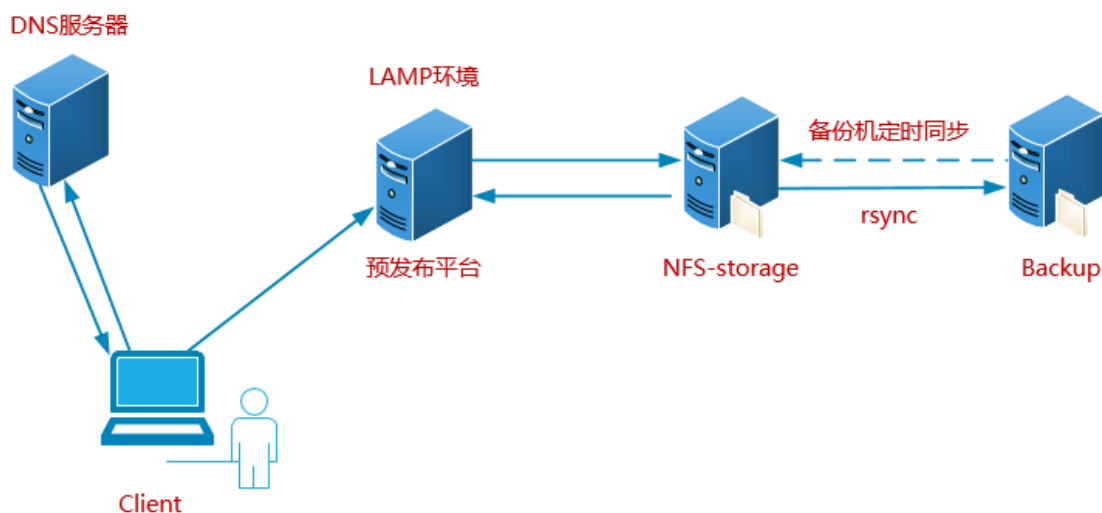


项目背景

初创公司，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（生产发生改变立马同步到预发布平台）

项目实战

一、环境准备

1. 需要准备的软件包

```
LAMP
Apache-->httpd
MySQL-->5.6.31
PHP-->7.2.17

apr-1.5.2.tar.bz2
apr-util-1.5.4.tar.bz2
httpd-2.4.37.tar.bz2
mysql-5.6.43
php-7.2.17.tar.xz

wordpress-4.7.3-zh_CN.tar.gz
```

2. 安装前环境准备

(一) 环境说明

IP	部署软件	主机名	角色
10.1.1.2	Apache(httpd)+MySQL+PHP	lamp.heima.cc	web服务器
10.1.1.3	Bind	dns-server.heima.cc	DNS服务器
10.1.1.1	elinks (用于测试)	client.heima.cc	客户端

(二) 环境准备

① 停止原来web服务

```
[root@lamp ~]# netstat -nlt|grep httpd
[root@lamp ~]# service httpd stop
```

② 关闭防火墙和selinux

```
[root@lamp ~]# service iptables stop
[root@lamp ~]# setenforce 0
```

③ 绑定IP和主机名到hosts

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

④ 安装所需依赖包

说明：每个人环境不同，可能所需软包不一样，灵活判断！

开发工具包

```
[root@lamp ~]# yum groupinstall "Development tools" -y
```

桌面开发工具包（图形化相关包）

```
[root@lamp ~]# yum groupinstall "Desktop Platform Development" -y
```

其他依赖包

```
[root@lamp ~]# yum install cmake pcre-devel ncurses-devel openssl-devel libcurl-devel
```

(三) 编译方式说明

Linux+Apache(2.4)+MySQL(5.6.31)+PHP(7.2.17)

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

1、安装需求

软件版本	安装目录	数据目录	端口
mysql-5.6.31	/usr/local/mysql	/usr/local/mysql/data	3306

2、安装步骤

(一) 创建mysql用户

```
[root@lamp ~]# useradd -r -s /sbin/nologin mysql
```

(二) 解压软件并==进入解压目录==

```
[root@lamp lamp]# pwd
/lamp
[root@lamp lamp]# tar -xf mysql-5.6.31.tar.gz
[root@lamp lamp]# cd mysql-5.6.31
[root@lamp mysql-5.6.31]# pwd
/lamp/mysql-5.6.31
```

(三) 根据需求配置

```
# vim cmake.sh
cmake . \
-DMAKE_INSTALL_PREFIX=/usr/local/mysql/ \
-DMYSQL_DATADIR=/usr/local/mysql/data \
-DENABLED_LOCAL_INFILE=1 \
-DWITH_INNOBASE_STORAGE_ENGINE=1 \
-DMYSQL_TCP_PORT=3306 \
-DDEFAULT_CHARSET=utf8mb4 \
-DDEFAULT_COLLATION=utf8mb4_general_ci \
-DWITH_EXTRA_CHARSETS=all \
-DMYSQL_USER=mysql

# chmod +x cmake.sh
# ./cmake.sh
编译:
根据上一步产生的Makefile进行编译-->mysqld(当前操作系统平台)
```

参数说明:

```
cmake . \
-DMAKE_INSTALL_PREFIX=/usr/local/mysql/ \
-DMYSQL_DATADIR=/usr/local/mysql/data \
-DENABLED_LOCAL_INFILE=1 \
-DWITH_INNOBASE_STORAGE_ENGINE=1 \
-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

-DWITH_INNOBASE_STORAGE_ENGINE=1 \	将InnoDB存储引擎静态编译到服务器
-DMYSQL_TCP_PORT=3306 \	端口
-DDEFAULT_CHARSET=utf8mb4 \	字符集
-DDEFAULT_COLLATION=utf8_general_ci \	字符校验规则
-DWITH_EXTRA_CHARSETS=all \	扩展字符集
-DMYSQL_USER=mysql	用户身份mysql

(四) 编译并安装

```
# make
# make install
```

(五) 后续配置

注意：安装完毕后==一定==要==进入到安装目录里==进行如下操作

① 更改数据目录权限

```
# 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
[root@lamp ~]# 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 拷贝启动脚本

```
[root@lamp mysql]# pwd
/usr/local/mysql
[root@lamp mysql]# cp support-files/mysql.server /etc/init.d/mysql
```

④ 启动数据库

```
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。
# kill -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
-bash: mysql: command not found

原因: 环境变量没有设置
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

再次测试成功:
[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: 5.6.43 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>
```

其他错误:

错误:

```
[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)
```

原因: /etc/my.cnf文件没有删

解决: 删除/etc/my.cnf

三、编译安装Apache

1. 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	安装路径
httpd-2.4.37	/usr/local/

(一) 解压软件并进入解压目录

```
[root@lamp lamp]# tar xf httpd-2.4.37.tar.bz2
[root@lamp lamp]# cd httpd-2.4.37
```

(二) 根据需求配置

```
[root@lamp httpd-2.4.37]# 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。

配置参数说明：

apache默认啥功能都没有，必须通过模块的进行添加！

```
# ./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-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
--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
```

其他错误说明：

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
```

(三) 编译并安装

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

四、编译安装php

1、解压软件并进入解压目录

```
[root@lamp lamp]# tar xf php-7.2.17.tar.xz
```

```
[root@lamp lamp]# cd php-7.2.17
```

2、根据需求配置

```
[root@lamp php-7.2.17]# vim php.sh
```

```
./configure \
```

```
--with-apxs2=/usr/local/apache2/bin/apxs \
```

```
--with-mysql \
```

```
--with-pdo-mysql \
```

```
--with-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
```

配置参数:

```
--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-mysql          php7中已被废弃
--with-mysqli
--with-pdo-mysql
以上三个是php的扩展,用于连接mysql数据库的
--with-iconv-dir
--with-freetype-dir
--with-jpeg-dir
--with-png-dir
--with-gd
--with-zlib
--with-libxml-dir
这些都是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
```

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
```

除此外, 还需要解决libxml2、libjpeg、libpng、freetype依赖问题

```
yum install libxml2-devel -y
yum install libjpeg-devel -y
yum install libpng-devel -y
yum install freetype-devel -y
```

3、编译并安装

编译

```
make
```

说明: make成功后, 会显示让你make test, 不用做

安装

```
make install
```

确认php成功安装:

```
[root@lamp php-7.2.17]# ls /usr/local/apache2/modules/libphp7.so
/usr/local/apache2/modules/libphp7.so
```

注意: 确认有这个libphp7.so模块文件, 就表示编译php成功

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 php7_module modules/libphp7.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 php7-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. 测试是否支持php

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

3. 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，可以去先停掉rpm版（或者改端口）httpd再启动

4. 测试LAMP环境

浏览器里输入：`http://10.1.1.13` //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/blog -p

2. 拷贝网站文件到指定目录
[root@lamp lamp]# tar -xf wordpress-4.7.3-zh_CN.tar.gz
[root@lamp lamp]# cp -a wordpress/* /www/blog/

3. 更改数据文件权限
[root@lamp lamp]# chown -R daemon.daemon /www/blog/

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 lamp]# 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
注释掉默认的虚拟主机配置，增加如下内容：
<VirtualHost *:80>
    DocumentRoot "/www/blog"
    ServerName www.blog.com
    ErrorLog "logs/blog-error_log"
    CustomLog "logs/blog-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blog库
[root@lamp myblog]# mysql -p123
. . .
mysql> create database blog default charset utf8; //创建myblog库来存放wordpress的数据
Query OK, 1 row affected (0.46 sec)

2) 浏览器安装博客

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 blog]# 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.blog.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.blog.net
2) firefox http://www.blog.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。

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）==