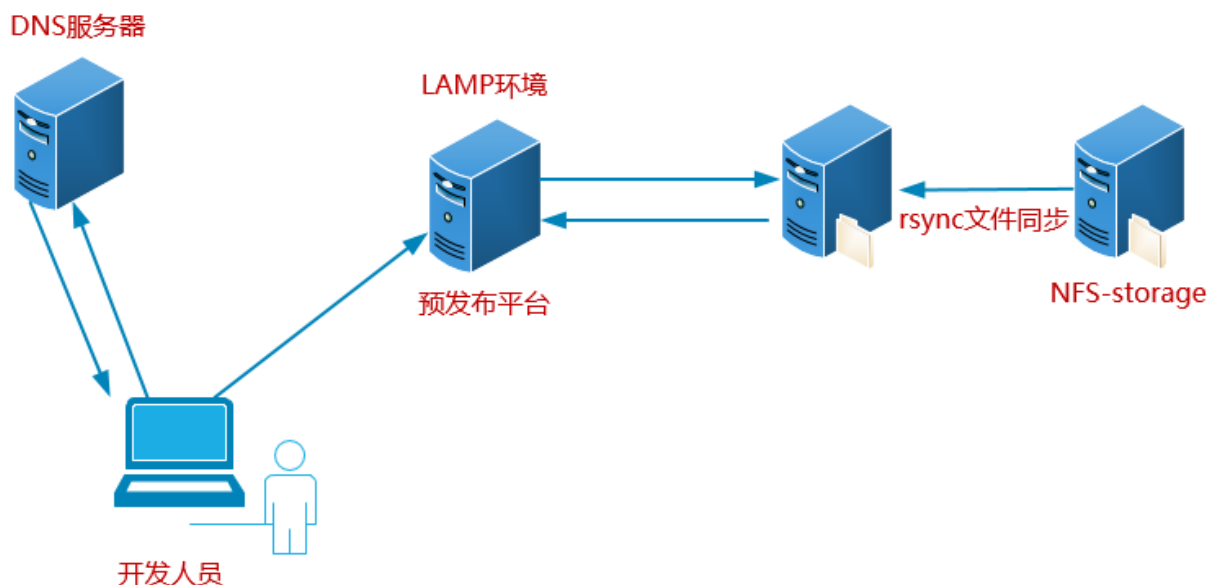


## 项目背景

随着业务不断增长，用户基数越来越大，为更好满足用户体验，开发人员提一个工单过来，需要运维人员给开发人员部署一套**预发布环境**（和生产环境保持一致），能够保证开发人员高效的进行预发布测试等工作。



## 具体要求

- 源码部署LAMP环境，和生产保持一致
- 静态资源文件同步生产环境

## 涉及知识点

- 源码部署LAMP环境（新，重点掌握）
- rsync同步静态资源（旧）
- 虚拟主机的配置（新）
- 用户认证和网络访问控制（新）
- Apache的安全认证（扩展）（新）

## 课程目标

- 了解常见的WEB服务器
- 理解Apache的工作原理
- 能够更改apache的默认数据根目录 /var/www/html
- 能够配置对网页实现用户名密码认证（掌握）
- 能够配置apache的虚拟主机（重点）
- 熟悉Apache的网络访问控制配置（掌握）
- 能够通过源码包的方式搭建LAMP环境（掌握）

## 理论知识

## 一、WEB服务概述

web服务是我们日常使用最多的服务，所有的网站都是以web服务的形式为我们呈现

### 1. WEB服务的特点

服务的架构：

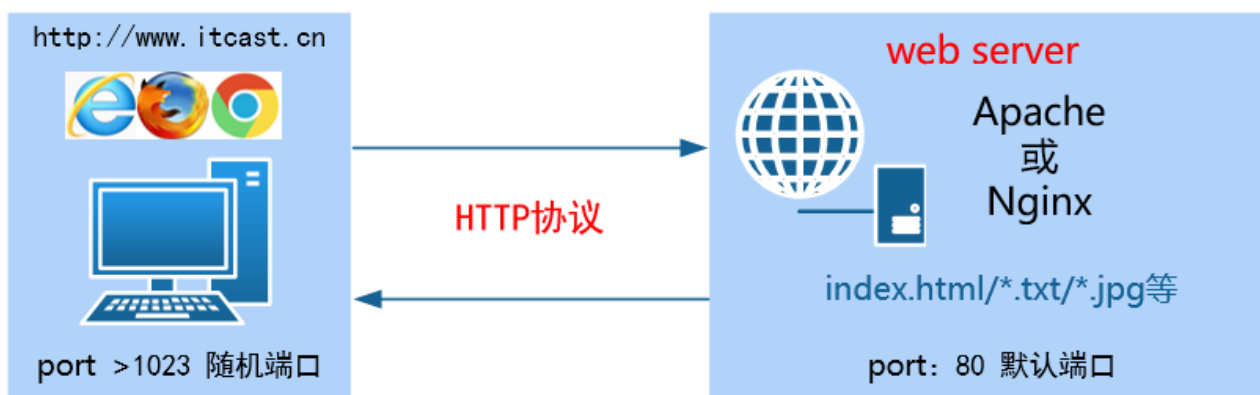
**C/S:** FTP SAMBA SSH (一般应用于局域网中)

**B/S:** apache、nginx (一般应用于互联网中)

- web服务分为客户端和服务端
- web客户端通常是我们所使用的**浏览器** (IE/Chrome/Firefox/Safari或者命令行浏览器等)
- **web服务端**就是我们所访问的网站提供的**web服务器**
  - 常见的web服务端程序有：  
**Apache**/Nginx/Tengine/Lighttpd/IIS等，不同的web服务针对不同的系统平台各自有优缺点

### 2. HTTP协议

- web服务端与客户端是通过HTTP协议 (HyperText Transfer Protocol) 超文本传输协议进行交互



- Apache或Nginx都**只支持静态页面**的解析

### 3. 静态页面和动态页面

- HTML语言
  - HTML (HyperText Markup Language) 超文本标记语言，是绝大多数网页使用的语言，几乎所有的网页都是以HTML作为源代码，通过浏览器解释后展现出来
  - HTML有固定语法，用来存储网页数据，定义不同元素，如文字、标题、图片等，现在的网页都用CSS语言来存储网页的表现形式
  - 现代主流的网页设计架构：
    - 内容存储：HTML
    - 网页架构：div
    - 网页样式：css
  - HTML形式的页面我们称之为**静态页面**，包含一些图片、文本、表格等
- 动态语言
  - 网站使用特定的语言编写的一些程序，在用户访问的时候**基于不同的条件生成**不同的HTML代码返回给用户浏览器，这样就实现网页的动态化。

- 常见的动态语言: .php .jsp .cgi .asp、python等

#### 4. WEB服务的中间件

- php: **PHP-FPM**、HHVM (HipHop Virtual Machine)
- jsp: **Tomcat**、JBoss、Resin、IBM WebSphere

#### 5. 常见的WEB组合方式

- **LAMP** (Linux + Apache + MySQL + PHP) //php作为Apache的模块
- **LNMP** (Linux + Nginx + MySQL + PHP) //php-fpm进程 (服务)
- Nginx /Apache+ Tomcat

## 二、Apache的介绍

Apache(httpd)是著名的web服务器软件，开源，由apache软件基金会负责管理开发。

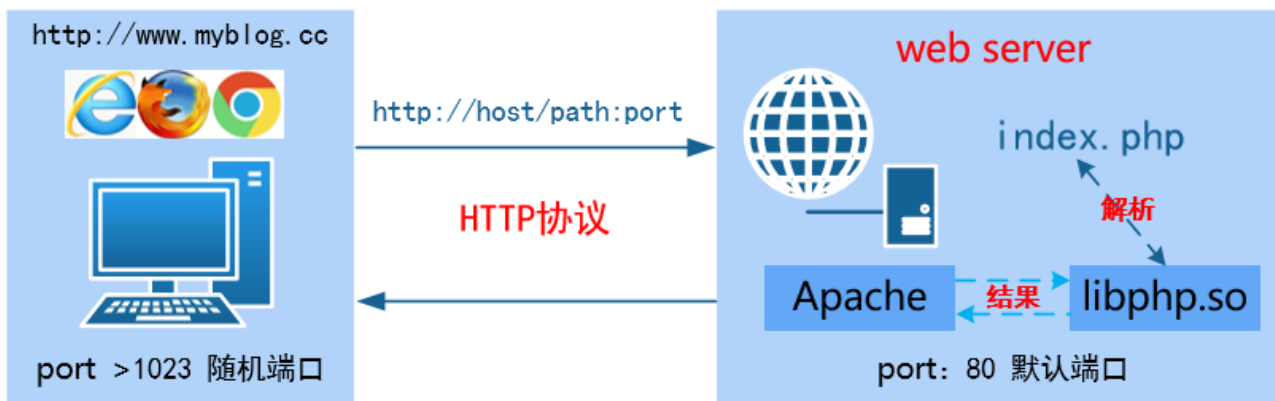


# Apache

#### 1. Apache服务特点

- 开放源码
- 跨平台
- 支持多种编程语言
- 采用模块化设计
- 安全稳定

#### 2. Apache的工作原理



#### 3. Apache的官网和手册

- [www.apache.org](http://www.apache.org)
- 英文手册: 需要安装

```
httpd-manual.noarch //安装后启动服务就可以本地查看
yum -y install httpd-manual.noarch
service httpd restart
查看官方手册:
IE: http://10.1.1.2/manual
```

- 中文手册参考: <http://www.jinbuguo.com/apache/menu22/index.html>

## 4. Apache的软件包

- 软件包: httpd-2.2

## 5. Apache相关文件介绍

服务端口:

80/tcp(http) 443/tcp http+ssl(https)

配置文件:

/etc/httpd/conf	主配置文件目录
/etc/httpd/conf.d/*.conf	子配置文件目录
/etc/httpd/conf.d/README	说明书
/etc/httpd/conf.d/welcome.conf	当没有首页index.html 显示红帽欢迎页面
/etc/httpd/conf/httpd.conf	主配置文件
/etc/httpd/logs	日志目录 /var/log/httpd/硬链接
/etc/httpd/modules	库文件 /usr/lib64/httpd/modules硬链接
/etc/httpd/run	pid信息
/etc/logrotate.d/httpd	日志轮循
/etc/rc.d/init.d/httpd	启动脚本
/etc/sysconfig/httpd	额外配置文件
/usr/lib64/httpd	
/usr/lib64/httpd/modules	库文件
/usr/lib64/httpd/modules/mod_actions.so	
/usr/sbin/apachectl	apache官方启动脚本
/usr/sbin/httpd	
/var/www	apache数据目录
/var/www/cgi-bin	存放apache的cgi脚本程序的数据目录
/var/www/html	存放apache的html数据目录
/var/www/error	
/var/www/error/HTTP_NOT_FOUND.html.var	404
/var/www/error/HTTP_FORBIDDEN.html.var	403

## 6. Apache主配置文件

ServerRoot "/etc/httpd"	//服务主目录
Listen 80	//监听端口
IncludeOptional conf.d/*.conf	//包含conf.d下的*.conf文件
User apache	//运行Apache的用户
Group apache	//运行Apache的用户组
DirectoryIndex index.html index.php	//设置默认主页
DocumentRoot /var/www/html/	//站点默认数据主目录

```

<Directory />                                //系统的根目录授权
    Options FollowSymLinks                    //支持软链接
    AllowOverride None                        //不支持.htaccess 访问列表 .htaccess文件提供了针对每个目录改变配置的方法
</Directory>

RHEL6:
<Directory "/var/www/html">                  //授权
    Options Indexes FollowSymLinks           //支持索引 支持软链接
    AllowOverride None                       //不支持 .htaccess 访问列表
    Order allow,deny                         //排序 先允许再拒绝
    Allow from all                           //允许all
</Directory>

RHEL7:
<Directory "/var/www/html">
    AllowOverride None
    Require all granted                      //允许所有人访问
</Directory>

```

## 实战演练

### 基本功操作

需求1:

访问一个静态页面，内容：hello world 新年快乐！

http://web-server-ip

思路:

1. 关闭防火墙和selinux
2. 配置yum源
3. 软件三部曲（静态页面数据根目录/var/www/html）
4. 了解配置文件（man 5 xx.conf apache查看手册）
5. 根据需求通过修改配置文件完成服务搭建
6. 启动服务测试验证

步骤:

- 1-4: 省略
  5. 根据需求修改配置文件
    - 1) 创建一个首页文件index.html

```
echo "hello world 新年快乐" > /var/www/html/index.html
```
- 注意：不需要修改配置文件

引申:

由于/var/www/html目录里存放文件放到了根分区，现在需要更改apache的默认数据根目录/www，如何完成？

思路：修改配置文件完成

1. 创建一个/www目录

```
mkdir /www
```

## 2. 在配置文件里来指定默认数据根目录

```
<Directory "/www">           //授权
    Options Indexes FollowSymLinks    //支持索引 支持软链接
    AllowOverride None                //不支持 .htaccess 访问列表
    Order allow,deny                  //排序 先允许再拒绝
    Allow from all                    //允许all
</Directory>
```

扩展: Apache作为文件共享服务使用

注意: 默认情况下, apache回到默认的数据目录里找index.html的首页文件, 如果首页文件不存在就会找测试页, 如果测试页不存在, 那么就会将目录里的文件共享出去

### 1. 共享文件

需求: 输入apache的IP可以直接看到共享文件

1) 在默认的共享目录里创建一堆文件

2) 移动默认的测试页welcome.conf文件

3) 启动服务即可

### 2. 通过别名的方式共享文件

1) 在配置文件里定义别名

```
Alias /test/ "/tmp/"
<Directory "/tmp/">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

2) 重启服务

3) 测试验证

http://10.1.1.2/test/

注意: 别名后面有"/", 那么访问时必须得加"/"

### 3. 通过软连接的方式共享文件

共享/home目录

1) 创建软连接

注意:

软连接文件必须放到apache的默认数据根目录里

```
ln -s /home/ /data/homes
```

2) 测试验证

http://10.1.1.2/homes/

## 需求2:

搭建2个静态页面网站, 内容分别为:

this is first test page!

this is second test page!

环境：一台服务 → 搭建两个网站

http://10.1.1.1/web1

http://10.1.1.1/web2

方法1：（不推荐）

步骤：

```
[root@web-server ~]# mkdir /www/web{1..2}
```

```
[root@web-server ~]# echo "this is first test page! " > /www/web1/index.html
```

```
[root@web-server ~]# echo "this is 2 test page! " > /www/web2/index.html
```

测试验证：

http://10.1.1.1/web1

http://10.1.1.1/web2

方法2：通过虚拟主机的方式实现（推荐）

基于IP的虚拟主机

基于端口的虚拟主机

基于域名的虚拟主机

eth0 10.1.1.1

eth1 192.168.159.133

在配置文件的最后面有虚拟主机的模板：

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

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

```
#   DocumentRoot /www/docs/dummy-host.example.com
```

```
#   ServerName dummy-host.example.com
```

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

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

```
#</VirtualHost>
```

## 1. 基于IP的虚拟主机

目标：

http://10.1.1.1            this is first test page!

http://192.168.159.131    this is second test page!

思路：

1) 必须配置2个IP地址

2) 配置2个基于IP的虚拟主机

步骤：

1) 配置IP地址

方法1：增加1个虚拟网卡

方法2：手动给eth0增加一个子接口

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
```

```
    link/ether 00:0c:29:41:a2:fc brd ff:ff:ff:ff:ff:ff
```

```
    inet 10.1.1.2/24 brd 10.1.1.255 scope global eth0
```

```
    inet6 fe80::20c:29ff:fe41:a2fc/64 scope link
```

```
        valid_lft forever preferred_lft forever
```

```
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
```

```
    link/ether 00:0c:29:41:a2:06 brd ff:ff:ff:ff:ff:ff
```

```
    inet 192.168.159.135/24 brd 192.168.159.255 scope global eth1
```

```
    inet6 fe80::20c:29ff:fe41:a206/64 scope link
```

```
        valid_lft forever preferred_lft forever
```

```
# ifconfig eth0:1 192.168.0.254/24      临时增加个子接口
```

## 2) 修改配置文件增加2个虚拟主机

```
[root@web-server conf]# vim httpd.conf
```

在配置文件后面增加以下内容:

```
<VirtualHost 10.1.1.2:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot /data/web1
    #ServerName dummy-host.example.com
    ErrorLog logs/web1-error_log
    CustomLog logs/web1-access_log common
</VirtualHost>
<VirtualHost 192.168.159.135:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot /data/web2
    #ServerName dummy-host.example.com
    ErrorLog logs/web2-error_log
    CustomLog logs/web2-access_log common
</VirtualHost>
```

## 3) 重启服务

```
[root@web-server conf]# service httpd restart
```

```
Stopping httpd:                                     [ OK ]
Starting httpd: httpd: apr_sockaddr_info_get() failed for app1-server
httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1
for ServerName
[Thu Sep 06 11:40:15 2018] [error] (EAI 2)Name or service not known: Failed to resolve server
name for 192.168.159.131 (check DNS) -- or specify an explicit ServerName
[Thu Sep 06 11:40:15 2018] [error] (EAI 2)Name or service not known: Failed to resolve server
name for 10.1.1.1 (check DNS) -- or specify an explicit ServerName
[ OK ]
```

原因: DNS无法解析

解决: 修改hosts文件, 将2个IP地址和主机名对应起来

```
[root@web conf]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.1.1.2    web.itcast.cc  web
192.168.159.135  web.itcast.cc  web
```

如果有以下错误:

```
[root@web-server conf]# service httpd restart
Stopping httpd:                                     [ OK ]
Starting httpd: httpd: apr_sockaddr_info_get() failed for app1-server
httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1
for ServerName
[ OK ]
```

原因: 主机名不是规范的主机名, 没有指定解析

解决:



修改配置文件，将以下行的#取掉

```
ServerName www.example.com:80
```

## 2. 基于端口的虚拟主机

环境：

1台服务，1个IP地址

思路：

80

10086

http://10.1.1.1:80

http://10.1.1.1:10086

步骤：

1) 修改配置文件

```
Listen 80
```

```
Listen 10086
```

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

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

```
    DocumentRoot /www/web1
```

```
    #ServerName dummy-host.example.com
```

```
    ErrorLog logs/80-error_log
```

```
    CustomLog logs/80-access_log common
```

```
</VirtualHost>
```

```
<VirtualHost *:10086>
```

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

```
    DocumentRoot /www/web2
```

```
    #ServerName dummy-host.example.com
```

```
    ErrorLog logs/10086-error_log
```

```
    CustomLog logs/10086-access_log common
```

```
</VirtualHost>
```

启动服务

```
[root@web-server conf]# netstat -nlt|grep 80
```

```
tcp        0      0 :::80                :::*                  LISTEN
        6708/httpd
```

```
[root@web-server conf]# netstat -nlt|grep 10086
```

```
tcp        0      0 :::10086             :::*                  LISTEN
        6708/httpd
```

## 3. 基于域名的虚拟主机（重点掌握）

环境：只有一台web服务器主机

```
www.myblog.cc          this is myblog.cc test page
```

```
www.abc.net            this is abc.net test page
```

思路：

1. 搭建DNS服务器或者客户端配置hosts文件

2. 在web-server端搭建2个web页面（通过基于域名虚拟主机方式）

步骤：

1. 搭建DNS服务器 10.1.1.250

参考：

```
[root@dns-server named]# cat myblog.cc.zone
```

```
$TTL 1D
```

```
@ IN SOA @ rname.invalid. (  
    0 ; serial  
    1D ; refresh  
    1H ; retry  
    1W ; expire  
    3H ) ; minimum
```

```
NS dns.myblog.cc.
```

```
dns A 10.1.1.250
```

```
www A 10.1.1.1
```

```
[root@dns-server named]# cat abc.net.zone
```

```
$TTL 1D
```

```
@ IN SOA @ rname.invalid. (  
    2018090308 ; serial  
    1M ; refresh  
    1M ; retry  
    1M ; expire  
    3M ) ; minimum
```

```
NS dns.abc.net.
```

```
dns A 10.1.1.250
```

```
www A 10.1.1.1
```

### 测试验证:

```
[root@dns-server named]# service named restart
```

```
Stopping named: [ OK ]
```

```
Starting named: [ OK ]
```

```
[root@dns-server named]# cat /etc/resolv.conf
```

```
nameserver 10.1.1.250
```

```
[root@dns-server named]# nslookup www.myblog.cc
```

```
Server: 10.1.1.250
```

```
Address: 10.1.1.250#53
```

```
Name: www.myblog.cc
```

```
Address: 10.1.1.1
```

```
[root@dns-server named]# nslookup www.abc.net
```

```
Server: 10.1.1.250
```

```
Address: 10.1.1.250#53
```

```
Name: www.abc.net
```

```
Address: 10.1.1.1
```

## 2. 在web-server上发布2个网站

```
[root@web-server web1]# echo "this is myblog.cc test page" > index.html
```

```
[root@web-server web2]# echo "this is abc.net test page" > index.html
```

```
NameVirtualHost *:80
```

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

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

```
DocumentRoot /www/web1
```

```
ServerName www.myblog.cc
```

```
ErrorLog logs/myblog-error_log
CustomLog logs/myblog-access_log common
</VirtualHost>
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot /www/web2
    ServerName www.abc.net
    ErrorLog logs/abc-error_log
    CustomLog logs/abc-access_log common
</VirtualHost>
```

### 重启服务测试验证

注意：如果是windows平台测试的话，修改你本机的hosts文件即可

### 课堂练习：

www.zhangsan.org	this is zhangsan.org test page
bbs.momowu.cc	this is bbs.momowu.cc test page

### 需求3：

对于第一个静态网站，需要特定的用户和密码才能访问，并且拒绝10.1.1.0/24网段访问除了10.1.1.254

#### 1. 创建密码文件并将用户加入其中

```
[root@web-server conf]# htpasswd -cm /etc/httpd/conf/.pass user01
New password:
Re-type new password:
Adding password for user user01
[root@web-server conf]# cat .pass
user01:$apr1$r5TsZKph$kxdIssOcWyR39jMW0bnbW.
[root@web-server conf]# htpasswd -bm /etc/httpd/conf/.pass user02 123
Adding password for user user02
[root@web-server conf]# cat .pass
user01:$apr1$r5TsZKph$kxdIssOcWyR39jMW0bnbW.
user02:$apr1$zzoCZcb.$zTaH7vfVP1QnfCPcVP0Xb.
```

#### 2. 修改配置文件开启认证的功能

```
<Directory "/www/web1">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
    AuthType Basic
    AuthName "Input your name and password:"
    AuthBasicProvider file
    AuthUserFile /etc/httpd/conf/.pass
    Require user user01
</Directory>
```

#### 3. 启动服务测试验证

```
[root@web-server conf]# service httpd restart
Stopping httpd: [ OK ]
Starting httpd: [Thu Sep 06 16:37:27 2018] [warn] _default_ VirtualHost overlap on port 80, the
first has precedence
[ OK ]
```

解决：开启如下开关

NameVirtualHost \*:80

如何允许多个人来访问？

方法1：将用户加入到组中，然后再允许组访问

1. 创建组文件

```
[root@web-server conf]# vim /etc/httpd/conf/groups
admin:user01 stu1 stu2
```

2. 修改配置文件

```
<Directory "/www/web1">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
AuthType Basic //启用认证
AuthName "Input your name and password:" //输入提示信息
AuthBasicProvider file
AuthUserFile /etc/httpd/conf/.pass //指定密码文件
AuthGroupFile /etc/httpd/conf/groups //指定组文件
Require group admin //允许admin的人来访问
</Directory>
```

注意：

属于admin组的成员必须在密码文件中存在。

方法2：允许密码文件里的所有用户访问

```
<Directory "/www/web1">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
AuthType Basic
AuthName "Input your name and password:"
AuthBasicProvider file
AuthUserFile /etc/httpd/conf/.pass
#AuthGroupFile /etc/httpd/conf/groups
#Require group admin
#Require user user02
Require valid-user //允许多个用户
Order deny,allow
Deny from 10.1.1.0/24
Allow from 10.1.1.254
</Directory>
```

网络访问控制：

RHEL6: http 2.2

Order allow,deny 如果allow和deny冲突, deny为准

Order deny,allow 如果allow和deny冲突, allow为准

#### 1、禁止部分ip不能访问网站

Order allow,deny

Allow from all

Deny from 192.168.0.254 172.16.2.10

#### 2、针对某个网段

Order allow,deny

Allow from all

Deny from 192.168.0.0/255.255.255.0

#### 3、针对域名

Order allow, deny

Allow from all

Deny from node1.itcast.cc \*.example.com

#### 4、拒绝大部分, 只允许某个ip

Order deny, allow

Deny from all

Allow from 192.168.0.254

#### 需求3: 只拒绝10.1.1.2主机访问我的网站

Order allow, deny

Allow from all

Deny from 10.1.1.2

#### 引申扩展:

RHEL7: http 2.4+

##### 案例1: 允许所有主机访问

```
<Directory "/var/www/html">
    AllowOverride None
    Require all granted
</Directory>
```

AllowOverride All            允许子目中的 .htaccess 中的设置覆盖当前设置  
AllowOverride None        不允许子目中的 .htaccess 中的设置覆盖当前设置

##### 案例2: 只允许网段192.168.0.0/24和192.168.10.254/24访问

```
<Directory "/var/www/html">
    AllowOverride None
    Require ip 192.168.0.0/24
    Require ip 192.168.10.254
</Directory>
```

##### 案例3: 只拒绝某些主机访问

```
<Directory "/var/www/html">
    AllowOverride None
    <RequireAll>
        Require not ip 10.1.1.254
        Require all granted
    </RequireAll>
```

```
</RequireAll>
</Directory>
```

## 项目实战

### 一、环境准备

#### 1. 需要准备的软件包

```
apr-1.5.2.tar.bz2
httpd-2.4.12.tar.bz2
php-5.6.23.tar.xz
wordpress-4.7.3-zh_CN.tar.gz
apr-util-1.5.4.tar.bz2
mysql-5.6.31.tar.gz
phpMyAdmin-4.4.11-all-languages.zip
```

#### 2. 安装前环境准备

##### 说明：

- 如果你在一台机器启动两个 apache 和 mysql，那么很可能造成一定的冲突，所以为了减少不必要的麻烦，首先停止卸载它们。
- 由于整个环境会涉及比较多的依赖关系包，所以我们先将一些依赖包装上。（根据不同的环境可能不仅限于这些包）

##### 1) 清空环境

```
[root@web soft]# netstat -nlt|grep 80
tcp        0      0 :::80          :::*           LISTEN
1651/httpd
```

```
[root@web soft]# service httpd stop
Stopping httpd:                                [ OK ]
```

##### 2) 关闭防火墙和selinux

```
[root@web soft]# service iptables stop
[root@web soft]# getenforce
Disabled
```

```
[root@web soft]# cd
```

##### 3) 临时更改主机名并且绑定到hosts文件中

```
[root@web ~]# hostname lamp.itcast.cc
[root@web ~]# vim /etc/hosts
[root@web ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.1.1.2    web.itcast.cc  web
192.168.159.135  web.itcast.cc  web
10.1.1.2    lamp.itcast.cc  lamp
```

##### 4) 安装相应的软件包

```
# yum groupinstall "Development tools" -y
# yum groupinstall "Desktop Platform Development" -y 桌面开发工具包（图形化相关包）
# yum install cmake pcre-devel ncurses-devel openssl-devel libcurl-devel -y
```

准备相应的软件：将所需要的软件包放到虚拟机里，我放到了/soft目录里

### 3. 编译方式

Linux+Apache(2.4)+Mysql(5.6.31)+Php(5.6.23)

Apache→MySQL→PHP 或者 MySQL→Apache→PHP

#### 说明：

1. apache必须要先于php安装。
2. apache和mysql之间并没有直接先后顺序的依赖,所以谁先谁后无所谓。
3. 在php-5.3版本前, mysql必须先于php的编译;因为php需要实现连接数据库的功能, 它通过mysql的接口才能编译出该功能;
4. 在php-5.3版本或者之后, php已经集成了一套连接mysql数据的代码, 并不依赖mysql的接口, 这个时候, mysql和php的编译顺序也就无所谓了。

## 二、编译安装MySQL

版本: mysql-5.6.31.tar.gz

需求:

1. 安装目录 /usr/local/mysql
2. 数据目录 /usr/local/mysql/data
3. 端口 默认3306
4. socket文件 /usr/local/mysql/mysql.sock

安装:

1. 官方网站下载相应的软件包

mysql-5.6.31.tar.gz

2. 解压软件包

```
[root@lamp soft]# tar xf mysql-5.6.31.tar.gz
```

3. 安装

- 0) 创建相应的用户

```
# useradd mysql -r -s /sbin/nologin
```

-r:创建系统用户

-s:指定默认的shell (非交互式)

说明: 如果有mysql用户, 可以省略。

```
[root@lamp mysql]# id mysql
```

```
uid=27(mysql) gid=27(mysql) groups=27(mysql)
```

- 1) 配置

根据需求配置:

```
[root@lamp soft]# cd mysql-5.6.31
```

```
[root@lamp mysql-5.6.31]# vim cmake.sh
```

```
cmake . \
```

```
-DCMAKE_INSTALL_PREFIX=/usr/local/mysql/ \
```

```
-DMYSQL_DATADIR=/usr/local/mysql/data \
```

```
-DENABLED_LOCAL_INFILE=1 \
-DWITH_INNOBASE_STORAGE_ENGINE=1 \
-DSYSCONFDIR=/usr/local/mysql/etc \
-DMYSQL_UNIX_ADDR=/usr/local/mysql/mysql.sock \
-DMYSQL_TCP_PORT=3306 \
-DDEFAULT_CHARSET=utf8 \
-DDEFAULT_COLLATION=utf8_general_ci \
-DWITH_EXTRA_CHARSETS=all \
-DMYSQL_USER=mysql
```

```
[root@lamp mysql-5.6.31]# chmod +x cmake.sh
```

```
[root@lamp mysql-5.6.31]# ./cmake.sh
```

编译:

```
make
```

安装:

```
make install
```

参数说明:

```
cmake . \
```

```
-DCMAKE_INSTALL_PREFIX=/usr/local/mysql/ \
```

```
-DMYSQL_DATADIR=/usr/local/mysql/data \
```

```
-DENABLED_LOCAL_INFILE=1 \
```

```
-DWITH_INNOBASE_STORAGE_ENGINE=1 \
```

```
-DSYSCONFDIR=/usr/local/mysql/etc \
```

```
-DMYSQL_UNIX_ADDR=/usr/local/mysql/mysql.sock \
```

```
-DMYSQL_TCP_PORT=3306 \
```

```
-DDEFAULT_CHARSET=utf8 \
```

```
-DDEFAULT_COLLATION=utf8_general_ci \
```

```
-DWITH_EXTRA_CHARSETS=all \
```

```
-DMYSQL_USER=mysql
```

安装路径

数据目录

开启加载外部文件功能; 1开启, 0关闭

将InnoDB存储引擎静态编译到服务器

配置文件路径

socket文件路径

端口

字符集

字符校验规则

扩展字符集

用户身份mysql

后续配置:

1. 更改目录权限让mysql用户可读写

```
# chown -R mysql:mysql /usr/local/mysql/
```

2. 初始化数据库

1) 清空环境

```
# rm -f /etc/my.cnf
```

说明: 如果/etc/my.cnf不存在可以省略此步

```
# ps -ef|grep mysqld 确保当前没有其他mysql程序运行, 如果有请kill掉。
```

```
# pkill -9 mysqld
```

2) 初始化数据库

```
# cd /usr/local/mysql/
```

```
[root@lamp mysql]# ./scripts/mysql_install_db --user=mysql
```

说明: 看到2个ok说明初始化完毕

3) 验证数据目录里是否有文件

```
[root@lamp mysql]# ll data/
```



```
total 110604
-rw-rw---- 1 mysql mysql 12582912 Nov 14 11:28 ibdata1
-rw-rw---- 1 mysql mysql 50331648 Nov 14 11:28 ib_logfile0
-rw-rw---- 1 mysql mysql 50331648 Nov 14 11:28 ib_logfile1
drwx----- 2 mysql mysql    4096 Nov 14 11:28 mysql
drwx----- 2 mysql mysql    4096 Nov 14 11:28 performance_schema
drwxr-xr-x 2 mysql mysql    4096 Nov 14 11:21 test
```

### 3. 拷贝mysql启动脚本到/etc/init.d/目录

```
[root@lamp mysql]# pwd
/usr/local/mysql
[root@lamp mysql]# ls
bin COPYING data docs include lib man my.cnf mysql-test README scripts share sql-
bench support-files
[root@lamp mysql]# cp support-files/mysql.server /etc/init.d/mysql
```

### 4. 启动mysql数据库

#### 1) 启动前确认3306端口没有被占用

```
[root@lamp data]# netstat -nlt|grep 3306
```

说明：如果3306端口被占用，请kill掉之前的mysqld程序，如果没有被占用，此步省略。

```
# pkill -9 mysqld
```

#### 2) 启动mysql数据库

```
[root@lamp mysql]# service mysql start
Starting MySQL.. SUCCESS!
```

### 5. 设置数据库管理员root用户密码

```
[root@lamp mysql]# /usr/local/mysql/bin/mysqladmin -u root password '123'
Warning: Using a password on the command line interface can be insecure.
```

### 6. 测试登录

```
[root@lamp mysql]# mysql -p123
Warning: Using a password on the command line interface can be insecure.
ERROR 2002 (HY000): Can't connect to local MySQL server through socket
'/var/lib/mysql/mysql.sock' (111)
```

解决：

```
[root@lamp mysql]# export PATH=/usr/local/mysql/bin/:$PATH
[root@lamp mysql]# echo $PATH
/usr/local/mysql/bin:/usr/lib64/qt-
3.3/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
```

永久解决：

```
[root@lamp mysql]# vim /etc/profile
```

在文件最后追加以下内容：

```
export PATH=/usr/local/mysql/bin/:$PATH
```

```
[root@lamp mysql]# source /etc/profile    立即生效
```

或者退出去重新登录生效

```
[root@lamp mysql]# mysql -p123
Warning: Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.6.31 Source distribution
```

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

### 三、编译安装Apache

#### 1. 编译安装依赖包apr

安装apr软件:

```
# tar xf apr-1.5.2.tar.bz2
```

```
# cd apr-1.5.2
```

```
# ./configure
```

```
# make
```

```
# make install
```

安装apr-util软件:

```
# tar xf apr-util-1.5.4.tar.bz2
```

```
# cd apr-util-1.5.4/
```

```
# ./configure --with-apr=/usr/local/apr/bin/apr-1-config 指定软件apr的路径
```

```
# make
```

```
# make install
```

思考:

**一个软件的库文件是有可能被其它软件所调用, 那么其它软件能否找到你的库文件呢?**

- 一般来说, 库文件安装到/lib,/lib64,/usr/lib/,/usr/lib64等, 都可以找得到; 但是如果一个软件A把库文件安装到/usr/local/A/lib下, 就要把这个路径加入到ldconfig命令可以找到的路径列表里面去, 别人才能找到。
- ldconfig是一个动态链接库管理命令; 主要用途是在默认搜索目录 (/lib,/lib64,/usr/lib/,/usr/lib64) 以及动态库配置文件/etc/ld.so.conf中所列的目录中搜索出可共享的动态链接库。

**问题:** 怎样将库文件的指定安装路径加入到ldconfig命令的搜索列表里?

方法1: 在/etc/ld.so.conf这个主配置文件里加上一行, 写上让别人要查找库文件的路径

```
echo "/usr/local/apr/lib/" >> /etc/ld.so.conf
```

方法2: 在/etc/ld.so.conf.d/目录下创建一个\*.conf结尾的文件, 里面加入该路径即可

```
# echo /usr/local/apr/lib/ > /etc/ld.so.conf.d/lamp.conf
```

```
# ldconfig 上面加入路径后, 就使用此命令让其生效
```

#### 2. 编译安装httpd

版本: httpd-2.4.12.tar.bz2

1. 下载

2. 解压

```
[root@lamp soft]# tar xf httpd-2.4.12.tar.bz2
```

### 3. 安装 (解压目录)

进入到解压目录里进行安装:

```
[root@lamp soft]# cd httpd-2.4.12
```

配置:

```
[root@lamp httpd-2.4.12]# vim apache.sh
./configure \
--enable-modules=all \
--enable-mods-shared=all \
--enable-so \
--enable-rewrite \
--with-pcre \
--enable-ssl \
--with-mpm=prefork \
--with-apr=/usr/local/apr/bin/apr-1-config \
--with-apr-util=/usr/local/apr/bin/apu-1-config
```

```
[root@lamp httpd-2.4.12]# chmod +x apache.sh
```

```
[root@lamp httpd-2.4.12]# ./apache.sh
```

说明: 如果之前相关的软件包都已安装, 基本不会有问题。

配置错误:

```
[root@lamp httpd-2.4.12]# ./apache.sh
checking how to run the C preprocessor... gcc -E
checking for gcc option to accept ISO C99... -std=gnu99
checking for pcre-config... false
configure: error: pcre-config for libpcre not found. PCRE is required and available from
http://pcre.org/
```

原因: 缺少软件包 pcre相关

解决:

```
[root@lamp httpd-2.4.12]# yum -y install pcre-devel
```

安装好依赖包后, 继续配置, 没有error说明配置完成

```
[root@lamp httpd-2.4.12]# ./apache.sh
```

...

```
checking for OpenSSL... checking for user-provided OpenSSL base directory... none
checking for OpenSSL version >= 0.9.8a... FAILED
configure: WARNING: OpenSSL version is too old
no
checking whether to enable mod_ssl... configure: error: mod_ssl has been requested but can not
be built due to prerequisite failures
```

解决:

```
yum -y install openssl-devel
```

编译:

```
make
```

安装:

```
make install
```

确认是否成功安装apache:

```
[root@lamp httpd-2.4.12]# ls /usr/local/apache2/  
bin build cgi-bin conf error htdocs icons include logs man manual modules
```

注意:

确认这个目录产生后, 说明apache编译安装成功

配置参数说明:

```
# ./configure  
--enable-modules=all          加载所有支持模块  
--enable-mods-shared=all      共享方式加载大部分常用的模块  
--enable-so                   启动动态模块加载功能  
--enable-rewrite              启用url地址重写功能  
--enable-ssl                  编译ssl模块, 支持https  
--with-pcre                   支持正则表达式  
--with-mpm=prefork            插入式并行处理模块, 称为多路处理模块, Prefork 是类UNIX平台上默认的MPM  
(1) prefork  
    多进程模型, 每个进程响应一个请求  
(2) worker  
    多进程多线程模型, 每个线程处理一个用户请求  
(3) event(最优)  
    事件驱动模型, 多进程模型, 每个进程响应多个请求  
--with-apr=/usr/local/apr/bin/apr-1-config 指定依赖软件apr路径  
--with-apr-util=/usr/local/apr/bin/apu-1-config
```

#### 四、编译安装php

版本: php-5.6.23.tar.xz

1. 下载软件

2. 解压

```
[root@lamp soft]# tar xf php-5.6.23.tar.xz
```

3. 进入到解压的目录里

```
[root@lamp soft]# cd php-5.6.23
```

1) 配置

```
[root@lamp php-5.6.23]# vim php.sh
```

```
./configure \  
--with-apxs2=/usr/local/apache2/bin/apxs \  
--with-mysql=/usr/local/mysql/ \  
--with-mysqli=/usr/local/mysql/bin/mysq_config \  
--with-pdo-mysql=/usr/local/mysql \  
--with-zlib \  
--with-zlib-dir=/usr/local/mysql/zlib \  
--with-curl \  
--enable-zip \  
--with-gd \  
--with-freetype-dir \  
--with-jpeg-dir \  
--with-png-dir \  
--enable-sockets \  
--with-xmllrpc \  
--enable-soap \  
--enable-opcache \  

```

```
--enable-mbstring \  
--enable-mbregex \  
--enable-pcntl \  
--enable-shmop \  
--enable-sysvmsg \  
--enable-sysvsem \  
--enable-sysvshm \  
--enable-calendar \  
--enable-bcmath
```

#### 配置错误:

```
checking for cURL support... yes  
checking for cURL in default path... not found  
configure: error: Please reinstall the libcurl distribution -  
easy.h should be in <curl-dir>/include/curl/
```

#### 解决:

```
yum -y install libcurl-devel
```

#### 2) 编译

```
make //make成功后, 会显示让你make test,不用做
```

#### 3) 安装

```
make install
```

#### 确认php成功安装:

```
[root@lamp php-5.6.23]# ls /usr/local/apache2/modules/libphp5.so
```

注意: 确认有这个libphp5.so模块文件, 就表示编译php成功

### php编译相关参数介绍:

```
--with-apxs2=/usr/local/apache2/bin/apxs //指定apxs路径  
apxs是一个为Apache HTTP服务器编译和安装扩展模块的工具,with-apxs2构建apache2.0版本的处理模块  
--with-config-file-path 和 --with-config-file-scan-dir //在指定 php 配置文件的路径  
--with-mysql 和 --with-mysqli //在指定你的 mysql 的位置和它的相关工具  
--with-iconv-dir  
--with-freetype-dir  
--with-jpeg-dir  
--with-png-dir  
--with-gd  
--with-zlib  
--with-libxml-dir //这些都是在启用对某种文件的支持  
--with-curl 和 --with-curlwrappers  
用于支持 curl 函数, 此函数允许你用不同的协议连接和沟通不同的服务器  
--with-openssl,--with-mhash,--with-mcrypt  
这都是和加密有关的参数, 启用它们是为了让php可以更好的支持各种加密。  
--enable-bcmath //高精度数学运算组件。  
--enable-shmop和 --enable-sysvsem //使得你的PHP系统可以处理相关的IPC函数。  
IPC是一个Unix标准通讯机制, 它提供了使得在同一台主机不同进程之间可以互相通讯的方法。  
--enable-inline-optimization //栈堆指针和优化线程。  
--enable-pcntl //多线程优化。  
  
with-apxs2 调用apache加载模块支持PHP  
gd 画图库
```

libiconv	字符变换转换
libmcrypt	字符加密
mcrypt	字符加密
mhash	哈希运算

## 五、后续配置

### 1. 配置apache和php联系

#### 1. 修改apache主配置文件

```
# vim /usr/local/apache2/conf/httpd.conf
```

//配置语言支持

```
LoadModule negotiation_module modules/mod_negotiation.so 此模块打开注释
```

```
Include conf/extra/httpd-languages.conf 打开此选项，扩展配置文件就生效了
```

```
...
```

//对虚拟主机的支持，后续再打开

```
#Include conf/extra/httpd-vhosts.conf
```

//加载php模块解析php页面，添加两行，告诉httpd把.php文件交给模块去编译

```
LoadModule php5_module modules/libphp5.so 找到这一句，在这句下面加上两句
```

//添加以下两行意思是以.php结尾的文件都认为是php程序文件，注意两句话的.php前面都是有一个空格的

```
AddHandler php5-script .php
```

```
AddType text/html .php
```

//默认主页加上index.php，并放在index.html前，支持php的首页文件

```
<IfModule dir_module>
```

```
    DirectoryIndex index.php index.html
```

```
</IfModule>
```

注意：默认的网站目录是：/usr/local/apache2/htdocs/

#### 2. 修改apache的子配置文件，优先支持中文

```
# vim /usr/local/apache2/conf/extra/httpd-languages.conf
```

```
DefaultLanguage zh-CN 打开注释，默认语言集改为中文
```

```
LanguagePriority zh-CN en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl pt pt-BR ru
```

```
sv tr zh-TW 语言集优先集，把zh-CN 写到前面
```

### 2. php支持连接本地数据库

#### 说明：

本地数据库一般是通过socket文件连接，而本地数据库的socket文件如果不在默认路径，就必须告诉php从哪里读取socket文件。

```
# cp /soft/php-5.6.23/php.ini-production /usr/local/lib/php.ini
vim /usr/local/lib/php.ini
.....
[MySQL]
mysql.default_port = 3306      改成对应的mysql的端口
mysql.default_socket = /usr/local/mysql/mysql.sock    对应的socket文件地址

[MySQLi]
mysqli.default_port = 3306
mysqli.default_socket = /usr/local/mysql/mysql.sock
```

### 3. 测试是否支持php

```
# cd /usr/local/apache2/htdocs
# rm -f index.html
# vim index.php
<?php
    phpinfo();
?>
```

### 4. 启动apache

为了方便启动，做成脚本启动脚本：

```
# cp /usr/local/apache2/bin/apachectl /etc/init.d/apache
# netstat -nlt|grep 80    确保没有80端口监听
# lsof -i :80
# service apache start
```

可能遇到的错误：

```
# /usr/local/apache2/bin/apachectl start
(98)Address already in use: make_sock: could not bind to address [::]:80
(98)Address already in use: make_sock: could not bind to address 0.0.0.0:80
no listening sockets available, shutting down
Unable to open logs
```

注意：

上面的报错表示80被占用，可以先停掉rpm版（或者改端口）httpd再启动

### 5.测试环境

浏览器里输入：`http://10.1.1.2` //看到php的测试页表示ok

到此，lamp的编译安装和基本配置完毕，下面就可以根据需求部署web应用了。

## 六、部署web应用

- 通过虚拟主机将web应用发布出去

```
1. 创建相应的数据目录来保存网站文件
[root@lamp ~]# mkdir /www/myblog -p
```

## 2. 拷贝网站文件到指定目录

```
[root@lamp soft]# tar -xf wordpress-4.7.3-zh_CN.tar.gz
[root@lamp soft]# cp -a wordpress/* /www/myblog/
```

## 3. 更改数据文件权限

```
[root@lamp soft]# chown -R daemon.daemon /www/myblog/
```

## 4. 配置虚拟主 (网站上线发布)

### 1) 主配置文件打开加载虚拟主机配置

```
[root@lamp ~]# cd /usr/local/apache2/conf/
[root@lamp conf]# vim httpd.conf
# Virtual hosts
Include conf/extra/httpd-vhosts.conf      //打开加载虚拟主机配置文件
```

### 2) 子配置文件目录里修改虚拟主机配置文件

```
[root@lamp extra]# pwd
/usr/local/apache2/conf/extra
[root@lamp extra]# vim httpd-vhosts.conf
```

追加如下内容:

```
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host2.example.com
    DocumentRoot "/www/myblog"
    ServerName bbs.momowu.cc
    ErrorLog "logs/bbs-error_log"
    CustomLog "logs/bbs-access_log" common
</VirtualHost>
```

## 5. 网站连接后台数据库配置

### 1) 在mysql数据库里创建myblog库

```
[root@lamp myblog]# mysql -p123
...
mysql> create database myblog default charset utf8; //创建myblog库来存放wordpress的数据
Query OK, 1 row affected (0.46 sec)
```

### 2) 指定网站连接数据库的用户名密码

```
[root@lamp soft]# cd /www/myblog/
[root@lamp myblog]# cp wp-config-sample.php wp-config.php
[root@lamp myblog]# vim wp-config.php
...
// ** MySQL 设置 - 具体信息来自您正在使用的主机 ** //
/** WordPress数据库的名称 */
define('DB_NAME', 'myblog');

/** MySQL数据库用户名 */
define('DB_USER', 'root');

/** MySQL数据库密码 */
define('DB_PASSWORD', '123');

/** MySQL主机 */
define('DB_HOST', '127.0.0.1');
```



```
/** 创建数据表时默认的文字编码 */
```

```
define('DB_CHARSET', 'utf8');
```

```
/** 数据库整理类型。如不确定请勿更改 */
```

```
define('DB_COLLATE', '');
```

```
...
```

补充: phpMyadmin网站相关配置文件

mysql的web管理网站phpMyAdmin相关文件

```
# cp config.sample.inc.php config.inc.php
```

```
# vim config.inc.php
```

```
....
```

```
/* Authentication type */
```

```
$cfg['Servers'][$i]['auth_type'] = 'cookie';
```

```
/* Server parameters */
```

```
$cfg['Servers'][$i]['host'] = '127.0.0.1'; ——>将localhost改为127.0.0.1
```

```
$cfg['Servers'][$i]['connect_type'] = 'tcp';
```

```
$cfg['Servers'][$i]['compress'] = false;
```

```
$cfg['Servers'][$i]['AllowNoPassword'] = false;
```

```
....
```

问题: 在重启apache的时候报如下错误:

```
[root@lamp myblog]# service apache restart
```

```
AH00557: httpd: apr_sockaddr_info_get() failed for LAMP
```

```
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using  
127.0.0.1. Set the 'ServerName' directive globally to suppress this message
```

```
httpd (no pid file) not running
```

原因: 自己的主机名和IP地址没有绑定在hosts文件中

## 七、测试验证

1. Linux客户端通过Firefox浏览器测试

1) 保证DNS能解析到www.myblog.net域名

方法1: 配置DNS

方法2: 更改hosts文件增加如下内容

```
10.1.1.2 www.myblog.net
```

2) firefox http://www.myblog.net

报错信息:

```
Forbidden
```

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

原因: 没有给网站目录授权

解决: 在web服务器端修改主配置文件授权

```
[root@lamp conf]# pwd
```

```
/usr/local/apache2/conf
```

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

```
....
```

```
<Directory />
    AllowOverride none
    Require all denied      默认拒绝所有, 改为Require all granted
</Directory>
重启apache测试验证
```

## 项目总结

### 1. 源码编译总结

- 配置阶段

```
# 通过指定参数, 来确认你想要安装的软件安装在哪里, 加上哪些功能和去掉哪些功能
./configure 或者 cmake
# 如果这一步报错, 基本都是缺少依赖包, 解决办法:
1> 使用yum去安装, 一般来说, rhel/centos做为一个成熟的linux操作系统, 常见的底层依赖包都自带了, 所以安装下面这两个组, 一般都会有你所需要的依赖包。
# yum groupinstall "Development tools" -y
# yum groupinstall "Desktop Platform Development" -y
2> 如果缺少依赖包在rhel/centos的yum源里找不到, 则上网下载第三方的软件, 先编译第三方软件, 再编译本软件
```

关于 `./configure` 参数选择的基本方法:

```
./configure --help 查看所有的编译参数
第一个重要参数
--prefix= 此参数指定安装目录(一般安装到/usr/local/或者/usr/local/软件名下)
注意: 如果指定了新的路径, 注意后续环境变量的修改

第二类重要参数:
--enable-xxx 打开一个功能(默认是关闭的)
--disable-xxx 关闭一个功能(默认是打开的)

第三类参数:
--with-xxx=DIR/file 指定一个目录或文件, 调用此目录或文件的功能
```

- 编译阶段

```
make
相当于你是根据你上一步定义好的文件 (Makefile), 把这个软件给做出来(这一步一般很少出错, 如果出错, 问题都比较麻烦。可能是一些兼容性的问题等等, 你可以尝试上网查询解决方法, 如果查不到, 只能换个环境或者换个软件版本或者换些编译参数重新编译)
```

- 安装阶段

```
make install
把做好的软件, 安装到你第一步所指定的安装目录里(这一步几乎不会出错的), 如果目录不存在会自动创建
```

## 2. 需求分析

所有复杂的任务都有一个简单的解决办法，理清思路，结合业务场景分析，测试，验证。