

LNMP架构部署动态网站环境



源码包程序



LNMP动态网站架构



搭建Discuz论坛



源码包程序

源码包程序

源码包程序

使用源码包来安装服务程序具有两个优势。

- 源码包的可移植性非常好，几乎可以在任何Linux系统中安装使用，而RPM软件包是针对特定系统和架构编写的指令集，必须严格地符合执行环境才能顺利安装（即只会去“生硬地”安装服务程序）。
- 使用源码包安装服务程序时会有一个编译过程，因此可以更好地适应安装主机的系统环境，运行效率和优化程度都会强于使用RPM软件包安装的服务程序。也就是说，可以将采用源码包安装服务程序的方式看作是针对系统的“量体裁衣”。

一般来讲，在安装软件时，如果能通过Yum软件仓库来安装，就用Yum方式；反之则去寻找合适的RPM软件包来安装；如果是在没有资源可用，那就只能使用源码包来安装了。

安装编译环境

第1步：下载及解压源码包文件

```
[root@localhost ~]# tar xzvf frp-0.21.0.tar.gz  
[root@localhost ~]# cd frp-0.21.0
```

第2步：设置环境变量

当编译工作结束后，如果系统环境符合安装要求，一般会自动在当前目录下生成一个Makefile安装文件

```
[root@localhost frp-0.21.0]# ./configure --prefix=/usr/local/program
```

第3步：编译源码生成二进制安装程序

接下来便可以使用make命令来根据Makefile文件内容提供的合适规则编译生成出真正可供用户安装服务程序的二进制可执行文件了。

```
[root@localhost frp-0.21.0]# make
```

第4步：运行二进制的服务程序安装包

```
[root@localhost frp-0.21.0]# make install
```

第5步：清理源码包临时文件

```
[root@localhost frp-0.21.0]# make clean
```

LNMP动态网站架构



LNMP动态网站架构

LNMP动态网站部署架构是一套由Linux + Nginx + MySQL + PHP组成的动态网站系统解决方案。LNMP中的字母L是Linux系统的意思，不仅可以是RHEL、CentOS、Fedora，还可以是Debian、Ubuntu等系统



LNMP动态网站架构

在使用源码包安装服务程序之前，首先要让安装主机具备编译程序源码的环境，他需要具备C语言、C++语言、Perl语言的编译器，以及各种常见的编译支持函数库程序。因此请先配置妥当Yum软件仓库，然后把下面列出的这些软件包都统统安装上

```
[root@localhost ~]# yum install -y apr* autoconf automake bison bzip2 bzip2*  
compat* cpp curl curl-devel fontconfig fontconfig-devel freetype freetype*  
freetype-devel gcc gcc-c++ gd gettext gettext-devel glibc kernel kernel-headers  
keyutils keyutils-libs-devel krb5-devel libcom_err-devel libpng libpng-devel  
libjpeg* libsepol-devel libselinux-devel libstdc++-devel libtool* libgomp  
libxml2 libxml2-devel libXpm* libtiff libtiff* make mpfr ncurses* ntp openssl  
openssl-devel patch pcre-devel perl php-common php-gd policycoreutils telnet  
t1lib t1lib* nasm nasm* wget zlib-devel unzip
```


LNMP动态网站架构

下载安装LNMP动态网站部署架构所需的16个软件源码包和1个用于检查效果的论坛网站系统软件包

```
[root@localhost ~]# cd /usr/local/src/  
[root@localhost src]# wget http://192.168.0.11:8889/lnmp.zip  
[root@localhost src]# unzip lnmp.zip  
[root@localhost src]# ls  
cmake-2.8.11.2.tar.gz  jpegsrc.v9a.tar.gz      libpng-1.6.12.tar.gz  mysql-  
5.6.19.tar.gz        pcre-8.35.tar.gz       tiff-4.0.3.tar.gz  
Discuz_X3.2_SC_GBK.zip  libgd-2.1.0.tar.gz     libvpx-v1.3.0.tar.bz2  nginx-  
1.6.0.tar.gz         php-5.5.14.tar.gz      yasm-1.2.0.tar.gz  
freetype-2.5.3.tar.gz  libmcrypt-2.5.8.tar.gz lnmp.zip              openssl-  
1.0.1h.tar.gz        t1lib-5.1.2.tar.gz    zlib-1.2.8.tar.gz  
[root@localhost src]#
```

LNMP动态网站架构

CMake是Linux系统中一款常用的编译工具。要想通过源码包安装服务程序，就一定要严格遵守上面总结的安装步骤——下载及解压源码包文件、编译源码包代码、生成二进制安装程序、运行二进制的服务程序安装包。

```
[root@localhost src]# tar xzvf cmake-2.8.11.2.tar.gz
[root@localhost src]# cd cmake-2.8.11.2
[root@localhost cmake-2.8.11.2]# ./configure
[root@localhost cmake-2.8.11.2]# make
[root@localhost cmake-2.8.11.2]# make install
```

配置MySQL服务

```
[root@localhost cmake-2.8.11.2]# cd ..  
[root@localhost src]# useradd mysql -s /sbin/nologin  
[root@localhost src]# mkdir -p /usr/local/mysql/var  
[root@localhost src]# chown -Rf mysql:mysql /usr/local/mysql
```

配置MySQL服务

接下来解压、编译、安装MySQL数据库服务程序。在编译数据库时使用的是cmake命令，其中，

-DCMAKE_INSTALL_PREFIX参数用于定义数据库服务程序的保存目录，

-DMYSQL_DATADIR参数用于定义真实数据库文件的目录，

-DSYSCONFDIR则是定义MySQL数据库配置文件的保存目录。(编译时间较长，为了防止ssh断开，建议screen运行！)

```
[root@localhost src]# tar xzvf mysql-5.6.19.tar.gz
[root@localhost src]# cd mysql-5.6.19
[root@localhost mysql-5.6.19]# cmake . -DCMAKE_INSTALL_PREFIX=/usr/local/mysql
-DMYSQL_DATADIR=/usr/local/mysql/var -DSYSCONFDIR=/etc
[root@localhost mysql-5.6.19]# make
[root@localhost mysql-5.6.19]# make install
```

配置MySQL服务

为了让MySQL数据库程序正常运转起来，需要先删除/etc目录中的默认配置文件。
然后在MySQL数据库程序的保存目录scripts内找到一个名为mysql_install_db的脚本程序。
执行这个脚本程序并使用--user参数指定MySQL服务的对应账号名称（在前面步骤已经创建）。

使用--basedir参数指定MySQL服务程序的保存目录。
使用--datadir参数指定MySQL真实数据库的文件保存目录。

这样即可生成系统数据库文件，也会生成出新的MySQL服务配置文件。

```
[root@localhost mysql-5.6.19]# rm -rf /etc/my.cnf
[root@localhost mysql-5.6.19]# cd /usr/local/mysql/
[root@localhost mysql]# ./scripts/mysql_install_db --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/var/
```

配置MySQL服务

把系统新生成的MySQL数据库配置文件链接到/etc目录中，然后把程序目录中的开机程序文件复制到/etc/rc.d/init.d目录中，以便通过service命令来管理MySQL数据库服务程序。记得把数据库脚本文件的权限修改成755以便于让用户有执行该脚本的权限

```
[root@localhost mysql]# ln -s my.cnf /etc/my.cnf
[root@localhost mysql]# cp ./support-files/mysql.server /etc/rc.d/init.d/mysqld
[root@localhost mysql]# chmod 755 /etc/rc.d/init.d/mysqld
```

配置MySQL服务

编辑刚复制的MySQL数据库脚本文件，把第46、47行的basedir与datadir参数分别修改为MySQL数据库程序的保存目录和真实数据库的文件内容。

```
[root@localhost mysql]# vi /etc/rc.d/init.d/mysqld  
46行 basedir=/usr/local/mysql  
47行 datadir=/usr/local/mysql/var
```

配置MySQL服务

启动mysql，并且设置开机启动

```
[root@localhost mysql]# service mysqld start
Starting MySQL. SUCCESS!
[root@localhost mysql]# chkconfig mysqld on
```


配置MySQL服务

把命令所保存的目录永久性地定义到PATH变量中，需要编辑/etc/profile文件并写入追加的命令目录，这样当物理设备在下一次重启时就会永久生效了。如果不想通过重启设备的方式来生效，也可以使用source命令加载一下/etc/profile文件，此时新的PATH变量也可以立即生效了。

```
[root@localhost mysql]# vi /etc/profile
74行 export PATH=$PATH:/usr/local/mysql/bin
[root@localhost mysql]# source /etc/profile
```

配置MySQL服务

MySQL数据库服务程序还会调用到一些程序文件和函数库文件。由于当前是通过源码包方式安装MySQL数据库，因此现在也必须以手动方式把这些文件链接过来。

```
[root@localhost mysql]# mkdir /var/lib/mysql
[root@localhost mysql]# ln -s /usr/local/mysql/lib/mysql /usr/lib/mysql
[root@localhost mysql]# ln -s /tmp/mysql.sock /var/lib/mysql/mysql.sock
[root@localhost mysql]# ln -s /usr/local/mysql/include/mysql /usr/include/mysql
```

配置MySQL服务

初始化数据库

```
[root@localhost mysql]# mysql_secure_installation
```

配置nginx服务

解决相关的软件依赖关系

提供Perl语言兼容的正则表达式库的软件包pcre

```
[root@localhost ~]# cd /usr/local/src
[root@localhost src]# tar xzvf pcre-8.35.tar.gz
[root@localhost src]# cd pcre-8.35
[root@localhost pcre-8.35]# ./configure --prefix=/usr/local/pcre --enable-utf8
[root@localhost pcre-8.35]# make
[root@localhost pcre-8.35]# make install
```

配置nginx服务

openssl软件包是用于提供网站加密证书服务的程序文件，在安装该程序时需要自定义服务程序的安装目录，以便于稍后调用它们的时候更可控。

```
[root@localhost pcre-8.35]# cd /usr/local/src
[root@localhost src]# tar xzvf openssl-1.0.1h.tar.gz
[root@localhost src]# cd openssl-1.0.1h
[root@localhost openssl-1.0.1h]# ./config --prefix=/usr/local/openssl
[root@localhost openssl-1.0.1h]# make
[root@localhost openssl-1.0.1h]# make install
```

配置nginx服务

openssl软件包安装后默认会在/usr/local/openssl/bin目录中提供很多的可用命令，我们需要像前面的操作那样，将这个目录添加到PATH环境变量中，并写入到配置文件中，最后执行source命令以便让新的PATH环境变量内容可以立即生效

```
[root@localhost openssl-1.0.1h]# vi /etc/profile
75行 export PATH=$PATH:/usr/local/openssl/bin:/usr/local/openssl/bin
[root@localhost openssl-1.0.1h]# source /etc/profile
```

配置nginx服务

zlib软件包是用于提供压缩功能的函数库文件。其实Nginx服务程序调用的这些服务程序无需深入了解，只要大致了解其作用就已经足够了

```
[root@localhost openssl-1.0.1h]# cd /usr/local/src/  
[root@localhost src]# tar xzvf zlib-1.2.8.tar.gz  
[root@localhost src]# cd zlib-1.2.8  
[root@localhost zlib-1.2.8]# ./configure --prefix=/usr/local/zlib  
[root@localhost zlib-1.2.8]# make  
[root@localhost zlib-1.2.8]# make install
```

配置nginx服务

创建一个用于执行Nginx服务程序的账户。账户名称可以自定义

```
[root@localhost zlib-1.2.8]# cd ..  
[root@localhost src]# useradd www -s /sbin/nologin
```

其中，--prefix参数用于定义服务程序稍后安装到的位置，
--user与--group参数用于指定执行Nginx服务程序的用户名和用户组。

配置nginx服务

在使用参数调用openssl、zlib、pcre软件包时，请写出软件源码包的解压路径，而不是程序的安装路径

```
[root@localhost src]# tar xzvf nginx-1.6.0.tar.gz
[root@localhost src]# cd nginx-1.6.0
[root@localhost nginx-1.6.0]# ./configure --prefix=/usr/local/nginx --without-http_memcached_module --user=www --group=www --group=www --with-http_stub_status_module --with-http_ssl_module --with-http_gzip_static_module --with-openssl=/usr/local/src/openssl-1.0.1h --with-zlib=/usr/local/src/zlib-1.2.8 --with-pcre=/usr/local/src/pcre-8.35
[root@localhost nginx-1.6.0]# make
[root@localhost nginx-1.6.0]# make install
```

配置nginx服务

要想启动Nginx服务程序以及将其加入到开机启动项中，也需要有脚本文件。

```
[root@localhost nginx-1.6.0]# curl https://raw.githubusercontent.com/xcracker/linux_shell/master/init.d/nginx > /etc/rc.d/init.d/nginx
[root@localhost nginx-1.6.0]# chmod 755 /etc/rc.d/init.d/nginx
[root@localhost nginx-1.6.0]# /etc/rc.d/init.d/nginx start
Reloading systemd:      [ 确定 ]
Starting nginx (via systemctl): [ 确定 ]
[root@localhost nginx-1.6.0]# chkconfig nginx on
[root@localhost nginx-1.6.0]# firewall-cmd --add-port=80/tcp --permanent
success
[root@localhost nginx-1.6.0]# firewall-cmd --reload
success
[root@localhost nginx-1.6.0]#
```

配置nginx服务

查看是否完成



配置PHP服务

yasm源码包是一款常见的开源汇编器

```
[root@localhost nginx-1.6.0]# cd ..  
[root@localhost src]# tar xzvf yasm-1.2.0.tar.gz  
[root@localhost src]# cd yasm-1.2.0  
[root@localhost yasm-1.2.0]# ./configure  
[root@localhost yasm-1.2.0]# make  
[root@localhost yasm-1.2.0]# make install
```

配置PHP服务

libmcrypt源码包是用于加密算法的扩展库程序

```
[root@localhost yasm-1.2.0]# cd ..  
[root@localhost src]# tar xzvf libmcrypt-2.5.8.tar.gz  
[root@localhost src]# cd libmcrypt-2.5.8  
[root@localhost libmcrypt-2.5.8]# ./configure  
[root@localhost libmcrypt-2.5.8]# make  
[root@localhost libmcrypt-2.5.8]# make install
```

配置PHP服务

libvpx源码包是用于提供视频编码器的服务程序

```
[root@localhost libmccrypt-2.5.8]# cd ..  
[root@localhost src]# tar xjvf libvpx-v1.3.0.tar.bz2  
[root@localhost src]# cd libvpx-v1.3.0  
[root@localhost libvpx-v1.3.0]# ./configure --prefix=/usr/local/libvpx --enable-  
shared --enable-vp9  
[root@localhost libvpx-v1.3.0]# make  
[root@localhost libvpx-v1.3.0]# make install
```

配置PHP服务

tiff源码包是用于提供标签图像文件格式的服务程序

```
[root@localhost libvpx-v1.3.0]# cd ..  
[root@localhost src]# tar xzvf tiff-4.0.3.tar.gz  
[root@localhost src]# cd tiff-4.0.3  
[root@localhost tiff-4.0.3]# ./configure --prefix=/usr/local/tiff --enable-shared  
[root@localhost tiff-4.0.3]# make  
[root@localhost tiff-4.0.3]# make install
```

配置PHP服务

libpng源码包是用于提供png图片格式支持函数库的服务程序

```
[root@localhost tiff-4.0.3]# cd ..  
[root@localhost src]# tar xzvf libpng-1.6.12.tar.gz  
[root@localhost src]# cd libpng-1.6.12  
[root@localhost libpng-1.6.12]# ./configure --prefix=/usr/local/libpng --enable-shared  
[root@localhost libpng-1.6.12]# make  
[root@localhost libpng-1.6.12]# make install
```


配置PHP服务

freetype源码包是用于提供字体支持引擎的服务程序

```
[root@localhost libpng-1.6.12]# cd ..  
[root@localhost src]# tar xzvf freetype-2.5.3.tar.gz  
[root@localhost src]# cd freetype-2.5.3  
[root@localhost freetype-2.5.3]# ./configure --prefix=/usr/local/freetype --  
enable-shared  
[root@localhost freetype-2.5.3]# make  
[root@localhost freetype-2.5.3]# make install
```

配置PHP服务

jpeg源码包是用于提供jpeg图片格式支持函数库的服务程序

```
[root@localhost freetype-2.5.3]# cd ..  
[root@localhost src]# tar xzvf jpegsrc.v9a.tar.gz  
[root@localhost src]# cd jpeg-9a  
[root@localhost jpeg-9a]# ./configure --prefix=/usr/local/jpeg --enable-shared  
[root@localhost jpeg-9a]# make  
[root@localhost jpeg-9a]# make install
```

配置PHP服务

libgd源码包是用于提供图形处理的服务程序,在编译libgd源码包时, 请记得写入的是jpeg、libpng、freetype、tiff、libvpx等服务程序在系统中的安装路径, 即在上面安装过程中使用--prefix参数指定的目录路径

```
[root@localhost jpeg-9a]# cd ..  
[root@localhost src]# tar xzvf libgd-2.1.0.tar.gz  
[root@localhost src]# cd libgd-2.1.0  
[root@localhost libgd-2.1.0]# ./configure --prefix=/usr/local/libgd --enable-shared --with-jpeg=/usr/local/jpeg --with-png=/usr/local/libpng --with-freetype=/usr/local/freetype --with-fontconfig=/usr/local/freetype --with-xpm=/usr/ --with-tiff=/usr/local/tiff --with-vpx=/usr/local/libvpx  
[root@localhost libgd-2.1.0]# make  
[root@localhost libgd-2.1.0]# make install
```

配置PHP服务

t1lib源码包是用于提供图片生成函数库的服务程序

安装后把/usr/lib64目录中的函数文件链接到/usr/lib目录中，以便系统能够顺利调取到函数文件

```
[root@localhost libgd-2.1.0]# cd ..
[root@localhost src]# tar xzvf t1lib-5.1.2.tar.gz
[root@localhost src]# cd t1lib-5.1.2
[root@localhost t1lib-5.1.2]# ./configure --prefix=/usr/local/t1lib --enable-shared
[root@localhost t1lib-5.1.2]# make
[root@localhost t1lib-5.1.2]# make install
[root@localhost t1lib-5.1.2]# ln -s /usr/lib64/libltdl.so /usr/lib/libltdl.so
[root@localhost t1lib-5.1.2]# cp -frp /usr/lib64/libXpm.so* /usr/lib
```

配置PHP服务

此时终于把编译php服务源码包的相关软件包都已经安装部署妥当了。

在开始编译php源码包之前，先定义一个名为LD_LIBRARY_PATH的全局环境变量，该环境变量的作用是帮助系统找到指定的动态链接库文件，这些文件是编译php服务源码包的必须元素之一。

编译php服务源码包时，除了定义要安装到的目录以外，还需要依次定义配置php服务程序配置文件的保存目录、MySQL数据库服务程序所在目录、MySQL数据库服务程序配置文件所在目录，以及libpng、jpeg、freetype、libvpx、zlib、t1lib等服务程序的安装目录路径，并通过参数启动php服务程序的诸多默认功能

```
[root@localhost t1lib-5.1.2]# cd ..
[root@localhost src]# tar -xzvf php-5.5.14.tar.gz
[root@localhost src]# cd php-5.5.14
[root@localhost php-5.5.14]# export LD_LIBRARY_PATH=/usr/local/libgd/lib
[root@localhost php-5.5.14]# ./configure --prefix=/usr/local/php --with-config-file-path=/usr/local/php/etc --with-mysql=/usr/local/mysql --with-mysqli=/usr/local/mysql/bin/mysql_config --with-mysql-sock=/tmp/mysql.sock --with-pdo-mysql=/usr/local/mysql --with-gd --with-png-dir=/usr/local/libpng --with-jpeg-dir=/usr/local/jpeg --with-freetype-dir=/usr/local/freetype --with-xpm-dir=/usr/ --with-vpx-dir=/usr/local/libvpx/ --with-zlib-dir=/usr/local/zlib --with-t1lib=/usr/local/t1lib --with-iconv --enable-libxml --enable-xml --enable-bcmath --enable-shmop --enable-sysvsem --enable-inline-optimization --enable-opcache --enable-mbregex --enable-fpm --enable-mbstring --enable-ftp --enable-gd-native-ttf --with-openssl --enable-pcntl --enable-sockets --with-xmlrpc --enable-zip --enable-soap --without-pear --with-gettext --enable-session --with-mcrypt --with-curl --enable-ctype
[root@localhost php-5.5.14]# make
[root@localhost php-5.5.14]# make install
```

配置PHP服务

在php源码包程序安装完成后，需要删除当前默认的配置文件，然后将php服务程序目录中相应的配置文件复制过来

```
[root@localhost php-5.5.14]# rm -rf /etc/php.ini
[root@localhost php-5.5.14]# ln -s /usr/local/php/etc/php.ini /etc/php.ini
[root@localhost php-5.5.14]# cp php.ini-production /usr/local/php/etc/php.ini
[root@localhost php-5.5.14]# cp /usr/local/php/etc/php-fpm.conf.default /usr/local/php/etc/php-fpm.conf
[root@localhost php-5.5.14]# ln -s /usr/local/php/etc/php-fpm.conf /etc/php-fpm.conf
```

配置PHP服务

php-fpm.conf是php服务程序重要的配置文件之一，我们需要启用该配置文件中第25行左右的pid文件保存目录，然后分别将第148和149行的user与group参数分别修改为www账户和用户组名称

```
[root@localhost php-5.5.14]# vi /usr/local/php/etc/php-fpm.conf
25行 pid = run/php-fpm.pid
148行 user = www
149行 group = www
```

配置PHP服务

配置妥当后便可把用于管理php服务的脚本文件复制到/etc/rc.d/init.d中了。为了能够执行脚本，请记得为脚本赋予755权限。最后把php-fpm服务程序加入到开机启动项中

```
[root@localhost php-5.5.14]# cp sapi/fpm/init.d.php-fpm /etc/rc.d/init.d/php-fpm
[root@localhost php-5.5.14]# chmod 755 /etc/rc.d/init.d/php-fpm
[root@localhost php-5.5.14]# chkconfig php-fpm on
```


配置PHP服务

为了安全，编辑php.ini配置文件，在305行的disable_functions参数后面追加上要禁止的功能。

```
[root@localhost php-5.5.14]# vi /usr/local/php/etc/php.ini
305行 disable_functions = passthru,exec,system,chroot,scandir,chgrp,chmod,shell_e
xec,proc_open,proc_get_status,ini_alter,ini_restore,dl,openlog,syslog,
readlink,symlink,popepassthru,stream_socket_server,escapeshellcmd,dll,popen,disk
_free_space,checkdnsrr,checkdnsrr,g etservbyname,getservbyport,disk_total_space,
posix_ctermid,posix_get_last_error,posix_getcwd,posix_getegid,posix_geteuid,posi
x_getgid,po six_getgrgid,posix_getgrnam,posix_getgroups,posix_getlogin,posix_get
pgid,posix_getpgrp,posix_getpid,posix_getppid,posix_getpwnam,posix_ getpwuid,pos
ix_getrlimit,posix_getsid,posix_getuid,posix_isatty,posix_kill,posix_mkfifo,posi
x_setegid,posix_set_euid,posix_setgid,posix_ setpgid,posix_setsid,posix_setuid,p
osix_strerror,posix_times,posix_ttyname,posix_uname
```

配置PHP服务

编辑Nginx服务程序的主配置文件，

把第2行的井号（#）删除，然后在后面写上负责运行Nginx服务程序的账户名称和用户组名称；

在第45行的index参数后面写上网站的首页名称。

最后是将第65～71行参数前的井号（#）删除来启用参数，主要是修改第69行的脚本名称路径参数，其中\$document_root变量即为网站信息存储的根目录路径，若没有设置该变量，则Nginx服务程序无法找到网站信息，因此会提示“404页面未找到”的报错信息。

在确认参数信息填写正确后便可重启Nginx服务与php-fpm服务

```
[root@localhost php-5.5.14]# vi /usr/local/nginx/conf/nginx.conf
2行   user   www;
45行  index  index.html index.htm index.php;
65行  location ~ /\.php$ {
66行  root   html;
67行  fastcgi_pass   127.0.0.1:9000;
68行  fastcgi_index  index.php;
69行  fastcgi_param  SCRIPT_FILENAME  $document_root$fastcgi_script_name;
70行  include       fastcgi_params;
71行  }
[root@localhost php-5.5.14]# systemctl restart nginx
[root@localhost php-5.5.14]# systemctl restart php-fpm
```

搭建Discuz论坛



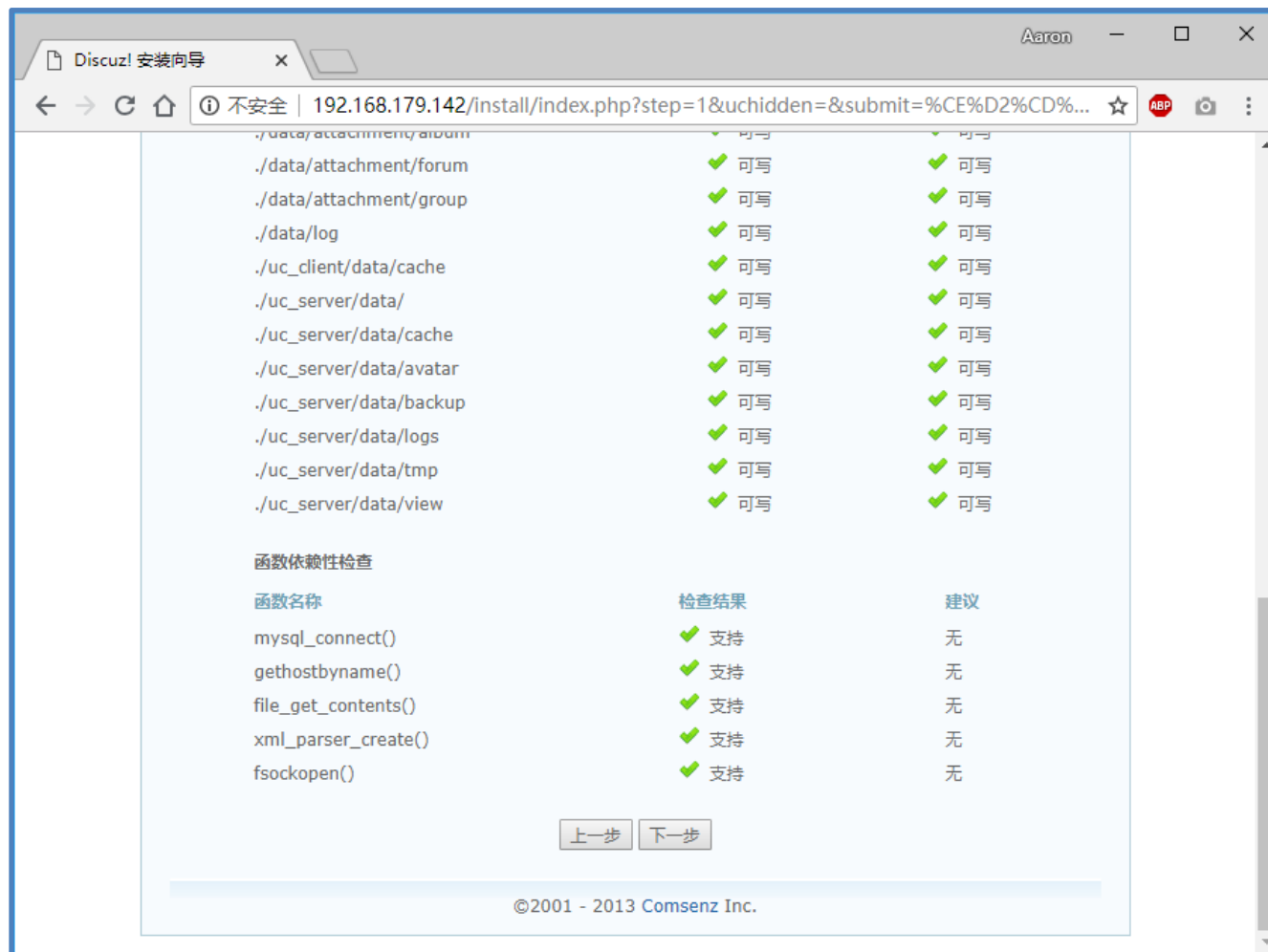
搭建Discuz论坛

```
[root@localhost php-5.5.14]# cd /usr/local/src/  
[root@localhost src]# unzip Discuz_X3.2_SC_GBK.zip  
[root@localhost src]# rm -rf /usr/local/nginx/html/{index.html,50x.html}*  
[root@localhost src]# mv upload/* /usr/local/nginx/html/  
[root@localhost src]# chown -Rf www:www /usr/local/nginx/html  
[root@localhost src]# chmod -Rf 755 /usr/local/nginx/html
```

搭建Discuz论坛



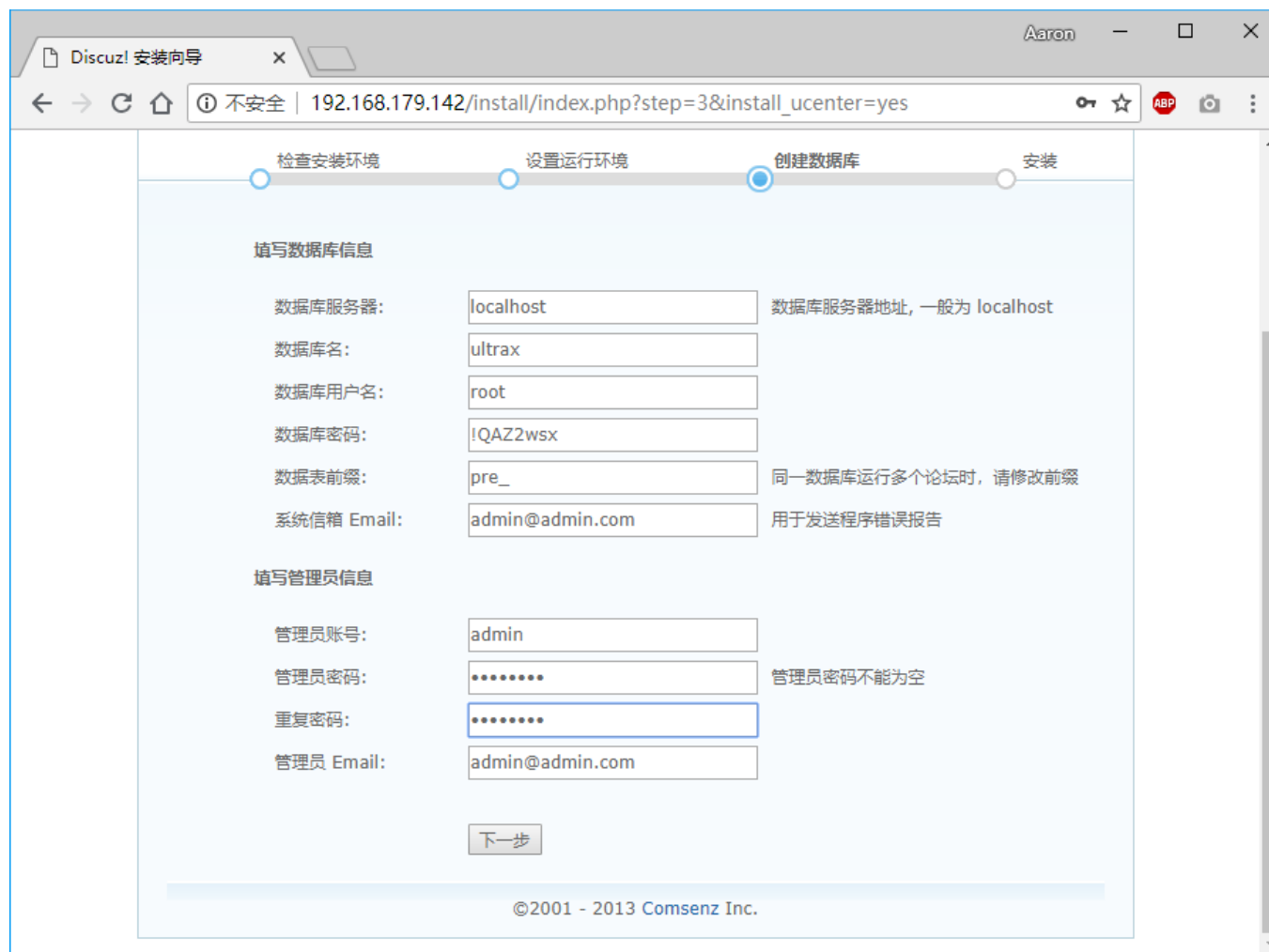
搭建Discuz论坛



搭建Discuz论坛



搭建Discuz论坛



The screenshot shows the Discuz! installation wizard in a web browser. The browser window has a single tab titled 'Discuz! 安装向导'. The address bar shows the URL '192.168.179.142/install/index.php?step=3&install_ucenter=yes'. The page features a progress bar at the top with four steps: '检查安装环境', '设置运行环境', '创建数据库' (which is the current step and highlighted with a blue circle), and '安装'. Below the progress bar, the form is divided into two sections: '填写数据库信息' and '填写管理员信息'. The '填写数据库信息' section contains six input fields: '数据库服务器' (localhost), '数据库名' (ultrax), '数据库用户名' (root), '数据库密码' (!QAZ2wsx), '数据表前缀' (pre_), and '系统信箱 Email' (admin@admin.com). The '填写管理员信息' section contains four input fields: '管理员账号' (admin), '管理员密码' (masked with dots), '重复密码' (masked with dots), and '管理员 Email' (admin@admin.com). A '下一步' button is located at the bottom of the form. The footer of the page reads '©2001 - 2013 Comsenz Inc.'

Discuz! 安装向导

← → ↻ ⬆ ① 不安全 | 192.168.179.142/install/index.php?step=3&install_ucenter=yes

检查安装环境 设置运行环境 创建数据库 安装

填写数据库信息

数据库服务器: localhost 数据库服务器地址, 一般为 localhost

数据库名: ultrax

数据库用户名: root

数据库密码: !QAZ2wsx

数据表前缀: pre_ 同一数据库运行多个论坛时, 请修改前缀

系统信箱 Email: admin@admin.com 用于发送程序错误报告

填写管理员信息

管理员账号: admin

管理员密码: 管理员密码不能为空

重复密码:

管理员 Email: admin@admin.com

下一步

©2001 - 2013 Comsenz Inc.

搭建Discuz论坛



搭建Discuz论坛



