conf]# systemctl restart httpd.service
```

访问网站，如下图：



说明：Alias /phpdata/ "/usr/local/phpdata/" # /phpdata/ 根据自己的要求起名，比如可以改为 /php/, 访问地址则为： <http://192.168.0.122/php/>

3.当一个目录没有默认首页时，访问<http://192.168.0.122/phpdata/>禁止显示目录列表。

操作步骤：

1.命令操作，如下：

```
[root@kai conf]# cp /boot/grub/ /usr/local/phpdata/ -r
```

```
[root@kai conf]# ll /usr/local/phpdata/
```

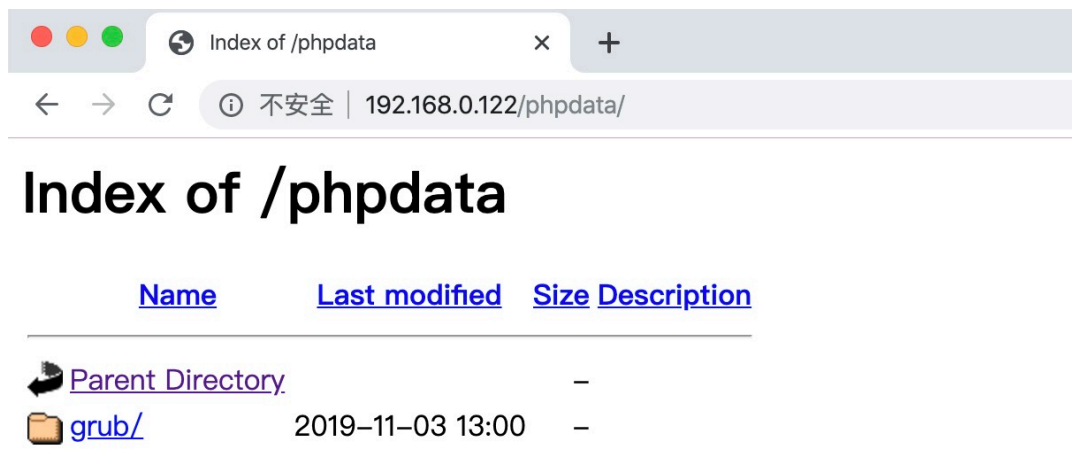
```
total 4
```

```
drwxr-xr-x 2 root root 27 Nov  3 13:00 grub
```

```
-rw-r--r-- 1 root root 26 Nov  3 12:10 index.html
```

```
[root@kai conf]# rm /usr/local/phpdata/index.html -rf
```

访问的页面如下图：



2.修改httpd.conf配置文件

```
[root@kai conf]# vim httpd.conf
```

注释掉如下：

```
Alias /phpdata/ "/usr/local/phpdata/"
```

```
<Directory "/usr/local/phpdata/">
```

```
    AllowOverride None
```

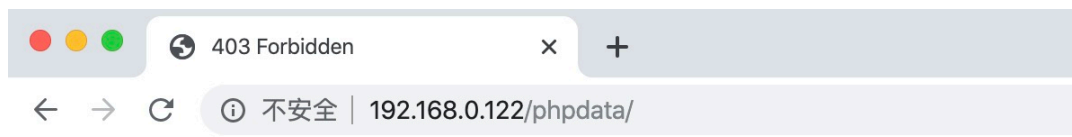
```
    #Options Indexes FollowSymLinks
```

```
    Require all granted
```

```
</Directory>
```

3.重启apache服务

访问页面如下图：



Forbidden

You don't have permission to access /phpdata/ on this server.

4.通过用户认证方式，对网站下/usr/local/phpdata/目录进行保护。设置/usr/local/phpdata/目录，只能通过用户名和密码方式进行访问。

操作步骤：

1.修改配置文件httpd.conf

```
[root@kai conf]# vim httpd.conf
```

```
Alias /phpdata/ "/usr/local/phpdata/"
```

```
<Directory "/usr/local/phpdata/">
```

```
    AllowOverride None
```

```
    #Options Indexes FollowSymLinks
```

```
    authtype basic
```

AuthType命令：指定

认证类型为：basic。

```
    authname "liuks web site"
```

AuthName命令：指定

认证区域名称。区域名称是在提示要求认证的对话框中显示给用户的。

```
    authuserfile /etc/httpd/conf/passwd.secret
```

AuthUserFile命令：指

定一个包含用户名和密码的文本文件，每行一对。

```
    require valid-user
```

(在AuthUserFile指定

的文件中任何用户都可以访问)

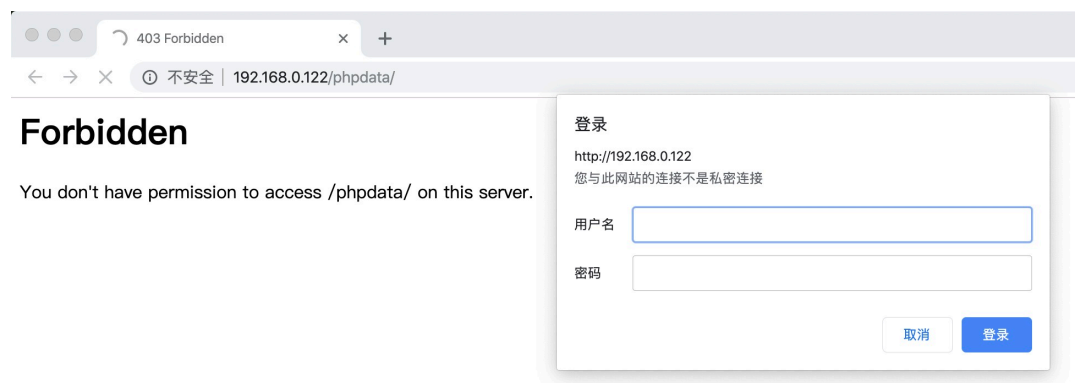
```
    #Require all granted
```

```
</Directory>
```

保存，退出。

2.重启apache服务

3.访问网页,如下图：



4.利用Apache附带的程序htpasswd，生成包含用户名和密码的文本.命令如

下:

注意: 第一个生成用户名和密码, 需要创建文件, 因此需要加-c参数, 第二次以后创建用户不需要c了, -m参数 是用 md5方式加密。

```
[root@kai conf]# htpasswd -cm /etc/httpd/conf/passwd.secret liuks
```

New password:

Re-type new password:

Adding password for user liuks

```
[root@kai conf]# htpasswd -m /etc/httpd/conf/passwd.secret lancer
```

New password:

Re-type new password:

Adding password for user lancer

```
[root@kai conf]# cat /etc/httpd/conf/passwd.secret
```

liuks:\$apr1\$Enl.rd.n\$UyRSi9wlfCQileir3FRMQ/

lancer:\$apr1\$SElwbVLS\$gp6Hpbl.2V7kG/t0qUUm6/

5.可以使用设定的用户名和密码登录 <http://192.168.0.122/phpdata/>网站了。

5.配置apache虚拟主机, 实现在一台服务器上运行多个网站。

apache虚拟主机实现有三种方法:

1.通过不同的IP地址

2.通过不同的域名

3.通过不同的端口

以上三种都需要把虚拟主机功能打开:

方法1:

通过不同IP地址, 解析不同的域名

操作步骤:

1.增加一个ip,并且查看下:

命令如下:

```
[root@kai conf]# ifconfig ens33:1 192.168.0.121 netmask 255.255.255.0
```

```
[root@kai conf]# ifconfig
```

```
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
```

```
    inet 192.168.0.122 netmask 255.255.255.0 broadcast
192.168.0.255
    inet6 fe80::393e:643:d037:411c prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:8c:9f:0b txqueuelen 1000 (Ethernet)
    RX packets 397075 bytes 32644136 (31.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 293975 bytes 46992739 (44.8 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
ens33:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.121 netmask 255.255.255.0 broadcast
192.168.0.255
    ether 00:0c:29:8c:9f:0b txqueuelen 1000 (Ethernet)
```

2.配置虚拟主机文件：

注意：虚拟主机配置文件在**conf.d**目录下创建。

创建一个**vhost.conf**文件，在里面添加如下配置：

```
[root@kai conf.d]# vim vhost.conf
```

```
<VirtualHost 192.168.0.122:80>
```

```
    ServerAdmin webmaster@dummy-host.example.com
```

```
    DocumentRoot /var/www/html/
```

```
    ServerName www.example.com
```

```
    ErrorLog logs/dummy-host.example.com-error_log
```

```
    CustomLog logs/dummy-host.example.com-access_log common
```

```
</VirtualHost>
```

```
<VirtualHost 192.168.0.121:80>
```

```
    ServerAdmin webmaster@dummy-host.example.com
```

```
    DocumentRoot /var/www/html/bbs/
```

```
    ServerName www.example.com
```

```
    ErrorLog logs/dummy-host.example.com-error_log
```

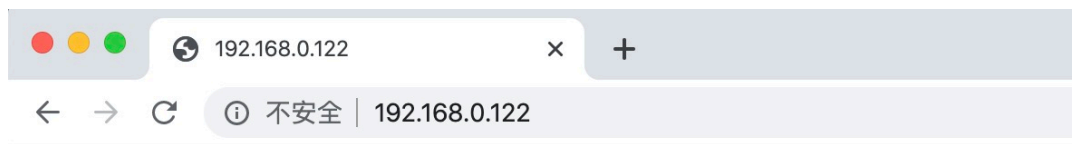
```
    CustomLog logs/dummy-host.example.com-access_log common
```

```
</VirtualHost>
```

3.重启apache服务

4.测试访问 <http://192.168.0.122> 和 <http://192.168.0.121> 网页

访问网页，如下图：



hello i am is 192.168.0.122's content



hello i am is 192.168.0.121's content

方法2:

通过不同的域名访问:

操作步骤:

1.编辑vhost.conf配置文件:

```
[root@kai conf.d]# vim vhost.conf
```

```
<VirtualHost *:80>
```

```
    ServerAdmin webmaster@dummy-host.example.com
```

```
    DocumentRoot /var/www/html/
```

```
    ServerName www.lks.com
```

```
    ErrorLog logs/dummy-host.example.com-error_log
```

```
    CustomLog logs/dummy-host.example.com-access_log common
```

```
</VirtualHost>
```

```
<VirtualHost *:80>
```

```
    ServerAdmin webmaster@dummy-host.example.com
```

```
    DocumentRoot /var/www/html/bbs/
```

```
    ServerName www.lancer.com
```


ErrorLog logs/dummy-host.example.com-error_log

CustomLog logs/dummy-host.example.com-access_log common

</VirtualHost>

2.由于没有真正的域名，所有本机配置下Host文件，如下配置：

[root@kai conf.d]# vim /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4

localhost4.localdomain4

::1 localhost localhost.localdomain localhost6

localhost6.localdomain6

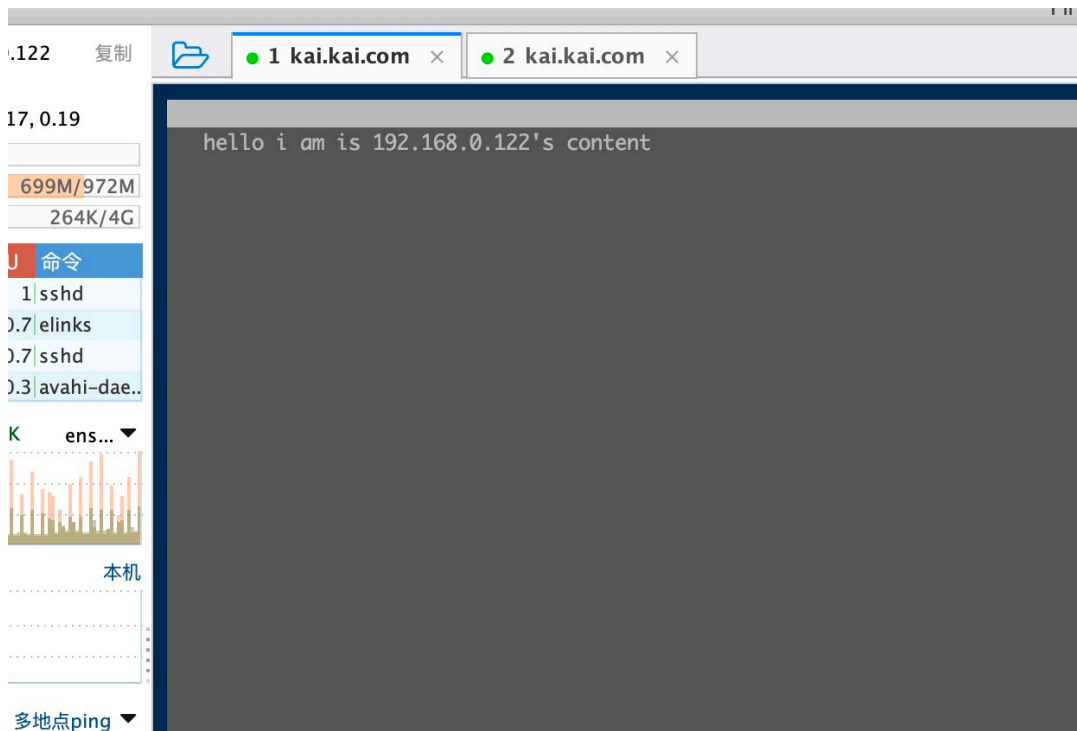
192.168.0.122 www.lks.com

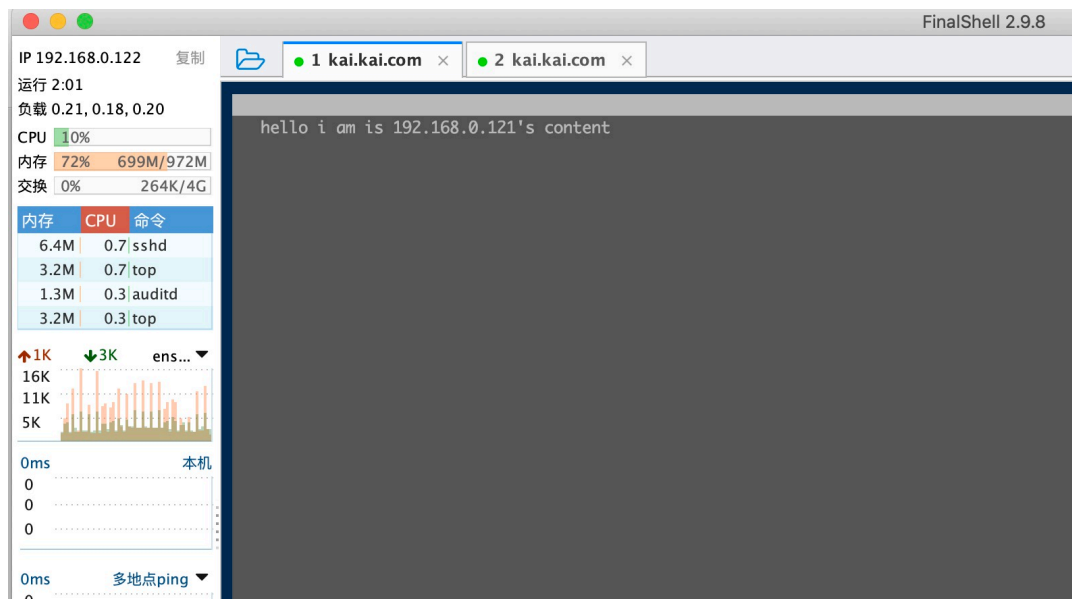
192.168.0.122 www.lancer.com

3.重启apache服务

通过elinks命令访问两个域名 www.lks.com 和 www.lancer.com

访问如下图：





方法3:

基于不同端口配置虚拟主机

操作步骤:

1.编辑vhost.conf文件, 如下修改:

```
[root@kai conf.d]# vim vhost.conf
```

```
<VirtualHost *:81>
```

```
    ServerAdmin webmaster@dummy-host.example.com
```

```
    DocumentRoot /var/www/html/
```

```
    ServerName www.example.com
```

```
    ErrorLog logs/dummy-host.example.com-error_log
```

```
    CustomLog logs/dummy-host.example.com-access_log common
```

```
</VirtualHost>
```

```
<VirtualHost *:82>
```

```
    ServerAdmin webmaster@dummy-host.example.com
```

```
    DocumentRoot /var/www/html/bbs/
```

```
    ServerName www.example.com
```

```
    ErrorLog logs/dummy-host.example.com-error_log
```

```
    CustomLog logs/dummy-host.example.com-access_log common
```

```
</VirtualHost>
```

2.配置主配置文件httpd.conf

[root@kai conf]# vim httpd.conf

添加两个端口 **81** 和 **82**

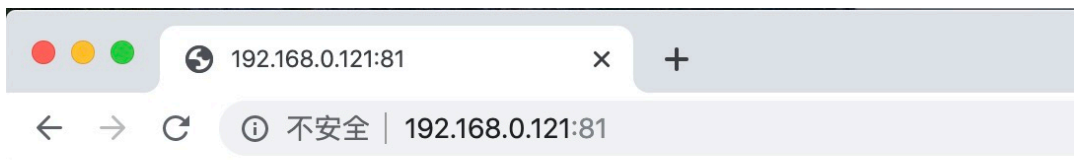
#Listen 80

Listen 81

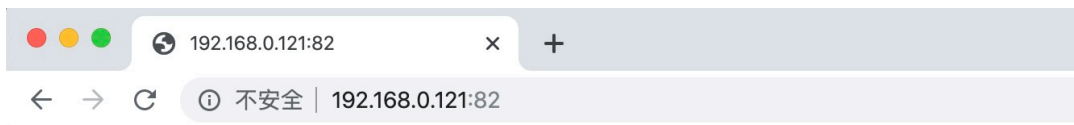
Listen 82

3.重启apache服务

4.访问网页，如下图：



hello i am is 192.168.0.122's content



hello i am is 192.168.0.121's content

OK，结束，希望大家熟练的操作一遍。

