# 生产环境下的LAMP环境搭建

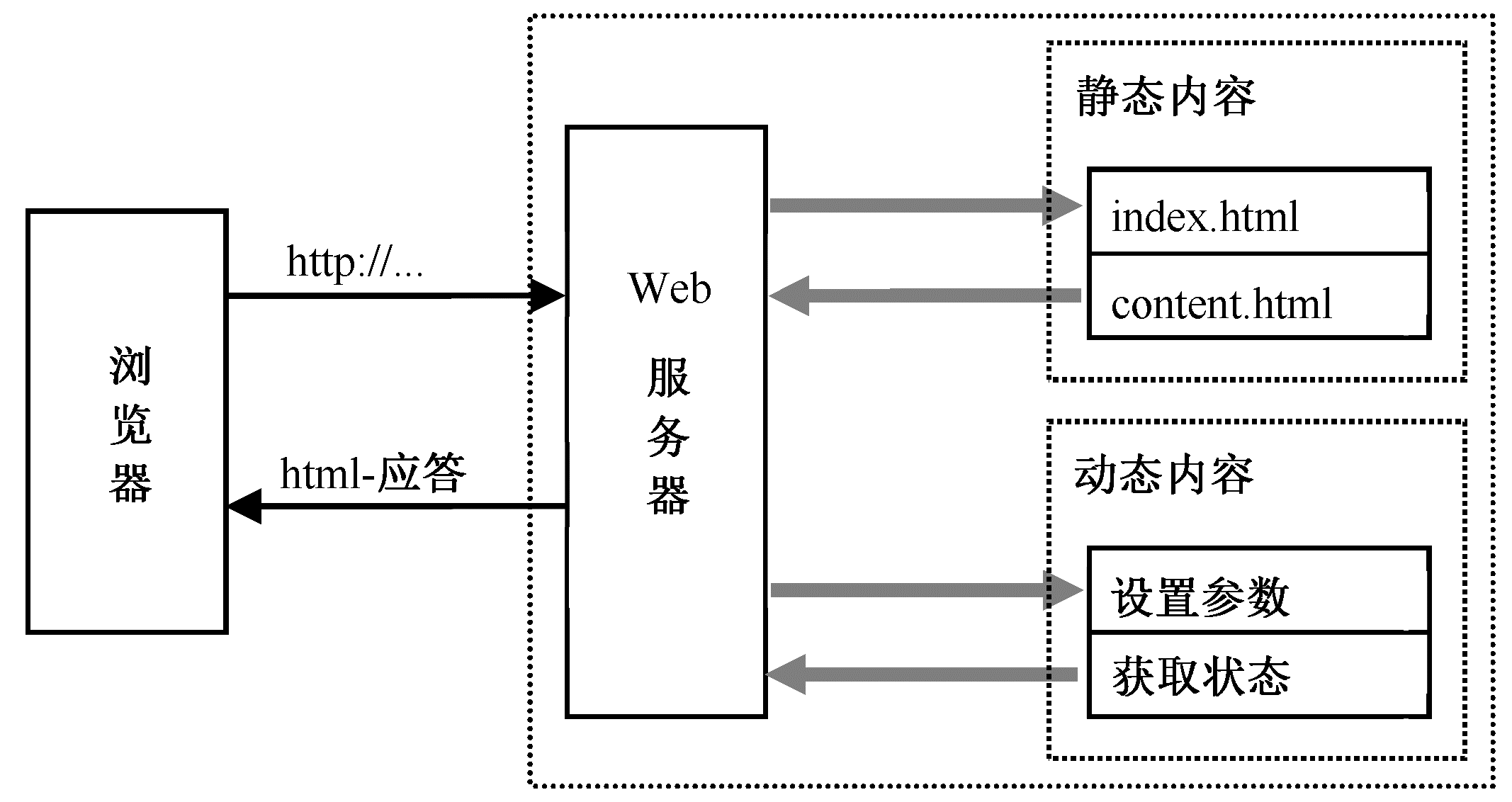
V20

学习猿地

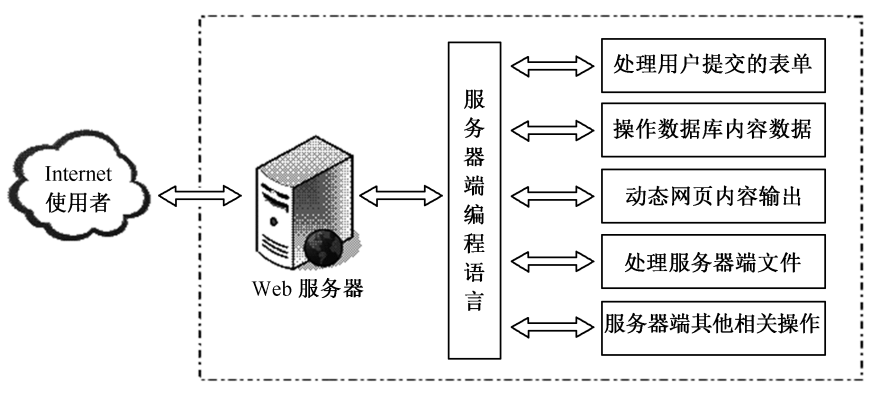
www.lmonkey.com

## 一、LAMP环境介绍

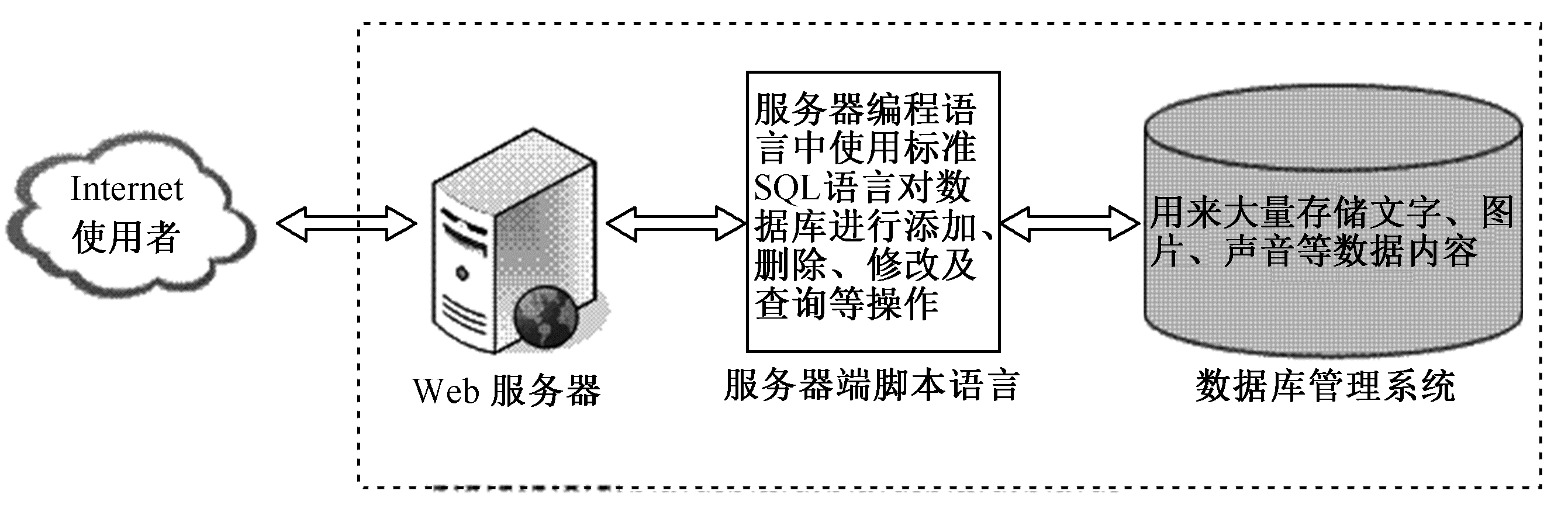
Web服务器的主要功能是提供网上信息浏览服务。所有网页的集合被称为网站，网站也只有发布到网上才能被他人访问到。所以开发人员需要将写好的网站上传到一台Web服务器［Web Server，也称为WWW（World Wide Web）服务器］上，并保存到Web服务器所管理的文档根目录中，才能完成对网站的发布。



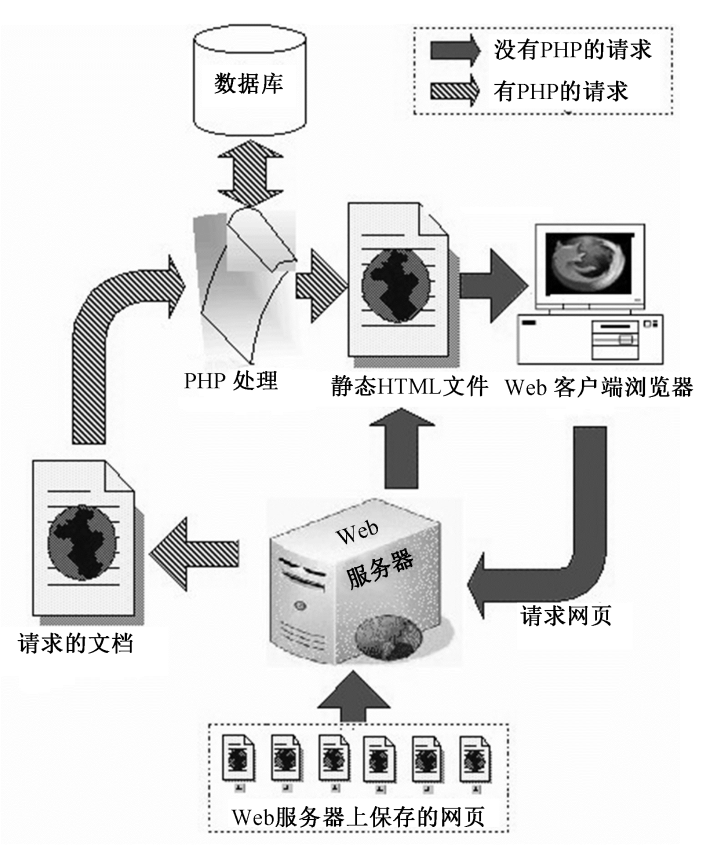
服务器端编程语言是提供访问商业逻辑的途径以供客户端应用的程序，是需要通过安装应用服务器解析的，而应用服务器又是Web服务器的一个功能模块，需要和Web服务器安装在同一个系统中。所以服务器端编程语言是用来协助Web服务器工作的编程语言，也可以说是对Web服务器功能的扩展，并外挂在Web服务器上一起工作，用在服务器端执行并完成服务器端的业务处理功能。当Web服务器收到一个HTTP请求时，就会将服务器下这个用户请求的文件原型响应给客户端浏览器，如果是HTML或是图片等浏览器可以解释的文件，浏览器将直接解释，并将结果显示给用户；如果是浏览器不认识的文件格式，则浏览器将解释成下载的形式，提示用户下载或是打开。如果用户想得到动态响应的结果，就要委托服务器端编程语言来完成了。



如果需要快速、安全地处理大量数据，则必须使用数据库管理系统。现在的动态网站都是基于数据库的编程，任何程序的业务逻辑实质上都是对数据的处理操作。数据库通过优化的方式，可以很容易地建立、更新和维护数据。数据库管理系统是Web开发中比较重要的构件之一，网页上的内容几乎都来自数据库。数据库管理系统也是一种软件，可以和Web服务器安装在同一台机器上，也可以不在同一台机器上安装，但都需要通过网络相连接。数据库管理系统负责存储和管理网站所需的内容数据，例如，文字、图片及声音等。当用户通过浏览器请求数据时，在服务器端程序中接收到用户的请求后，在程序中使用通用标准的结构化查询语言（SQL）对数据库进行添加、删除、修改及查询等操作，并将结果整理成HTML发回到浏览器上显示。



LAMP是Linux + Apache + MySQL + PHP的标准缩写。Linux操作系统、网站服务器Apache、数据库MySQL和PHP程序模块的连接，形成一个非常优秀的网站数据库的开发平台，是开源免费的自由软件，与JavaEE和ASP.NET架构形成了三足鼎立的竞争态势，是较受欢迎的开源软件网站开发平台。LAMP组合具有简易性、低成本、高安全性、开发速度快和执行灵活等特点，使得其在全球发展速度较快、应用较广，越来越多的企业将平台架构在LAMP之上。不管是否是专业人士，都可以利用LAMP平台工具来设计和架设网站及开发应用程序，目前主流的网站都在使用LAMP作为自己的系统运行平台。



LAMP这个特定名词最早出现在1998年，是Linux操作系统、Apache网页服务器、MySQL数据库管理系统和PHP程序模块4种技术名称开头字母缩写组成的。LAMP并不是某一家公司的产品，而是一组经常用来搭建动态网站或者服务器的开源软件组合。它们本身都是各自独立的软件，但是因为经常被结合在一起使用，并拥有越来越高的兼容度，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组合在发展速度上超过了JavaEE和ASP.NET等同类开发平台的商业软件。在LAMP平台上开发的项目在软件方面的投资成本较低、运行稳定，因此受到整个IT界的关注。

## 二、课程内容介绍

**目标：**为PHP项目部署最佳环境（可以使用Laravel框架，可以使用Redis做缓存，使用Memcache存储临时会话，可以使用Composer安装常用的代码库）。

**要求：**环境部署中的所有软件都是目前最高的稳定版本。

**真实：**不用Window，不用虚拟机安装，使用云服务器（阿里云）部署安装。

|  |  |  |  |
| --- | --- | --- | --- |
| 软件名称 | 最新版本 | 发布日期 | 官网 |
| Linux | CentOS 8.0-1905 | 2019-9-24 | <https://www.centos.org/> |
| Apache | 2.4.41 | 2019-08-14 | <http://www.apache.org/> |
| MySQL | 8.0.17 | 2019-7-22 | <https://www.mysql.com/> |
| PHP | 7.3.10 | 2019-9-26 | <https://www.php.net/> |

|  |  |  |
| --- | --- | --- |
| 部署方式 | 优点 | 缺点 |
| 集成环境（宝塔，PHPStudy等） | 安装超简单，几个命令搞定。  可以通过Web图形操作管理 | 自己定义性差，存在安全风险 |
| yum安装 | 安装部署便捷，快速 | 软件版本固定，且版本较低 |
| 源码安装 | 版本可自选，可自定义性强 | 难度较大，维护相对困难 |

采用源码包安装，不仅仅是为了搭建完美环境， 对学习也是很有帮助， 可以深入了解自己的环境架构， 对开发有了全局了解。对Apache、MySQL和PHP等软件有更深刻的认识。

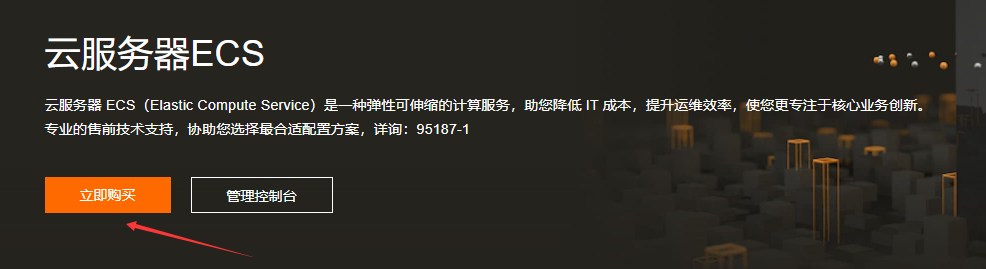
## 三、阿里云环境准备

1. 访问[https://www.aliyun.com](https://www.aliyun.com/)， 注册登录。

2. 选择云服务器ECS



1. 点击立即购买



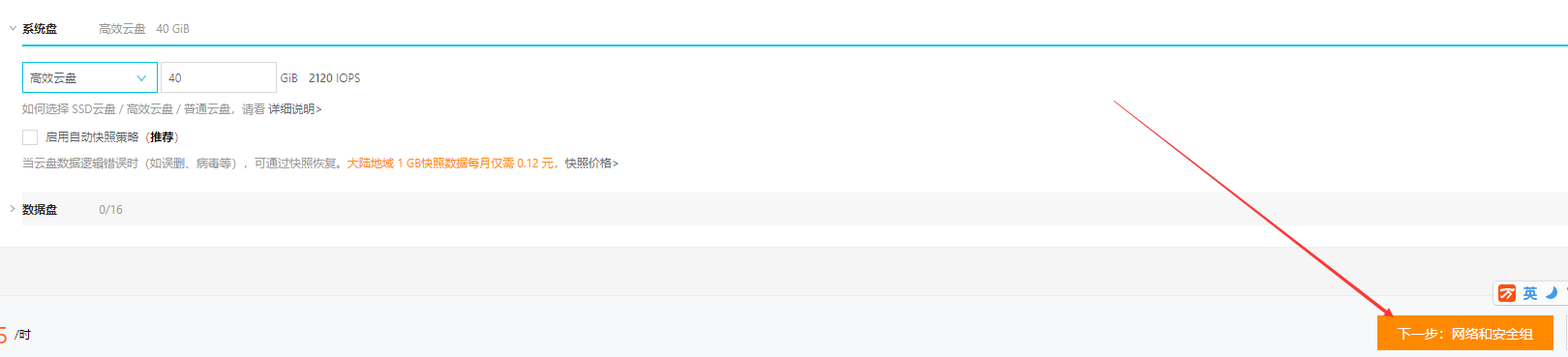
1. 选择服务器的基础配置



按量付费，选择华北3（张家口）有优惠的服务器。



实验环境可以先选择CPU1核， 内存2GB即可， 项目运行后期可增加。





网络不用改默认即可， 因为是按流量付费，所以带宽可以选择最高。



1. 系统配置， 选择自己定义密码， 设置root密码， 要记住，不然还要在控制台重置密码。



1. 设置释放时间， 根据自己的项目和学习需求指定。



1. 进入管理控制台。



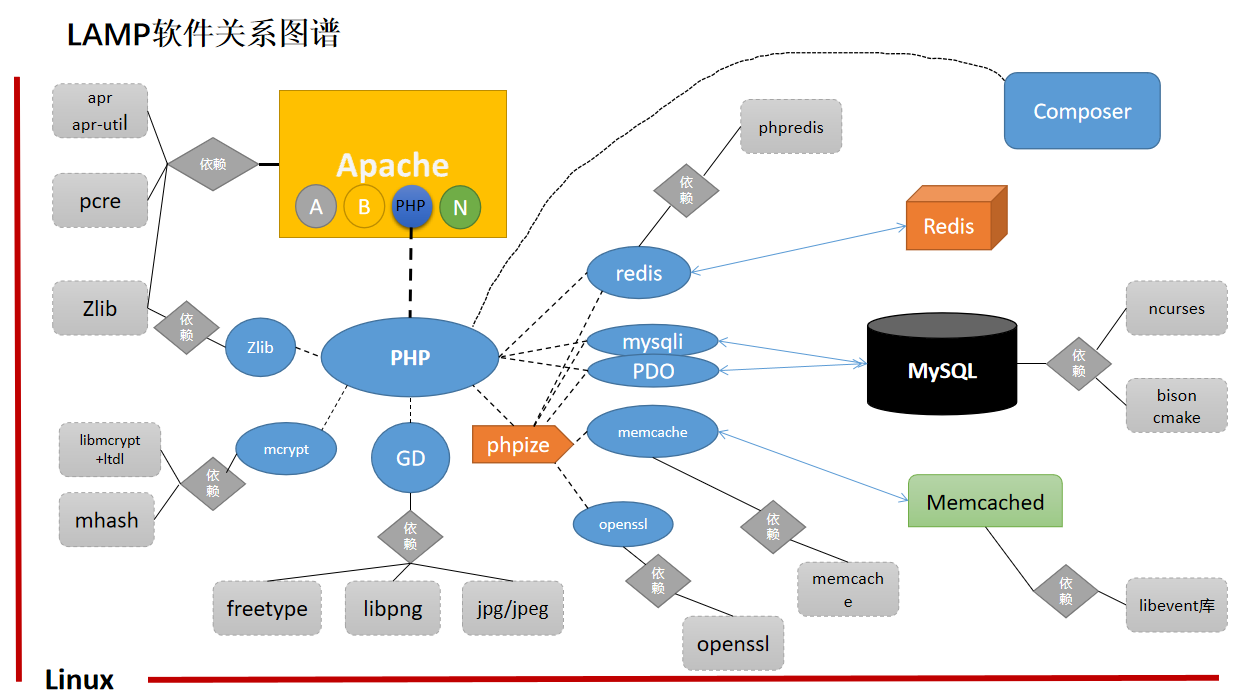
1. 可以查看到自己购买的服务器实例



1. 下载Xshell连接通过公网IP连接，远程管理。

可以在学习猿地（www.lmonkey.com）的“猿材料”中下载最新的XSHELL中文版。

## 四、LAMP环境软件之间的关系图谱及下载



|  |  |
| --- | --- |
| 软件源码 | 介绍 |
| apr-1.6.5.tar.gz | Apache高一点的版本将这些通用的函数独立出来并发展成为一个新的项目。APR的开发就从Apache中独立出来，Apache仅仅是使用APR而已。主要为上层的应用程序提供一个可以跨越多操作系统平台使用的底层支持接口库。 |
| apr-util-1.6.1.tar.gz | apr-util也是Apache获立出来的一个开发包，该文件中也是包含了一些常用的开发组件。这些组件与apr目录下的相比，它们与apache的关系更加密切一些。比如存储段和存储段组，加密等等。 |
| pcre-8.43.tar.gz | PCRE是一个用C语言编写的正则表达式函数库，被广泛使用在许多开源软件之中，最著名的莫过于Apache HTTP服务器和PHP脚本语言。 |
| zlib-1.2.11.tar.xz | zlib 是通用的压缩库，提供了一套 in-memory 压缩和解压函数，并能检测解压出来的数据的完整性(integrity)。zlib 也支持读写 gzip (.gz) 格式的文件。 |
| httpd-2.4.41.tar.gz | Apache最新版源源码包 |
| mcrypt-2.6.8.tar.gz | mcrypt 是php里面重要的加密支持扩展库。 |
| libmcrypt-2.5.8.tar.gz | 多密码加密库（在PHP中使用），libmcrypt是一个密码库，可以方便地将各种密码组合在一起以方便使用。 |
| mhash-0.9.9.9.tar.gz | Mhash是基于离散数学原理的不可逆向的php加密方式扩展库。mhash的可以用于创建校验数值，消息摘要，消息认证码，以及无需原文的关键信息保存（如密码）等。 |
| freetype-2.10.0.tar.gz | freetype库是一个开源的字体引擎,支持多种字符集编码(utf-8等). |
| libpng-1.6.7.tar.gz | libpng是官方的PNG参考库。它支持几乎所有PNG功能，具有可扩展性 |
| jpegsrc.v9c.tar.gz | jpeglib是处理文件后辍名为"．jpg"或"．jpeg"图片的库 |
| php-7.3.10.tar.gz | PHP软件的源码包 |
| pecl-memcache-4.0.4.tar.gz | PHP的memcache扩展模块源码 |
| libevent-2.1.11-stable.tar.gz | libevent作为轻量级的网络库,势必要处理多个事件的请求,若只有几个事件的发生,简单的多线程就可以实现。 |
| memcached-1.5.19.tar.gz | memcache服务器软件源码包 |
| phpredis-develop.zip | PHP的redis扩展库源码 |
| redis-5.0.6.tar.gz | Redis服务器软件源码包 |
| ncurses-6.1.tar.gz | 提供字符终端处理库，MySQL安装的依赖包 |
| mysql-8.0.17.tar.gz | MySQL服务器的源码包 |
| phpMyAdmin-5.0.0.zip | PHP项目，Web方式管理远程服务器的MySQL数据库。 |

主体采用源码包安装即可， 一些需要的库可直接使用yum安装。

在Linux服务器中使用wget直接在学习猿地的“猿材料”中下载需要的最新源码包。

|  |
| --- |
| # wget https://cdn2.lmonkey.com/e7977e11d6abd86e36466784beaf24c5/190e0d241f4648908245a2ff732f9080.zip |

直接下载到/root目录下即可， 使用unzip解压到当前目录， 如果没有安装unzip工具，可以使用yum安装。

编译安装

注意：每个源码包配置编译安装完成后，确认安装目录下是否生成安装文件（并确定目录是否正确）

建议将安装路径指定为[--prefix=/usr/local/软件名]格式

源码软件包安装报错确认与解决方案

|  |
| --- |
| echo $? #安装软件过程中由于频繁刷屏，建议在每个步骤结束后执行此命令  ./configure #此步骤报错多是依赖关系没解决或是编译工具未安装（注意关键词提示）  make #此步骤多是编译时选项参数书写错误、不存在、漏写等问题  #一般需要检查上一个步骤：./configure --help |

注意：若遇到报错，最简答的办法是，找到问题解决后重新解压软件，重新安装，步骤最简洁。

安装 gcc、gcc-c++、make等编译工具

|  |
| --- |
| # yum -y install gcc gcc-c++ make  安装c++ 编译器：yum -y install gcc-c++ ，再次编译通过。 |

安装libxml进行安装测试

|  |
| --- |
| # tar -xf libxml2-2.9.9.tar.gz  # cd /root/lamp/libxml2-2.9.9  # ./configure --prefix=/usr/local/libxml2/  # make  # make install |

## 五、安装 Apache

1. 解压安装Apache需要的所有软件和依赖包到当前目录下，即/root/lamp/下

|  |
| --- |
| # tar -xf httpd-2.4.41.tar.gz  # tar -xf apr-1.6.5.tar.gz  # tar -xf apr-util-1.6.1.tar.gz  # tar -xf pcre-8.43.tar.gz  # tar -xf zlib-1.2.11.tar.xz |

1. 源码包 2.4.\*版本中默认没有集成 apr的依赖包，所以需要提前解决依赖问题

|  |
| --- |
| # cp -a /root/lamp/apr-1.6.5 /root/lamp/httpd-2.4.41/srclib/apr  # cp -a /root/lamp/apr-util-1.6.1 /root/lamp/httpd-2.4.41/srclib/apr-util  缺少 expat库, 需要先安装否则在编译apr时会出错。apr-util致命错误:expat.h:没有那个文件或目录  #yum install -y expat-devel |

解压 apr和 apr-util，复制整个目录并取消目录上的版本号到指定位置，./configure时会检测

1. Apache默认需要依赖 pcre软件，但由于 Apache软件版本较高，则系统预安装的 pcre无法使用，所以需要人为手动安装适合版本

|  |
| --- |
| # cd /root/lamp/pcre-8.43  # ./configure  # make  # make install |

### 安装 zlib

|  |
| --- |
| # cd /root/lamp/zlib-1.2.11  # ./configure |
| # make  # make install |

1. Apache的加密传输模块 mod\_ssl，需要安装此软件产生

|  |
| --- |
| # yum -y install openssl-devel |

1. httpd软件安装

|  |  |
| --- | --- |
| # cd /lamp/httpd-2.4.41  # ./configure \  --prefix=/usr/local/apache2 \  --sysconfdir=/usr/local/apache2/etc \  --with-included-apr \  --enable-so \  --enable-deflate=shared \  --enable-expires=shared \  --enable-rewrite=shared \  --enable-ssl  # make  # make install |  |

--enable-ssl选项是为了后期实现 https提前设置的参数。

1. 启动 Apache测试

|  |
| --- |
| # /usr/local/apache2/bin/apachectl start  # ps aux | grep httpd |

使用进程查看命令确认 Apache是否启动，是否产生进程

|  |
| --- |
| # netstat -tlun |

使用网络进程查看命令确认 Apache是否启动，是否开启了 80监听端口

注意事项：在 CentOS 7操作系统上 Apache默认监听了 Ipv6地址的 80端口，没有监听 Ipv4的地址，所以需要修改下配置文件使其监听。

|  |
| --- |
| Listen 0.0.0.0:80 |

报错提示：若启动时提示/usr/local/apache2/modules/mod\_deflate.so无权限，可关闭 SELinux解决，类似此类.so文件不能载入或没有权限的问题，都是 SELinux问题，MySQL和 Apache都可能有类似问题。

警告提示：发现启动服务提示：AH00558: httpd: Could not reliably determine the server's fully

qualified domain name, using localhost.localdomain. Set the 'ServerName' directive globally

to suppress this message

解决办法：打开主配置文件 httpd.conf

|  |
| --- |
| 搜索 ServerName  （约在 200行左右）  改为 ServerName localhost:80（并且去掉前面的#注释） |

验证：通过浏览器输入地址访问：http://服务器 ip，若显示“It works”即表明 Apache正常工作。

## 六、安装 MySQL

### 安装 ncurses

|  |
| --- |
| # yum -y install ncurses-devel  # cd /root/lamp/ncurses-6.1  # ./configure --with-shared --without-debug --without-ada --enable-overwrite  # make  # make install |

若不安装 ncurses编译 MySQL时会报错

### 安装 cmake3

CMake是一个非常强大的编译自动配置工具，支持各种平台， MySQL被Oracle收了以后版本改为Cmake取代configure。 而新版的MySQL8以后需要使用新版的cmake3来编译自动配置。

|  |
| --- |
| yum -y install cmake3 bison |

最好加上文本语法分析器bison这个作为依赖一起安装。

### 升级gcc

升级gcc，MySQL的8.0.17最新版本，需要最新的版的gcc, CentOS7中默认的版本太低，不能编译新版的MySQL。

|  |
| --- |
| # yum -y install centos-release-scl  yum -y install devtoolset-8-gcc devtoolset-8-gcc-c++ devtoolset-8-binutils  scl enable devtoolset-8 bash  echo "source /opt/rh/devtoolset-8/enable" >>/etc/profile  export CC=/opt/rh/devtoolset-8/root/usr/bin/gcc export  CPP=/opt/rh/devtoolset-8/root/usr/bin/cpp export  CXX=/opt/rh/devtoolset-8/root/usr/bin/c++  以下为替换系统GCC  ln -s /opt/rh/devtoolset-8/root/usr/bin/\* /usr/local/bin/  hash -r  gcc --version |

### 加大交换分区，需要最少1G

|  |
| --- |
| mkdir -p /var/cache/swap/  dd if=/dev/zero of=/var/cache/swap/swap0 bs=1M count=1024  chmod 0600 /var/cache/swap/swap0  mkswap /var/cache/swap/swap0  swapon /var/cache/swap/swap0  但是这样并不能在系统重启的时候自动挂载交换分区, 这样我们就需要修改 fstab.  修改 /etc/fstab 文件, 新增如下内容:  /var/cache/swap/swap0 none swap sw 0 0 |

|  |
| --- |
| #free -m 最后一行就是交换分区 |

### 安装 MySQL

|  |
| --- |
| # tar -xf mysql-8.0.17.tar.gz  #useradd -r -s /sbin/nologin mysql |

为 MySQL软件创建运行用户，创建为系统用户，并限制此用户登录操作系统

|  |  |
| --- | --- |
| #cd /root/lamp/mysql-8.0.17/  #cmake3 \  -DDOWNLOAD\_BOOST=1  -DWITH\_BOOST=/boot/ \  -DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql \  -DCMAKE\_C\_COMPILER=/opt/rh/devtoolset-8/root/usr/bin/gcc -DCMAKE\_CXX\_COMPILER=/opt/rh/devtoolset-8/root/usr/bin/g++ \  -DMYSQL\_UNIX\_ADDR=/tmp/mysqld.sock \  -DDEFAULT\_CHARSET=utf8 \  -DDEFAULT\_COLLATION=utf8\_general\_ci \  -DWITH\_EXTRA\_CHARSETS:STRING=utf8,gbk \  -DWITH\_MYISAM\_STORAGE\_ENGINE=1 \  -DWITH\_INNOBASE\_STORAGE\_ENGINE=1 \  -DWITH\_MEMORY\_STORAGE\_ENGINE=1 \  -DWITH\_READLINE=1 \  -DFORCE\_INSOURCE\_BUILD=1 \  -DENABLED\_LOCAL\_INFILE=1 \  -DMYSQL\_DATADIR=/data/mysql \  -DMYSQL\_USER=mysql \  -DMYSQL\_TCP\_PORT=3306  # make  # make install | #使用BOOST库，  #使用BOOST库的位置  #安装位置  #用指定的gcc编辑器  用指定的g++编辑器  #指定 socket（套接字）文件位置  #扩展字符支持  #默认字符集  #默认字符校对  #安装 myisam存储引擎  #安装 innodb存储引擎  #安装 memory存储引擎  #支持 readline库  #启用加载本地数据  #指定数据文件的位置  #指定 mysql运行用户  #指定 mysql端口 |

MySQL安装后需要调整相应配置文件和参数才能正常运行

1. 修改 MySQL目录的用户归属

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

1. 初始化数据库

|  |
| --- |
| #mkdir -p /data/mysql/  #chown -R mysql:mysql /data/mysql/  #/usr/local/mysql/bin/mysqld --initialize --user=mysql \  --basedir=/usr/local/mysql --datadir=/data/mysql |

注意：在初始化数据时，有一个初始秘密一要记下来， 要通过个密码进行第一次登录。

[Server] A temporary password is generated for root@localhost: **v.Mf<E8q0wlc**

1. 修改配置文件，并初始化授权表

|  |
| --- |
| #vi /etc/my.cnf  [mysqld]  datadir=/data/mysql  basedir=/user/local/mysql  socket=/tmp/mysqld.sock  # Disabling symbolic-links is recommended to prevent assorted security risks  symbolic-links=0  # Settings user and group are ignored when systemd is used.  # If you need to run mysqld under a different user or group,  # customize your systemd unit file for mariadb according to the  # instructions in http://fedoraproject.org/wiki/Systemd  [mysqld\_safe]  log-error=/var/log/mariadb/mariadb.log  pid-file=/var/run/mariadb/mariadb.pid  #  # include all files from the config directory  #  !includedir /etc/my.cnf.d |

1. 创建并授权[mysqld\_safe]的两个选项目录和文件

|  |
| --- |
| # mkdir /var/log/mariadb  # touch /var/log/mariadb/mariadb.log  # chown -R mysql:mysql /var/log/mariadb/  # mkdir /var/run/mariadb/ 最好改个目录，重启这个目录会消失，就不能启动mysql  # chown -R mysql:mysql /var/run/mariadb/ |

1. 启动 MySQL服务

用原本源代码的方式去使用和启动 mysql

|  |
| --- |
| /usr/local/mysql/bin/mysqld\_safe --user=mysql & |

1. 登录 MySQL

|  |
| --- |
| # /usr/local/mysql/bin/mysql -u root -p  Enter password: **v.Mf<E8q0wlc**  mysql>  进入之后必须先更改密码才能操作  mysql>alter user root@'localhost' identified by "123456";  mysql>show databases;  mysql>use test;  mysql>show tables;  mysql>exit |

1. 关闭 MySQL服务

关闭 mysql, 最好别用killall方式

|  |
| --- |
| # /usr/local/mysql/bin/mysqladmin -uroot -p shutdown  Enter password:123456 |

## 七、安装 PHP

### 安装 libxml2

|  |
| --- |
| # tar -xf libxml2-2.9.9.tar.gz  # cd /root/lamp/libxml2-2.9.9  # ./configure --prefix=/usr/local/libxml2/  # make  # make install |

### 安装 libmcrypt

|  |
| --- |
| # tar -xf libmcrypt-2.5.8.tar.gz  # cd /root/lamp/libmcrypt-2.5.8  # ./configure --prefix=/usr/local/libmcrypt/  # make  # make install |

安装 libltdl，也在 libmcrypt源码目录中，非新软件

|  |
| --- |
| # cd /root/lamp/libmcrypt-2.5.8/libltdl  # ./configure --enable-ltdl-install  # make  # make install |

### 安装 mhash

|  |
| --- |
| # tar -xf mhash-0.9.9.9.tar.gz  # cd /root/lamp/mhash-0.9.9.9  # ./configure  # make  # make install |

### 安装 mcrypt

|  |
| --- |
| # tar -xf mcrypt-2.6.8.tar.gz  # cd /lamp/mcrypt-2.6.8  # export LD\_LIBRARY\_PATH=/usr/local/libmcrypt/lib:/usr/local/lib |

变量：LD\_LIBRARY\_PATH用于指定 libmcrypt和 mhash的库的位置

|  |
| --- |
| # ./configure --with-libmcrypt-prefix=/usr/local/libmcrypt  # make  # make install |

### 安装 libpng

|  |
| --- |
| # tar -xf libpng-1.6.7.tar.gz  # cd /root/lamp/libpng-1.6.7  # ./configure --prefix=/usr/local/libpng  # make  # make install |

### 安装 jpeg6

|  |
| --- |
| #tar -xf jpegsrc.v9c.tar.gz  # cd /root/lamp/jpeg-9c/ |
| # ./configure --prefix=/usr/local/jpeg9/ --enable-shared --enable-static  # make  # make install |

--enable-shared与--enable-static参数分别为建立共享库和静态库使用的 libtool

### 安装 freetype

|  |
| --- |
| # tar -xf freetype-2.10.0.tar.gz  # cd /root/lamp/freetype-2.10.0  # ./configure --prefix=/usr/local/freetype/  # make  # make install  #yum install freetype-devel |

PHP 会出现Configure: error: freetype.h not found 错误解决方法如下：

使用:yum install freetype-devel补充依赖包即可

### 安装 PHP

|  |  |
| --- | --- |
| # tar -xf php-7.3.10.tar.gz  # cd /root/lamp/php-7.3.10  # ./configure \  --prefix=/usr/local/php/ \  --with-config-file-path=/usr/local/php/etc/ \  --with-apxs2=/usr/local/apache2/bin/apxs \  --with-libxml-dir=/usr/local/libxml2/ \  --with-jpeg-dir=/usr/local/jpeg9/ \  --with-png-dir=/usr/local/libpng/ \  --with-freetype-dir=/usr/local/freetype/ \  --with-mcrypt=/usr/local/libmcrypt/ \  --with-gd \  --enable-soap \  --enable-mbstring=all \  --enable-sockets \  --with-pdo-mysql=/usr/local/mysql \  --with-pdo-mysql=mysqlnd \  --with-gd --without-pear  # make  # make install | #指定配置文件目录  #指定 apache动态模块位置  #指定 libxml位置  #指定 jpeg位置  #指定 libpng位置  #指定 freetype位  #指定 libmcrypt位置  #启用 gd库  #支持 soap服务(api)  #支持多字节，字符串  #支持套接字  #启用 mysql的 pdo模块支持  Mysql Native Driver 简称:mysqlnd  #不安装 pear(Composer代替) |

PHP安装后需要调整相应配置文件和参数才能正常运行。

1. 生成 php配置文件

|  |
| --- |
| # mkdir /usr/local/php/etc  # cp /root/lamp/php-7.3.10/php.ini-production /usr/local/php/etc/php.ini |

2.修改 Apache配置文件，使其识别\*.php文件，并能通过 php模块调用 php进行页面解析

|  |
| --- |
| # vim /usr/local/apache2/etc/httpd.conf  AddType application/x-httpd-php .php |

3.重启 Apache服务

|  |
| --- |
| # /usr/local/apache2/bin/apachectl stop  # /usr/local/apache2/bin/apachectl start |

4.测试 php页面是否能正常解析（即 apache和 php连通性）

|  |
| --- |
| # vim /usr/local/apache2/htdocs/test.php  <?php  phpinfo();  ?> |

通过浏览器输入地址访问：http://Apache服务器地址/test.php

### 为 PHP安装 openssl模块

|  |
| --- |
| #cd /root/lamp/php-7.3.10/ext  # cd openssl/  #mv config0.m4 config.m4  # /usr/local/php/bin/phpize  # ./configure --with-openssl --with-php-config=/usr/local/php/bin/php-config  # make  # make install |

phpize是用来扩展php扩展模块的，通过phpize可以建立php的外挂模块。

yum -y install autoconf 执行phpize时生成configure文件。

Autoconf是一个用于生成shell脚本的工具，可以自动配置软件源代码以适应多种类似POSIX的系统。为了让你的软件包在所有的不同系统上都可以进行编译。

### 为 PHP安装 memcache扩展模块

|  |
| --- |
| #tar -xf pecl-memcache-4.0.4.tar.gz  # cd pecl-memcache-4.0.4  # /usr/local/php/bin/phpize  # ./configure --with-php-config=/usr/local/php/bin/php-config  # make  # make install |

### 12.修改 php配置文件

使其识别并调用 openssl和 memcache两个模块

|  |
| --- |
| # vi /usr/local/php/etc/php.ini  extension\_dir="/usr/local/php/lib/php/extensions/no-debug-zts-20180731/" |

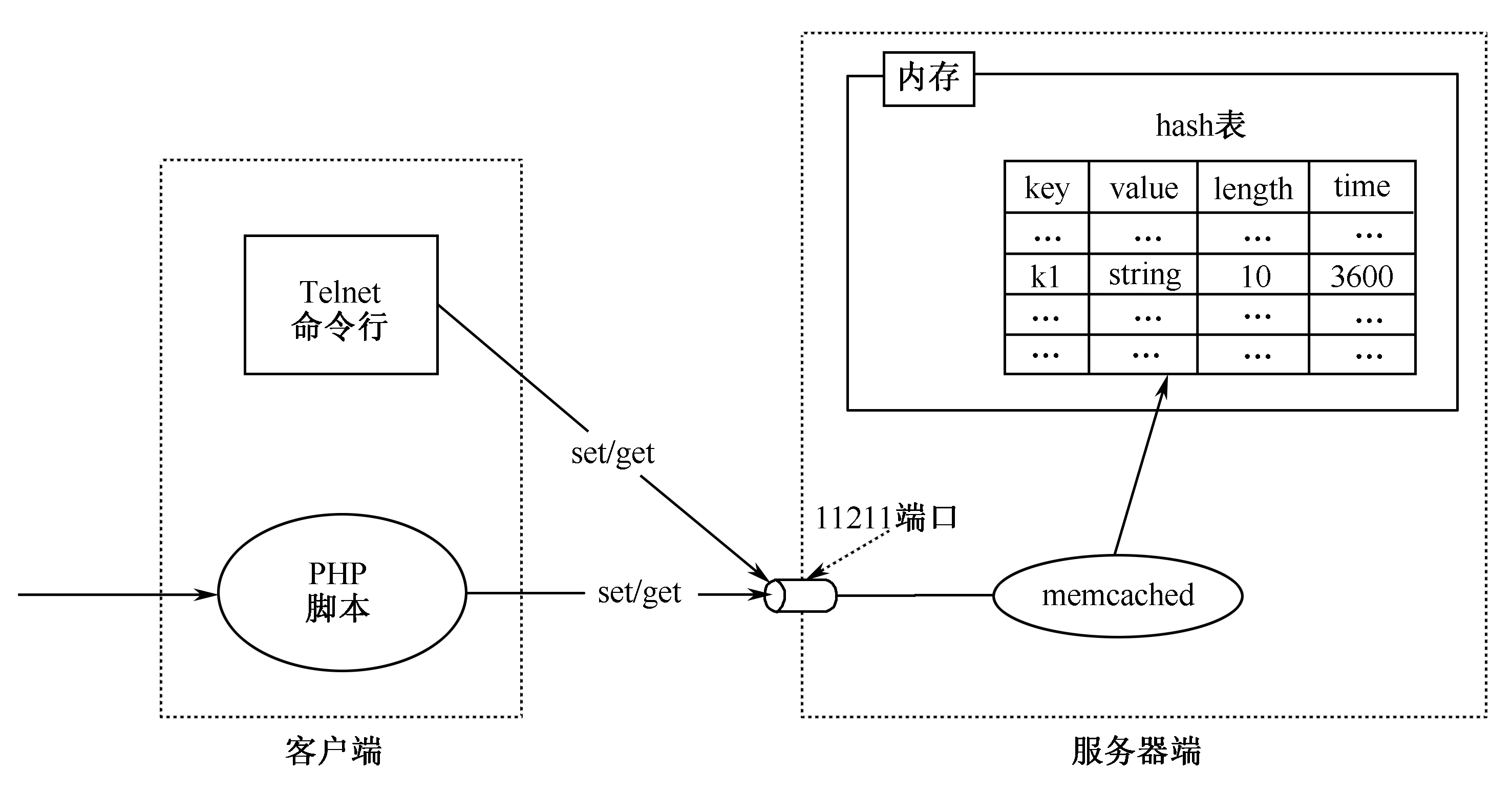
取消分号注释，并添加以上路径（此路径来自于模块安装命令的结果）

|  |
| --- |
| extension="openssl.so";  extension="memcache.so"; |

添加以上两个库文件的调用。

重启 apache，刷新 phpinfo 页面，并查看是否有两个新增的模块

### 安装 memcached服务



MemCache用到了libevent库，libevent是安装MemCache的唯一前提条件，是一套跨平台的事件处理接口的封装，memcached使用libevent来进行网络并发连接的处理，能够在很大并发的情况下，仍旧保持快速的响应能力。

|  |
| --- |
| # tar -xf memcached-1.5.19.tar.gz  # yum -y install libevent-devel  # cd /root/lamp/memcached-1.5.19  # ./configure --prefix=/usr/local/memcache  # make  # make install  # useradd -r -s /sbin/nologin memcache |

添加 memcache用户，此用户不用登录，不设置密码

|  |
| --- |
| # /usr/local/memcache/bin/memcached -umemcache & |

启动 memcache服务，并设置为后台运行

|  |
| --- |
| # netstat -an | grep :11211 |

检查 memcache是否正常启动，并监听了 11211端口

可以安装 yum -y install telnet 连接测试 telnet localhost 11211

成功连接memcached服务器后，与memcached通信的客户端命令并不多，并且使用方法都非常简单。仅有5个常用的命令（区分大小写），如下所示。

 stats：当前所有memcached服务器运行的状态信息。

 add：添加一个数据到服务器。

 set：替换一个已经存在的数据。如果数据不存在，则和add命令相同。

 get：从服务器端提取指定的数据。

 delete：删除指定的单个数据。如果要清除所有数据，可以使用flush\_all指令。

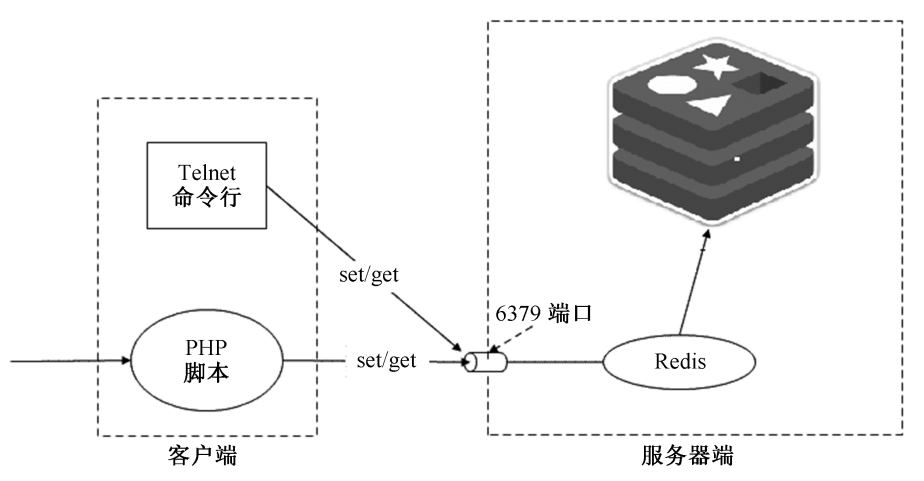
如果以上命令执行发生错误，MemCache协议会对错误部分做出提示。主要有3个错误提示的指令，如下所示。

 ERROR：普通的错误信息，如指令错误。

 CLIENT\_ERROR <错误信息>：客户端错误。

 SERVER\_ERROR <错误信息>：服务器端错误。

### 安装 Redis服务



|  |
| --- |
| # tar -xf redis-5.0.6.tar.gz  # cd /root/lamp/redis-5.0.6 |

安装

|  |
| --- |
| # make  #cd src  #make install PREFIX=/usr/local/redis |

移动配置文件到安装目录下

|  |
| --- |
| # cd /root/lamp/redis-5.0.6  # mkdir /usr/local/redis/etc  #mv redis.conf /usr/local/redis/etc |

配置redis为后台启动

|  |
| --- |
| # vi /usr/local/redis/etc/redis.conf    //将daemonize no 改成daemonize yes |

开启redis

|  |
| --- |
| #/usr/local/redis/bin/redis-server /usr/local/redis/etc/redis.conf |

同样可以使用telnet连接测试

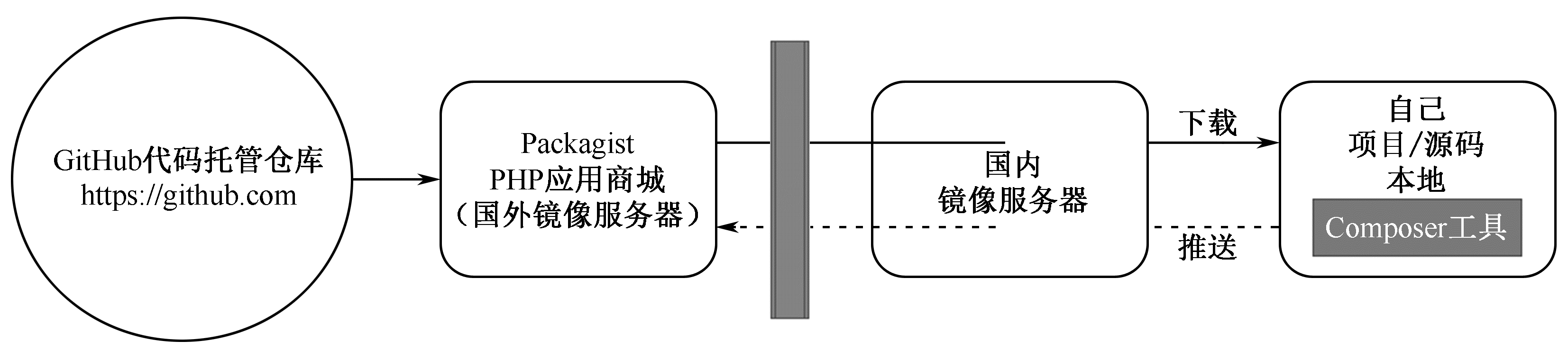
### 为 PHP安装 redis扩展模块

|  |
| --- |
| #unzip phpredis-develop.zip  #cd /root/lamp/phpredis-develop  # /usr/local/php/bin/phpize  # ./configure --with-php-config=/usr/local/php/bin/php-config  # make  # make install |

|  |
| --- |
| # vi /usr/local/php/etc/php.ini  extension\_dir="/usr/local/php/lib/php/extensions/no-debug-zts-20180731/" |

|  |
| --- |
| extension="openssl.so";  extension="memcache.so";  extension="redis.so"; |

### 安装PHP依赖管理工具Composer



在一般情况下，安装包的数据多数是从 https://github.com上下载的，安装包的元数据是从http://packagist.org上下载的。国内镜像服务器所做的就是缓存所有安装包和元数据到国内的机房并通过国内的 CDN 进行加速，这样下载安装代码和更新加速的过程更加稳定。国内镜像服务器URL为https://pkg.phpcomposer.com。项目中常用的Composer代码包如表24-1所示。

项目中常用的Composer代码包

|  |  |  |
| --- | --- | --- |
| 包名 | 说明 | 地址 |
| guzzlehttp/guzzle | 功能强大的HTTP请求库 | [https://packagist.org/packages/guzzlehttp/guzzle](https://packagist.org/packages/guzzlehttp/guzzle" \t "https://blog.csdn.net/Webben/article/details/_blank) |
| hashids/hashids | 数字ID转字符串，支持多语言 | [https://packagist.org/packages/hashids/hashids](https://packagist.org/packages/hashids/hashids" \t "https://blog.csdn.net/Webben/article/details/_blank) |
| intervention/image | 图片处理，获取图片信息、上传、格式转换、缩放、裁剪等 | [https://packagist.org/packages/intervention/image](https://packagist.org/packages/intervention/image" \t "https://blog.csdn.net/Webben/article/details/_blank) |
| phpmailer/phpmailer | 邮件发送 | [https://packagist.org/packages/phpmailer/phpmailer](https://packagist.org/packages/phpmailer/phpmailer" \t "https://blog.csdn.net/Webben/article/details/_blank) |
| phpoffice/phpexcel | excel操作类 | [https://packagist.org/packages/phpoffice/phpexcel](https://packagist.org/packages/phpoffice/phpexcel" \t "https://blog.csdn.net/Webben/article/details/_blank) |
| monolog/monolog | 日志操作， composer官方就是用它做例子的 | [https://packagist.org/packages/monolog/monolog](https://packagist.org/packages/monolog/monolog" \t "https://blog.csdn.net/Webben/article/details/_blank) |
| catfan/medoo | 简单易用的数据库操作类，支持各种常见数据库 | [https://packagist.org/packages/catfan/medoo](https://packagist.org/packages/catfan/medoo" \t "https://blog.csdn.net/Webben/article/details/_blank) |
| league/route | 路由调度 | https://packagist.org/packages/league/route |
| Carbon/Carbon | 时间操作 | [https://packagist.org/packages/nesbot/carbon](https://packagist.org/packages/nesbot/carbon" \t "https://blog.csdn.net/Webben/article/details/_blank) |

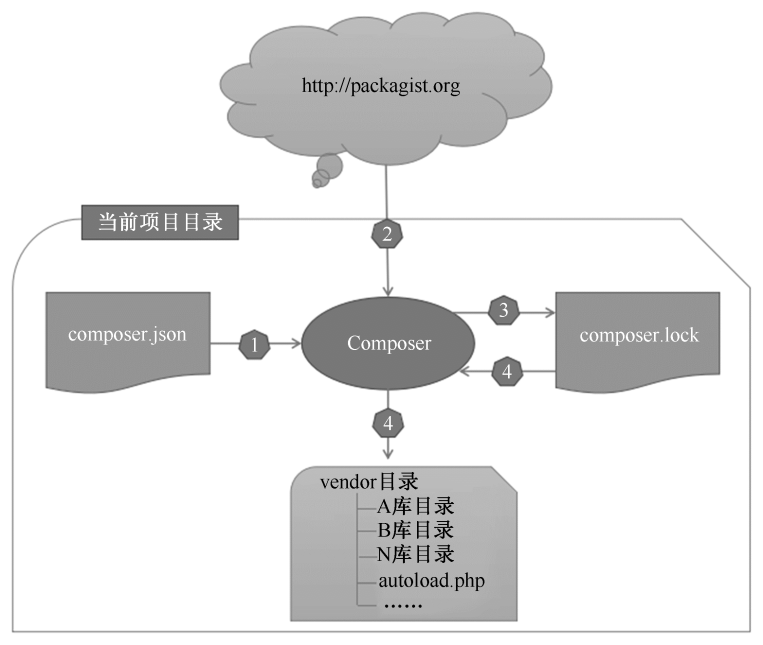
|  |
| --- |
| //下载composer压缩包  #wget https://getcomposer.org/composer.phar  //移动到全局  #cp composer.phar /usr/local/bin/composer  //更改权限  #chmod u+x /usr/local/bin/composer |

因为Composer是命令行的操作工具，所以安装完成后，在命令行中输入“composer-V”命令，或直接输入“composer”命令回车执行成功。

Composer 设置国内镜像加速命令:

|  |
| --- |
| # /usr/local/bin/composer \  config -g repo.packagist composer https://packagist.phpcomposer.com  查看Composer全局设置命令  # /usr/local/bin/composer config -gl |

在使用composer install、composer update、composer require这3个命令时，都会下载PHP类库。也都有可能要经过以下几个步骤。



Composer执行命令需要经历的步骤

（1）Composer读取composer.json文件，这个json在当前执行项目的目录下。

（2）Composer通过读取的json数据去Packagist.org获取各个包的包名、作者、下载URL等信息。

（3）将从Packagist.org获取的元数据存放到当前目录下的composer.lock中。

（4）Composer读取composer.lock中的元数据，根据元数据下载包，并且放到当前目录下的vendor目录里面。

其中，composer update 会将步骤1、步骤2、步骤3、步骤4都执行一遍，所以下载的类库是composer.json配置中匹配搭配的最新类库，而composer install只执行步骤4。 composer require 会将配置写入composer.json，然后执行步骤1、步骤2、步骤3、步骤4。如果想了解不同的包的配置是怎么写的，可以在Packagist.org中找到，因为每个开源项目都有安装和使用方法。

## 八、安装 phpMyAdmin

phpMyAdmin是通过PHP代码连接MySQL数据库进行管理的。使用PHP的mysqli扩展模块，如果环境中没有安装，请先使用phpize安装。

进入MySQL数据库： \s 或 status

UNIX socket: /tmp/mysqld.sock

# vi /usr/local/php/etc/php.ini

mysqli.default\_socket = /tmp/mysqld.sock

两个改成一样。

由于新版本的mysql账号密码解锁机制不一致导致的

解决办法：

删除创建的用户和授权，

找到mysql配置文件并加入

default\_authentication\_plugin=mysql\_native\_password

变为原来的验证方式，然后从新创建用户并授权即可

或

mysql -uroot -p

use mysql;

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY '123456';

|  |
| --- |
| # unzip phpMyAdmin-5.0.0-alpha1-all-languages.zip  # cp -a /root/lamp/phpMyAdmin-5.0.0-alpha1-all-languages  /usr/local/apache2/htdocs/phpmyadmin  # cd /usr/local/apache2/htdocs/phpmyadmin  # cp -a config.sample.inc.php config.inc.php |

|  |
| --- |
| # vim config.inc.php （可改，可不改）  $cfg['Servers'][$i]['auth\_type'] = 'cookie';  $cfg['Servers'][$i]['auth\_type'] = 'http';  #设置 auth\_type为 http ，即设置为 HTTP身份认证模式（新增即可） |

通过浏览器输入地址访问：http://Apache服务器地址/phpmyadmin/index.php

用户名为 root ，密码为 MySQL设置时指定的 root密码 123456

## 九、设置 Apache、MySQL、Memcache、Redis开机自启

借助系统自带脚本/etc/rc.local(CentOS7,以上改下执行权限，不然开机不执行)，此脚本开机后会自动加载，我们可以将源码安装的服务启动命令写入该脚本，间接实现开机自启动。

|  |
| --- |
| # vi /etc/rc.local  /usr/local/apache2/bin/apachectl start  /usr/local/mysql/bin/mysqld\_safe --user=mysql &  /usr/local/memcache/bin/memcached -umemcache &  /usr/local/redis/bin/redis-server /usr/local/redis/etc/redis.conf |

## 十、项目安装上线：

1、 把 php项目拷贝到网站默认目录下：/usr/local/apache2/htdocs/\*\*

2、 使用 phpMyAdmin创建网站所需数据库

注意事项：注意目录权限和归属，防止权限过大或者权限过小

切记：修改配置文件之前一定要备份一个版本。

3、从学习猿地的“猿代码”中，下载一个blog项目， 安装一下。

|  |
| --- |
| # wget https://cdn2.lmonkey.com/e7977e11d6abd86e36466784beaf24c5/854c8c81638042eba99508179fef2eda.zip |