

第1章 LNMP架构概述

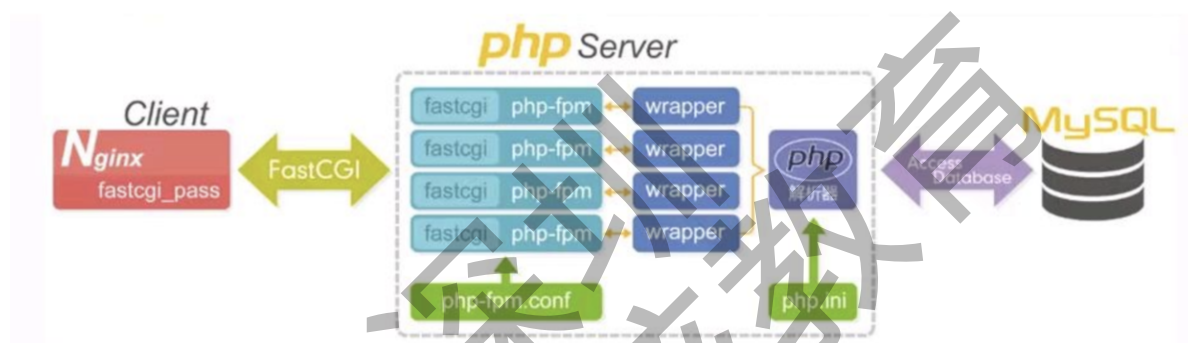
1.什么是LNMP

- 1 LNMP 是一套技术的组合， L=Linux、 N=Nginx、 M=MySQL、 P=PHP

2.LNMP架构是如何工作的

- 1 首先 Nginx 服务是不能处理动态请求，那么当用户发起动态请求时，Nginx 又是如何进行处理的。
- 2 当用户发起 http 请求，请求会被 Nginx 处理，如果是静态资源请求 Nginx 则直接返回，如果是动态请求 Nginx 则通过 fastcgi 协议转交给后端的 PHP 程序处理，具体如下图所示

3.Nginx与Fast-CGI工作流程



比喻：

- 1 Nginx: 拉货的货车
- 2 FastCGI: 高速公路
- 3 php-fpm: 每个路道的收费站
- 4 wrapper: 搬货的师傅
- 5 systemctl start php-fpm: 管理整个高速工作所有的收费站
- 6
- 7 1.用户通过 http 协议发起请求，请求会先抵达 LNMP 架构中的 Nginx
- 8 2.Nginx 会根据用户的请求进行判断，这个判断是有 Location 进行完成
- 9 3.判断用户请求的是静态页面， Nginx 直接进行处理
- 10 4.判断用户请求的是动态页面， Nginx 会将该请求交给 fastcgi 协议下发
- 11 5.fastcgi 会将请求交给 php-fpm 管理进程，php-fpm 管理进程接收到后会调用具体的工作进程 warrap
- 12 6.warrap 进程会调用 php 程序进行解析,如果只是解析代码 php 直接返回
- 13 7.如果有查询数据库操作，则由 php 连接数据库(用户 密码 IP)发起查询的操作
- 14 8.最终数据由 mysql->php->php-fpm->fastcgi->nginx->http->user

第2章 LNMP安装部署

1.创建www统一用户

```
1 [root@web-7 ~]# groupadd www -g 666
2 [root@web-7 ~]# useradd www -s /sbin/nologin -M -u 666 -g 666
3 [root@web-7 ~]# id www
4 uid=666(www) gid=666(www) 组=666(www)
```

2.使用官方仓库安装nginx

```
1 [root@web-7 ~]# cat /etc/yum.repos.d/nginx.repo
2 [nginx-stable]
3 name=nginx stable repo
4 baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
5 gpgcheck=1
6 enabled=1
7 gpgkey=https://nginx.org/keys/nginx_signing.key
8
9 [nginx-mainline]
10 name=nginx mainline repo
11 baseurl=http://nginx.org/packages/mainline/centos/$releasever/$basearch/
12 gpgcheck=1
13 enabled=0
14 gpgkey=https://nginx.org/keys/nginx_signing.key
15
16 [root@web-7 ~]# yum install nginx -y
```

3.启动Nginx并加入开机自启动

```
1 [root@web-7 ~]# systemctl start nginx
2 [root@web-7 ~]# systemctl enable nginx
```

4.使用第三方拓展源安装php7.1

```
1 [root@web-7 ~]# yum remove php-mysql-5.4 php php-fpm php-common
2 [root@web-7 ~]# rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
3 [root@web-7 ~]# rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release.rpm
4 [root@web-7 ~]# yum -y install php71w php71w-cli php71w-common php71w-devel php71w-embedded php71w-gd php71w-mcrypt php71w-mbstring php71w-pdo php71w-xml php71w-fpm php71w-mysqlnd php71w-opcache php71w-pear php71w-redis php71w-pear-mongodb
```

5.配置php-fpm用户与nginx用户保持一致

```
1 [root@web-7 ~]# sed -i '/^user/c user = www' /etc/php-fpm.d/www.conf
2 [root@web-7 ~]# sed -i '/^group/c group = www' /etc/php-fpm.d/www.conf
```

6.启动php-fpm并加入开机自启动

```
1 [root@web-7 ~]# systemctl start php-fpm
2 [root@web-7 ~]# systemctl enable php-fpm
```

7.安装Mariadb数据库

```
1 [root@web-7 ~]# yum install mariadb-server mariadb -y
```

8.启动Mariadb数据库并加入开机自启动

```
1 [root@web-7 ~]# systemctl start mariadb
2 [root@web-7 ~]# systemctl enable mariadb
```

9.配置Mariadb账号密码

```
1 [root@web-7 ~]# mysqladmin password 'oldboy123'
2 [root@web-7 ~]# mysql -uroot -poldboy123
```

10.mysql基础命令

```
1 #查看当前mysql有哪些用户
2 MariaDB [(none)]> select user,host from mysql.user;
3 +-----+-----+
4 | user | host |
5 +-----+-----+
6 | root | 127.0.0.1 |
7 | root | ::1 |
8 |      | localhost |
9 | root | localhost |
10 |      | web-7 |
11 | root | web-7 |
12 +-----+-----+
```

登录mysql

```
1 [root@web-7 ~]# mysql -uroot -poldboy123
```

使用指定IP登录

```
1 [root@web-7 ~]# mysql -uroot -poldboy123 -h127.0.0.1
```

查询当前数据库有哪些用户

```
1 select user,host from mysql.user;
2 查询    mysql库里的user表里的    user和host字段的内容
```

查看当前有哪些数据库

```

1 MariaDB [(none)]> show databases;
2 +-----+
3 | Database          |
4 +-----+
5 | information_schema |
6 | mysql              |
7 | performance_schema |
8 | test               |
9 +-----+

```

查看库里面有哪些表

```

1 MariaDB [(none)]> show tables from mysql;
2 +-----+
3 | Tables_in_mysql    |
4 +-----+
5 | columns_priv        |
6 | db                  |
7 | event               |
8 | func                |
9 | general_log         |
10 | help_category       |
11 | help_keyword        |
12 | help_relation       |
13 | help_topic          |
14 | host                |
15 | ndb_binlog_index    |
16 | plugin              |
17 | proc                |
18 | procs_priv          |
19 | proxies_priv        |
20 | servers              |
21 | slow_log            |
22 | tables_priv         |
23 | time_zone           |
24 | time_zone_leap_second |
25 | time_zone_name      |
26 | time_zone_transition |
27 | time_zone_transition_type |
28 | user                |
29 +-----+
30 24 rows in set (0.00 sec)

```

查看表里面有哪些字段

```

1 MariaDB [mysql]> desc user;
2 +-----+-----+-----+-----+-----+-----+
3 | Field          | Type          | Null | Key |
4 | Default | Extra |
5 | Host          | char(60)      | NO   | PRI |
6 | User          | char(16)      | NO   | PRI |

```

7	Password	char(41)	NO		
8	Select_priv	enum('N','Y')	NO		N
9	Insert_priv	enum('N','Y')	NO		N
10	Update_priv	enum('N','Y')	NO		N
11	Delete_priv	enum('N','Y')	NO		N
12	Create_priv	enum('N','Y')	NO		N
13	Drop_priv	enum('N','Y')	NO		N
14	Reload_priv	enum('N','Y')	NO		N
15	Shutdown_priv	enum('N','Y')	NO		N
16	Process_priv	enum('N','Y')	NO		N
17	File_priv	enum('N','Y')	NO		N
18	Grant_priv	enum('N','Y')	NO		N
19	References_priv	enum('N','Y')	NO		N
20	Index_priv	enum('N','Y')	NO		N
21	Alter_priv	enum('N','Y')	NO		N
22	Show_db_priv	enum('N','Y')	NO		N
23	Super_priv	enum('N','Y')	NO		N
24	Create_tmp_table_priv	enum('N','Y')	NO		N
25	Lock_tables_priv	enum('N','Y')	NO		N
26	Execute_priv	enum('N','Y')	NO		N
27	Rep_l_slave_priv	enum('N','Y')	NO		N
28	Rep_l_client_priv	enum('N','Y')	NO		N
29	Create_view_priv	enum('N','Y')	NO		N
30	Show_view_priv	enum('N','Y')	NO		N
31	Create_routine_priv	enum('N','Y')	NO		N
32	Alter_routine_priv	enum('N','Y')	NO		N
33	Create_user_priv	enum('N','Y')	NO		N
34	Event_priv	enum('N','Y')	NO		N
35	Trigger_priv	enum('N','Y')	NO		N

```

36 | Create_tablespace_priv | enum('N','Y') | NO | | N
37 | ssl_type | enum('', 'ANY', 'X509', 'SPECIFIED') | NO | | 
38 | ssl_cipher | blob | NO | | 
39 | x509_issuer | blob | NO | | 
40 | x509_subject | blob | NO | | 
41 | max_questions | int(11) unsigned | NO | | 0
42 | max_updates | int(11) unsigned | NO | | 0
43 | max_connections | int(11) unsigned | NO | | 0
44 | max_user_connections | int(11) | NO | | 0
45 | plugin | char(64) | NO | | 
46 | authentication_string | text | NO | | 
47 +-----+-----+-----+-----+-----+-----+
48 42 rows in set (0.00 sec)

```

退出

```
1 | exit
```

第3章 LNMP环境配置

1. Fastcgi语法

设置 fastcgi 服务器的地址，该地址可以指定为域名或 IP 地址，以及端口

```

1 | Syntax: fastcgi_pass address;
2 | Default: -
3 | Context: location, if in location
4 |
5 | #语法示例
6 | fastcgi_pass localhost:9000;
7 | fastcgi_pass unix:/tmp/fastcgi.socket;

```

设置 fastcgi 默认的首页文件，需要结合 fastcgi_param 一起设置

```

1 | Syntax: fastcgi_index name;
2 | Default: -
3 | Context: http, server, location

```

通过 fastcgi_param 设置变量，并将设置的变量传递到后端的 fastcgi 服务器

```
1 Syntax: fastcgi_param parameter value [if_not_empty];
2 Default: -
3 Context: http, server, location
4
5 #语法示例
6 fastcgi_index index.php;
7 fastcgi_param SCRIPT_FILENAME /code$fastcgi_script_name;
```

2.最终Nginx连接FastCGI服务器配置如下

```
1 [root@web-7 ~]# cat /etc/nginx/conf.d/php.conf
2 server {
3     server_name www.oldboy.com;
4     listen 80;
5     root /code;
6     index index.php index.html;
7
8     location ~ \.php$ {
9         root /code;
10        fastcgi_pass 127.0.0.1:9000;
11        fastcgi_index index.php;
12        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
13        include fastcgi_params;
14    }
15 }
```


3.在/code目录下创建info.php文件并访问测试

```
1 [root@web-7 ~]# mkdir /code
2 [root@web-7 ~]# chown -R www:www /code/
3 [root@web-7 ~]# cat /code/info.php
4 <?php
5     phpinfo();
6 ?>
```

4.检查nginx语法并重启

```
1 [root@web-7 ~]# nginx -t
2 nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
3 nginx: configuration file /etc/nginx/nginx.conf test is successful
4 [root@web-7 ~]# systemctl restart nginx
```

5.绑定hosts然后在浏览器访问

PHP Version 7.1.30	
	
System	Linux web01 3.10.0-327.el7.x86_64 #1 SMP Thu Nov 19 22:10:57 UTC 2015 x86_64
Build Date	Jun 2 2019 08:07:10
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/bz2.ini, /etc/php.d/calendar.ini, /etc/php.d/ctype.ini, /etc/php.d/curl.ini, /etc/php.d/dom.ini, /etc/php.d/exif.ini, /etc/php.d/fileinfo.ini, /etc/php.d/ftp.ini, /etc/php.d/gd.ini, /etc/php.d/gettext.ini, /etc/php.d/gmp.ini, /etc/php.d/iconv.ini, /etc/php.d/igmp.ini, /etc/php.d/json.ini, /etc/php.d/mbstring.ini, /etc/php.d/mcrypt.ini, /etc/php.d/mongodb.ini, /etc/php.d/mysqli.ini, /etc/php.d/mysqli_mysql.ini, /etc/php.d/openssl.ini, /etc/php.d/pdo.ini, /etc/php.d/pdo_mysql.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/phar.ini, /etc/php.d/posix.ini, /etc/php.d/redis.ini, /etc/php.d/shmop.ini, /etc/php.d/simplexml.ini, /etc/php.d/sockets.ini, /etc/php.d/sqlite3.ini, /etc/php.d/sysvmsg.ini, /etc/php.d/sysvsem.ini, /etc/php.d/sysvshm.ini, /etc/php.d/tokenizer.ini, /etc/php.d/xml.ini, /etc/php.d/xml_wddx.ini, /etc/php.d/xmlreader.ini, /etc/php.d/xmlwriter.ini, /etc/php.d/xsl.ini, /etc/php.d/zlib.ini, /etc/php.d/zlib.ini
PHP API	20160303
PHP Extension	20160303
Zend Extension	320160303
Zend Extension Build	API320160303,NTS
PHP Extension Build	API20160303,NTS
Debug Build	no

6.测试php和数据库访问是否正常

```
1 [root@web-7 ~]# cat /code/mysql.php
2 <?php
3     $servername = "localhost";
4     $username = "root";
5     $password = "oldboy123";
6
7     // 创建连接
8     $conn = mysqli_connect($servername, $username, $password);
9
10    // // 检测连接
11    if (!$conn) {
12        die("Connection failed: " . mysqli_connect_error());
13    }
14    echo "php 连接 MySQL 数据库成功";
15    ?>
```

7.通过浏览器访问数据库页面

← → ↻ ⓘ 不安全 | www.oldboy.com/mysql.php

php 连接 MySQL 数据库成功

第4章 部署博客Wordpress

1.配置Nginx虚拟主机站点

```
1 [root@web-7 ~]# cat /etc/nginx/conf.d/wordpress.conf
2 server {
3     listen 80;
4     server_name blog.oldboy.com;
```



```
5     root /code/wordpress;
6     index index.php index.html;
7
8     location ~ \.php$ {
9         root /code/wordpress;
10        fastcgi_pass 127.0.0.1:9000;
11        fastcgi_index index.php;
12        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
13        include fastcgi_params;
14    }
15 }
```

2.重启nginx

```
1 [root@web-7 ~]# systemctl restart nginx
```

3.下载解压wordpress到代码目录

```
1 [root@web-7 ~]# mkdir /code
2 [root@web-7 ~]# cd /code/
3 [root@web-7 /code]# wget https://cn.wordpress.org/wordpress-4.9.4-zh_CN.tar.gz
4 [root@web-7 /code]# tar xf wordpress-4.9.4-zh_CN.tar.gz
5 [root@web-7 /code]# chown -R www:www /code/wordpress
```

4.创建wordpress数据库

```
1 [root@web-7 /code]# mysql -uroot -poldboy123
2 > create database wordpress;
3 > exit
```

5.浏览器访问wordpress并部署

深井教案
老男孙

WordPress > 调整配置文件

blog.oldboy.com/wp-admin/setu...



欢迎使用WordPress。在开始前，我们需要您数据库的一些信息。请准备好如下信息。

1. 数据库名
2. 数据库用户名
3. 数据库密码
4. 数据库主机
5. 数据表前缀 (table prefix, 特别是当您要在一个数据库中安装多个WordPress时)

我们会使用这些信息来创建一个wp-config.php文件。如果自动创建未能成功，不用担心，您要做的只是将数据库信息填入配置文件。您也可以在文本编辑器中打开wp-config-sample.php，填入您的信息，并将其另存为wp-config.php。需要更多帮助？[看这里](#)。

绝大多数时候，您的网站服务提供商会给您这些信息。如果您没有这些信息，在继续之前您将需要联系他们。如果您准备好了...

现在就开始！



请在下方填写您的数据库连接信息。如果您不确定，请联系您的服务提供商。

数据库名	<input type="text" value="wordpress"/>	将WordPress安装到哪个数据库？
用户名	<input type="text" value="root"/>	您的数据库用户名。
密码	<input type="password" value="oldboy123"/>	您的数据库密码。
数据库主机	<input type="text" value="localhost"/>	如果localhost不能用，您通常可以从网站服务提供商处得到正确的信息。
表前缀	<input type="text" value="wp_"/>	如果您希望在同一个数据库安装多个WordPress，请修改前缀。

提交



不错。您完成了安装过程中重要的一步，WordPress现在已经可以连接数据库了。如果您准备好了的话，现在就...

现在安装

欢迎

欢迎使用著名的WordPress五分钟安装程序！请简单地填写下面的表格，来开始使用这个世界上最具扩展性、最强大的个人信息发布平台。

需要信息

您需要填写一些基本信息。无需担心填错，这些信息以后可以再次修改。

站点标题

oldzhang

用户名

wordpress

用户名只能含有字母、数字、空格、下划线、连字符、句号和“@”符号。

密码

123456

隐藏

非常弱

重要：您将需要此密码来登录，请将其保存在安全的位置。

确认密码

☒ 确认使用弱密码

您的电子邮件

526195417@qq.com

请仔细检查电子邮件地址后再继续。

对搜索引擎的可见性

☐ 建议搜索引擎不索引本站点

搜索引擎将本着自觉自愿的原则对待WordPress提出的请求。并不是所有搜索引擎都会遵守这类请求。

安装WordPress



成功！

WordPress安装完成。谢谢！

用户名

wordpress

密码

您设定的密码。

登录



用户名或电子邮件地址

wordpress

密码

●●●●●●

☐ 记住我的登录信息

登录

忘记密码?

← 返回到oldzhang

OLDZHANG

又一个WordPress站点

2019年7月30日 由WORDPRESS

第一篇博客

这是我的第一篇博客!

编辑

搜索...



近期文章

第一篇博客

近期评论

一位WordPress评论者发表在《第一篇博客》

第5章 部署问答网站Wecenter

1.配置Nginx虚拟站点

```
1 [root@web-7 ~]# cat /etc/nginx/conf.d/wecenter.conf
2 server {
3     listen 80;
4     server_name zh.oldboy.com;
5     root /code/zh;
6     index index.php index.html;
7
8     location ~ /\.php$ {
9         root /code/zh;
10        fastcgi_pass 127.0.0.1:9000;
11        fastcgi_index index.php;
12        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
13        include fastcgi_params;
14    }
15 }
```

2.下载解压wecenter并授权

```
1 [root@web-7 ~]# mkdir /code/zh -p
2 [root@web-7 ~]# cd /opt/
3 [root@web-7 /opt]# rz
4 [root@web-7 /opt]# ls
5 weCenter_3-3-2.zip
6 [root@web-7 /opt]# unzip weCenter_3-3-2.zip -d /code/zh/
7 [root@web-7 /opt]# chown -R www:www /code/
```

3.创建wecenter数据库

```
1 [root@web-7 ~]# mysql -uroot -poldboy123
2 > create database zh;
3 > exit;
```

4.浏览器访问

openssl 支持 ✓

编码转换 ✓

上传限制 2M (此处建议值 > 8M)

目录权限 /code/zh/system/ ✓

配置目录权限 /code/zh/system/config/ ✓

缓存目录权限 /code/zh/cache/ ✓

临时目录权限 /code/zh/tmp/ ✓

上传目录权限 /code/zh/uploads/ ✓

下一步

遇到问题? 联系我们 | Copyright © - VieCenter 3.3.2, All Rights Reserved

• 配置系统

需要您提供必要的系统配置信息

数据库主机 localhost 通常为 localhost

数据库帐号 root

数据库密码 oldboy123

数据库端口 一般情况下不需要填写

数据库名称 zh

数据表前缀 aws_ 同数据库安装多个本程序时需要更改

数据表类型 InnoDB 请根据服务器状态选择数据表类型

开始安装

• 添加管理员

数据库导入成功, 创建管理员账户

用户名 admin

密码

E-mail 526195417@qq.com

完成

• 安装成功

欢迎使用 WeCenter 问答交流平台, 为了增强安全性, 请将 install/index.php 文件删除

[访问网站首页](#)

5. 清除首页安装文件

```
1 [root@web-7 ~]# rm -rf /code/zh/install
```

第6章 拆分数据库

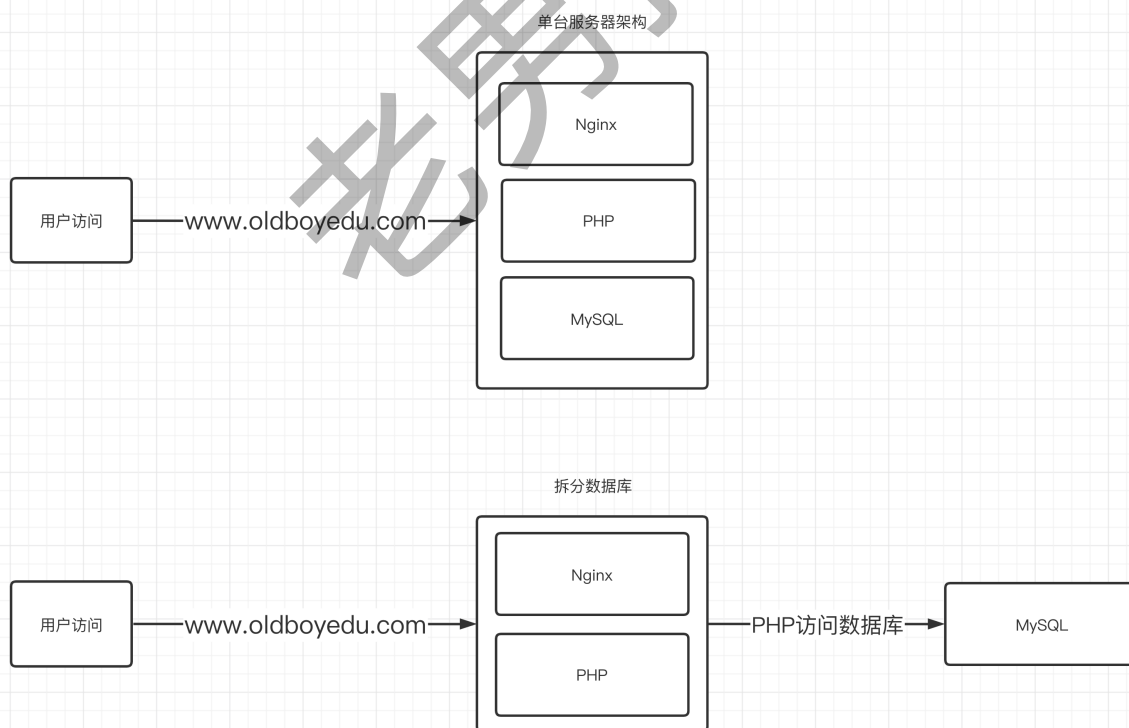
1. 为什么要进行数据库的拆分

- 1 由于单台服务器运行 LNMP 架构会导致网站访问缓慢, 当内存被吃满时, 很容易导致系统出现 oom, 从而 kill 掉MySQL 数据库, 所以需要将 web 和数据库进行独立部署

2. 数据库拆分后解决了什么问题

- 1 缓解 web 网站的压力
- 2 增强数据库读写性能
- 3 提高用户访问的速度

3. 数据库拆分架构演变过程



4. 数据库拆分环境搭建


```
1 web-7 172.16.1.7    nginx+php
2 db-51 172.16.1.51  mysql
```

5.数据库拆分详细步骤

5.1 web-7上备份数据库并拷贝到db-51上

备份 web-7 上的数据库,oldboy123 是数据库密码

```
1 [root@web-7 ~]# mysqldump -uroot -p'oldboy123' -A --single-transaction >
  mysql-all.sql
```

将 web-7 上备份的数据库拷贝至 db-51 服务器上

```
1 [root@web-7 ~]# scp mysql-all.sql 10.0.1.51:/tmp
```

5.2 db-51恢复数据库

将 web-7 服务器上推送的数据库备份文件恢复至 db-51 服务器新数据库中

```
1 [root@db-51 ~]# yum install mariadb mariadb-server -y
2 [root@db-51 ~]# systemctl start mariadb
3 [root@db-51 ~]# systemctl enable mariadb
4 [root@db-51 ~]# mysqladmin password 'oldboy123'
5 [root@db-51 ~]# mysql -uroot -p'oldboy123' < /tmp/mysql-all.sql
6 [root@db-51 ~]# mysql -uroot -p'oldboy123' -e "show databases;"
7 +-----+
8 | Database |
9 +-----+
10 | information_schema |
11 | mysql |
12 | performance_schema |
13 | test |
14 | wordpress |
15 | zh |
16 +-----+
```

5.3 db-51数据库进行授权

```
1 在新数据库上授权，允许所有网段，通过 all 账户连接并操作该数据库
2 grant all privileges on *.* to 'all'@%' identified by 'oldboy123';
3
4 授权所有权限 grant all privileges
5 授权所有库所有表 *.*
6 将授权赋予给哪个用户，这个用户只能通过哪个网段过来(%所有) 'all'@%'
7 授权该用户登录的密码 identified by
```

操作命令：

```
1 [root@db-51 ~]# mysql -uroot -p'oldboy123'
2 > grant all privileges on *.* to 'all'@'%' identified by 'oldboy123';
3 > flush privileges;
4 >exit;
```

测试使用IP地址能不能登陆

```
1 [root@db-51 ~]# mysql -uall -p'oldboy123' -h 10.0.1.51
2 > show databases;
3 +-----+
4 | Database          |
5 +-----+
6 | information_schema |
7 | mysql              |
8 | performance_schema |
9 | test               |
10 | wordpress           |
11 | zh                  |
12 +-----+
```

5.4 web-7修改代码连接到新数据库

修改wordpress

```
1 [root@web-7 ~]# vim /code/wordpress/wp-config.php
2 /** MySQL数据库用户名 */
3 define('DB_USER', 'all');
4
5 /** MySQL数据库密码 */
6 define('DB_PASSWORD', 'oldboy123');
7
8 /** MySQL主机 */
9 define('DB_HOST', '172.16.1.51');
```

修改wecenter

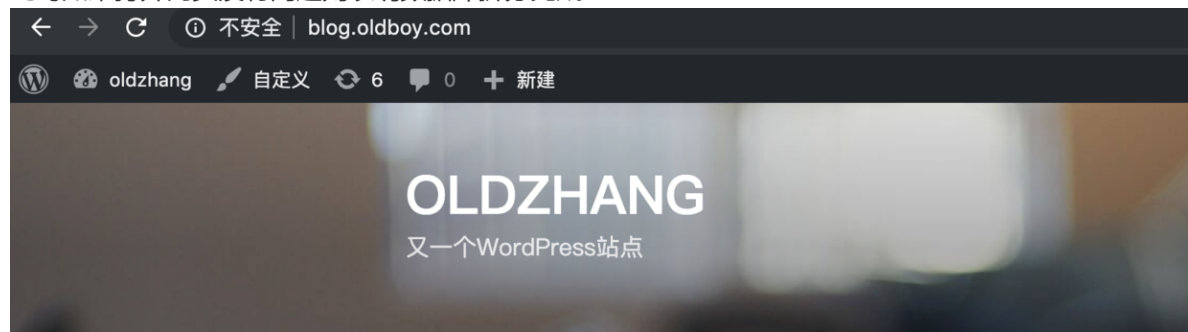
```
1 [root@web-7 ~]# vim /code/zh/system/config/database.php
2 'host' => '172.16.1.51',
3 'username' => 'all',
4 'password' => 'oldboy123',
5 'dbname' => 'zh',
```

停止web-7上的数据库

```
1 [root@web-7 ~]# systemctl stop mariadb.service
```

5.5 测试访问

此时如果打开网页没有问题则表明数据库拆分完成



文章

2019年7月30日 编辑

第一篇博客

这是我的第一篇博客！



第7章 拓展WEB服务器

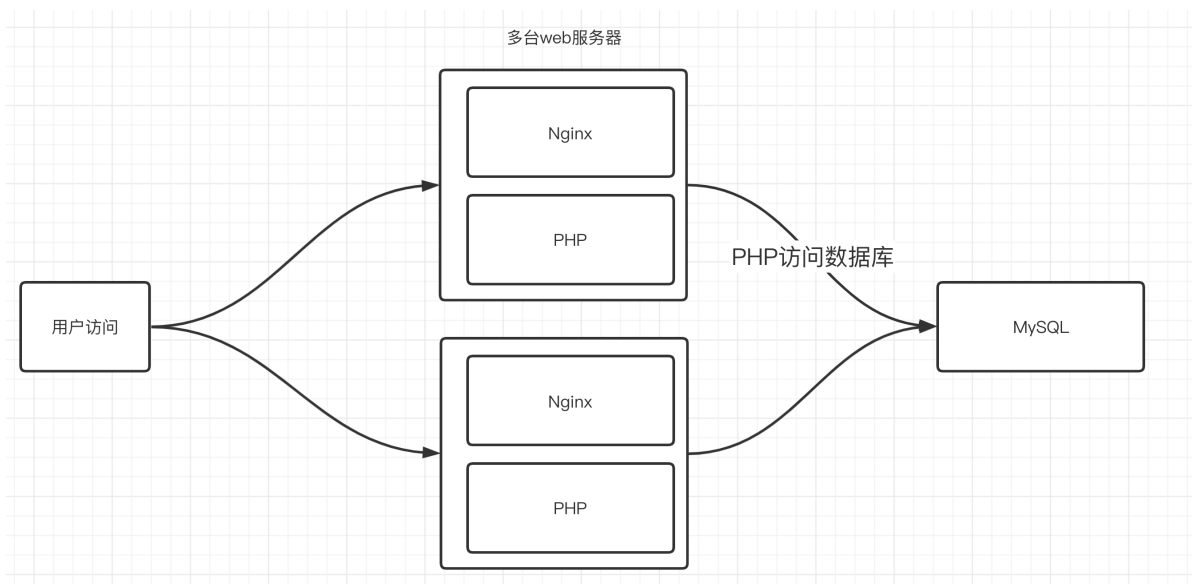
1.为什么要拓展多台web节点

单台 web 服务器能抗住的访问量是有限的，配置多台 web 服务器能提升更高的访问速度

2.拓展多台web解决了什么问题

- 1.单台 web 节点如果故障，会导致业务 down 机
- 2.多台 web 节点能保证业务的持续稳定，扩展性高
- 3.多台 web 节点能有效的提升用户访问网站的速度

3.多台web服务器架构组成



4.多台web服务器思路

1. 可以使用ansible批量部署多台web服务器
2. 配置内网私有yum仓库
3. 按照web-7的步骤安装好nginx和php，然后远程拷贝代码到新机器

5.正常配置步骤

5.1 创建用户名密码

- 1 [root@web-8 ~]# groupadd www -g 666
- 2 [root@web-8 ~]# useradd www -s /sbin/nologin -M -u 666 -g 666

5.2 安装NP

可以直接从web-7上拷贝yum源到本机yum目录

- 1 [root@web-8 ~]# scp -rp root@172.16.1.7:/etc/yum.repos.d/* /etc/yum.repos.d/
- 2 [root@web-8 ~]# scp -rp root@172.16.1.7:/etc/pki/rpm-gpg/* /etc/pki/rpm-gpg/
- 3 [root@web-8 ~]# yum install nginx -y
- 4 [root@web-8 ~]# yum -y install php71w php71w-cli php71w-common php71w-devel php71w-embedded php71w-gd php71w-mcrypt php71w-mbstring php71w-pdo php71w-xml php71w-fpm php71w-mysqlnd php71w-openssl php71w-pear php71w-redis php71w-pear-mongodb

5.3 拷贝web-7的nginx配置文件到web-8

- 1 [root@web-8 ~]# scp -rp root@172.16.1.7:/etc/nginx /etc/

5.4 拷贝web-7的php配置文件到web-8

- 1 [root@web-8 ~]# scp -rp root@172.16.1.7:/etc/php-fpm.d /etc/

5.5 拷贝代码目录到web-8

```
1 [root@web-7 ~]# tar czf code.tar.gz /code
2 [root@web-7 ~]# scp code.tar.gz root@172.16.1.8:/tmp
```

5.6 web-8上将代码解压到相应目录

```
1 [root@web-8 ~]# tar xzf /tmp/code.tar.gz -C /
```

5.7 web-8上启动nginx和php-fpm并加入开机自启动

```
1 [root@web-8 ~]# systemctl start nginx php-fpm
2 [root@web-8 ~]# systemctl enable nginx php-fpm
```

5.8 web访问测试

修改hosts为web-8的地址，然后浏览器访问测试

第8章 将静态资源挂载到共享存储

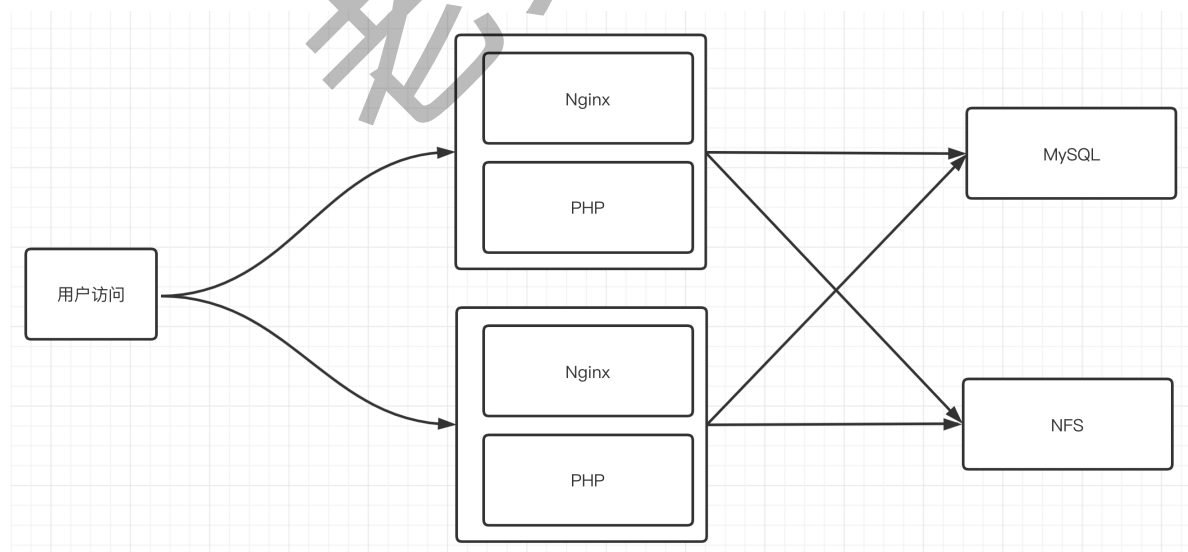
1.为什么要拆分静态资源到独立服务器

当后端的 web 节点出现多台时，会导致用户上传的图片、视频附件等内容仅上传至一台 web 服务器，那么其他的web 服务器则无法访问到该图片

2.新增一台nfs存储解决了什么问题

- 1.保证了多台 web 节点静态资源一致。
- 2.有效节省多台 web 节点的存储空间。
- 3.统一管理静态资源，便于后期推送至 CDN 进行静态资源加速

3.多台web节点架构组成



4.多台web节点环境规划

```
1 web-7      nginx+php
2 web-8      nginx+php
3 db-51      mysql
4 nfs        nfs
```

5.共享存储配置

5.1 安装配置NFS

```
1 [root@nfs01 ~]# yum install nfs-utils -y
2 [root@nfs01 ~]# cat /etc/exports
3 /data/blog 172.16.1.0/24(rw,sync,all_squash,anonuid=666,anongid=666)
4 /data/zh 172.16.1.0/24(rw,sync,all_squash,anonuid=666,anongid=666)
```

5.2 创建共享目录并进行授权

```
1 [root@nfs01 ~]# mkdir /data/{blog,zh} -p
2 [root@nfs01 ~]# chown -R www.www /data
```

5.3 启动nfs服务并加入开机自启

```
1 [root@nfs01 ~]# systemctl restart nfs-server
```

6.web-7端操作步骤如下

6.1 web-7节点挂在nfs

```
1 [root@web-7 ~]# yum install nfs-utils -y
2 [root@web-7 ~]# showmount -e 172.16.1.31
3 Export list for 172.16.1.31:
4 /data/zh 172.16.1.0/24
5 /data/blog 172.16.1.0/24
```

6.2 查找Wordpress 静态资源存放的位置

在wordpress的文章里添加一张图片，然后查看图片URL资源



这里地址为：

<http://blog.oldboy.com/wp-content/uploads/2019/07/DNS图.png>

6.3 备份web-7服务器上Wordpress 的静态资源

```
1 [root@web-7 ~]# cd /code/wordpress/wp-content
2 [root@web-7 /code/wordpress/wp-content]# cp -a uploads/ uploads_bak/
```

6.4 web-7客户端执行挂载操作

```
1 [root@web-7 /code/wordpress/wp-content]# mount -t nfs 172.16.1.31:/data/blog
/code/wordpress/wp-content/uploads/
2 [root@web-7 /code/wordpress/wp-content]# cp -rp uploads_bak/* uploads/
```

6.5 将挂载信息加入开机自启

```
1 [root@web-7 ~]# tail -1 /etc/fstab
2 172.16.1.31:/data/blog /code/wordpress/wp-content/uploads nfs defaults 0 0
```

7.web-8端操作

web-8端操作和web-7一样

```
1 [root@web-8 ~]# cd /code/wordpress/wp-content
2 [root@web-8 /code/wordpress/wp-content]# cp -a uploads/ uploads_bak/
3 [root@web-8 /code/wordpress/wp-content]# mount -t nfs 172.16.1.31:/data/blog
/code/wordpress/wp-content/uploads/
4 [root@web-8 /code/wordpress/wp-content]# cp -rp uploads_bak/* uploads/
5 [root@web-8 ~]# tail -1 /etc/fstab
6 172.16.1.31:/data/blog /code/wordpress/wp-content/uploads nfs defaults 0 0
```

深圳教習
老男仔